



UNIVERZITA MATEJA BELA
Fakulta prírodných vied, Katedra informatiky



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta prírodovedne-humanitní a pedagogická



MEDZINÁRODNÁ KONFERENCIA O VYUČOVANÍ INFORMATIKY

DidInfo 2023

28. ROČNÍK KONFERENCIE

ISBN 978-80-557-2038-8

EAN 9788055720388

ISSN 2454-051X

DOI <https://doi.org/10.24040/2023.9788055720388>



Táto publikácia je šírená pod licenciou Creative Commons Attribution 4.0 International Licence CC BY.

BANSKÁ BYSTRICA
2023

Predsedovia programového výboru:

doc. Ing. Jarmila Škrinárová, PhD. *Univerzita Mateja Bela, SK*

doc. PaedDr. Jirí Vaníček, PhD. *Jihočeská Univerzita, CZ*

Programový výbor konferencie:

doc. RNDr. Gabriela Andrejková, CSc. *Univerzita Pavla Jozefa Šafárika, SK*

Mgr. Jan Berki, PhD. *Technická univerzita v Liberci, CZ*

doc. RNDr. Miroslava Černochová, CSc. *Univerzita Karlova, CZ*

prof. Dr. Valentina Dagiene *Vilnius University, LT*

Mgr. Adam Dudáš, PhD. *Univerzita Mateja Bela, SK*

PaedDr. Ján Guniš, PhD. *Univerzita Pavla Jozefa Šafárika v Košiciach, SK*

doc. Mgr. et Mgr. Marie Hubálovská, PhD. *Univerzita Hradec Králové, CZ*

prof. RNDr. Štěpán Hubálovský, PhD. *Univerzita Hradec Králové, CZ*

prof. Dr. Mirjana Ivanovic *University of Novi Sad, SR*

Ing. Jana Jacková, PhD. *Katolícka univerzita v Ružomberku, SK*

prof. RNDr. Ivan Kalaš, PhD. *Univerzita Komenského, SK*

doc. RNDr. Zuzana Kubincová, PhD. *Univerzita Komenského, SK*

doc. RNDr. Gabriela Lovászová, PhD. *Univerzita Konštantína Filozofa, SK*

Ing. Božena Mannová, PhD. *České vysoké učení technické, CZ*

RNDr. Alžbeta Michalíková, PhD. *Univerzita Mateja Bela, SK*

RNDr. Pavel Pešat, PhD. *Univerzita J. E. Purkyně, CZ*

Prof. Dr. Kate Sanders *Rhode Island College, USA*

prof. Ing. Veronika Stoffová, CSc. *Trnavská univerzita v Trnave, SK*

doc. RNDr. Petr Šaloun, PhD. *Technická univerzita Ostrava, CZ*

doc. RNDr. Ľubomír Šnajder, PhD. *Univerzita Pavla Jozefa Šafárika, SK*

doc. RNDr. Pavel Töpfer, CSc. *Univerzita Karlova, CZ*

doc. Ing. Ľudovít Trajtel', PhD. *Univerzita Mateja Bela, SK*

Dr. Livia Tudor *Petroleum-Gas University, RO*

PaedDr. Patrik Voštinár, PhD. *Univerzita Mateja Bela, SK*

Organizačný výbor konferencie:

PaedDr. Patrik Voštinár, PhD.

Mgr. Adam Dudáš, PhD.

RNDr. Alžbeta Michalíková, PhD.

Katarína Gavalďová

Recenzenti:

doc. RNDr. Gabriela Andrejková, CSc. *Univerzita Pavla Jozefa Šafárika, Košice, SK*

Mgr. Jan Berki, PhD. *Technická Univerzita v Liberci, CZ*

Ing. Jindra Drábková, PhD. *Technická Univerzita v Liberci, CZ*

Mgr. Adam Dudáš, PhD. *Univerzita Mateja Bela, Banská Bystrica, SK*

PaedDr. Ján Guniš, PhD. *Univerzita Pavla Jozefa Šafárika v Košiciach, SK*

Ing. Dana Horváthová, PhD. *Univerzita Mateja Bela, Banská Bystrica, SK*

PaedDr. Roman Hrušecký, PhD. *Univerzita Komenského, Bratislava, SK*

doc. RNDr. Miroslav Iliáš, PhD. *Univerzita Mateja Bela, Banská Bystrica, SK*

Ing. Jana Jacková, PhD. *Katolícka univerzita v Ružomberku, SK*

doc. RNDr. Ľudmila Jašková, PhD., *Univerzita Komenského, Bratislava, SK*

prof. RNDr. Ivan Kalaš, PhD. *Univerzita Komenského, SK*

doc. Mgr. Ján Karabáš, PhD. *Univerzita Mateja Bela, Banská Bystrica, SK*

doc. RNDr. Zuzana Kubincová, PhD., *Univerzita Komenského, Bratislava, SK*

doc. RNDr. Gabriela Lovászová, PhD. *Univerzita Konštantína Filozofa, Nitra, SK*
RNDr. Miroslav Melicherčík, PhD. *Univerzita Mateja Bela, Banská Bystrica, SK*
RNDr. Alžbeta Michalíková, PhD. *Univerzita Mateja Bela, Banská Bystrica, SK*
RNDr. Pavel Pešat, PhD., *Technická Univerzita, Liberec, CZ*
doc. RNDr. Ľubomír Salanci, PhD. *Univerzita Komenského, Bratislava, SK*
PaedDr. Mgr. Vladimír Siládi, PhD. *Univerzita Mateja Bela, Banská Bystrica, SK*
prof. Ing. Veronika Stoffová, CSc. *Trnavská univerzita v Trnave, SK*
doc. Ing. Jarmila Škrinárová, PhD. *Univerzita Mateja Bela, SK*
doc. RNDr. Ľubomír Šnajder, PhD., *Univerzita Pavla Jozefa Šafárika, Košice, SK*
doc. PaedDr. Monika Tomcsányiová, PhD. *Univerzita Komenského, Bratislava, SK*
doc. Ing. Ľudovít Trajtel', PhD., *Univerzita Mateja Bela, Banská Bystrica, SK*
doc. PaedDr. Jiří Vaníček, PhD., *Jihočeská Univerzita, České Budějovice, CZ*
Mgr. Michal Vagač, PhD. *Univerzita Mateja Bela, Banská Bystrica, SK*
PaedDr. Patrik Voštinár, PhD. *Univerzita Mateja Bela, Banská Bystrica, SK*

Editori: Mgr. Adam Dudáš, PhD., RNDr. Alžbeta Michalíková, PhD.,
PaedDr. Patrik Voštinár, PhD., doc. Ing. Jarmila Škrinárová, PhD.
Copyright © 2023 autori
ISBN: 978-80-557-2038-8
EAN: 9788055720388
ISSN: 2454-051X
DOI: <https://doi.org/10.24040/2023.9788055720388>
Elektronický, konferenčný zborník

Predhovor

Stretávame sa po dvoch rokoch a konečne naživo, na pôde Fakulty prírodných vied Univerzity Mateja Bela v Banskej Bystrici na 28. ročníku konferencie DidInfo 2023.

Pred dvomi rokmi sa konferencia kvôli pandémie musela konať online a v minulom roku sa z rovnakých dôvodov nekonala vôbec. O to viac si teraz vážime osobné stretnutia a diskusie, ktoré nám chýbali. Konferencia je už tradične zameraná na vyučovanie informatiky.

Je mi čťou privítať všetkých účastníkov. Špeciálne vítam medzi významné osobnosti, ktoré prijali pozvanie a pripravili inšpiratívne prednášky: doc. PaedDr. Jiřího Vaníčka, PhD. z Pedagogickej fakulty Jihočeskej univerzity v Českých Budejoviciach a prof. Ivana Kalaša z Univerzity Komenského v Bratislave.

Za hlavný cieľ konferencie považujeme rozvoj osobností žiaka a učiteľa v predmete Informatika, ktorý vnímame v kontexte rozvoja algoritmického myslenia, nadobúdania tradičných i nových zručností, ale najmä schopnosti riešiť úlohy.

Pozrime sa na program tohtoročnej konferencie detailnejšie. Pre oblasť vyučovania informatiky na základných školách sa môžeme tešiť na pekné príklady výučby programovania, vzdelávacie aplikácie Smart City s použitím BBC micro:bit a edukačnej robotiky. Ocenenia hodné sú práce týkajúce sa projektov Informatika s Emilom – živý zošit, Emil správca a Spievajme so Scratchom. Inšpirácie pre vyučovanie informatiky na stredných školách nájdeme napr. vo výučbe logických funkcií pomocou Minecraft Education, programovaním s Lego Lab EV3.

Rovnako zaujímavé sú príspevky, týkajúce sa vyučovania informačných systémov, počítačovej bezpečnosti, virtuálnej učebne, edukačnej robotiky, Industry 4.0, programovania železnice s vláčikom a projektom Erasmus s názvom Fertile. Veríme, že Vás zaujmú výsledky projektu „Algoritmické myslenie v predmete Informatika v sekundárnom vzdelávaní“. Kolegovia z českej republiky nám predstavia nové rámcové vzdelávacie programy a učebnice informatiky.

Súťaže majú svoje nezastupiteľné miesto, kde vzájomné porovnávanie riešení s inými žiakmi pôsobí motivačne. Mimoriadne dôležité sú analýzy súťažných úloh z rôznych pohľadov. Veľká pozornosť je venovaná súťažiam iBobor, Bobřík, Scratch Cup, IT v Nitre, resp. iBobor pre študentov so zrakovým znevýhodnením.

Tradičným lákadlom konferencie sú workshopy. Tento rok sme pripravili 4 workshopy, ktoré sú zamerané na Umelá inteligencia v školskej informatike - Kreatívne multimediálne nástroje a CoSpaces - prostredie na vytvorenie AR a VR aplikácií počas výučby na základných a stredných školách, Umelá inteligencia v školskej informatike – Programátorské nástroje a Spoločnosť EDUXE - LEGO Education SPIKE Prime pre 2. stupeň základných a stredných škôl.

Želáme konferencii naplnenie stanovených cieľov, aktívnym účastníkom úspech a množstvo inšpirácie do ďalšieho roka!

Marec 2023

doc. Ing. Jarmila Škrinárová, PhD.

Predsedyňa programového výboru
konferencie DidInfo 2023

Obsah

Pozvané prednášky

Jiří Vaníček	
Programovací bobří úlohy s tvorbou programu pomocí bloků.....	9
Jarmila Škrinářová	
Algoritmické myslenie v predmete Informatika v sekundárnom vzdelávaní.....	11
Ivan Kalaš	
Dobré správy pre informatiku.....	14

Príspevky

Peter Agh	
Koncept a princípy nových učebníc Informatiky pre 4.-5.ročník podľa nových Rámcových vzdelávacích programov v Českej republike.....	16
Jan Bezděka, Zbyněk Filipi	
Podpora výuky logických funkcí pomocí prostředí Minecraft Education.....	22
Andrej Blaho, Ivan Kalaš, Milan Moravčík	
Živý zošit: Informatika a informatické základy digitálnej gramotnosti na 1. stupni.....	26
Katarína Brinziková, Ľubomír Šnajder, Ján Guniš	
BBC micro:bit SMART CITY v projektovom vyučovaní alebo „o krok bližšie k žiakom“.....	32
Mária Čujdíková, Ľudmila Jašková, Mária Stankovičová	
Súťaž iBobor - analýza obťažnosti úloh určených pre nevidiacich študentov stredných škôl.....	40
Mária Čujdíková, Ivan Kalaš	
Tabuľky: Ako ich žiaci interpretujú v kontexte prostredia Emil správca.....	46
Mária Čujdíková, Mária Stankovičová	
Rozvoj informatických kompetencií u žiakov so zrakovým postihnutím formou Letnej počítačovej školy.....	54
Václav Dobiáš	
Výsledky sociálně znevýhodněných žáků v soutěži Bobřík informatiky 2022.....	59
Jindra Drábková	
Kurz Data, informace a modelování pro učitele informatiky na gymnáziu.....	64
Filip Frank, Tomáš Jakeš	
Transformace úloh z blokového prostředí Lego Lab EV3 do blokového prostředí Scratch.....	69

Ján Guniš, Ľubomír Šnajder, Csaba Török, Ľubomír Antoni	
Kompetencie učiteľov informatiky v oblasti programovania.....	74
Petr Hanzal	
Subjektívny vnímanie obtížnosti programovacích úloh v závislosti na výuce programovania.....	80
Roman Hrušecký, Ivan Kalaš, Klára Dvorská	
Bezpečnosť v digitálnom veku – koncepcia vzdelávania na ZŠ a SŠ.....	85
Jakub Krcho, Karolína Miková	
Hodnotenie žiakov v edukačnej robotike – výsledky prvej etapy výskumu.....	90
Zuzana Kubincová, Karolína Miková	
Preferencie pedagógov pre personalizované vyučovanie informatiky – prieskum.....	96
Daniel Lessner, Martin Prade	
Informační systémy v českém kurikulu informatiky.....	102
Gabriela Lovászová, Viera Michaličková	
IT v Nitre: Súťaž v programovaní pre stredoškóľakov.....	108
Miroslav Melicherčík, Vladimír Siládi, Michal Povinský	
Propedeutické využitie digitálnej modelovej železnice k programovaniu mikrokontroléra Arduino.....	112
Karolína Miková, Zuzana Kubincová	
Aplikovanie vyvíjanej metodiky na rozvoj informatického myslenia v kombinácii s divadlom - prvotné výsledky.....	118
Karolína Miková, Barbora Stenová	
Ako posúdiť vhodnosť robotickej stavebnice - LEGO SPIKE Prime.....	124
Ľubomír Salanci	
Analýza kurzu programovania pre matematikov.....	129
Mária Stankovičová	
Porozumenie zadaní úloh z programovania s grafickým výstupom nevidiacim študentom.....	134
Veronika Stoffová	
Dištančné vyučovanie prakticky orientovaných predmetov.....	136
Václav Šimandl, Jan Pršala	
Úspěšnost žáků při řešení informatických úloh v průběhu školního dne.....	142
Ľubomír Šnajder, Katarína Brinziková, Ján Guniš	
Aktivita EU Code Week Spievajme so Scratchom.....	148
Monika Tomcsányiová	
Informatické koncepty pre žiakov nižšieho sekundárneho vzdelávania so záujmom o programovanie.....	155

Zdeněk Ulrych, Miroslav Malaga	
Vzdělávání budoucích strojních a průmyslových inženýru v oblastech důležitých pro Industry 4.0.....	160
Jiří Vaníček, Karolína Miková	
Sady bobřích úloh se sestavováním programu z bloků.....	166
Patrik Voštinár, Róbert Címer	
IT activities for marginalized children.....	172
Viola Vrbová, Zbyněk Filipi	
Deskové hry jako prostředek pro rozvoj inforatického myšlení studentů učitelství informatiky.....	177
Miroslav Zíka	
Virtuální třída jako nástroj na zdokonalování didaktických dovedností studentů učitelských programů.....	182

Workshopy

Zuzana Tkáčová	
Umelá inteligencia v školskej informatike - Kreatívne multimedialne nástroje.....	188
Patrik Voštinár	
CoSpaces - prostredie na vytvorenie AR a VR aplikácií počas výučby na ZŠ a SŠ.....	188
Spoločnosť EDUXE	
LEGO Education SPIKE Prime pre 2. stupeň základných a stredných škôl.....	188
Zuzana Tkáčová	
Umelá inteligencia v školskej informatike – Programátorské nástroje.....	188

Sponzori



POZVANÉ PREDNÁŠKY

Programovací bobří úlohy s tvorbou programu pomocí bloků

Programming bebras tasks with creating programming code using blocks

Jiří Vaniček

Jihočeská univerzita v Českých Budějovicích,

Pedagogická fakulta

Jeronýmova 10

37115 České Budějovice

Česko

vanicek@pf.jcu.cz



ABSTRACT

The lecture summarizes the experience with the development of bebras algorithmic tasks, the solution of which is based on assembling a program from blocks. A module enabling the implementation of such tasks was developed and verified and templates for four microworlds forming frameworks for programming tasks were created. This module allows to assemble the program, run it and check correctness of the solution and also communicate with contest application. Based on these templates, more than 100 contest tasks, which were included in the Bebras Challenge in the Czech republic in 2021 and 2022 or the preparation for the contest and which are also applicable in lessons as part of the new computer science curriculum, were created. Additional software module which saves data from each step of students' process of creating the program as a solution of a task allow us to analyze students approach to learning basics of programming and search which misconceptions impede them to understand basic programming concepts.

Keywords

Programming. Algorithmization. Block oriented environment. Bebras challenge. Microworld.

ROZŠÍŘENÝ ABSTRAKT

Školní soutěž Bebras Challenge, Bobřík informatiky se svojí dlouhou tradicí (v ČR od roku 2008) a velkým množstvím účastníků (v roce 2022 v ČR více než 183 000, celosvětově 2,8 milionu) měl a má velký dopad na formování představ žáků a učitelů, co je nová informatika. Bebras je platformou, která celosvětově přináší množství nových informatických úloh a formuje tak školní kurikulum. V úlohách se při online testu žáci ocitají v situaci, v níž mají najít příslušný informatický koncept a uplatněním informatického myšlení určit odpověď. Protože soutěž

probíhá formou online testu, jsou odpovědi realizovány výběrem jedné z nabízených možností, označováním objektů na ploše nebo manipulací s objekty po ploše.

Vhodné bobří úlohy mohou rozvíjet algoritmicizaci. Typickými algoritmicizacími úlohami, které se v soutěži objevují, jsou čtení zapsaného algoritmu a posouzení, co vykoná, hledání počátečního či koncového stavu po vykonání daného algoritmu, porovnání více algoritmů oproti zadání úlohy, posouzení pravidel k provedení výpočtu, hledání chyby v algoritmu nebo jeho optimalizace. Mezi soutěžními úlohami především z technických důvodů doposud chyběly úlohy, na něž se odpovídá sestavením programu, což v soutěži nabídka algoritmicizací úloh ochuzovalo.

Hledali jsme řešení, které by umožnilo do stávajícího systému soutěže implementovat úlohy, v nichž by soutěžící stejně jako v programovacích prostředích používaných na školách, jako je Scratch, Makecode, Hour of code a další mohli sestavovat svůj program z bloků. Naším cílem bylo inovovat soutěž Bobřík informatiky typem takovýchto úloh a tím zvýšit pestrost a atraktivitu soutěžních testů.

Na základě analýzy dostupných blokově orientovaných prostředí pro výuku programování jsme vytvořili softwarový modul, který do testové aplikace integruje komponenty Blockly pro sestavování programu z bloků, realizuje spuštění vytvořeného programu a jeho interpretaci ve vytvořené šabloně nějaké mikrosvětě a též zajišťuje komunikaci se softwarem soutěžního testu ke kontrole správnosti programu a ukládání do soutěžní databáze. Pokaždé, když žák stiskem tlačítka Spustit požádá o vykonání sestaveného programu, systém zároveň uloží žákův program a též informace o tom, zda řešení vyhovelo všem požadavkům zadání, zda žák vytvořil správný program.

Pro tento modul jsme vytvořili čtyři šablony mikrosvětů, které v administrátorské části modulu umožňují autorům snadno vytvářet nové konkrétní úlohy a jejich varianty bez potřeby je programovat. Dva z těchto mikrosvětů jsou tradiční, s řadou historií prověřených úloh, a dva relativně nové, s novými možnostmi. Tyto mikrosvěty stručně popíšeme:

Robot Karel – program posouvá postavu ve čtvercovém poli, otáčí jí oběma směry o pravý úhel, detekuje objekty na aktuálním poli, přidává a odebírá tyto předměty a detekuje překážky. K základním příkazům mikrosvěta jsou doplněny bloky pro opakování (pevný počet, s podmínkou) a rozhodování. Typickou úlohou je dojít na místo, vyhnout se překážkám, sesbírat předměty apod.

Želví grafika – program posouvá postavu na ploše a otáčí ji o daný úhel, případně kreslí stopu jejího pohybu. Protože program nic nedetekuje, k základním příkazům mikrosvěta je doplněn pouze cyklus s pevným počtem opakování. Typickou úlohou je nakreslit geometrický obrazec podle vzoru.

Film – program animuje pohyb postavy, která mění svoji polohu a velikost v čase. Mikrosvět má příkaz Postava, který vykreslí objekt na hrací plochu na místě daném parametry pozice x, y, velikostí postavy a otočení postavy oproti základnímu směru. Tyto parametry je možno popsat výrazem, který obsahuje proměnnou čas. Animace je realizována tak, že po spuštění programu se hodnota proměnné čas plynule mění od 0 do 100 a pro každou z těchto hodnot se vykoná příkaz Postava. Tento příkaz je doplněn blokem pro vytváření matematických výrazů se základními početními úkony a blokem Když, umožňujícím kontrolu podmínky času (např. zda čas překročil určitou hodnotu) a tím umožňující skládání jednoduchých pohybů do složitějších. Typickou úlohou je vykonat animaci podle vzoru.

Proměnné – tento mikrosvět postrádá postavu a grafický výstup, programuje se tu výpis číselných hodnot, obdobný tisku na list papíru. Používají se číselné proměnné a základními bloky jsou přiřazení hodnoty do proměnné, změna její hodnoty, blok pro samotnou hodnotu proměnné, pro vypsání hodnoty proměnné do jednoho řádku a také blok pro číselnou konstantu. K základním blokům jsou přidány bloky se strukturami Opakuj krát, Opakuj dokud, Když-tak-jinak a dále bloky s podmínkami porovnávacími číselné hodnoty a se základními početními operacemi. Typickou úlohou je vypsát řadu čísel, zadanou výpisem nebo jejím popisem.

Ke každému z těchto mikrosvětů bylo vytvořeno více než 15 úloh, které byly použity v ostré soutěži nebo při přípravě na ni. Protože soutěžní software vytváří test jako sadu úloh, umožnil nám vytvářet sady navazujících úloh jednoho mikrosvěta (obdobných sadám úloh z Hodiny kódu). Pro každý mikrosvět byla vytvořena jedna sada o více než 10 úlohách, které posloužily k seznámení žáků s daným mikrosvěttem před vlastní soutěží a poskytly též učitelům náplň do hodin informatiky.

Vyvinutý modul, šablony 4 mikrosvětů a více než 100 úloh byly ověřeny na více než 1300 českých školách během posledních dvou ročníků soutěže v národním a postupovém ústředním kole a také ve formě přípravných sad úloh. Celkem si test s alespoň dvěma programovacími úlohami zahrálo více než 420 000 žáků.

Pro potřebu zkoumání, jak žákům řešení těchto úloh jde, jsme vyvinuli ještě zpětnovazební modul, který při každém požadavku na spuštění sestaveného programu odešle na server údaje o soutěžícím, úloze a čase, zápis sestaveného programu z bloků a aktuální konstelaci objektů na ploše. To umožnilo při daném množství účastníků nasbírat velké množství dat, které jsme využili k analýze, jakým způsobem začátečníci programují, jak si poradí

s konceptem opakování a které miskoncepty jim brání řešit úlohy a jak tyto miskoncepty žáci překonávají.

Klíčová slova

Programování. Algoritmizace. Blokově orientované prostředí. Bobřík informatiky. Mikrosvět.

Algoritmické myslenie v predmete Informatika v sekundárnom vzdelávaní

Algorithmic thinking in the Informatics subject in secondary education

Jarmila Škrinárová
Univerzita Mateja Bela
v Banskej Bystrici
Fakulta prírodných vied,
Katedra informatiky
Tajovského 40,
97401 Banská Bystrica
Slovensko

jarmila.skrinarova@umb.sk

Gabriela Lovászová
Univerzita Konštantína Filozofa
v Nitre
Fakulta prírodných vied,
Katedra informatiky
Tr. A. Hlinku 1,
949 01 Nitra
Slovensko

glovaszova@ukf.sk

Jana Jacková
Katolícka Univerzita v
Ružomberku
Pedagogická fakulta,
Katedra informatiky
Hrabovská cesta 1
034 01, Ružomberok
Slovensko

jana.jackova@ku.sk

Adam Dudáš
Univerzita Mateja Bela
v Banskej Bystrici
Fakulta prírodných vied,
Katedra informatiky
Tajovského 40,
97401 Banská Bystrica
Slovensko

adam.dudas@umb.sk

ABSTRACT

In this lecture, we present the results of the KEGA project titled Implementation of new trends in informatics in the teaching of algorithmic thinking and programming in the subject of informatics in secondary education. The outputs of the project are intended for current and future teachers of informatics and are available on the website entitled "Algorithmic thinking in the subject of informatics in secondary education". The page is available at <https://lms.umb.sk/course/view.php?id=3839>, it mainly contains project outputs intended for computer science teachers.

Project outputs: analysis of available methodologies, methodical and teaching materials, new teaching subjects, publications, events: workshops, seminars and competitions, thesis, feedback and collaboration.

Keywords

Programming. Algorithmic thinking. Future teachers of informatics. Artificial intelligence for kids. Educational robotics. Educational games.

ROZŠÍRENÝ ABSTRAKT

V tejto prednáške prezentujeme výsledky projektu KEGA s názvom Implementácia nových trendov v informatike do výučby algoritmického myslenia a programovania v predmete informatika v sekundárnom vzdelávaní. Výstupy projektu sú určené pre súčasných aj budúcich učiteľov informatiky a sú dostupné na webovej stránke.

Súčasťou informatického vzdelávania je nadobúdanie kritických zručností, potrebných pre 21. storočie. Kľúčové je získanie a rozvoj algoritmického a výpočtového myslenia [2 - 5]. Ďalším cieľom je rozvoj zručností v informačných a komunikačných technológiách. Preto je dôležité poskytovať kvalitné vzdelávanie v informatike na základných, stredných, ale aj materských školách. Zatiaľ čo počet škôl, ktoré ponúkajú kvalitné informatické vzdelávanie sa zvyšuje, stále existujú rozdiely v prístupe a v skúsenostiach s kvalitným informatickým vzdelávaním. Preto autori Lee et al. [1] odporúčajú skorý prístup, rôzne inovatívne prístupy a vhodné poradenstvo pre učenie sa študentov k informatickému vzdelávaniu. Tiež je potrebné zvýšiť a zlepšiť možnosti profesionálneho rozvoja pre učiteľov informatiky.

Uvedená motivácia viedla tri slovenské pracoviská k príprave a realizácii projektu riešiaceho implementáciu nových trendov v informatike do výučby algoritmického myslenia a programovania v predmete informatika v sekundárnom vzdelávaní. Ide o pracoviská, ktoré pripravujú budúcich učiteľov informatiky - Katedra informatiky FPV Univerzity Mateja Bela v Banskej Bystrici, Katedra informatiky FPV Univerzity Konštantína Filozofa v Nitre a Katedra informatiky, PdF Katolícka Univerzita v Ružomberku. Projekt sme realizovali v rokoch 2020 až 2022.

Globálny cieľ sme rozpracovali do čiastkových cieľov projektu: vytvoriť a overiť inovatívne metodiky a učebné materiály pre výučbu vybraných tematických celkov v predmete informatika na základných a stredných školách.

Výstupy projektu sú určené pre súčasných aj budúcich učiteľov informatiky a sú dostupné na webovej stránke s názvom „Algoritmické myslenie v predmete Informatika v sekundárnom vzdelávaní“. Stránka je dostupná na <https://lms.umb.sk/course/view.php?id=3839>, obsahuje najmä výstupy projektu, ktoré sú určené učiteľom informatiky.

Výstupy projektu:

- Analýza dostupných metodík.
- Metodické a učebné materiály.
- Zavedené nové predmety.
- Publikácie.
- Organizácia konferencie DIDINFO.
- Podujatia: workshopy, semináre a súťaže.
- Záverečné práce.
- Spätná väzba a spolupráca.

V analýze dostupných metodík sme sa zamerali na prehľad knižných a online metodických materiálov zameraných na algoritmické myslenie [8] a programovanie resp. k témam umelá inteligencia, robotika [6, 7], mobilné, webové a cloudové technológie.

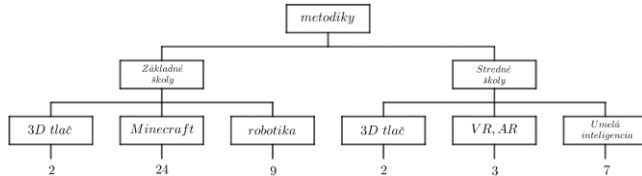
Zamerali sme sa na dostupnosť metodík s použitím MIT App Inventor 2 [9], BBC micro:bit, Minecraft: Education Edition a umelú inteligenciu pre deti.

1 Metodické a učebné materiály

Vytvorili sme 35 metodík a komplexných metodických materiálov, ktoré obsahujú niekoľko čiastkových overených metodík pre

základné školy a 12 metodík a komplexných, overených metodických učebných materiálov pre stredné školy.

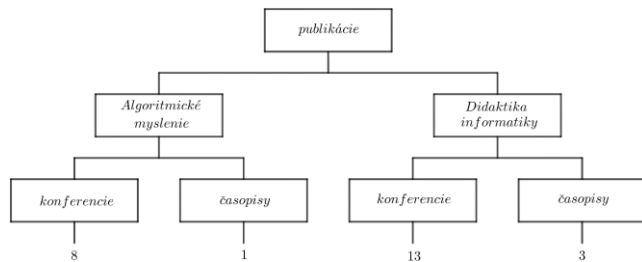
Z obsahového hľadiska sme sa zamerali na témy metodík, ktoré riešia 3D tlač, Minecraft Education Edition, edukačnú robotiku, virtuálnu a rozšírenú realitu a umelú inteligenciu pre deti. Tematické rozdelenie s uvedenými počtami metodík je zobrazené na obrázku 1.



Obrázok 1: Tematické členenie metodík s uvedenými počtami

2 Publikácie

Vedecké a odborné práce, vytvorené v rámci projektu, boli publikované prevažne na zahraničných konferenciách a v 4 indexovaných časopisoch s kvartilmi Q1 alebo Q2. Spolu ide o 26 publikácií, z ktorých je aktuálne 20 (a ďalších 5 bude) evidovaných v databázach WoS alebo Scopus, pozri obrázok 2.



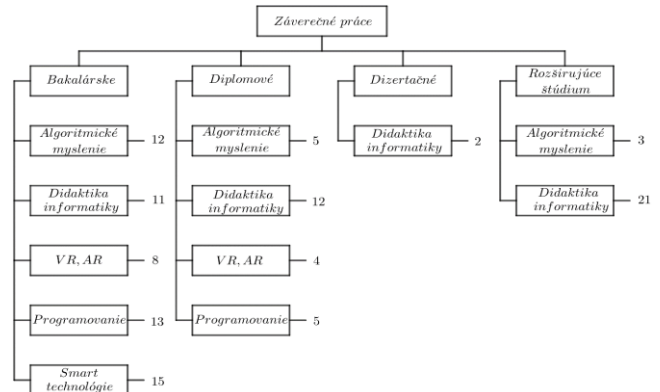
Obrázok 2: Členenie publikácií s uvedenými počtami

3 Podujatia: workshopy, semináre a súťaže

Kurzy, konferencia a podujatia, vytvorené počas riešenia projektu, boli zamerané na nové moderné témy, ktoré rozvíjajú algoritmické a výpočtové myslenia, zatriktívňujú vyučovanie informatiky a programovanie zvlášť.

4 Záverečné práce a konferencie

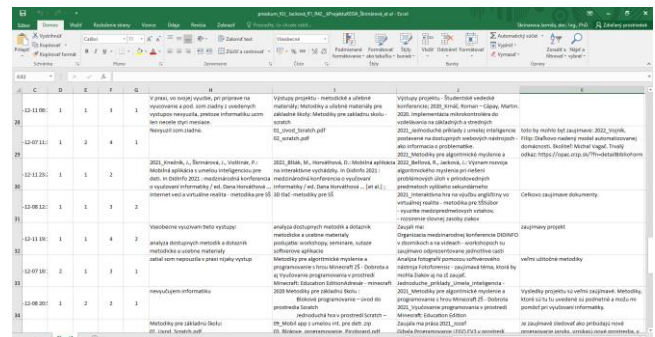
Pod vedením členov riešiteľského kolektívu projektu bolo úspešne obhájených spolu 111 záverečných prác. Z toho 2 dizertačné práce z teórie vyučovania informatiky, 26 diplomových prác, 59 bakalárskych prác z Učiteľstva informatiky, alebo Aplikovanej informatiky a 24 záverečných prác Rozširujúceho štúdia učiteľstva informatiky. Tematické členenie záverečných prác ukazuje obrázok 3.



Obrázok 3: Tematické členenie a počty záverečných prác

5 Spätná väzba a spolupráca so školami

Vzhľadom k tomu, že sú uvedené výstupy projektu adresované a prístupné učiteľom informatiky, zverejnili sme informácie o projekte a učiteľia, resp. študenti rozširujúceho štúdia informatiky z KU v Ružomberku nám poskytli základnú spätnú väzbu k uvedeným materiálom, pozri obrázok 4.



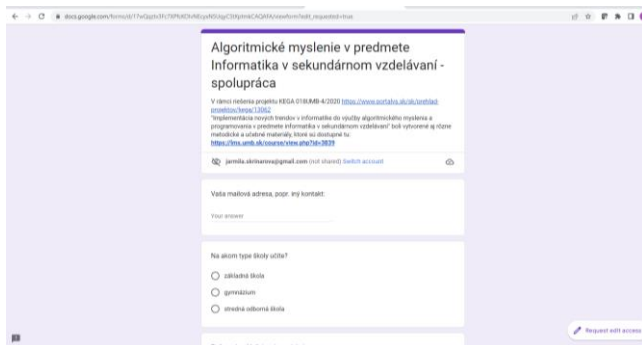
Obrázok 4: Prieskum o materiáloch projektu

Posmelení kladnou spätnou väzbou sme sa rozhodli ponúknuť učiteľom spoluprácu. S učiteľmi, ktorí prejavia o implementovanie konkrétnych metodík do svojho vyučovania môžeme koordinovať a zosynchronizovať postupy a spolu s nimi navrhnúť spoločnú špecifikáciu kvalitatívnych aj merateľných cieľov a rovnako aj rovnaký spôsob vyhodnocovania riešení. Rovnaké overovania na rôznych školách pomôžu nielen pri porovnávaní sa, ale máme za to, že spolupráca prispeje k akcelerácii rozvoja osobností žiakov a učiteľov.

Preto sme pripravili Google formulár, v ktorom vás žiadame o rovnakú spätnú väzbu ako je uvedená na obrázku 5.

Odkaz na Google formulár nájdete na konci stránky projektu, alebo na:

https://docs.google.com/forms/d/17wQqzt3F7XPfcKOiVNEcyN5UqyCStXptmkCAQFA/viewform?edit_requested=true



Obrázok 5: Prieskum o materiáloch projektu vo forme Google formulára

Záver

Cieľom tejto prednášky je prezentovať výstupy projektu, ktoré veríme prispejú k skvalitneniu vyučovania informatiky na školách. Preto sú všetky výstupy projektu dostupné, pre učiteľov informatiky, na webovej stránke projektu: <https://lms.umb.sk/course/view.php?id=3839>.

Rôznorodé metódy sú orientované na rozvoj algoritmickeho myslenia. Algoritmicke a výpočtové myslenie je dôležité nielen pre riešenie informatických úloh, ale je aplikovateľné v bežnom živote. Z nášho pohľadu sú užitočné metódy a aplikácie, vhodné na získavanie poznatkov, pomocou programovania s pripravenými úlohami, najmä pre oblasť umelej inteligencie, robotiky, IoT, virtuálnej a rozšírenej reality. Očakávaný prínos vnímame v možnom rýchlym a priamom prenose nových tém, poznatkov a metód do vyučovania informatiky na ZŠ a SŠ cez absolventov učiteľského štúdia a v ich tvorivom prístupe, ako to môžeme vidieť na kvalitných záverečných prácach. Veríme v užitočnosť výsledkov projektu pre učiteľov z praxe, ktorí absolvovali workshopy a kurzy. Predpokladáme ďalšiu spoluprácu s učiteľmi z praxe.

Kľúčové slová

Programovanie. Algoritmicke myslenie. Budúci učelia informatiky. Umelá inteligencia pre deti. Edukačná robotika. Edukačné hry.

BIBLIOGRAFICKÉ ODKAZY

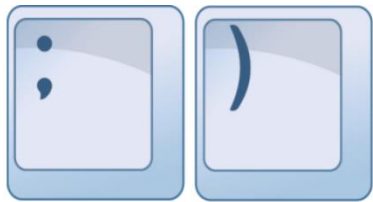
- [1] Lee, S. J., Francom, G. M., Nuatomue, J., Computer science education and K-12 students' computational thinking: A systematic review. In *International Journal of Educational Research*. Volume 114, 2022, 102008, ISSN 0883-0355. <https://doi.org/10.1016/j.ijer.2022.102008>.
- [2] Chen, K. Z., Chi, H.H. Novice young board-game players' experience about computational thinking. In *Interactive Learning Environments*. <https://doi.org/10.1080/10494820.2020.1722712>
- [3] Chevalier, M., Giang, C., Piatti, A., & Mondada, F. Fostering computational thinking through educational robotics: a model for creative computational problem solving. In *International Journal of STEM Education*, 7(39). <https://doi.org/10.1186/s40594-020-00238-z>
- [4] Chiazzese, G., Fulantelli, G., Pipitone, V., Taibi, D. Engaging primary school children in computational thinking: Designing and developing videogames. In *Education in the Knowledge Society*, 19(2), 63–81.
- [5] Chou, P. Using ScratchJr to foster young children's computational thinking competence: A case study in a third-grade computer class. In *Journal of Educational Computing Research*, 58(3), 570–595.
- [6] Chiazzese, G., Arrigo, M., Chifari, A., Lonati, V., & Tosto, C. Educational robotics in primary school: Measuring the development of computational thinking skills with the Bebras tasks. In *Informatics*, 6(4), 43. <https://doi.org/10.3390/informatics6040043>
- [7] Çınar, M., & Tüzün, H. Comparison of object oriented and robot programming activities: The effects of programming modality on student achievement, abstraction, problem solving, and motivation. In *Journal of Computer Assisted Learning*, 37(2), 370–386.
- [8] Antoni, L. et al. Attribute Exploration in Formal Concept Analysis and Measuring of Pupils' Computational Thinking. In: Paralič, J., Sinčák, P., Hartono, P., Mařík, V. (eds) *Towards Digital Intelligence Society. DISA 2020*. In *Advances in Intelligent Systems and Computing*, vol 1281. Springer, Cham. https://doi.org/10.1007/978-3-030-63872-6_8
- [9] ŠNAJDER, L. *Výučba programovania mobilných aplikácií v App Inventor 2*. [online]. Dostupné na: <http://ics.upjs.sk/~snajder/ai2/>

Dobré správy pre informatiku

Good news for computer science

Ivan Kalaš
KDMFI FMFI
Univerzita Komenského v Bratislave
Slovensko
ivan.kalas@fmph.uniba.sk

Pozvaná prednáška bola prezentovaná bez uvedenia rozšíreného abstraktu.



PRÍSPEVKY

Koncept a princípy nových učebníc Informatiky pre 4. - 5. ročník podľa nových Rámcových vzdelávacích programov v Českej republike

Concepts and Principles of new Informatics textbooks for 4th-5th grade supporting new Framework Education Programmes in Czech Republic

Peter Agh
nezávislý konzultant
Česká republika
peter.agh.nz@gmail.com

ABSTRACT

In this paper, we describe new textbooks published in Czech Republic. Textbooks are dedicated for 4th and 5th grade. Textbooks covers three of four themes of new framework education programmes. Topics include: data and their representation (information coding), structuring and analyzing data, modelling, information systems and understanding of digital technologies. It also supports basics of security (safe vs. risk behaviour) and precepts of artificial intelligence understanding. Textbooks contains a minimum of theory, they focus on activities and teach and develop precepts through experiences. It can be combined with any programming or robotics course.

Keywords

Computational thinking. New framework education programmes in Czech Republic. Teaching informatics for 4th and 5th grade. New textbooks.

ABSTRAKT

V tomto článku popisujeme nové učebnice informatiky vydané v Českej republike. Učebnice sú určené pre štvrtú a piatu triedu základnej školy. Učebnice pokrývajú tri zo štyroch tematických okruhov nových Rámcových vzdelávacích programov. Zahŕňajú témy: dáta, informácie a ich reprezentácia (kódovanie informácie), štruktúrované dáta a vyvodzovanie z dát, modelovanie, informačné systémy a pochopenie princípov digitálnych technológií. Tiež podporujú základy bezpečnosti (bezpečného chovania) a budovanie prekonceptov pre pochopenie umelej inteligencie. Učebnice obsahujú minimum výkladu, učia a budujú prekoncepty získavaním skúseností pomocou rozmanitých aktivít.

Kľúčové slová

Informatické myslenie. Nové rámcové vzdelávacie programy v Českej republike. Nové učebnice pre štvrtú a piatu triedu základnej školy.

1 ÚVOD

Od septembra 2021 v Českej republike platia pre výuku informatiky nové rámcové vzdelávacie programy (RVP; obdoba slovenských ŠVP), pozri [1,2]. Revidované programy do značnej miery menia doterajšiu náplň výuky informatiky. Pred revíziou sa učitelia zameriavali hlavne na používateľskú perspektívu -

vyučovanie obsluhy počítača, kancelárskeho softvéru apod. Učitelia v rôznej miere vyučovali programovanie. V novom RVP sa z informatiky vyčlenili tzv. *digitálne kompetencie* (kam spadá napríklad tvorba digitálneho obsahu, vyhľadávanie informácií a ďalšie) a presunuli sa medzi tzv. *kľúčové kompetencie*. Kľúčové kompetencie pritom nie sú alokované na konkrétny predmet, digitálne kompetencie sa teda môžu (a majú) rozvíjať v rámci rôznych predmetov.

Nová náplň predmetu informatika sa sústreďuje na rozvoj *informatického myslenia* a porozumenie princípom *digitálnych technológií*. Na štvrtú a piatu triedu sa v novom RVP vzťahujú tieto tematické okruhy, ktorých zjednodušený popis tu uvádzame (podrobný popis je možné nájsť napríklad v [1]):

- *Dáta, informácie a modelovanie*, kde si žiaci vyskúšajú zber a záznam dát, ako aj reprezentáciu (kódovanie) rôznych foriem a druhov dát (texty, obrázky a ďalšie).
- *Algoritmizácia a programovanie*, kde sa žiaci učia algoritmizácii a skúsia si základy programovania v niektorom z blokových programovacích jazykov. Niektoré, zložitejšie koncepty, napríklad podmienky sa učia až na druhom stupni (v ČR od 6. triedy).
- *Informačné systémy*, kde si žiaci na prvom stupni skúšajú rozlišovať (vnímať a popisovať) čiastkové časti v rozmanitých systémoch, ktoré na prvý pohľad pôsobia ako jeden celok. Ďalšou témou tohto okruhu je práca so štruktúrovanými dátami, kde namiesto ad-hoc zápisu dát žiaci dáta štruktúrujú do tabuliek, zoznamov apod.
- *Digitálne technológie*, ktorá sa venuje digitálnym zariadeniam, internetu práci v zdieľanom prostredí ako aj vybraným základom bezpečnosti. Aj keď RVP to explicitne neuvádza, je možné rozlišovať dve dimenzie: praktické zručnosti (napríklad ovládanie počítača či iného digitálneho zariadenia) na jednej strane a pochopenie (napríklad fungovania internetu) na strane druhej.

Okrem očakávaných výstupov RVP sa k jednotlivým tematickým okruhom uviedlo aj učivo, ktoré výstupy ďalej RVP konkretizuje (pozri [1]). Ďalšiu predstavu o to, ako RVP naplniť, dávajú učebnice vytvorené v rámci projektu PRIM (pozri ďalej) či modelové ŠVP (Školní vzdělávací programy) pozri [3].

Nový koncept informatiky považujeme za inovačný vo viacerých smeroch, pričom pre potreby tohto článku zmienime tieto:

- Zmena obsahového zamerania predmetu informatika – miesto dôrazu na digitálne kompetencie sa nové RVP zameriava na informatické myslenie a porozumenie princípom digitálnych technológií.
- Témy ako modelovanie, kódovanie informácie či práce s dátami sú ako v RVP uvedené explicitne. Taktiež, vo vzorových učebniciach (pozri nižšie) sú učené samostatne. Nepredpokladá sa, že tieto témy sú pokryté programovaním. Z pohľadu RVP sú to plnohodnotné témy, ktoré prinášajú ďalšie konceptuálne nástroje, pomáhajúce informačné a digitálne technológie uchopiť a efektívnejšie s nimi pracovať.
- 1. stupeň (v ČR aj 5. ročník) nepokrýva dané témy komplexne. Žiaci prostredníctvom hier, zážitkových aktivít a ďalších cvičení (ktoré môžu prebiehať aj bez počítača) budujú prekoncepty, prípravu, ktorá sa rozvíja v ďalších stupňoch. Príkladom je tematický okruh informačných systémov, kde sa prirodzene nečaká, že desaťročné dieťa bude programovať informačný systém, ale už v tomto veku sa môže učiť štruktúrovať dáta, vnímať časti systému a ich vzťahy apod.

Na podporu výučby novej informatiky prebehlo -a prebieha- viacero projektov a aktivít. V kontextu tohto článku zmienime projekt PRIM (pozri napr. [4]). Jedným z jeho výstupov sú vzorové učebnice, voľne prístupné online na odkazovanej stránke. Pre 1. stupeň (4. a 5. triedu) je okrem učebníc programovania a robotiky k dispozícii aj učebnica Základy informatiky pro 1. stupeň ZŠ autorov Berki a Drábková [5] či učebnica Práce s Daty autorov Filippi, Mainz a Fadrhonc [6].

V máji a júni 2022 boli vydané ďalšie učebnice [8,9,10]:

- *Informatika 1. diel*
- *Informatika 2. diel*
- *Hľadá sa Puffy* (pracovný zošit)

Tieto sú predmetom tohto príspevku.

2 PREDSTAVENIE UČEBNÍČ

Zmienené učebnice sú určené pre tieto vekové skupiny:

- **Informatika 1.diel**, doporučená pre 4. ročník
- **Informatika 2. diel**, doporučená pre 5. ročník
- **Hľadá sa Puffy** (pracovný zošit), doporučená pre 5. ročník



Obrázok 1: Nové učebnice

Učebnice tvoria alternatívu k zmieneným dvom učebniciam projektu PRIM pre 1.stupeň a rozširujú tak podporu novej informatiky pre učiteľa.

Obidva diely učebnice *Informatika* a pracovný zošit *Hľadá se Puffy* pokrývajú tieto témy a učivo (uvádzame skrátené názvy):

- *Kódovanie, šifrovanie a prenos dát*
- *Modelovanie*
- *Dáta a informácie*
- *Štruktúrované dáta* (okruh Informačné systémy)
- *Systémy okolo nás* (okruh Informačné systémy)
- *Digitálne technológie* (porozumenie princípom, nie nácviku ovládania)

Pričom uvedené témy spadajú do troch zo štyroch tematických okruhov nového RVP (Dáta, informácie a modelovanie, Informačné systémy a Digitálne technológie), pričom v oblasti Digitálnych technológií sa venujú rovne porozumenia, nie nácviku ovládania.

Škola si pre naplnenie RVP k tomu musí doplniť:

- Programovanie či robotiku. Učiteľ má už aj pre 1.stupeň k dispozícii viacero (aj voľne dostupných) kurzov. Učebnice čiastočne podporujú tému algoritmickej pomoci unplugged cvičení, ktoré môže učiteľ použiť na prípravu či doplnenie (sprestrenie) svojho kurzu programovania či robotiky.
- Nácvik praktickej práce s počítačom či digitálnym zariadením (napr. prihlásenie, prepínanie aplikácii, ale aj práca v zdieľanom prostredí a ďalšie) .

Nad rámec RVP sa učebnice venujú Základom bezpečnosti. Budujú základy pre neskoršie pochopenie princípov umelej inteligencie.

K samotným učebniciam je navyše k dispozícii aj niekoľko pracovných listov, ktoré podporujú rozvoj vybraných digitálnych kompetencií (najmä tvorbu digitálneho obsahu, napríklad textu či prezentácie). Cieľom nie je toto téma pokryť, ale učiteľa podporiť.

Učebnice vyšli v máji až júni 2022 v nakladateľstve Fraus [7]. Učebnice majú platnú doložku Ministerstva školství, mládeže a tělovýchovy České republiky.

3 UKÁŽKY Z UČEBNÍČ

V tejto kapitole uvedieme niekoľko ukážok z učebníc, aby čitateľ získal predstavu o obsahu a forme. Zároveň si bude môcť pod popisom všeobecných didaktických princípov (kapitola 4) predstaviť ich konkrétnu implementáciu.

3.1 Kódovanie informácie

T tejto téme sa primárne venujú učebnice *Informatika 1. diel* a *2. diel*. Téma sa vyučuje ako v štvrtom tak aj v piatom ročníku.

V učebnici pre 4. ročník sa začína so (zjednodušenou) Cézarovou šifrou, ako jednoduchým príkladom kódovania.

Nasleduje *cvičenie s vlajkami*. Cezar stojí na kopci a chce dávať príkazy svojim armádam. Rozhodne sa použiť vlajky. Existuje len niekoľko možných príkazov, napríklad „Zaútočte“. Prvá úloha žiakov je vymyslieť systém kódovania (teda ktorá vlajka bude znamenať ktorý príkaz). Následne kódujú konkrétnu správu. Príklad systému kódovania navrhnutému žiakmi je na nasledujúcom obrázku.

Povel	Vlajka	Povel	Vlajka
ZAÚTOČTE		ARMÁDA A	
UTEČTE		ARMÁDA B	
NA SEVER		ARMÁDA C	
NA JIH		RÁNO	
NA VÝCHOD		VEČER	
NA ZÁPAD		V POLEDNE	

Obrázok 2: Cvičenie vlajky – ukážka riešenia

Táto téma je rozvíjaná v ďalších cvičeniach. Napríklad, v nasledujúcom cvičení *Cézar v noci* chce Cézar ukázať tieto príkazy, ale v noci. Vlajky nie je vidieť, preto používa sekvencie zvukov, pričom si vybral používať zvuky “Tleskni” alebo “Dupni”. Napríklad, pre príkaz „Zaútočte“ si zvolí sekvenciu “Tleskni-Dupni-Dupni-Tleskni”. Dostávame sa tak k binárnemu kódu.

V ďalších cvičeniach sa žiaci postupne dostanú ku kódovaniu rôznych typov informácie. Na záver tejto témy v učebnici pre 4. ročník sa vysvetlí, ako tieto cvičenia súvisia s informatikou.

V 5. triede sa toto téma opakuje a ďalej rozvíja.

3.2 Práca s dátami, Štruktúrované dáta

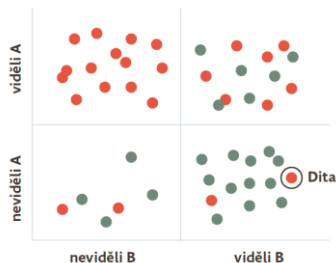
Z pohľadu RVP sa jedná o dve témy z dvoch okruhov, ktoré ale považujeme za efektívne rozvíjať spoločne.

Témam sa primárne venuje pracovný zošit *Hľadá sa Puffy*.

V učebnici sú úlohy, ktoré ukazujú záznam dát pomocou rôznych foriem a prostriedkov (texty, obrázky apod.) či zber dát. Úlohy tiež pracujú s rôznymi štruktúrami (tabuľka, graf, zoznam a ďalšie).

Okrem “tradičných” úloh má učebnica úlohy, ktoré napríklad vyžadujú dáta interpretovať (v jazyku detí, “premýšľať o dátach”). Nejedná sa pritom o ľubovoľné logické úlohy, ale o úlohy, ktoré majú informatický (a prípadne aj matematický) presah. Tieto pracujú s rôznymi konceptami, napríklad sledovať v dátach pravidelnosť (opakovanie vzoru), rozlišovať absolútnu hodnotu vs. pomer a ďalšie.

Príkladom je úloha *Kino* zo zošita *Hľadá sa Puffy*: Na grafe na obr. 3 sú znázornení žiaci červenej a zelenej triedy. Každému žiakovi na grafe odpovedá práve jedna bodka. Každá z tried bola na jednom z filmov „A“ a „B“. Úloha (po prípravných úlohách) je odhadnúť, ktorá trieda pravdepodobne videla ktorý film. O tom graf priamo nehovorí (nepopisuje, kto videl ktorý film v škole, ale kto daný film videl). Jedna z prípravných úloh potom je diskusia, ako je možné, že niečo nevidel ani jeden film.



Obrázok 3: Kino (skrátaná ukážka)

S mnohými štruktúrami (tabuľky, grafy) sa pracuje aj v matematike. Cvičenia ale často pokrývajú často len základné úkony (čítanie grafu, doplnenie tabuľky, jednoduché úvahy), pričom pre rozvoj informatického myslenia potrebujeme aj ďalšie typy úloh. Napríklad, v matematike pracujeme v kontexte jednej tabuľky, ale informatika často spája údaje z viacerých tabuliek. Túto operáciu si dieťa vyskúša v rámci cvičenia *Najdi v datech* v zošite *Hľadá se Puffy*. V rámci tohto cvičenia dieťa zisťuje, kedy sa postava z príbehu mohla vrátiť z koncertu tým, že spája údaje z tabuliek o koncertoch a mestách, kde sa koncerty konajú (pozri ďalší obrázok).

Datum	Mesto	Začátek	Koncert trvá	Mesto	Nejrychlejší možný čas autem
5. 10.	Nový Celet	20.00	2 hodiny	Adamov	2 hodiny
14. 10.	Boroskin	19.00	2 hodiny	Darkin	1 hodina 30 minút
15. 10.	Máchov	17.30	1 hodinu	Máchov	3 hodiny
17. 10.	Darkin	15.00	2 hodiny	Tokio	12 hodín

Obrázok 4: Spájanie dát (skrátaná ukážka)

3.3 Modelovanie

Tejto téme sa primárne venujú učebnice *Informatika prvý a druhý diel*. Čiastočne sa objavuje aj v zošite *Hľadá sa Puffy*.

Téma sa vyučuje ako v štvrtom tak aj v piatom ročníku.

Pred samotnou ukážkou uvedieme niektoré z našich pohľadov na modelovanie:

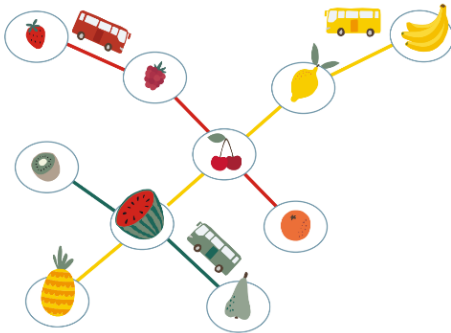
- Aj keď RVP samotné to neuvádza explicitne, za prínosné považujeme rozlišovať dva predmety didaktiky: modelovacie zručnosti a modely samotné.
- Na 1. stupni pracujeme s vizuálnymi modelmi. Model pritom nechápeme len ako „obrázok“ ale ako užitočné zjednodušenie reality. Modelovacie zručnosti sú potom určité operácie nad modelmi. Preto, nestačí mať ľubovoľné cvičenia pracujúce s obrázkami modelmi, ale je potrebné nájsť cvičenia, ktoré rozvíjajú modelovacie zručnosti. Príkladom je prechod od konkrétnej podoby cesty k jej topologickej abstrakcii, pozri cvičenie nižšie.
- Modelovacie zručnosti sa rozvíjajú činnosťou nad konkrétnymi modelmi. K tomu by bolo možné rôzne modely z rôznych oblastí. V rámci učebnice pracujeme s modelmi, ktoré majú informatický presah a tým podporiť rozvoj informatických konceptov v budúcnosti.

Vo všetkých troch učebniciach sa vyskytujú rôzne modely a úlohy, ktoré s modelmi pracujú. Uvedieme dve ukážky.

Prvá ukážka (obr. 4) je z učebnice pre 4. triedu. Jedná sa o *cvičenie s dopravou*, ktoré slúži ako úvod do tradičných úloh s modelovaním cestnej siete pomocou grafov (v zmysle teórie grafov).

1 DOPRAVA

Ve městě mají autobusové zastávky. Jezdí mezi nimi červený, zelený a žlutý autobus.



- Jak budeš cestovať z Pomerančové do Hruškové?
- Honza začal popisovať, jak jezdí červený autobus. Zapisuje písmena, napríklad „J“ je Jahodová. Doplň, čo napísal:



- Honza popsal i ďalšie autobusy. Oprav chyby, ktoré udělal.



Obrázok 5: Doprava (skrátaná ukážka)

Ďalší príklad modelovania uvedieme z učebnice pre 5. ročník. Žiaci hrajú hru na hádanie slov: je daná skupina 8 slov, z ktorých si jeden žiak vyberie tajné slovo. Druhý žiak ho má uhádnuť, pričom sa smie pýtať iba jedno slovo áno-nie.

Žiaci najskôr niekoľkokrát hrajú hru, pričom sa snažia dôjsť na návod, ako tajné slovo uhádnuť. Napokon majú svoj návod popísať rozhodovacím stromom. Príklad je na nasledujúcom obrázku:



Obrázok 6: Háďanie slov

Toto cvičenie neprecvičuje len modelovanie, ale napríklad aj hľadanie spoločných a odlišných vlastností (čo je súčasťou tematického celku Informačné systémy) či podporuje koncept vetvenia.

3.4 Digitálne technológie

Ďalšia ukážka cvičenia je z učebnice *Informatika 2. diel*. Pred týmto cvičením žiaci robia iné cvičenia, v ktorých sa dozvedia, že počítače môžu mať rôznu formu (a že napríklad aj mobil považujeme za druh počítača). Ďalším krokom je budovať prekoncept, že “skryté počítače sa nachádzajú všade okolo nás” (v spotrebičoch, zariadeniach...), čo môžeme podporiť napríklad cvičením na nasledujúcom obrázku. V ňom si deti majú vybrať jeden z bežných predmetov, a predstaviť si, ako by ho v budúcnosti mohol riadiť počítač (tak, aby to bolo užitočné).



Věc	Co užitečného by mohla dělat, když ji bude ovládat počítač?	Aby ji počítač ovládal, bude potřebovat určité informace. Například:

Obrázok 7: Počítače skryté okolo nás

3.5 Bezpečnosť

Téma bezpečnosti sa uvedie v 4. ročníku a rozvíja v 5. ročníku. Na obr. 8 je ukážka z cvičenia pre 4. ročník. Pripravuje porozumenie, že niektoré dáta sú dôverné (citlivé) a je nebezpečné ich prezradiť. V cvičení vystupuje Cezar, ktorý pošle nezašifrovanú správu. Deti sa pýtame, čo sa stane, keď pošla chytiť - ktoré údaje je nebezpečné prezradiť?

Zdravím Tě, generále!

Doufám, že je u vás dobré počasí.
Pošlu Ti na pomoc špiona, používá jméno Adam Mrkvička.
Jeho fotografii přikládám.

Přijde k Tobě zítra ve 12 hodin.
Umi dobře počítat. Věděl hned, kolik je 15 krát 15.
A má rád kuřecí maso. Ale když sní arašídny, může umřít.

Pošlu Ti také svou červenou armádu.
Přeji Ti hodně zdraví, už jsi dlouho nemocný.
Tvůj Cezar

Obrázok 8: Citlivé údaje (skrátaná ukážka)

Za vymysleným príbehom nasleduje ukážka konkrétneho profilu na sociálnej sieti a diskusia o citlivosti údajov zverejnených v tomto profile. V 5. ročníku sa toto téma ďalej rozvinie.

Považujeme za užitočné, ak si žiaci okrem učebnice absolvujú aj iný kurz bezpečnosti. Ten jednak môže informácie v učebnici ďalej doplniť (napríklad o praktickú ukážku nastavenia dvojfaktorového zabezpečenia sociálnej siete) a zároveň sa domnievame, že kľúčové myšlienky bezpečnosti je vhodné učiť opakovanne a rôznym prístupom.

3.6 Ďalšie typy aktivít

Učebnice obsahujú aj ďalšie typy úloh, napríklad efektívne algoritmy, vysvetlenie princípů internetu, systémy okolo nás (z časti informačné systémy) a ďalšie. Tieto v tomto príspevku z priestorových dôvodov neuvádzame, záujemcov odkazujeme na učebnice a príručky k nim.

4 DIDAKTICKÉ PRINCÍPY A KONCEPT UČEBNÍC

Po úvodnom predstavení a uvedení príkladov z učebníc sa v tejto kapitole budeme venovať popisu konceptu učebníc, a to z rôznych perspektív:

- Samotné cvičenia a ďalšie aktivity (koncepty a princípy aplikované pri ich tvorbe)
- Štruktúra učebníc (umiestnenie cvičení a aktivít do jednotlivých učebníc a vzájomné vzťahy učebníc)
- Spracovanie učebníc a súvisiacich materiálov
- Princípy a predpoklady výuky s učebnicami

4.1 Didaktické princípy cvičení a aktivít

Pri tvorbe cvičení a aktivít sme uplatnili niekoľko princípov uvedenej v tejto sekcii.

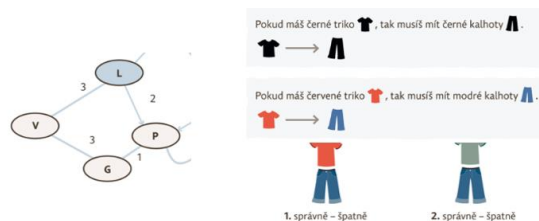
Miesto pasívneho výkladu uprednostňujeme *aktívne formy práce - cvičenia a hry*, v ktorých deti získavajú potrebné skúsenosti. Napríklad, výkladu o binárnom kódovaní predchádza päť šifrovacích a kódovacích hier, ktoré deti na uvedený výklad pripravujú.

Aj keď sa jedná o jednoduché hry a aktivity, *tieto majú často hlbší informatický presah*. Vysvetľujú určité aspekty digitálneho sveta, aplikujú a rozvíjajú určité spôsoby myslenia (informatické myslenia) a určité spôsoby vyjadrovania (napríklad informatické modely). Tiež pripravujú (budujú prekoncepty) pre 2. stupeň, pozri napríklad rozhodovacie stromy, ďalej efektívny vs. neefektívny algoritmus, hra, v ktorej program môže napodobňovať ľudské správanie aj keď nemá ľudské vedomie a ďalšie.

Učebnice nie sú voľnou zbierkou cvičení. *Séria cvičení často rozvíja určitý koncept* – pozri príklad cvičení s kódovaním informácie.

Jednotlivé cvičenia sa tiež môžu vzájomne podporovať - príklad prepojení troch cvičení demonštruje nasledujúci obrázok.

- Prvé cvičenie je tradičná úloha, kedy máme mapu s mestami, ktoré sú prepojené cestami, a hľadáme dopravné spojenie.
- V druhom cvičení sa rieši obdobný problém, ale zavedú sa jednosmerné cesty - po ktorých sa dá ísť len v smere šípky.
- Tretie cvičenie z našej ukážky pracuje s pravidlami – sú dané pravidlá obliekania a máme posúdiť, či daná postava je alebo nie oblečená v súlade s týmito pravidlami. Príkladom pravidla je pravidlo *Keď máš červené tričko, musíš mať červené nohavice*. Platí pritom, že pravidlo nie je symetrické, platí len v smere šípky.
- Druhé cvičenie tak -okrem iného- pripravuje deti na tretie cvičenie. V inom prostredí a to konkrétnejšom) ukazuje reláciu (ktorá nie je symetrická (v topo jednosmernej ceste sa nedá ísť oboma smermi, pravidlo neplatí oboma smermi).



Obrázok 9: Orientované cesty a pravidlá

Snažíme sa využívať synergie. Aj keď v novom RVP sú niektoré témy oddelené, tak tam, kde je to výhodné, niektoré cvičenia a aktivity témy prepájajú. Nejedná sa pritom o snahu „vidieť“ staré aktivity v novom RVP. V niektorých aktivitách či sérii aktivít podporujeme viacero oblastí zároveň, napríklad od popisov prirodzených systémov (v RVP okruh Informačné systémy) plynulo prechádzame k popisu umelých (v danom cvičení digitálnych) systémov (v RVP okruh Digitálne technológie). Iným príkladom je rozhodovací strom, uvedený v príkladoch v kapitole 3, ktorý je popísaný vizuálnym modelom, ale pri jeho vytváraní rozlišujeme spoločné a rozdielne vlastnosti objektov, objekty hierarchicky usporiadame apod.

Cieľom je zapojiť žiakov rôznej úrovne. *Cvičenia sú často gradované*. Prevedenie gradácie je obvykle tým, že cvičenie obsahuje viacero úloh od jednoduchej cez priemerné až po úlohy náročné. Zároveň, učebnice obsahujú aj *voliteľné cvičenia navyše*, ktoré môže využiť aktívny žiak alebo učiteľ, ak sa chce danej oblasti venovať hlbšie.

4.2 Štruktúra a vzájomné vzťahy učebníc

Väčšinu tém nájdeme ako v učebnici Informatika 1, tak aj v učebnici Informatika 2, teda ich učíme oba ročníky, štvrtý aj piaty. Obsah pre štvrtý ročník predstavuje úvod do tém, v piatom ročníku sa znalosti žiakov opakujú a prehlbujú. Napríklad, namiesto jedného dlhšieho bloku modelovania, učeneného v štvrtej i piatej triede, máme dva kratšie bloky venované modelovaniu v oboch ročníkoch.

Motiváciou tohto prístupu je látku častejšie opakovať, zvýšiť pestrosť hodín (nerobiť jednu tému príliš dlho). Zároveň to umožňuje lepšie zacieliť cvičenia na danú vekovú skupinu, teda robiť v 5. ročníku náročnejšie cvičenia ako v 4. ročníku.

Niektoré témy učíme spoločne s inými témami, aj keď RVP ich uvádza oddelene. Napríklad, diskusiu o výhodách digitálnych technológií považujeme za vhodné prepojiť s diskusiou o nevýhodách či rizikách týchto technológií.

Predpokladá sa, že škola bude používať obidve učebnice.

Pracovný zošit Hľadá sa Puffy sa sústreďuje na témy Práca s dátami (časť z tematického okruhu Data, informácie a modelovanie) a Štruktúrované dáta (časť z tematického okruhu Informačné systémy). Pracovný zošit je určený pre 5. ročník. Nemá pevnú väzbu na učebnice Informatika – je možné s ním pracovať pred prácou s učebnicou Informatika 2, po práci s touto učebnicou, či obe učebnice používať paralelne.

Táto učebnica *podporuje aj prácu s dátami v matematike*, preto je možné ju využiť na podporu tohto predmetu. Naše odporúčanie je práve z celkových 8 vyučovacích hodín, ktoré tento zošit vyžaduje, brať 4 hodiny z hodinového fondu matematiky a 4 z hodiny z hodinového fondu informatiky. Kládne to však vyššiu náročnosť na školu.

4.3 Spracovanie učebníc a súvisiacich materiálov

Učebnice sú papierové, vybrané cvičenia sú digitalizované. V digitálnej forme sú k dispozícii aj cvičenia a pracovné listy navyše. Nesnažime sa mať príliš veľa materiálov s podobnými aktivitami, pretože to nepovažujeme za potrebné. Upozorňujeme učiteľov na to, aby pri prípravu plánu počítali takisto s programovaním (robotikou) a nácvikom praktickej práce s digitálnym zariadením (v rozsahu, ktorý si škola zvolila v predmete Informatika učiteľ). Jedným z dôvodov, prečo kladieme dôraz na časovú efektívnosť výuky týchto tém je práve ponechať dostatok priestoru aj pre tieto ďalšie témy.

Učebnica *Hledá se Puffy* je spracovaná ako *detektívny príbeh*, v rámci ktorého žiaci riešia rozmanité úlohy. Primárne sa jedná o prácu s dátami a štruktúrovanými dátami, ale niektoré cvičenia sa venujú modelom či práce s pravidlami.

Ku každej z učebníc bola vytvorená *príručka pre učiteľa* [11, 12, 13]. Tá obsahuje výsledky, typy pre organizáciu hodiny, odporúčenia či ďalšie informácie. Príručky sú voľne prístupné online.

Učebnice obsahujú aj rozširujúci obsah (označený ako „Rozšírení“), ktorý môžu predstavovať cvičenia navyše či výklad navyše.

Súčasťou tvorby učebnice bol nielen výber, ktoré cvičenia do učebnice zaradiť, ale ktoré cvičenia či výklad pre 1. stupeň nezaraďovať. Príkladom nezaraďovaných pojmov je *kilobajt*, *procesor*, *RAM* a ďalšie, ktoré nie sú uvedené v RVP 1. stupňa. Niektoré z nich nie sú v učebnici vôbec, iné sa nachádzajú v rozširujúcom učive či ako námety pre ďalšiu prácu v príručke pre učiteľa. V príručke tiež upozorňujeme na témy, ktoré sú z pohľadu RVP pre 1. stupeň nepovinné, ale môžu byť pre žiakov významné, napríklad *vírus* a *antivírus*.

4.4 Princípy výuky

Predpokladom používania učebníc je schopnosť spolupracovať so spolužiakom a spolužiačkou, prostredie kde sa žiaci neboja spraviť chybu a vedenie k postoju, že sa máme „postaviť čelom“ aj k úlohám, na ktoré nemáme naučený postup riešenia.

Učebnice kladú dôraz na aktívnu prácu žiakov miesto pasívneho výkladu. Niektoré úlohy sú pre žiakov nové, resp. vyžadujú nový spôsob práce či uvažovania. To vyžaduje proaktívny postoj žiakov. A ten zas vyžaduje prostredie, v ktorom sa žiaci neboja chybovať, pretože chyby sú pri riešení nových typov úloh či náročnejších úloh prirodzené až nevyhnutné. Učebnica obsahuje aj náročnejšie úlohy a nepredpokladá sa, že každý žiak vyrieši všetky úlohy a cvičenia.

Niektoré aktivity je potrebné vykonať v dvojici či v skupine, pozri napríklad cvičenie na hádanie slov uvedené v kapitole 3.3. Ale aj u individuálne riešených úloh je vhodné dovoliť žiakom vzájomne si radiť či pomáhať si. Žiaci by preto mali mať určitú skúsenosť a návyky, ako spolupracovať.

5 ZHRNUTIE

V tomto príspevku sme predstavili tri nové učebnice, ktoré vyšli v Českej republike. Učebnice sú určené pre štvrtú a piatu triedu základnej školy. Učebnice podporujú väčšinu tém nových RVP (Rámcových vzdelávacích programov v Českej republike) pre informatiku. Pracovný zošit *Hledá sa Puffy* zároveň podporuje aj matematiku (práca s dátami).

Popísali sme princípy a koncept učebníc a ich používania. Uviedli sme niekoľko príkladov cvičení z učebníc.

Z priestorových dôvodov je náš popis stručný. Zaujemca o podrobnejšie môže nahliadnuť do príručiek k učebniciam, absolvovať niektorý z kurzov a prirodzene tiež nahliadnuť do samotných učebníc.

BIBLIOGRAFICKÉ ODKAZY

- [1] Nové RVP. Dostupné tiež na: <https://revize.edu.cz/nova-informatika-v-rvp-zv>
- [2] RŮŽIČKOVÁ, D. Informatika a revize rámcových vzdělávacích programů. In *DidInfo 2020* [online]. Liberec, 2020. ISBN 978-80-7494-532-8, ISSN 2454-051X. Dostupné tiež na: http://www.didinfo.net/images/DidInfo/files/Didinfo_2020.pdf
- [3] VANÍČEK, J., BERKI, J. Modelové školní vzdělávací programy pro novou informatiku v Česku. In *DidInfo 2021* [online]. Banská Bystrica, 2021, s. 158 - 161. ISBN: 978-80-557-1823-1. Dostupné tiež na: http://www.didinfo.net/images/DidInfo/files/DIDINFO_2021_zbornik.pdf
- [4] *Informatické myšlení* Dostupné tiež na: <https://imysleni.cz/>
- [5] BERKI, J., DRÁBKOVÁ, J. *Základy informatiky pro 1. stupeň základní školy* [online]. Liberec: Technická univerzita v Liberci, 2020. ISBN 978-80-7494-520-5. Dostupné tiež na: <https://imysleni.cz/ucebnice/zaklady-informatiky-pro-1-stupen-zs>
- [6] FILIPI, Z., MAINZ, D., FADRHZONZ, J. *Práce s daty pro 5.-7. ročník základní školy* [online]. Plzeň: Západočeská univerzita v Plzni, 2020, ISBN 978-80-261-0917-4. Dostupné tiež na: <https://imysleni.cz/ucebnice/prace-s-daty-pro-5-az-7-tridu-zakladni-skoly>
- [7] Informatika na webu nakladatelství Fraus. Dostupné tiež na: <https://www.informatika.fraus.cz/>
- [8] AGH, P. *Informatika 1. díl*. Plzeň: Fraus, 2022, 1. vydání. ISBN 978-80-7489-783-2
- [9] AGH, P. *Informatika 2. díl*. Plzeň: Fraus, 2022, 1. vydání. ISBN 978-80-7489-784-9
- [10] AGH, P. *Hledá se Puffy*. Plzeň: Fraus, 2022, 1. vydání. ISBN 978-80-7489-785-6
- [11] AGH, P. *Príručka učitele k učebnici Informatika 1. díl*. Plzeň: Fraus, 2022, 1. vydání. Dostupné tiež na: <https://ucebnice.fraus.cz/catalog/cs/i-stupen-informatika-hleda-se-puffy/p100586ick1sinfick1s.html?downloads=>
- [12] AGH, P. [online]. *Príručka učitele k učebnici Informatika 2. díl*. Plzeň: Fraus, 2022, 1. vydání. Dostupné tiež na: <https://ucebnice.fraus.cz/catalog/cs/i-stupen-informatika-hleda-se-puffy/p100586ick1sinfick1s.html?downloads=>
- [13] AGH, P. *Príručka učitele k pracovnímu sešitu Hledá se Puffy* [online]. Plzeň: Fraus, 2022, 1. vydání. Dostupné tiež na: <https://ucebnice.fraus.cz/catalog/cs/i-stupen-informatika-hleda-se-puffy/p100586ick1sinfick1s.html?downloads=>

Podpora výuky logických funkcí pomocí prostředí Minecraft Education

Support for teaching logic functions using Minecraft Education

Jan Bezděka
KVD FPE ZČU
Klatovská tř. 51
PSČ 30100 Plzeň
Česko
bezdekaj@kv.d.zcu.cz

Zbyněk Filipi
KVD FPE ZČU
Klatovská tř. 51
PSČ 30100 Plzeň
Česko
filipiz@kv.d.zcu.cz

ABSTRACT

In the article, we present a way how to better teach the problems of logical functions and also the construction of logical circuits. The aim is to make the reader aware of the possibility of alternative teaching methods of this problems in a more inspiring way. The primary focus is on secondary school students, but the acquired knowledge is also transferable to hobby circles in primary schools or only to classes in which logical functions are taught. The starting point was to look for a suitable internal motivation. The use of the well-known and widespread computer game Minecraft Education seems to be ideal from this point of view. After a brief introduction to the game itself and to its history, we describe the experience with the teaching of logical functions and the way of their teaching with the help of didactical analysis. The next part of the article is an introduction to the minimum of special components needed for the construction of logical functions and circuits in the environment of Minecraft Education. The final part contains an introduction to the possibilities of teaching more complex logic circuits including empirical experience and the showing of the possibilities for theoretical circuit extensions.

Keywords

Logic Function. Logic Circuit. Minecraft Education.

ABSTRAKT

V článku představujeme způsob, jak lépe učit problematiku logických funkcí a následně také konstrukci logických obvodů. Cílem je seznámit čtenáře s možností alternativní výuky této problematiky více motivující formou. Primární zaměření je na žáky středních škol, ale získané poznatky jsou přenositelné na zájmové kroužky základních škol či jen samostatnou výuku logických funkcí. Východiskem bylo hledání vhodné vnitřní motivace. Použití známé a značně rozšířené počítačové hry Minecraft Education se z tohoto hlediska jeví jako ideální. Po stručném seznámení s hrou samotnou a její historií se věnujeme popisu zkušeností s výukou logických funkcí a způsobu jejich vyučování za pomoci didaktické analýzy. Následuje seznámení s minimem speciálních komponent potřebných pro konstrukci logických funkcí a obvodů v prostředí Minecraft Education. Obsahem závěrečné části je seznámení s možnostmi výuky složitějších logických obvodů včetně empirických zkušeností a uvedení možností pro budoucí možná rozšíření obvodů.

Klíčová slova

Logická funkce. Logický obvod. Minecraft Education.

1 ÚVOD

Logické funkce a v obecné rovině pak i číslicová technika jsou důležitou součástí informatiky. Jde o rovinu fyzických zařízení a jejich součástí, ale také o úroveň programovou a v širším měřítku dokonce i algoritmační [1]. Při jejich výuce ovšem často narážíme na několik problémů. Koncept logických funkcí je i pro studenty na vysoké škole teoretický a těžko uchopitelný, přičemž praktické využití není na první pohled zřejmé. Při výuce na střední škole je pak tento problém akcentován ještě více. Žáci se rovněž potřebují naučit pracovat a přemýšlet v binární v soustavě.

Existuje několik způsobů, jak výuku logických funkcí ztraktivnit a přiblížit začátečníkům. Obvykle používáme simulace, které pomohou vizualizovat fungování obvodů. Další možností je zejména v případě logických funkcí realizace jejich činnosti pomocí kontaktní logiky, což opět přispěje k lepšímu pochopení funkce obvodu [1]. Problémem všech těchto řešení je nutnost dodat počáteční vnější motivaci, aby se žáci tématu věnovali a dokázali ho tak lépe pochopit. Právě absence vnitřní motivace je hlavním důvodem nízké úspěšnosti [2].

Silným motivátorem je použití prostředí počítačových her [3], nejlépe takového, které studenti znají a ideálně také používají. V případě výuky číslicové techniky se jako ideální prostředí nabízí hra Minecraft konkrétně ve verzi Education. V tomto případě je vysoká pravděpodobnost, že žáci hru znají a ovládají její prostředí. Stejnou situaci můžeme díky stovkám miliónů měsíčně aktivních hráčů předpokládat i u předchozí generace tedy současných žáků středních a studentů vysokých škol [4].

2 VÝUKA V MINECRAFT EDUCATION

Hru Minecraft v roce 2009 pod názvem Cave game vydalo studio Mojang AB. Po rychlém úspěchu došlo záhy na přejmenování hry na současný název Minecraft. Z hlediska využití při výuce je zásadní rok 2014, kdy celé studio zakoupil Microsoft a s tím převzal i další vývoj hry [4].

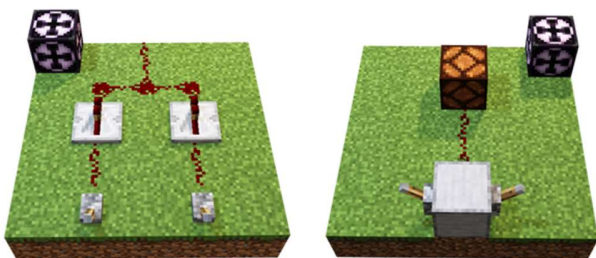
V roce 2016 byla vydána verze Education, která je primárně zacílena na školství. Učitel má k dispozici vlastní „učitelský režim“ pro spravování světa. V tomto režimu dále může měnit obecné nastavení pro žáky, zablokovat je apod. K této edici je poskytována také silná podpora, a to jak od společnosti Microsoft, tak

i komunity¹. K dispozici jsou připravené materiály a výukové video tutoriály. Minecraft lze použít pro výuku napříč předměty a učitel může díky tomu pohodlně uplatnit principy mezipředmětových vazeb. Velmi silná je zde podpora programování. My se budeme dále výhradně věnovat podpoře výuky číslicové techniky a logických funkcí.

V současné době se můžeme setkat se dvěma vývojovými větvemi hry Minecraft. Konkrétně se jedná o Java verzi primárně určenou pro osobní počítače a verzi Bedrock, která je naprogramována v jazyce C++ a je multiplatformní. Verze Bedrock umožňuje vzájemnou interakci uživatelů osobních počítačů konzolí a dalších zařízení. I proto doporučujeme používat právě ji. Je navíc obsažena v licenci Microsoft 365 A3 pro školy, která na školách v rámci České republiky značně rozšířená a umožňuje studentům využití hry i na počítači v domácnosti.

2.1 Logické funkce

Prvním termínem z konceptové vrstvy oboru informatiky, který jsme se rozhodli za pomoci Microsoft Education přenést do tematické vrstvy znalostí žáků byly logické funkce. Zásadní výhodou je skutečnost, že logické funkce lze sestavit pomocí různých kombinací použitých bloků a také techniky jejich sestavování. Není tedy neobvyklé, že více žáků dojde k cíli různými způsoby (Obrázek:1), což napomáhá naplňování cílů z kompetenční vrstvy.



Obrázek 1: Různé možnosti realizace log. funkce OR

V rámci předběžného ověřování na střední škole jsme se často setkávali s přirozeně vzniklou diskusí ohledně použitých materiálů a postupů, které jednotliví žáci použili. V případě, že diskuse nevznikne samovolně je možné ji inicializovat z pozice učitele, přičemž již jen vzájemná analýza konstrukcí, které žáci použili přispívá k vyšší míře porozumění.

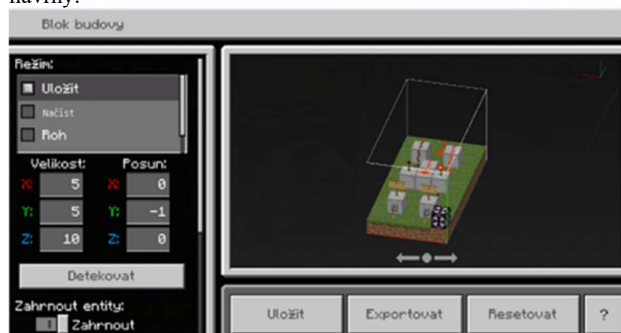
Při praktickém použití pro seznámení s logickými funkcemi se nám osvědčilo nejprve kombinovat popis logické funkce, její pravdivostní tabulku a ukázkou, jak je tuto funkci možno realizovat pomocí spínacové logiky (Obrázek: 2).



Obrázek 2: Podklady pro vysvětlení log. funkce

Poté co žáci navrhli a otestovali vlastní zapojení logické funkce, se seznámili s využitím speciální možnosti hry, která se nazývá „Blok budovy (Structure block)“ (Obrázek: 3).

Tato funkce umožňuje seskupení již hotových bloků a jejich samostatné uložení. Žákem vytvořenou hotovou logickou funkcí lze tedy tímto způsobem uložit a v případě potřeby ji nahrát či kopírovat. Blok budovy lze dokonce i přenášet mezi jednotlivými světy. Můžeme jej také vyexportovat jako 3D objekt ve formátu .glb (například pro možnost samostatného prezentování či 3D tisku). Není tak třeba blok v případě složitějších konstrukcí neustále vytvářet. Díky tomu je eliminována nudná opakující se činnost a žáci se mohou více soustředit na vlastní návrh. Zároveň s tím, jak si vytvářejí nové logické funkce si navrhují portfolio ve formě vlastní knihovny a mají je nadále k dispozici pro další návrhy.



Obrázek 3: Blok budovy (Structure block)

Je nutné poznamenat, že v prostředí Minecraft jsou k dispozici i logické funkce předpřipravené od tvůrců hry, které jsou realizované jako samostatný blok. Při jejich využití by ovšem žáci nebyli dostatečně aktivní a přechod z konceptové vrstvy oboru do tematické vrstvy žákovské zkušenosti by se neuskutečnil v požadované kvalitě. Samostatný návrh fungování logické funkce považujeme obzvláště u žáků technicky zaměřených oborů na středních školách za důležitý.

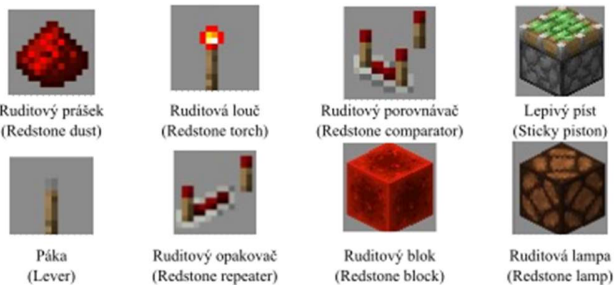
2.2 Použití speciální komponenty Minecraft Education

Před samotným zahájením činnosti v prostředí Minecraft Education je vhodné se seznámit s potřebnými speciálními komponentami. Je možné, že některé běžnější komponenty budou žáci již umět aktivně používat. Následně popsaných několik komponentů (Obrázek: 4) považujeme za nezbytné minimum pro konstrukci logických obvodů. Detailnější seznámení a také studium dalších použitelných rozšíření je možné například pomocí [6] nebo zhlédnutím některého z mnoha dostupných video tutoriálů na portálu Youtube.

Rudí prášek (Redstone) používáme jako vodič, který můžeme aplikovat na většinu povrchů. Prášek nelze aplikovat na tekutiny plyny a sklo, toho je možné využít v případě potřeby izolantu. Podobně jako odpor u standardního vodiče i zde je vlastností ubývání síly signálu, kdy maximální dosah je 16 bloků. Poté je třeba signál zesílit, popřípadě obnovit. Úroveň síly signálu je indikována intenzitou barvy vodiče, ale signalizace je viditelná

¹ Materiály komunity lze nalézt na stránce:
<https://skolstvi.ms/home/minecraft-education-edition>

pouze ve stavu kdy vodič vede signál. Vlastní propojení je realizováno umístěním bloků na dotek vedle sebe.



Obrázek 4: Použité komponenty

Ruditová louč (Redstone torch) umožňuje využití negovaného výstupu signálu, rovněž je možno ji použít jako přepínatelný zdroj energie.

Ruditový porovnávač (Redstone comparator) umožňuje porovnávat nebo odčítat signál v závislosti na způsobu použití.

Lepivý píst (Sticky piston) tento blok je výhodné používat ve spojení z Ruditovým blokem. Umožňuje přenos signálů ve svislém směru a následně tak vytvářet konstrukce, které mají více pater. Tento postup snižuje zastavené místo a v případě vhodného návrhu rovněž zřehledňuje konstrukce.

Páka (Lever) slouží jako přepínač mezi stavy zapnuto a vypnuto, respektive logická 1 a logická 0. Jeho aktivaci zajišťuje hráč.

Ruditový opakovač (Redstone repeater) používá se k zesílení signálu (minimálně po každých 15 blocích Redstone), má však také využití při zpoždění signálu, popřípadě lze použít jako usměrňovač (pro vedení signálu jedním směrem).

Ruditový blok (Redstone blok) tento blok má využití primárně jako trvalý zdroj energie (napětí).

Ruditová lampa (Redstone lamp) blok je možné použít jako indikaci stavu log 0 nebo log 1 (sonda) nebo například pro konstrukci displeje.

2.3 Logické obvody

Poté co žáci navrhli a pochopili jednotlivé logické funkce bylo možné začít konstruovat složitější obvody. Žáci se seznámili se způsoby vytváření logických obvodů na základě navržené pravdivostní tabulky. Jedná se o konstrukci tabulky na základě požadavku na fungování obvodu. Následně je vhodné obvody postavit v prostředí Minecraft. V případě, že je v kompetenční vrstvě cílem, aby se žáci naučili minimalizovat logické funkce, popřípadě konstruovat zapojení pomocí stejnorodé struktury, je vhodné probrat a vyzkoušet tuto problematiku v této fázi. Dalším krokem, který ovšem záleží na úrovni, které by žáci měli dosáhnout, je budování složitějších obvodů.

Jako první složitější obvod jsme v našem případě zvolili obvod binární sčítačky, která je v konceptové vrstvě oboru navázána na sčítání v binární soustavě. Protože bylo patrné, že žáci nemají doposud uvedenou dovednost dostatečně zvnitřněnou v tematické vrstvě, přestože s ní měli zkušenost z předchozího studia, přistoupili jsme k hromadnému opakování. V jeho rámci jsme využili řízenou diskusi, na níž jsme navázali ilustrační úlohy, jimiž jsme připomněli i rozdíl mezi binárním sčítáním a sčítáním

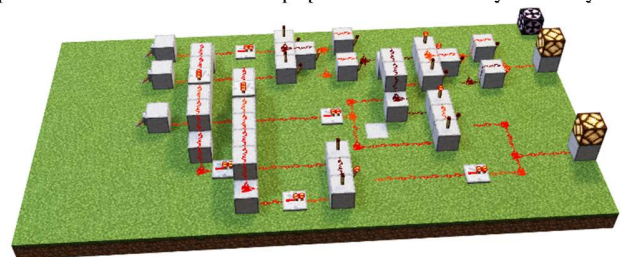
modulo 2 (XOR). Žáci v první iteraci dokázali posléze samostatně navrhnout poloviční dvoubitovou sčítačku.

Následně jsme iniciovali diskusi vedoucí k řešení problémů s přenosem do vyššího řádu. Cílem byl návrh úplné jednobitové sčítačky (Obrázek: 5). Zde bylo důležité zůstat opět primárně v roli mediátora a zasahovat minimálně už jenom kvůli faktu, že obvod lze postavit více způsoby. Je poměrně jednoduchý ale umožňuje kombinovat dříve nabyté znalosti a prakticky je použit.

V této části návrhu již bylo možno pozorovat výrazné rozdíly mezi jednotlivými postupy, které žáci použili. V případě na první pohled nejjednoduššího postupu, kdy žáci vytvářeli úplnou sčítačku za pomoci logických funkcí OR, AND, NOT a následného tradičního postupu konstrukce logického obvodu bylo výsledné zapojení velké a obsahovalo značné množství bloků.

V této fázi je zásadní, zda mají žáci k dispozici internet a mohou z něj čerpat. V takovém případě obvykle vyhledali již hotový zjednodušený návrh, který nadále realizovali. V případě, že internet k dispozici nebyl, přistoupili někteří žáci k minimalizaci logické funkce za pomoci Karnaughovy mapy. V několika případech dokázali žáci navrhnout jednodušší obvod i na základě analýzy jeho funkce a předchozích zkušeností. Zde obvykle předcházelo upozornění vyučujícího, že i tento postup je možný a existuje výrazně jednodušší zapojení obvodu.

V okamžiku, kdy jsme dosáhli fáze, že žáci jsou schopni samostatně navrhnout obvod, začali jsme uvažovat nad možnostmi, jak posílit znalosti a dovednosti v tematické vrstvě u žáků. Z několika postupů, které jsme vyzkoušeli nám nejvyšší míru efektivity poskytl následující scénář. Žáci jsou vyzváni, aby ve svém zapojení vytvořili jednu chybu, kterou následně jiný spolužák musí odhalit. Tento postup je velice dobře přenositelný i v případě týmové výuky. Výhodou je, že žák využívá nabyté znalosti při vytváření chyby tak, aby co nejvíce zkomplikoval její hledání spolužákovi a znovu také při hledání chyby v cizím zapojení. Sekundární výhodou tohoto postupu byla také aktivizace žáků vyplývající ze soutěžního prostředí, přičemž tuto motivaci může vyučující ještě posílit například vyhlášením vhodné ceny pro vítěze nebo případně vítězný tým.



Obrázek 5: Žáky navržený obvod sčítačky

Možným rozvinutím tohoto úkolu je například návrh úplné paralelní čtyřbitové sčítačky, kterému se věnují například v [5].

Dalším zařízením z konceptové vrstvy oboru, které jsme vytvořili, přičemž publikaci ohledně jeho užití jsme nezaznamenali, byla tvorba čítače a jeho následné rozšíření na čítač se zkráceným cyklem (modulo). Za pomoci několika čítačů modulo m jsme dále vytvořili hodiny. Indikace času byla realizována pouze v binárním tvaru, ve formě indikace výstupů jednotlivých klopných obvodů v rámci čítače. Zde se nabízí další pokračování práce, ať již ve formě rozšíření pro nadané žáky, či pokračování k dalším oborům v rámci číslicové techniky. Je například možné navrhnout

dekodér a prostředí Minecraft Education umožňuje rovněž vytvořit vlastní displej [5]. Tato potenciální rozšíření a zdokonalování původního obvodu nabízejí ideální příležitost pro týmovou nebo projektovou výuku, což přináší do budoucí praxe žáků další nezanedbatelné benefity.

2.4 Předvýzkum - praktické zkušenosti

V rámci použití Minecraft Education pro výuku logických funkcí jsme nejprve provedli předvýzkum, který měl za úkol zodpovědět několik otázek. Jednak ověřit, zda je tento způsob výuky efektivní a naplní naše předpoklady, dále zjistit časovou náročnost výuky realizované tímto způsobem a v neposlední řadě ověřit, zda žáci budou pracovat na zadaném úkolu a nebudou prostředí zneužívat k jiným aktivitám.

Předvýzkum byl realizován ve třetím ročníku čtyřletého studijního oboru: Mechanik elektrotechnik (dvě třídy, 26 a 22 žáků) a Telekomunikace (jedna třída, 19 žáků) v rámci předmětu Číslicová technika.

Před samotným zahájením testování byla zjišťována předchozí zkušenost s hrou Minecraft, kterou projevíli všichni žáci. Zajímavou informací, kterou jsme zjistili, ovšem bylo, že 46 žáků hru Minecraft nehrálo ani jednou v uplynulých dvou letech. Pouze 7 žáků mělo předchozí zkušenost s verzí Education. Na základě těchto zjištění jsme na úvod první hodiny zařadili seznámení s odlišnostmi mezi standardní a Education verzí. Ovládání a využívání Minecraft prostředí nečinilo žákům žádné problémy, ale většina z nich měla jen omezené znalosti o funkci Redstone obvodů. Doporučujeme tudíž do úvodní hodiny zavést seznámení se základními Redstone komponentami alespoň v rozsahu viz kapitola 2.2.

Pro výuku Číslicové techniky jsou v rámci školy standardně používány vždy bloky dvou vyučovacích hodin, což bylo ideální i pro naše potřeby. Ve všech třech případech bylo použito rozdělení tříd pro vlastní výuku na poloviční skupiny vzhledem k limitům počítačové učebny na 15 míst, díky čemuž se vyučující mohl věnovat žákům v dostatečné míře. Žáci byli rozděleni do dvojic.

Pro vlastní výuku a seznámení se s konstrukcí logických funkcí se ukázala jako ideální hodinová dotace tří bloků (6 x 45 min) přičemž poslední byl použit pro opakování a prohlubování použití jednotlivých funkcí. Pro následnou stavbu logického obvodu konkrétně logické sčítačky je dostatečná dotace dvou výukových bloků. V prvním bloku zvládli všichni žáci návrh a stavbu poloviční sčítačky a část dokonce i stavbu úplné sčítačky (nejrychlejší čas byl 57 minut) pro kterou byl určen další výukový blok. Následným krokem byla stavba úplné čtyřbitové paralelní sčítačky. Tento postup se osvědčil z hlediska více rychlostní výuky.

Z hlediska organizace výuky byl zvolen lokální server spuštěný na počítači učitele. To je výhodné i z hlediska ukládání a následného využití při dalším bloku výuky a umožňuje to žákům využít předešlé výtvořky. Žáci pracovali v jednom světě a mohli se navzájem pozorovat a ovlivňovat. Vzhledem k možnosti omezení práv např. pro použití bloků s TNT a podobně jsme v tomto uspořádání nezaznamenali větší obtíže a pozitivně nás překvapila i míra samoregulace žáků. Zajímavé bylo sledovat i reakce žáků, kdy si například své funkce oplotili a v případném volném čase i vizuálně vylepšovali například pomocí popisných tabulí různých ohraničení a podobně. Problém se vzájemným poškozováním výtvořů může nastat v případě stavby náročnějších obvodů pro které jsou třeba „Bloky budov“. Aby je mohli žáci používat je třeba jim v prostředí přiřadit vyšší práva (operátor) což jim ale zároveň umožní prakticky plnou kontrolu prostředí. V našem případě jsme

ale s tímto žádné závažné problémy nezaznamenali. Prakticky jedinou negativní zkušeností bylo zneužívání funkce chatu, pro jehož použití doporučujeme stanovit jasná pravidla.

3 ZÁVĚR

V režimu konstruování logických funkcí je možné a výhodné použít Minecraft Education i na úrovni žáků základních škol v neformálním vzdělávání v kroužcích výpočetní techniky nebo programování. Složitější konstrukce s logickými obvody lze s výhodou použít při výkladu na středních školách, případně při výuce vysokoškolských studentů informatiky na pedagogických fakultách.

Využití prostředí Minecraft Education pro výuku číslicové techniky nabízí rozsáhlé možnosti. V současné době pracujeme také na ověřování možnosti podpory při výuce funkce jednotlivých částí procesoru v rámci předmětu mikroprocesorová technika na technické střední škole. Na portálu Youtube je možné zhlédnout funkční ověřené konstrukce paměti, displejů, aritmeticko logické jednotky a dokonce celého procesoru včetně možnosti jeho jednoduchého programování.

BIBLIOGRAFICKÉ ODKAZY

- [1] SPOLJARIC, M. et. al. Interactive approach to Digital Logic. In: KORICIC, M. et. al. *MIPRO 2020: 43rd international convention : proceedings*. Rijeka: Croatian Society for Information, Communication and Electronic Technology, 2020. ISBN 978-953-233-099-1.
- [2] NAKONEČNÝ, M. *Motivace lidského chování*. Praha: Academia, 1996. ISBN 80-200-0592-7.
- [3] HUSNI, M. et. al. Computer Game-Based Learning Media as a Strategy to Increase Students' Motivation Learning. *At-Taqaddum* [online]. 2021, 13(1), 85-102 [cit. 2023-02-20]. ISSN 2527-9726.
- [4] SÁNCHEZ-LÓPEZ, I. et. al. Metaverse and education: the pioneering case of Minecraft in immersive digital learning. *El Profesional de la información* [online]. 2020, 31(6). [cit. 2023-02-20]. ISSN 1699-2407.
- [5] HÖLTGEN, S & FECKER, T & PLEIKIES, S & WORMSBECHER, N & DIVANI, S. A Case of Toy Computing. Implementing Digital Logics with "Minecraft", 2020. 10.13140/RG.2.2.33343.33446
- [6] FR JELLEY, Craig. Minecraft: Průvodce k využití Ruditu. Ilustroval Ryan MARSH, přeložil Radek KUBÁČ. V Praze: Egmont ČR, 2017. ISBN 9788025240748.
- [7] DEZUANNI, M. Minecraft and children's digital making: implications for media literacy education, *Learning, Media and Technology* 43, 3, pp. 236–249, 2018. doi:10.1080/17439884.2018.1472607, URL <https://doi.org/10.1080/17439884.2018.1472607>.
- [8] PRAYAGA, Lakshmi & DAVIS, James & WHITESIDE, Alexander & RIFFLE, Andrew. An Exploration In The Use Of Minecraft To Teach Digital Logic To Secondary School Students, 2016. 2.

Živý zošit: Informatika a informatické základy digitálnej gramotnosti na 1. stupni

Living Workbook: Informatics and the Base for Digital Literacy at Primary

Andrej Blaho

KAI FMFI

Univerzita Komenského v Bratislave
Slovensko

andrej.blaho@fmph.uniba.sk

Ivan Kalaš

KDMFI FMFI

Univerzita Komenského v Bratislave
Slovensko

ivan.kalas@fmph.uniba.sk

Milan Moravčík

EDIX, s. r. o.

Bratislava,
Slovensko

milan.moravcik@edix.sk

ABSTRACT

An important topic in the didactics of informatics is the awareness of the differences between own content and digital literacy (DL). Although closely related, informatics cannot be reduced to the development of computer skills and DL. On the other hand, however, it is natural that school informatics must contribute to this area by building the informatics foundations of DL, especially at primary level. How to characterise these foundations? In our paper we approach the question indirectly. We analyse one particular validated educational content for the 3rd Year, The Living Workbook, as part of the Informatics with Emil method. We have just completed its iterative development and validation. With its concept and content, The Living Workbook materialises our idea of how a modern constructionist pedagogy can be engaged in Informatics in Year 3, with sound space to build the informatics foundations of DL.

Keywords

Informatics in primary education. Digital literacy. Informatics with Emil. Living Workbook. Informatics foundations of digital literacy.

ABSTRAKT

Dôležitou témou didaktiky informatiky je uvedenie si rozdielov medzi vlastným obsahom a digitálnou gramotnosťou (DG). Aj keď ide o úzko súvisiace komponenty vzdelávania, informatiku nemôžeme redukovať na získavanie počítačových zručností a rozvoj DG. Je ale prirodzené, že školská informatika musí prispieť aj v tejto oblasti, a to budovaním *informatických základov DG*, predovšetkým práve na 1. stupni ZŠ. Ako môžeme charakterizovať tieto základy? V príspevku k položenej otázke pristupujeme nepriamo. Namiesto vymedzenia spomínaného pojmu analyzujeme jeden konkrétny vzdelávací obsah pre 3. ročník, a to Živý zošit, ktorý je súčasťou metódy Informatika s Emilom. Práve sme ukončili jeho iteratívny vývoj a overovanie. Živý zošit svojou koncepciou a obsahom vyjadruje našu predstavu, ako sa dá modernou konštruktivistickou pedagogikou realizovať v 3. ročníku informatika, s priestorom na budovanie informatických základov DG.

Kľúčové slová

Informatika na 1. stupni ZŠ. Digitálna gramotnosť. Informatika s Emilom. Živý zošit. Informatické základy digitálnej gramotnosti.

1 ÚVOD

Diskusie o informatike v primárnom vzdelávaní sú v súčasnosti v rozvinutých krajinách skutočne intenzívne. Diskutuje sa o formáte tohto vzdelávania, o tom, či má ísť o povinný predmet alebo nie, či začínať v prvom, druhom či treťom ročníku, ako nastaviť vzdelávací obsah tak, aby bol skutočne... *informatikou pre všetkých žiakov* [1], či sa zamerať na rozvoj digitálnej gramotnosti (skrátene

DG) a počítačových zručností alebo dať dôraz na programovanie a algoritmičné myslenie a pod.

V našom článku sa budeme venovať poslednej zo spomenutých tém. Domnievame sa totiž, že hodnota moderného informatického vzdelávania spočíva v komplexnom, vývinovo primeranom a vyváženom obsahu, ktorý:

- (1) rozvíja informatické myslenie ako nástroj na riešenie problémov informatickými postupmi, a zároveň
- (2) buduje *informatické základy digitálnej gramotnosti*.

Pri snahe o presnejšie vymedzenie cieľa (2) budeme postupovať nepriamo, a to analýzou konkrétneho vzdelávacieho obsahu. Naším cieľom bude odpovedať na otázku, aké informatické postupy a zručnosti si žiaci rozvíjajú pri riešení aktivít Živého zošita v 3. ročníku ZŠ. To považujeme za náš príspevok k vymedzeniu toho, čo sú informatické základy DG.

Tento text dokumentuje jeden zo záverečných krokov projektu Informatika s Emilom. Výskumnou stratégiou, ktorú v ňom od začiatku využívame, je *výskum vývojom* [2]. Jeho dôležitú etapu tvorí finálna spätná analýza procesu výskumu, vývoja a vytvorenej pedagogickej intervencie, pozri našu podobnú analýzu napr. v [3]. Vzhľadom na limitovaný rozsah sa tu však zameriame iba na jeden jej aspekt, a to analýzu výslednej intervencie z hľadiska prínosu k informatickému vzdelávaniu.

2 PROJEKT INFORMATIKA S EMILOM

Pri vývoji Informatiky s Emilom pre 1. stupeň (resp. pre pripravovaný 1. a 2. cyklus) sme sa rozhodli vytvoriť nielen samotný vzdelávací obsah, softvérové prostredia a pracovné zošity pre žiakov, ale aj modernú konštruktivistickú metódu pre vyučovanie informatiky, spolu s metodickou podporou učiteľov. Celý zámer sme rozdelili do troch pilierov:

- Emil 3 a 4 zameraný primárne na rozvoj programovania, s početnými prepojeniami na ďalšie oblasti informatiky, matematiky a iné vzdelávacie oblasti,
- Robotika s Emou, pozri [4, 5, 6], ktorej aktivity kombinujú skupinovú prácu žiakov, rozvoj informatického myslenia programovaním fyzického zariadenia (CS unplugged) a produktívneho prepojenia s rozvojom matematického myslenia, tvorivosti, komunikácie a ďalších oblastí, a to už od 1. ročníka ZŠ,
- Živý zošit (zatiaľ pre 3. ročník), pozri [7], zameraný práve na informatické základy DG. Túto zložku Informatiky s Emilom po dvojročnom overovaní práve pripravujeme na publikovanie.

Vo všetkých troch zložkách projektu je našim sekundárnym cieľom získať vo výslednom vzdelávacom obsahu vhodné inštrumenty na výskum a ich prostredníctvom sa dozvedieť viac o poznávacom

processe žiakov v informatike, pozri napr. [3, 5, 8]. Podobný vývo-
jovo výskumný postup vidíme aj u ďalších autorov, pozri napr. [9].

3 Živý zošit

Je výsledkom našej snahy využiť pedagogické princípy Emila (ob-
javovanie v dvojiciach, bez spätnej väzby softvérového prostredia,
spoločné diskusie riadené pedagógom a ďalšie) aj na zvyšné kom-
ponenty informatického vzdelávania. Práve spomínané celotriedne
diskusie sú dôležitou súčasťou spoločného konštruovania pozna-
nia, a tiež spätnej väzby pre učiteľa. (Metodická príručka k Živému
zošitu radí učiteľom primárneho vzdelávania najmä to, ako, o čom
a kedy robiť tieto diskusie.)

Podobne ako Emil a Ema, aj Živý zošit je členený do skupín aktivít,
ktoré označujeme písmenami **A, B, C...** Princíp skúmania nových
javov, situácií a informatických pojmov, nástrojov a postupov pod-
porujeme v každej skupine aktivít vhodnou gradáciou.

Živý zošit je postavený na báze vektorovej grafiky – všetky prvky
v aktivitách sú vizuálne objekty s rôznymi atribútmi. To žiakom
umožňuje s týmito prvkami – obrázkami, textami, útvarmi, čiarami
a oblasťami s obrysom, tabuľkami a ďalšími – manipulovať, vytvá-
rať ich, rušiť alebo rôzne modifikovať.

Po etape skúmania nového nástroja alebo javu sa žiaci v druhej
časti každej vyučovacej hodiny presunú v Živom zošite, Obrázok 1
vpravo hore, do „tvorivej dielne“, kde môžu podľa vlastného zvá-
ženia a námetov spolužiakov upravovať aktivity alebo si vytvárať
svoje vlastné. Pri **tvorbe** teda žiaci pomocou už preskúmaných ná-
strojov upravujú scénu, objekty i zadanie zvolených aktivít a
vytvárajú tak svoj vlastný digitálny interaktívny obsah, ktorý si
môžu uložiť a zadať ho na riešenie svojim spolužiakom a spolu-
žiakom v triede. Vytvorená aktivita prakticky zodpovedá vlastnej
malej aplikácii či malej hre. Domnievame sa, že Živý zošit dobre
ilustruje filozofiu konštrukcionizmu [10], čiže efektívne učenie sa
skúmaním a objavovaním, a najmä tvorbou zmysluplných produk-
tov, ktoré majú žiaci chuť zdieľať s ostatnými.



Obrázok 1: Rozpracovaná aktivita F2. Prezentuje tiež celé prostredie Živého zošita.

V tomto príspevku budeme analyzovať informatický obsah a pos-
tupy na budovanie informatických základov digitálnej gramotnosti
v aktivitách Živého zošita pre 3. ročník ZŠ. V nich žiaci objavajú
a preskúmajú 11 nových nástrojov. V aktivitách skupín **A až D**,
pozri Obrázok 2 hore, žiaci postupne objavajú a preskúmajú šesť
nástrojov: **ťahaj**; **vlož obrázok**; **zmenši**, **zväčši** a **otoč**; **vlož útvar**;
kresli a **vyfarbí**. V aktivitách skupín **E až H**, pozri dolný riadok
Obrázka 2, žiaci postupne objavajú a preskúmajú ďalších päť

nástrojov: **kopíruj**; **preklop**; **vlož text**; **choď na web** a **vlož
tabuľku**. Používajú však aj všetky nástroje z predchádzajúcich
aktivít.

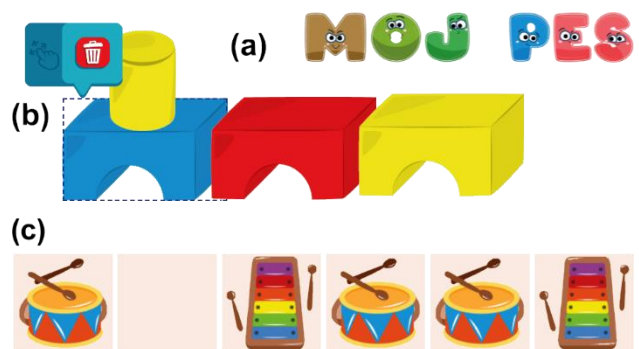


Obrázok 2: Nástroje, ktoré žiaci preskúmajú v 3. ročníku

A Obrázky

Žiaci v dvojiciach skúmajú nové softvérové prostredie, podobné
Emilovi. Spoznávajú tu prvé nástroje **ťahaj** a **vlož obrázok**. Skú-
maním, skúšaním a diskutovaním objavujú, ako sa ktorý nástroj
správa, čo robí. Najprv len presúvajú na ploche už pripravené
prvky obrázkov, aby z nich vytvorili známy výtvor. Z veselých
písmen (pozri Obrázok 3a), ktorými sú grafické objekty umiestnené
na ploche, skladajú slová. V matematickej úlohe presúvajú zápalky
tak, aby vznikla pravdivá rovnosť. Počínajúc aktivitou **A4** už majú
žiaci k dispozícii aj nástroj **vlož obrázok**. Keď si ho zvolia, otvorí
sa im panel s ponukou obrázkov, buď iba ako **jednoduchý zoznam
možností** na výber (napr. obrázky hračiek v **A4** alebo zvierat v **A5**)
alebo s možnosťou výberu rôznych farebných variantov pre každý
drevený prvok stavebnice. S nami vytvorenými obrázkami sa dá
hneď manipulovať, alebo sa tiež dajú kedykoľvek zrušiť. Žiaci
v týchto aktivitách objavujú a skúmajú, že :

- obrázky na ploche sa prekrývajú – podľa toho, v akom poradí
vznikli. Toto poradie sa nedá meniť, takže pri vytváraní nových
obrázkov (od aktivity **A4**) musia premýšľať aj o tom, čo má byť
nad čím. Toto poradie pri vytváraní hrá kľúčovú úlohu napr.
v aktivitách **A6** a **A7**, neskôr aj v ďalších skupinách, pozri
Obrázok 3b,
- obrázky niekedy svojím tvarom vyjadrujú aj textovú infor-
máciu a pri skladaní týchto písmen do slov (ťaháním na ploche)
žiaci uvažujú aj o inom type poradia, o správnom umiestnení
v slove,



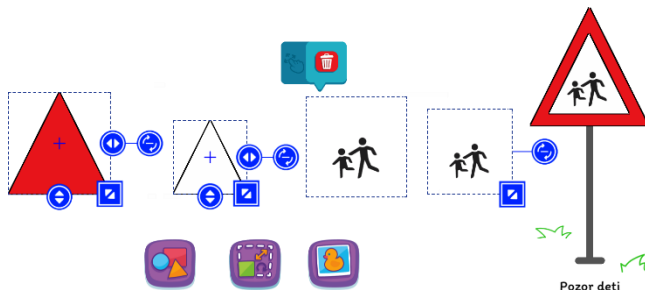
Obrázok 3: V grafických objektoch obrázok objavujú žiaci rôzne typy poradia

- obrázky sú niekedy na ploche zoradené podľa určitého opaku-
júceho sa vzoru, niektoré prvky však v postupnosti chýbajú,
pozri Obrázok 3c. Žiaci premýšľajú o ďalšej podobe poradia
a chýbajúce prvky dopĺňajú pomocou nástroja **vlož obrázok**.

Pri tvorbe v tejto skupine žiaci napr. vkladajú do plochy písmená z knižnice obrázkov podľa vlastného zámeru, potom ich ale preusporiadajú – ako úlohu na riešenie pre niekoho iného. Zo zápalkiek vytvárajú vlastnú „zápalkovú rovnosť“ a na záver ju „zašifrujú“ tým, že jednu alebo dve zápalky presunú na iné miesta.

B Útvary

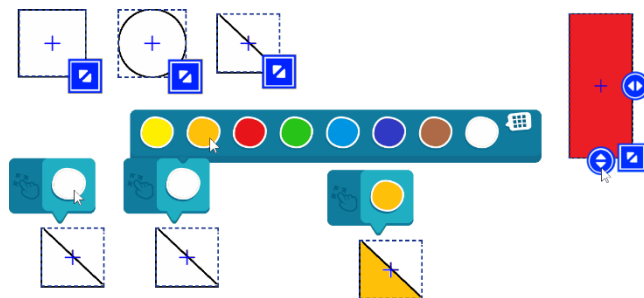
V tejto skupine aktivít žiaci ďalej rozširujú svoje zručnosti v práci s grafickými objektami, konkrétne so základnými geometrickými útvarmi štvorec, obdĺžnik, kruh, elipsa a trojuholník rovnoramenný a pravouhlý. Manipulujú s nimi na ploche, zmeňujú ich, zväčšujú a otáčajú, upravujú a opravujú ich veľkosť, proporcie a farbu výplne. Vytvárajú ich novým nástrojom **vlož útvar** do plochy, volia si pritom pre vytváraný útvar dva parametre: tvar a farbu výplne. Uvažujú o poradí pri ich vytváraní kvôli prekryvaniu (ktorý útvar je nad ktorým), kombinujú grafické objekty útvar a obrázok do jedného celku, pozri Obrázok 4.



Obrázok 4: Niekoľko krokov a použité nástroje pri riešení B6

V skupine B vyzerá gradácia pri objavovaní a skúmaní takto:

- žiaci najprv manipulujú na ploche s vopred pripravenými útvarmi (v B1 sú to obdĺžniky s rovnakou šírkou a rôznou výškou, v B2 trojuholník, kruhy a elipsy, niektoré aj v inom než základnom otočení). Jediný nástroj je **táhaj**,
- pribudne nový nástroj **zmeňši/zväčši** na proporčnú zmenu veľkosti objektu, pozri Obrázok 5 vľavo hore,
- keď žiaci niektorý útvar označia (kliknutím naň so zapnutým nástrojom **táhaj**, ktorý vlastne slúži aj na jednoduché označenie objektu bez ťahania), zobrazí sa im panel jeho nastavení (čiže atribútov objektu), zatiaľ iba s farbou výplne, ktorú tu môžu nielen vidieť, ale aj zmeniť, pozri Obrázok 5 dolu,
- keď teraz žiaci zvolia nástroj **zmeňši/zväčši**, pribudne im možnosť útvar zmenšovať a zväčšovať aj neproporčne, pozri Obrázok 5 celkom vpravo,



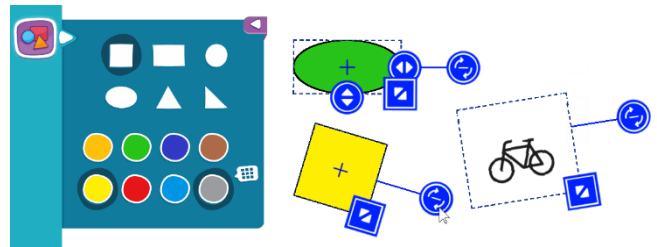
Obrázok 5: Proporčná a neproporčná zmena veľkosti, panel nastavení geometrického útvaru, zatiaľ iba s farbou výplne

- pribudne nástroj **vlož útvar** s dvomi parametrami – tvarom nového útvaru a farbou jeho výplne, pozri Obrázok 6 vľavo,

- pri vytváraní dopravných značiek v B6 pribudne aj známy nástroj **vlož obrázok**. Pri nástroji **zmeňši/zväčši** pribudne možnosť označený útvar aj otáčať (budeme mu preto odteraz hovoriť **zmeňši, zväčši a otoč**), pozri Obrázok 6 vpravo.

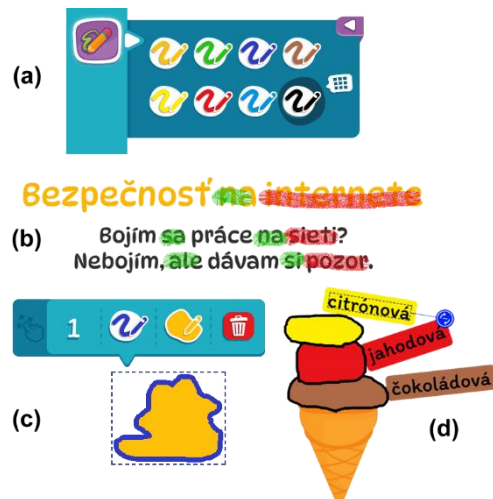
C Kreslenie

Z pohľadu grafických objektov doposiaľ žiaci pracovali iba s grafickými útvarmi a s predpripravenými obrázkami z knižnice. V tejto skupine aktivít sa budú stretávať aj s ďalšími typmi grafických obrázkov: čiarami (nakreslenými „voľnou rukou“) a oblasťami, ktoré sú vždy ohraničené čiarou a vyplnená farbou výplne, a tiež ich kompozíciami poskladanými z čiar a farebných oblastí. Žiaci objavujú možnosti, ktoré ponúkajú dva nové nástroje – **kresli** a **vyfarbi**.



Obrázok 6: Parametre nástroja vlož útvar v B4, otáčanie vložených útvarov a obrázkov

Žiaci najprv manipulujú s farebnými časťami obrázka, ktoré umiestňujú na správne miesto. Každá z týchto častí je uzavretá oblasť vopred pripravená voľným kreslením a vyfarbená jednou farbou. V ďalšej aktivite dokresľujú na plochu čiary nástrojom **kresli**, zatiaľ bez parametra, teda s predvolenou hrúbkou a farbou. Tieto nastavenia žiaci zatiaľ nemôžu dodatočne zmeniť. Môžu však nakreslený a označený čiarový objekt zrušiť. V C3 pribudne nástroju **kresli** prvý parameter, a to farba čiar, Obrázok 7a. Žiaci dokresľujú čiarami rôznych farieb chýbajúce časti písmen na „plagát“ s obrázkami a názvami domácich zvierat. Potom hrubou čiarou so štruktúrou zvýrazňovača označujú časti pripraveného textu na tému bezpečnosti na internete, pozri Obrázok 7b. Hľadajú cestu na mape a svoje riešenie kreslia ako čiaru. Ďalej objavia nástroj **vyfarbi** s parametrom farba na vyfarbenie oblasti uzavretej nakreslenou



Obrázok 7: Nástroje, niektoré parametre a panel nastavení pri kreslení.

čiarou. Začínajú aktívne používať pomocný nástroj na zrušenie práve vykonaného kroku, aj keď bol prístupný už od aktivity A1. Vtedy však žiaci presúvali grafické objekty a každé nesprávne umiestnenie sa dalo prirodzene „opraviť“ aj ďalším presunutím.

V aktivite C7 žiaci objavia, že ak pri kreslení čiar (jedným ťahom) úplne ohraničia určitú oblasť, tá sa automaticky vyfarbí aktuálne zvolenou farbou výplne – parametrom nástroja **kresli**. Ak nakreslenú a vyfarbenú oblasť označia, na paneli nastavení pre tento objekt môžu farbu výplne dodatočne zmeniť. K dispozícii tu majú aj nástroj **zmenši, zväčši a otoč**, aj keď iba na to, aby mohli otáčať kartičky s označením zmrzlín, pozri Obrázok 7c. V aktivite C8 už má nástroj **kresli** tri parametre, a to hrúbku a farbu čiar a farbu výplne. Aj panel nastavení pre objekt, ktorý si žiaci nakreslia, už teraz zobrazuje a umožňuje modifikovať tieto tri hodnoty, pozri Obrázok 7d.

D Kopírovanie

Aktivity tejto skupiny sú zamerané na kopírovanie grafických objektov a ich umiestňovanie na plochu podľa určitých logických podmienok. Jediným novým nástrojom tu je **kopíruj**, ktorý nemá žiadne parametre. Keď žiaci zvolia tento nástroj, môžu urobiť kópiu ktoréhokoľvek objektu na ploche (okrem tých, ktoré sú tvorcami aktivít „uzamknuté“). Kópia objektu vznikne s malým posunom od predlohy a je označená. Žiaci s ňou teda môžu ihneď manipulovať – umiestniť ju inam alebo jej zmeniť farbu v paneli nastavení.

V aktivite D1 žiaci pridávajú kopírovaním ďalšie husi a celý letiaci krídol organizujú do tvaru písmena V. V ďalšej aktivite majú podľa troch prvých korálikov na náhrdelníku opakovať ten istý vzor až do konca. Jediný objekt, ktorý sa však dá kopírovať, je nezafarbený korálik, ktorý musia opakovane kopírovať a na paneli nastavení novým kópiám voliť správne farby. Alternatívou je použiť nástroj **vyfarbi**. V D3 majú žiaci nakopírovať tri vtáčiky a rozmiestniť ich do okienok podľa určitej logickej podmienky (susednosti). Aktivita D4 je prvá, v ktorej sa budú nakopírované obrázky vo výslednej scéne prekryvať. Žiaci preto musia zvažovať poradie, v akom vzor obrázky kopírujú (napr. oči snehuliaka nemôžu vzniknúť skôr ako jeho telo, pretože by boli ukryté za ním). V D5 musia žiaci ukladať nakopírované obrázky podľa určitej logickej podmienky (na každom poschode a v každom stĺpci okien musia byť záclony rôznych farieb). V D6 žiaci kopírujú výtvarne podané písmená do slov a v záverečnej aktivite kopírujú rôzne zeleniny podľa ceny (akoby v obchode) do nákupu za danú výslednú sumu eur.

V etape tvorby žiakom prídadne aj možnosť použiť iné obrázky z knižnice a vytvoriť viac alebo menej podobné varianty jednotlivých aktivít. Napr. pri tvorbe v D7 môžu do ponuky obchodu pridať ďalšie ale iné zeleniny a ovocie, kopírovať alebo poprehadzovať cenovky (zatiaľ ich nedokážu prepisovať, až do aktivít písmena F) a zmeniť výsledné sumy za nákup v zadaní aktivity.

E Súmernosť

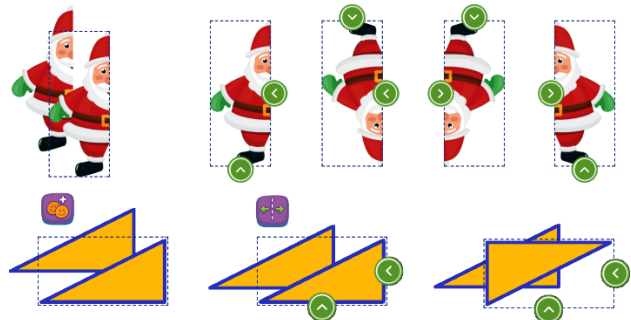
V aktivitách tejto skupiny žiaci pracujú so súmernými obrázkami a útvarmi, presnejšie grafickými objektami osovo súmernými podľa zvislej osi alebo podľa vodorovnej osi.

V metodickom materiáli učiteľom odporúčame, aby si pred hodinou na celé listy papiera vytlačili niekoľko písmen ako **A**, **B**, **H** alebo **N** a so žiakmi najprv diskutovali o tom, či písmená dokážeme prehnúť napoly tak, aby bol na oboch poloviciach prakticky „rovnaký“ ale preklopený obrázok. Žiaci potom objavujú, že niečo podobné môžu robiť aj pomocou nástroja **preklop**, pozri Obrázok 8. Skúmajú, ktoré obrázky sú súmerné a podľa ktorej osi, ako môžu

dorobiť súmerný obrázok z jednej polovice tým, že ju skopírujú a vhodne preklopia. Dokresľujú obrázok podľa osi súmernosti, vytvárajú zvislo aj vodorovne súmerné mozaiky. Vo vlastnej tvorbe potom používajú nástroj **preklop** (spolu so všetkými už známymi nástrojmi) na to, aby pre iných spolužiakov vytvorili podobné zadania.

Z pohľadu informatiky je tu najdôležitejšie, že si žiaci:

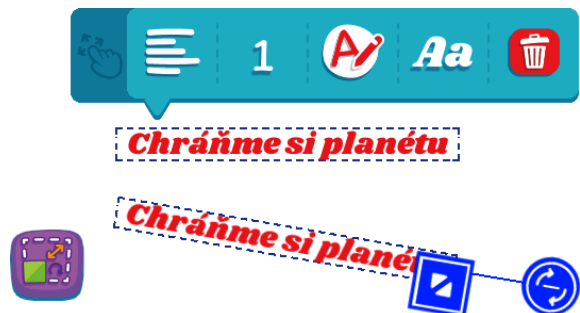
- budujú pohľad na **obrázok ako na štruktúru**, ktorá má určité vlastnosti. Rozvíjajú si svoju vnímanie toho, či je obrázok poskladaný z opakujúcich sa častí (možno rôzne poprekápaných), či je súmerný ako celok, ako by sa dal vytvoriť, doplniť, rozdeliť a pod., a tiež
- hľadajú správny postup operácií (preklopení) podľa toho, aký obrázok potrebujú vytvoriť.



Obrázok 8: Experimenty s nástrojom preklop – vhodné pre žiakov 3. ročníka vďaka dobrej vizuálnej „čitateľnosti“ operácie

F Text

V tejto skupine aktivít žiaci spoznávajú základy práce s textom, objavujú rôzne operácie, ktoré s textom robíme, a tiež vlastnosti a nastavenia, ktoré objekty text majú. Upravujú texty po stránke obsahovej, vytvárajú ich pomocou nástroja **vlož text** (najprv slovo, potom niekoľko slov v jednom alebo viacerých riadkoch), premiestňujú texty na ploche podľa určitých podmienok a pod.



Obrázok 9: Úplný panel nastavení pre objekt text. Aj text sa dá otáčať pomocou nástroja zmenši, zväčši a otoč

Keďže na 1. stupni na informatike žiaci na mnohých školách pracujú s tabletmi, v týchto aktivitách spoznávajú aj **prácu s virtuálnou klávesnicou**. Spolupráca v dvojiciach a pravidelné diskusie celej skupiny riadené pedagógom zabezpečia, že objavovanie písania (a to aj s diakritikou) nerobí žiakom problémy a netreba mu venovať špeciálny výklad. Učia sa vnímať text podobne ako iné vizuálne objekty, takže ho napr. aj otáčajú (pozri Obrázok 9), kopírujú či zväčšujú. Najprv pri vytváraní textu neuvažujú o jeho nastaveniach (vopred ich už zvolili autori). Potupne však v gradácii

objavia možnosť nastaviť a dodatočne meniť farbu textu, potom aj typ písma a jeho veľkosť, a napokon aj zarovnanie odseku. Tomu zodpovedá aj rozrastajúci sa obsah **panela s parametrami nástroja vlož text**, a tiež rozsah **panela s nastaveniami** pri označení textového objektu. Napr. na obrázku 1 sme videli aktívny nástroj **vlož text** v situácii, kedy žiaci pred vytvorením nového textového objektu nastavujú na paneli s parametrami nástroja iba farbu písma.

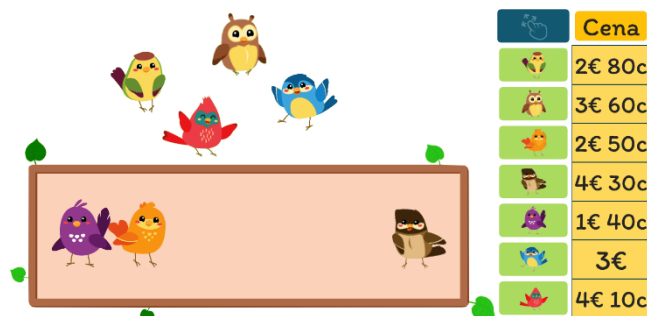
V aktivitách skupiny **F** žiaci spoznávajú aj ďalší nový nástroj, a to **chod' na web**. Ten otvára nové okno so zjednodušeným prehliadačom lokálneho, obsahovo obmedzeného internetu. Žiaci na ňom vyhľadávajú špecifické informácie potrebné pri riešení svojich aktivít, ako napr. názvy planét Slnecnej sústavy, príslovia, anglické názvy zvierat a pod. V iných aktivitách tu zasa hľadajú grafické objekty ako pozadie alebo dekoratívne obrázky a prenášajú si ich do aktivít, ktoré vytvárajú. Učia sa takto bezpečne orientovať na webe, sledovať odkazy na stránkach, vyhľadávať informácie či vkladať stadiaľ rôzne grafické objekty do svojho produktu.








V aktivite **F6** žiaci **vpisujú** do krížovky odpovede, každé písmeno vkladajú samostatne do pripraveného textového okienka s dĺžkou obmedzenou na jediný znak. Takéto **textové vstupy** s vopred nastaveným maximálnym počtom písmen, prípadne aj ďalšími obmedzeniami potom využívame aj v ďalších aktivitách. Vo vlastnej tvorbe v skupine **F** potom žiaci vytvárajú kompozície z textov a obrázkov, texty upravujú, otáčajú, preklápagú atď. Napr. v **F7** využívajú všetky doposiaľ známe nástroje pri tvorbe vlastnej pozvánky na svoju narodeninovú oslavu.

G Tabuľky

V tejto skupine aktivít žiaci pracujú s tabuľkami ako základnými štruktúrami údajov, rozvíjajú si orientovanie v jednoduchých tabuľkách so záhlavím, riadkami a stĺpcami. V políčkach (bunkách) tabuľky a v políčkach záhlaví poznajú a používajú čísla alebo obrázky. Dopĺňajú niektoré číselné údaje do frekvenčnej tabuľky na základe inak reprezentovaných údajov na ploche – napr. v podobe obrázkov. Rovnako dopĺňajú aj niektoré hodnoty v záhlaví – označenie stĺpca alebo riadka. Interpretujú údaje vo frekvenčnej tabuľke pomocou inej reprezentácie, napr. kopírovaním a rozdeľovaním euro mincí medzi tri deti a pod.

V aktivite **G3** žiaci vidia na ploche tabuľku s rôznymi vtáčikmi a ich cenami, ak by sa kupovali v obchode, pozri Obrázok 10. Úlohou je usporiadať ich do rámy podľa ceny od najlacnejšieho po najdrahší. Jedná sa teda o vytvorenie inej reprezentácie na základe údajov v tabuľke. Na to však musia žiaci uvažovať o tabuľke ako celok a v istom zmysle si ju usporiadať, resp. vyhľadať v nej najmenší prvok podľa ceny, potom druhý najmenší prvok alebo pod.



	Cena
	2€ 80c
	3€ 60c
	2€ 50c
	4€ 30c
	1€ 40c
	3€
	4€ 10c

Obrázok 10: Čiastočne rozpracovaná aktivita **G3**.
Jediným nástrojom je tu ťahanie

V ďalších aktivitách tejto skupiny žiaci vyplňajú tabuľku nie len na základe frekvencie (*Kolko dvojeurových mincí má Jana? Kolko ich má Filip?* a pod.), ale aj na základe ďalších kritérií, podmienok alebo výpočtov (*Kolko mincí má Jana dokopy? Kolko má peňazí celkovo? A Filip? A Sofia?* a pod.). V aktivite **G6** žiaci interpretujú ďalšiu dvojrozmernú frekvenčnú tabuľku a vyfarbujú podľa nej stavbu z „drevených“ prvkov. Interpretácia však nevedie k jednému možnému ofarbeniu hradu. Tým vytvárame podnetné príležitosti na následnú diskusiu v triede. Celá skupina aktivít vrcholí v etape skúmania tým, že žiaci sami **vytvoria na ploche novú tabuľku**, určia pre ňu vhodný počet riadkov a stĺpcov (zmenou nastavení tabuľky), pomenujú riadky a stĺpce a tabuľku vyplnia podľa údajov reprezentovaných daným textom.

K ďalším aspektom práce s tabuľkami – konkrétne s kódovacími tabuľkami a slovníkmi – sa žiaci opäť vrátia v skupine aktivít **H**, a tiež napr. v Emilovi 3, kde údaje z tabuľkovej reprezentácie prenášajú do Emilovej scény tým, že ho priamo riadia príkazmi. Progres žiakov v oblasti práce s tabuľkami a to, ako zručne a rôznorodo manipulujú s touto štruktúrou už v 3. ročníku, nás skutočne prekvapuje a teší. O našom bežiacom projekte zameranom na skúmanie práve tohto javu, teda ako žiaci rozumejú tabuľkám a ako interpretujú ich obsah, referujeme v [11].

H Tajné správy

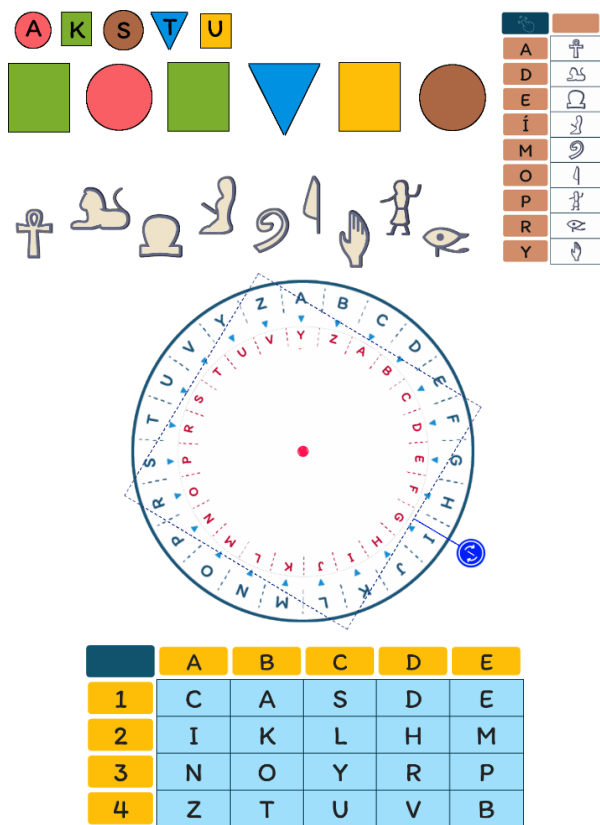
Témou tejto skupiny aktivít sú **kódy a šifry, šifrovanie a dešifrovanie**. So žiakmi v 3. ročníku sa nám však zdá postačujúce používať len slová a spojenia tajné správy, šifry a šifrovanie, a to aj pre ľubovoľný kód – aj preto, že slovo kódovanie teraz získalo skreslený a nadužívaný význam.

Žiaci začnú podľa rady v zadaní zafarbovali každé druhé písmeno v dlhých nezrozumiteľných slovách – a vyskočí im z nich zrozumiteľná správa. Potom používajú **slovník (kódováciu tabuľku)** typu *písmeno – zafarbený útvar* alebo *písmeno – hieroglyf* na preklad slova jedným alebo druhým smerom, pozri Obrázok 11 hore. Riešia logickú úlohu na skúmanie zašifrovaných slov. Musia pritom uvažovať o **stavbe a vlastnostiach zašifrovaných reťazcov** a slov, ako napr. dĺžke slova, opakujúcom sa znaku, či sa ten istý znak vyskytuje aj v inom slove na ploche a pod.

Žiaci potom objavujú názornú cézarovu šifru, pozri Obrázok 11 v strede, najprv na dešifrovanie správy so známym otočením (posunom), potom s malou pomôckou a tým, že môže nástrojom **otoč** cézarov disk aj skutočne otáčať a nájsť správne nastavenie pre dešifrovanie. V aktivite **H7** nájdú žiaci dvojrozmernú kódovaciu tabuľku so stĺpcami A, B, C... a riadkami 1, 2, 3, pozri Obrázok 11 dolu, a majú za úlohu dekodovať meno slávneho vedca. S takýmto tabuľkovým kódom (reprezentáciou) pracujeme aj v Robotike s Emou [4], ktorá je súčasťou vzdelávacieho obsahu Informatiky s Emilom. Naším cieľom prirodzene je, aby sa žiaci k rovnakým konceptom a postupom z informatiky, matematiky či digitálnej gramotnosti vracali opakovane a v rôznych kontextoch.

4 DISKUSIA

Vzdelávacie obsah Živého zošita pre 3. ročník ZŠ teraz (po ukončení vývoja a overovania na spolupracujúcej ZŠ v Modre) podrobujeme analýze z rôznych pohľadov, aby sme pomenovali, ako jednotlivé skupiny aktivít prispievajú k rozvoju inforatického myslenia, a zároveň aj inforatických základov digitálnej gramotnosti. Zároveň práve prebieha aj nezávislé overovanie pedagógmi na niekoľkých základných školách v Čechách.



Obrazok 11: Rôzne kódy, šifry a slovníky, aké žiaci objavili v aktivitách H

Spomínanú analýzu robíme z perspektívy novo pripravovanej reformy vzdelávacieho programu [12] na Slovensku a nedávno prijatej revízie vzdelávacieho programu v Čechách [13]. To nám napr. pomáha identifikovať, ktorým aspektom informatiky sa máme v Živom zošite ďalej venovať v 4. ročníku, aby sme obsahom celej Informatiky s Emilom (teda aj Emila a Emy) splnili všetky požiadavky kurikula. Druhoplánovo tu sledujeme aj pripravovanú reformu matematického programu na Slovensku a teší nás, že oblasť (nazývaná teraz komponent) *Závislosti, vzťahy a práca s údajmi* dostáva teraz takú významnú a explicitnú pozornosť.

Posudzovanie vzdelávacieho obsahu očami štátnych vzdelávacích programov však v sebe vždy nesie aj určité riziká. Ľahko sa dá totiž prehliadnuť potenciálne významný príspevok konkrétneho obsahu k menej hmatateľným a ťažšie pomenovateľným komponentom vzdelávania. V prípade celej Informatiky s Emilom tu máme na mysli dve oblasti. Prvá je príspevok vzdelávacieho obsahu k rozvoju informatického myslenia ako takého, napr. rozvoj logiky a strategického myslenia, a tiež *štruktúrneho pohľadu* na dáta a postupy, rastúca pripravenosť preskúmať správanie, rozpoznať možnosti a akceptovať obmedzenia v danej situácii a hľadať riešenie problému „v rámci možností“.

Druhou takouto ťažšie uchopiteľnou oblasťou je rozvoj rôznych zručností vyššieho rádu, akými je napr. spolupráca, komunikácia a verbalizácia svojich postupov – Informatika s Emilom vytvára svojou pedagogickou metódou bohaté príležitosti aj na ich rozvoj. Ak sa však vrátíme späť k premýšľaniu o naplnení cieľov informatického vzdelávania, našu analýzu, ktorú sme tu prezentovali považujeme okrem iného aj za náš príspevok k nepriamemu vymedzeniu, ako súvisí školská informatika a digitálna gramotnosť a čo

máme na mysli, ak hovoríme o rozvoji informatických základov digitálnej gramotnosti.

POĎAKOVANIE

Ďakujeme všetkým žiakom a učiteľom spolupracujúcich základných škôl a pedagogických fakúlt na Slovensku a v Česku, ktorí sa podieľali na niekoľkoročnom overovaní Živého zošita. Ďakujeme doc. PaedDr. M. Tomcsányiovej, PhD. za množstvo námetov na aktivity, ktorými prispela k vzniku tohto vzdelávacieho obsahu.

BIBLIOGRAFICKÉ ODKAZY

- [1] SENTANCE, S. Moving to mainstream: Developing computing for all. In: *Proc. of the 14th Workshop in Primary and Secondary Computing Education*, 2019, pp: 1-2, DOI: doi.org/10.1145/3361721.3362117
- [2] PLOMP, T., NIEVEEN, N. (eds.) *Educational design research. Part A: An Introduction*. 204 p. SLO, 2013, Netherland, slo.nl/publish/pages/2904/educational-design-research-part-a.pdf
- [3] BLAHO, A., KALAŠ, I., MORAVČÍK, M. Programming in Year 4: An analysis of the design research process. In *Proceedings of the 14th International Conference on Computer Supported Education, Vol 2: CSEdu*, pp. 425-433. DOI: [10.5220/0010988000003182](https://doi.org/10.5220/0010988000003182)
- [4] HRUŠECKÁ, A., KALAŠ, I. Robotika s Emou pre 1. stupeň základnej školy. 72 s. Indícia 2021, Bratislava.
- [5] KALAS, I., HRUSECKA, A. Levels of Control in Primary Robotics. To appear in post-conference book, WCCE 2022, Springer.
- [6] MIKOVÁ, K. et al. Gradation of Cognitive Operations of Blue-Bot Control in the Primary Education. In *Robotics in Education. RiE 2021*, pp.3-13, AISC vol 1359. Springer, Cham 2021. DOI: [10.1007/978-3-030-82544-7_1](https://doi.org/10.1007/978-3-030-82544-7_1)
- [7] BLAHO, S. Nová metóda pre novú informatiku. In: *Studia Scientifica Facultatis Paedagogicae Universitas Catholica Ružomberok* [online]. Ružomberok: VERBUM, 2023, 22 (1), s. 7-17. DOI: doi.org/10.54937/ssf.2023.22.1.7-17
- [8] KALAŠ, I., HORVÁTHOVÁ, K. Programming Concepts in Lower Primary Years and Their Cognitive Demands. In: *Passey, D. et al. (eds) Digital Transformation of Education and Learning – Past, Present and Future. OCCE 2021. IFIP Advances in ICT*, vol 642. Springer, Cham 2022. DOI: [10.1007/978-3-030-97986-7_3](https://doi.org/10.1007/978-3-030-97986-7_3)
- [9] VANÍČEK, J., DOBIÁŠ V., ŠIMANDL, V.: Understanding loops: What are the misconceptions of lower secondary pupils?, *Informatics in Education* (2022), DOI: [10.15388/infedu.2023.20](https://doi.org/10.15388/infedu.2023.20)
- [10] KALAŠ, I. et al. Konštrukcionizmus. Od Piageta po školu v digitálnom veku. In: *Proc of DidInfo 2011*, s.7-18. Banská Bystrica: Univerzita Mateja Bela, Fakulta prírodných vied.
- [11] ČUJDÍKOVÁ, M., KALAŠ, I. Tabuľky: Ako ich žiaci interpretujú v kontexte prostredia Emil Správca. Accepted for *DidInfo 2023*, Banská Bystrica: Univerzita Mateja Bela, Fakulta prírodných vied.
- [12] Národný inštitút vzdelávania a mládeže (NIVAM). Projekt nového kurikula, <https://vzdelavanie21.sk/>
- [13] MŠMT *Rámcový vzdelávací program pro základní vzdělávání – Informatika*, Praha 2021, online: <https://revize.edu.cz/files/informatika-2021.pdf>

BBC micro:bit SMART CITY v projektovom vyučovaní alebo „o krok bližšie k žiakom“ BBC micro:bit SMART CITY in project based learning or “one step closer to the students”

Katarína Brinziková
Prírodovedecká fakulta UPJŠ
Jesenná 5
041 54 Košice
Slovenská republika
katarina.brinzikova@student.upjs.sk

Eubomír Šnajder
Prírodovedecká fakulta UPJŠ
Jesenná 5
041 54 Košice
Slovenská republika
lubomir.snajder@upjs.sk

Ján Guniš
Prírodovedecká fakulta UPJŠ
Jesenná 5
041 54 Košice
Slovenská republika
jan.gunis@upjs.sk

ABSTRACT

In this article, we present our own experiences, findings and perspective on project based learning using the BBC micro:bit kit at Jarmočná 96 Elementary School, Ždaňa. Both the Ždaňa-City and Ždaňa-City:TheNextGeneration projects helped to improve the material security of the school, introduced another form of experiential learning into teaching, engaged the students, taught them the basics of programming in a non-violent playful way, and increased public awareness of the school. The result of the project is an intelligent interactive city containing traffic lights, streetlamps, barriers, information panels, a wind turbine, an intelligent plant sprinkler, robotic cars, etc. The development of the city required and at the same time developed the knowledge of students from several natural and social science subjects.

Keywords

BBC micro:bit. Programming. Smart City. Project ENTER. STEAM.

ABSTRAKT

V tomto článku uvádzame vlastné skúsenosti, zistenia a pohľad na projektové vyučovanie využitím stavebnice BBC micro:bit na Základnej škole Jarmočná 96, Ždaňa. Projekty Ždaňa-City a Ždaňa-City:TheNextGeneration pomohli zlepšiť materiálne zabezpečenie školy, zaviedli do výučby ďalšiu formu zážitkového učenia, zaujali žiakov, naučili ich základy programovania nenásilnou hravou formou, zvýšili aj povedomie verejnosti o škole. Výsledkom projektu je inteligentné interaktívne mestečko obsahujúce semaforey, pouličné lampy, závory, informačné panely, veternú turbínu, inteligentný zavlažovač rastlinky, robotické autíčka a i. Vývoj mestečka si vyžadoval a zároveň rozvíjal poznatky žiakov z viacerých prírodovedných a spoločenskovedných predmetov.

Kľúčové slová

BBC micro:bit. Programovanie. Inteligentné mesto. Projekt ENTER. STEAM.

1 ÚVOD

Školstvo prežíva v posledných rokoch veľkú digitalizačnú premenu. Aj infromatické vzdelávanie si vyžaduje a zažíva veľké zmeny v obsahu a formách výučby. Školy sa z rôznych zdrojov vybavujú novým hardvérom a softvérom so senzormi, aktuátormi, ktoré majú veľký potenciál skvalitniť výučbu nielen informatiky, ale aj ostatných vyučovacích predmetov. Prepojenie obsahu učiva viacerých predmetov a zameranie na tvorbu v praxi využiteľných produktov umožňuje realizácia projektového vyučovania. Práca na projektoch je pre realizátora veľmi náročná, od napísania prvej myšlienky až po odovzdanie záverečnej správy. V článku uvádzame naše skúsenosti, zistenia, odporúčania z realizácie projektového vyučovania s využitím edukačných stavebníc BBC micro:bit (ďalej micro:bit) a Lego stavebnice na Základnej škole, Jarmočná 96, Ždaňa.

2 PROJEKTOVÉ VYUČOVANIE

Projektové vyučovanie má v procese výučby neľahkú pozíciu. Učiteľ si pri slove projekt väčšinou predstaví náročnosť prípravy a samotnej realizácie, (ne)schopnosť žiakov vedieť pracovať na projektoch, obavy zo spolupráce s učiteľmi-kolegami, s nepochopením požadovaného cieľa a realizácie sprievodných úloh a podobne. A navyše mnoho učiteľov len ťažko dokáže „obetovať“ vyučovaciu hodinu zo svojho Učebného plánu¹, ktorý je umenie dodržiavať. Dobehtnutie zameškaného učiva má za následok tlak len a len na samotných žiakov. Málokto učiteľ si však uvedomuje, že projektové vyučovanie patrí medzi vyučovacie koncepcie, ktoré sú charakterizované najvyšším stupňom samostatnosti poznávacej činnosti žiakov [1]. V problémovom vyučovaní žiaci riešia problémy, sú nútení rozmýšľať, skúmať, objavovať, tvoriť. Problémy, ktoré riešia žiaci sú komplexné, nemusia vychádzať z obsahu učiva, ale predovšetkým zo života, z mimoškolských skúseností, ich riešenie si vyžaduje poznatky z viacerých vied (tradičných vyučovacích predmetov), problémy riešia skupiny žiakov (kooperatívne vyučovanie) najmä z vlastného záujmu a bez vonkajšej motivácie [2]. Projekt vedie k tvorbe jedinečného riešenia alebo osobitého

¹ Učebné plány sú súčasťou školských vzdelávacích programov a rozpracúvajú rámcové učebné plány štátneho vzdelávacieho programu podľa jednotlivých ročníkov s určením celkového

tyždenného počtu vyučovacích hodín pre príslušný ročník školského vzdelávacieho programu.

produktu a vyžaduje od žiaka autorský vklad. Podporuje ich kreativitu a samostatnosť. Učiteľia by mali považovať úspešné riešenie projektov žiakmi za najvyšší a najdôležitejší cieľ svojho predmetu [3]. Charakteristickým znakom projektového vyučovania je prebratie zodpovednosti za riešenie projektu samotnými žiakmi, pričom žiaci si majú možnosť sami voliť, ako spĺnia projektovú úlohu a v rámci skupiny si zvolia, akej konkrétnej činnosti sa budú venovať [4, 5]. Učiteľ je počas realizácie projektov žiakmi facilitátorom – ten kto pomáha študujúcim, uľahčuje im učenie (nie ako tradičný autoritatívny učiteľ) [1]. Projektové hodiny sú potom ďaleko viac motivačné, zapamätateľné, tvorivejšie, aktívnejšie, ďaleko viac poznávacie. Niekedy aj napriek týmto kladným stránkam, učiteľ zaujme postoj, že práca na projekte je príliš náročná. Okrem spomínaných záporov projekt musí byť premyslene organizovaný a riadený, učiteľ musí citlivo odhadnúť mieru voľnosti a mieru zodpovednosti žiakov, projektové vyučovanie navyše nerespektuje zásadu postupnosti vyučovania poznatkov, zásadu primeranosti [6].

Projektové vyučovanie chápeme ako prostriedok, ktorý umožní „byť o krok bližšie k žiakom“. Bližšie k predmetu – bližšie k poznaniu. Vhodne navrhnutý projekt vytvára hodnoty človeka, žiaci sa učia komunikácii, rešpektovaniu iných názorov, argumentovaniu. Projekt odhaľuje skryté vlastnosti žiaka, rozvíja vedomosti a zručnosti, formuje k tímovej zodpovednosti.

3 ŠKOLY A GRANTOVÁ PODPORA

Školy často nemajú financie na nákup učebných pomôcok, hardvéru, potrebného k modernému vzdelávaniu. Majú však možnosť získať finančné prostriedky pomocou rôznych grantových programov alebo výziev. Tie často podporujú tvorivých učiteľov, ktorí chcú so žiakmi pracovať moderne, ktorí vedú svojimi aktivitami, inovatívnymi prístupmi a vyučovacími metódami zaujať. Grantové prostriedky následne učiteľia využívajú na nákup žiadaných moderných pomôcok, ktoré súvisia s vypracovanou témou pre konkrétnu grantovú výzvu. Je potrebné, aby učiteľ mal projekt dômyselne premyslený a prepracovaný, aby jasne definoval problém, zdôvodnil potreby projektu, formuloval ciele, ale vyzdvihol aj dopad projektu. Mnohokrát je to pre učiteľa náročná cesta, pretože príprava, realizácia projektu, dokumentácia, záverečná správa projektu si vyžadujú nemalé úsilie a čas aj mimo pracovnej doby. Mnohých učiteľov táto skutočnosť naozaj odradí.

3.1 Grantový program ENTER

V roku 2020 Nadačný fond Telekom pri Nadácii Pontis vyhlásil grantovú výzvu: ENTER pre školy 2020 – Digitálna generácia (ďalej len projekt ENTER). Cieľom grantového programu bolo sprístupniť žiakom a žiačkam druhého stupňa základných škôl hrovou, zaujímavou a interaktívnou formou svet digitálnych technológií a programovania prostredníctvom micro:bitov. Micro:bity môžu byť využité nielen na hodinách informatiky, ale aj na všetkých ostatných predmetoch, pretože ich využitie je veľmi praktické a variabilné [7].

Pre autorku článku ako učiteľku informatiky to bola jedinečná príležitosť ako získať finančné prostriedky pre základnú školu prioritne na nákup stavebníc BBC micro:bitov. Objavila sa jedna veľká otázka: „Čo vymyslíme?“. Prirodzene tým skutočným hlavným zámerom bolo zamerať sa na žiakov, na ich vnútornú motiváciu, prirodzenú zvedavosť, kritické myslenie, spoluprácu, rešpektujúci prístup a v neposlednom rade na tvorivé riešenie problémov.

O tom, čo je to micro:bit sa v tomto článku nebudeme zaoberať. Veľmi podrobne je o ňom napísané v Zborníku DIDINFO 2021

v krátkom príspevku s názvom *BBC micro:bit ako súčasť projektu ENTER*, tiež na internetových stránkach *Projektu ENTER* [8, 9, 10].

3.2 Projekt ENTER a Ždaňa-City

Svet programovania ponúka široké spektrum možností na realizáciu sa žiakov. V pedagogickej praxi vnímame, že žiaci sú tvoriví, nápadiť a niekedy ich stačí len vhodne nasmerovať a hneď objavujú svoje skryté schopnosti. Výučba s micro:bitmi je prirodzene zážitková, rozvíja vedomosti a zručnosti v rôznych oblastiach, ktoré možno preniesť na konkrétne situácie zo života. Na to, aby projekt mohol byť napísaný bolo potrebné zistiť, aký hardvér ponúkajú e-shopy zamerané na micro:bity. Doslova to bol prieskum slovenského trhu, kde sa ponúkala v celku bohatá zásobáreň hardvéru a nápadov. Oslovili nás najmä semaforey, pouličné lampy, závory robotické autička a sady Smart Home Kit, Smart Science IoT Kit. Objekty mesta sa začali prirodzene prepájať a nápady začali prichádzať samé – svetelné a textové billboardy, fyzikálne zákony, prirodné javy, inteligentné mesto, inteligentná domácnosť, robotika, atď. Zvolili sme si široko využiteľnú tému – MESTO. Táto téma budovania inteligentných riešení bola zhodou okolností obracaná takmer v každom koncepte Smart Cities v oblastiach správy a rozvoja miest a regiónov.

3.3 Ciele projektu

Ďalšou etapou bolo stanoviť ciele, vysloviť, čo chceme aktivitami dosiahnuť, aké kvalitatívne a kvantitatívne výsledky očakávame a podobne. Uvádzame výber zo stanovených cieľov:

- Žiak 1. stupňa ZŠ vie opísať základné časti micro:bitu, vytvoriť jednoduchý program v prostredí Microsoft MakeCode, nahrávať program z počítača do micro:bitu, vie bezpečne manipulovať s micro:bitom.
- Žiak 2. stupňa ZŠ má porozumieť postaveniu programovania v informatike a v bežnom živote, vie k micro:bitu pripojiť rozširujúci hardvér, dokáže riešiť základné algoritmické problémy, riešenia vie zapísať a odlaďiť v konkrétnom programovacom prostredí, vie vsadiť micro:bit do reálneho prostredia a modelovať rôzne fyzikálne zákony a hry, dokáže navrhnúť a zrealizovať jednoduchý koncept interaktívneho inteligentného mesta.
- Učiteľ dokáže vzbudiť záujem žiakov o programovanie, zistiť prostredníctvom prieskumu záujem o informatiku a programovanie pred a po realizácii projektu, vie žiakov motivovať a viesť celou tvorbou projektu ako mentor alebo facilitátor, dokáže so žiakmi zrealizovať interaktívnu výstavu projektu, rozvíjať prezentačné zručnosti žiakov, zviditeľniť projekt prostredníctvom médií, ozrejmiť využiteľnosť projektu aj po skončení realizácie projektu.

3.4 Realizácia projektu

Pre dosiahnutie každého cieľa bola určená jedna aktivita, ktorá sa mala realizovať v presne stanovenom mesiaci. Grant umožnil zakúpenie základných sád micro:bitov a príslušenstva, ktoré boli a v súčasnosti aj sú využité na informatickom krúžku, na hodinách informatiky, ale aj na hodinách chémie, fyziky, matematiky a i. Projekt bol realizovaný polovicu času svojej tvorby dištančne, pretože do hry vstúpila pandémia a s ňou i zatváranie škôl. Realizácia bola možná aj na základe rozhodnutia pani riaditeľky, ktorá zahrnula do rozvrhu vyučovanie informatiky. Na mnohých školách sa tak nestalo, o čom sa hovorilo aj na Rodičovskom združení ENTER [11] a následne v rôznych médiách napr. Tretina základných škôl na 2. stupni počas dištančného vzdelávania vôbec nevyučovala informatiku, ukazuje prieskum [12].

Programovanie micro:bitov prvé mesiace prebiehalo prioritne v prostrediach Microbit Classroom, Microsoft Teams v Priradených úlohách. V oboch prostrediach mohol učiteľ vidieť tvorbu žiakov v aktuálnom čase. Žiaci boli témou projektu nadšení. Pred nimi sa objavila výzva, ktorá im bola známa (veď v meste už bol každý, aj križovatky už videli, pouličné lampy a závory tiež), ale predsa to bolo niečo, čo až tak do hĺbky nikdy programátorsky nepoznali a neriešili. Žiaci sa snažili programovať na mobilných zariadeniach, na počítačoch, notebookoch a čo je najdôležitejšie, výsledky svojej tvorby mohli aspoň čiastočne vidieť ihneď, pretože v programovacom prostredí sa nachádza simulátor. Ten ale nedokázal simulovať pohyb robotov, svietiť skutočnými farbami semafora a aj práve preto si niektorí žiaci naozaj priali, aby už boli v škole. No vtedy mestečko ešte neexistovalo. Po návrate do školy ostávali už len tri mesiace do odovzdania projektu. Žiaci tretieho ročníka sa pustili do tvorby mestečka, a stavebnicu Lego oživovali svojimi nápadiami, montovali micro:bity k semaforom a pouličným lampám (pozri Obrázok 1). Tvorivosť nenechala na seba dlho



Obrázok 1: Tretiaci v akcii

čakať. Žiaci boli nápadití a vznikalo mnoho skvelých situácií. Napr. autorka projektu sa pýta žiaka: „Čo práve vyrábaš? Čo z toho bude?“ a žiak hovorí: „No predsa parkovisko! Tu by mohla byť rampa. Pani učiteľka, nemáme nejakú rampu?“ A pani učiteľka, dovtedy ani netušila, že micro:bitová závora nájde hneď takéto využitie. Aj takýmto spôsobom vzniklo množstvo budov a scén v mestečku.

Žiaci tretieho ročníka jeden deň doniesli z prírodných a iných materiálov (zo šišiek, z konárikov, z polystyrénu, ...) lesné táborisko (pozri Obrázok 2), čo otvorilo nové obzory a rozšírilo mestečko o ďalšie skvelé prvky, kde boli využité sady IoT s rôznymi rozširujúcimi senzormi.



Obrázok 2: Táborisko

4 INTELIGENTNÉ INTERAKTÍVNE MESTEČKO

Prepojením činností prezenčnej aj dištančnej výučby vzniká INTELIGENTNÉ INTERAKTÍVNE MESTEČKO **Ždaňa-City** (v súčasnosti už aj **Ždaňa-City:TheNextGeneration**). Je to model micro:bitového mesta², ktoré má

- funkčné križovatky (rádiovo riadené semaforey – ak sa na jednom semafore rozsvieti červená, na druhom naskočí zelená, navyše so sekundovým odpočítavaním, alebo semafor pre chodcov aj so zvukovým efektom pre hendikepovaných ľudí),
- sklápacie závory,
- budovy (v ktorých sa v náhodných intervaloch rozsvetuje svetlo),
- pouličné lampy (ktoré reagujú na svetelný senzor – čím väčšia tma, tým jasnejšie svetlo),
- autíčka (roboty ovládané na diaľku iným micro:bitom alebo mobilom),
- autonómne autíčka (so senzorom, ktorý sníma a sleduje nakreslenú čiaru),
- veternú turbínu, snímač teploty v priestore, kompas, meranie vlhkosti pôdy a podobne.

Niektoré prvky mestečka sú opísané v nasledovných podkapitolách.

4.1 Semaforey s rovnakým zobrazením svetelnej signalizácie

Semaforey s rovnakým zobrazením svetelnej signalizácie a číselného odpočítavania na LED matici (hardvér **STOP:bit**) využívajú rádiovú komunikáciu, kvôli správnej synchronizácii svetelnej signalizácie a číselného odpočítavania. V programe sa využíva volanie funkcie z dôvodu správneho otočenia čísel. V mestečku sú tieto semaforey umiestnené pred prechodom pre chodcov (pozri Obrázok 3).

Medzipredmetové vzťahy: občianska výchova – bezpečnosť na cestách, fyzika – LED, zapojenie obvodov, matematika – premenná, postupnosť.



Obrázok 3: Semaforey s rovnakým zobrazením

4.2 Semaforey so striedavým zobrazením svetelnej signalizácie

Semaforey so striedavým zobrazením svetelnej signalizácie (hardvér **STOP:bit**) sú umiestnené na ulici s čiastočne obmedzenou premávkou, kde je potrebné regulovať dopravu obojsmerne. Ak sa na semafore rozsvieti červená, odosiela signál druhému micro:bitu, aby rozsvietil oranžovú a vzápätí zelenú (pozri Obrázok 4).

Medzipredmetové vzťahy: občianska výchova – bezpečnosť na cestách, fyzika – LED, zapojenie obvodov, matematika – premenná, postupnosť.

² Video Ždaňa-City: <https://lnk.sk/cxiw>



Obrázok 4: Semaforey so striedavým zobrazením

4.3 Pouličné lampy – rádiová komunikácia

Pouličné lampy (hardvér **LAMP:bit**) komunikujú rádiovou komunikáciou. Hlavný vysielateľ je umiestnený pod zdrojom svetla (stolová lampa). Ak má vysielateľ svetla dostatok, tak odosiela signál prijímačom, teda všetkým pouličným lampám, že je deň a svietiť nepotrebujú. Ak ale hlavný vysielateľ nemá dostatok svetla, tak sa všetky lampy v rovnakom čase rozsvietia (pozri Obrázok 5).

Medzipredmetové vzťahy: občianska výchova – bezpečnosť na cestách, fyzika – LED, zapojenie obvodov.



Obrázok 5: Pouličné lampy – rádiová komunikácia

4.4 Závory

Závory (hardvér **ACCESS:bit**) – ako by to vyzeralo, keby mestečko nemalo parkovacie zóny. Táto závara funguje na jednoduchom princípe otvárania a zatvárania závory v určitých časových intervaloch so sprievodným upozorňujúcim textom (pozri Obrázok 6).

Medzipredmetové vzťahy: občianska výchova – bezpečnosť na cestách, fyzika – servo motory, zapojenie obvodov, matematika – časový interval.

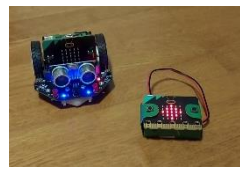


Obrázok 6: Závory

4.5 Robotické autíčko

Robotické autíčko (hardvér **Micro:Maqueen Lite**) – rýchlosť autíčka je nízka, pretože je určené na pohyb po uliciach mestečka. Nemôže mať demolačnú rýchlosť. Pohyb umožňuje diaľkové ovládanie, využívajúce rádiovú komunikáciu. Pri naklonení sa aj na ovládači a aj na autíčku zobrazuje smer, ktorým sa ubera. Navyše autíčko aktívne využíva smerové svetlá (pozri Obrázok 7).

Medzipredmetové vzťahy: občianska výchova – bezpečnosť na cestách, fyzika – LED, servo motory, zapojenie obvodov, robotika.



Obrázok 7: Robotické autíčko

4.6 Tanečníčka roztleskavačka

Tanečníčka roztleskavačka (hardvér **Ring:bit Car**) je umiestnená na konci pretekárskej dráhy. Keď pretekár dorazí do cieľa, tak sa robot rozbliká. Rýchlosť otáčok servo-motorov je vyberaná z intervalu, čo má za následok náhodný nepredvídateľný pohyb – tanec môže sprevádzať melódia (pozri Obrázok 8).

Medzipredmetové vzťahy: fyzika, hudobná výchova, robotika, matematika.



Obrázok 8: Tanečníčka roztleskavačka

4.7 Robot ovládaný mobilnou aplikáciou

Robot ovládaný mobilnou aplikáciou – pripojenie cez Bluetooth (hardvér **:MOVE mini Buggy**) – na správne fungovanie sú potrebné mobilné aplikácie Kitronik a Microbit. Ak bolo pripojenie úspešné, tak sa autíčko usmievalo. Na tejto pretekárskej dráhe sa robili skutočné preteky, kde sa žiakom stopoval a zapisoval dosiahnutý čas (pozri Obrázok 9).

Medzipredmetové vzťahy: fyzika, hudobná výchova, robotika.

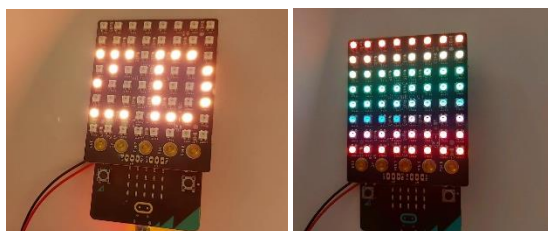


Obrázok 9: Pretekárske robotické autíčko

4.8 Informačný panel

Informačný panel (hardvér **ZIP Tile**) – mestečko má aj kultúrne vyžitie vo forme futbalového ihriska, nad ktorým sa týči displej – matica veľkosti 8x8. Na nej sme zobrazovali rôzne údaje od stavu futbalového zápasu až po teplotu v mestečku (pozri Obrázok 10).

Medzipredmetové vzťahy: fyzika, biológia, občianska výchova



Obrázok 10: Informačný panel

4.9 Veterná turbína

Veterná turbína (hardvér **Smart Science IoT Kit**) – mestečko bolo doplnené o časť vytvorenú prevažne z prírodných materiálov – má charakterizovať les, prírodu, život v prírode. Obyvateľ lesnej chaty využíva energiu z prírodných zdrojov. Pri rieke je symbolicky umiestnená vodná elektrárňa i veterná turbína (pozri Obrázok 11). Medzipredmetové vzťahy: fyzika, biológia, chémia.



Obrázok 11: Veterná turbína

4.10 Inteligentné zavlažovanie rastlinky

Inteligentné zavlažovanie rastlinky (hardvér **Smart Home Kit**) – senzor vlhkosti je vložený do pôdy, micro:bit meria a vyhodnocuje stav pôdy a podľa nameraného výsledku sa rozhodne či spustí činnosť čerpadla, ktoré má rastlinku polievať. Prvotné utopenie rastlinky žiakov nútilo zamyslieť sa ako nastaviť dávkovanie vody. Medzipredmetové vzťahy: fyzika, biológia, chémia, matematika (pozri Obrázok 12).



Obrázok 12: Samozavlažovanie rastlinky.

Programy vytvorené v MakeCode čitateľ nájde na webovom sídle Základnej školy, Jarmočná 96, Ždaňa [13]. Alebo čoskoro v pripravovanom e-learningovom kurze Moodle UPJŠ.

Prepojenie viacerých predmetov – matematika, fyzika, občianska náuka, chémia, biológia, výtvarná výchova, hudobná výchova, ... žiakom ukázalo, aké je dôležité mať všeobecný prehľad. Rôzne vedomosti žiaci spojili do jedného spoločného projektu, v ktorom využili micro:bity, stavebnicu Lego a iné prírodné materiály.

5 VYHODNOTENIE PROJEKTU A JEHO PROPAGÁCIA

Autorka článku tvrdí, že hlavný cieľ projektu sa žiakom aj jej samotnej podarilo splniť nad rámec spokojnosti a s ním aj ďaleko viac. „Prekonali sme samých seba, keď sme aj napriek náročnej pandemickej situácii, ktorá nás zasiahla zvládli zdanlivo nemožné – okrem toho, že projekt pomohol zlepšiť materiálne zabezpečenie školy, zaviedol do výučby ďalšiu formu zážitkového učenia, zaujal žiakov, naučil ich základy programovania nenásilnou hrovou formou, nerobil rozdiely medzi žiakmi, zvýšil aj povedomie verejnosti o školy – boli sme pozvaní na **Rodičovské združenie s ENTERom**, objavili sme sa v mnohých médiách (pozri Obrázok 13), inšpirovali mnohých učiteľov cez lektorovanie Učíme

s hardvérom, prezentovali projekt na Klube učiteľov informatiky cez UPJŠ v KE, dokonca sme s projektom postúpili na medzinárodný festival vedy **Science on Stage** do Prahy (marec 2022) (pozri Obrázok 14).



Obrázok 13: Rodičovské združenie s ENTERom



Obrázok 14: Science on Stage Praha 2022

Výsledok projektu sa prejavil ako úžasná interaktívna učebná pomôcka, v ktorej vidíme dlhodobú perspektívu. Mestečko sme nechceli zbúrať. Navyše žiaci mali množstvo ďalších výborných nápadov, preto našou ďalšou úlohou bolo získať ďalšie sady micro:bitov a HW.

Zapojili sme sa do 2. ročníka grantovej výzvy ENTER pre školy. [14] a zakúpili ďalšie sady – dron AIR:bit, sklenník, smart hodinky, WiFi moduly a podobne (pozri Obrázok 15).



Obrázok 15: QR kód Ždaňa-City:TheNextGeneration

Využitie micro:bitov je neobmedzené a učiť programovanie, ktorého programy sú prenosené do reálneho života (nie iba obrazkový výstup) má obrovský význam a úspech. Navyše kombinácia micro:bitu, prostredia Scratch a IoT ponúka ďalšie a ďalšie možnosti využitia. Žiaci si navyše sami tvoria príbeh mestečka – dopĺňajú ho ďalšími prvkami stavebnice LEGO a tým definujú ďalšie problémy a navrhujú ich riešenia. Najnovšie chceli žiaci mestečko chrániť – situáciu v mestečku i mimo neho monitorujeme prieskumným dronom Air:bit s WiFi kamerou. Žiaci

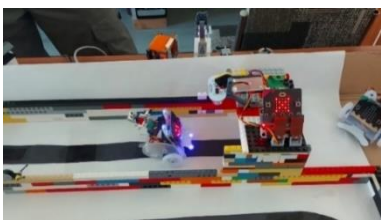
majú množstvo skvelých nápadov, s ktorými sa chcú zapojiť aj do **ENTER programiády** [8]. Nasledujúce obrázkové ukážky predstavujú aké rozšírenia v mestečku vznikli:

- **Predajňa s alarmom** – ak je alarm aktívny a dvere sa otvoria, tak sa spustí alarm. Na krúžku vznikla vtipná animácia (pozri Obrázok 16 **Error! Reference source not found.**).



Obrázok 16: Predajňa s alarmom

- **Vylepšená pretekárska dráha** – autonómny pohyb autička po dráhe, v celi sa nachádza sonar:bit, ktorý ak zachytí pohyb autička, tak spustí víťaznú melódiu a odošle signál roztlieskavačke, ktorá začne farebne blikať a náhodne sa pohybovať, čo pôsobí ako tanec (**Ring:bit Car**) (pozri Obrázok 17).



Obrázok 17: Vylepšená pretekárska dráha

- **Semafor pre chodcov** so zvukovým znamením pre hendikepovaných (**STOP:bit**) (pozri Obrázok 18).



Obrázok 18: Semafor pre chodcov

- **Sklenník** – kontroluje vlhkosť pôdy, vzduchu, svetlom podporuje rýchly rast a polieva podľa potreby (**Smart Greenhouse Kit**) (pozri Obrázok 19).



Obrázok 19: Sklenník

- **Dron** – zvuk motorov pripomínal bzučanie tisícok včiel, preto vznikla myšlienka simulácie letu včielky (**AIR:bit**) (pozri Obrázok 20).



Obrázok 20: Dron

- Smart programovacie hodinky (**Smart Coding Kit: Wearable Power Supply Extension Kit**) (pozri Obrázok 21).



Obrázok 21: Smart programovacie hodinky

- časť mestečka – kolotoč, semafor, pouličné lampy, policajné auto, v pozadí pretekárska autonómna dráha (pozri Obrázok 22).



Obrázok 22: Časť mestečka

Na základnej škole, kde sme projekt realizovali sú žiaci z rôznorodých prostredí, ktoré im ponúkajú rôzne vzorce správania, žiaci s vývinovými poruchami i žiaci s ADHD. Všetci potrebujú kvalitné vzdelanie, osobitý prístup. Chceme okrem základov programovania micro:bitov zahrnúť do výučby využitie rozširujúcich HW sád a návrh a riešenia STEAM projektov (Science, Technology, Engineering, Art, Mathematics čo v preklade znamená Veda, Technika, Inžinierstvo, Umenie, Matematika). Tieto aktivity by sme chceli realizovať aj na informatickom krúžku. Plánujeme žiakom zapožičiavať sady micro:bitov aj domov. A našim cieľom je, aby sa naučili vážiť si veci, vzájomnej súhre, tvoreniu, rešpektovaniu názorov iných, argumentácii, zlepšovať prezentačné zručnosti pri propagácii projektu, zvýšiť povedomie blízkeho okolia školy a pritiahnúť pozornosť okolitých škôlok a škôl a vyvolať tak záujem o vzdelávanie na našej škole.

Najkrajšia myšlienka, ktorá zaznievala počas dňa otvorených dverí mestečka, keď sa žiaci stali súčasťou mestečka bola „**chcel by som byť postavičkou v tomto mestečku**“, no nesplnili by ste im aj tento sen? Použitie GreenScreen na krúžku (pozri Obrázok 23).



Obrázok 23: Obyvatelia mestečka

6 ZÁVER

Pre učiteľa nie je nič krajšie ako keď žiaci spomínajú na projekt s úsmevom, keď zážitky a poznatky z neho rezonujú ešte ďalšie roky. Keď učiteľ spomenie žiakom, že ideme pracovať na projekte a žiaci nadšené vykrikujú a hneď sa bez vyzvania učiteľa delia do skupín a plánujú. Keď zazvoní a oni chcú ešte ostať na vyučovaní, keď vedomosti, poznatky a zručnosti z neho využívajú aj na iných predmetoch, alebo pri tvorivej práci doma. Navyše, keď sa projekt dostane nielen na národnú, ale aj na medzinárodnú úroveň, tak je to veľká pocta aj pre učiteľa a jeho školu. Chceme podporiť učiteľov, aby zakomponovali do vyučovania projekty, aby sa nebáli vyjsť zo svojho „starého ja“ a dali priestor žiakom, oni to dokážu.

Väčšina príbehov má „to be continued“. Ani ten náš nekončí. Pomocou senzorov z micro:bitových sád vieme meraním a zaznamenaním získať množstvo dát, s ktorými vieme ďalej pracovať. Navyše pripravujeme e-learningový kurz v prostredí LMS MOODLE s názvom Tvorba e-learningovej podpory pre učiteľov informatiky v oblasti programovania SMART CITY s BBC micro:bit. K tomuto kurzu už máme aj informačný plagát (pozri Príloha).

POĎAKOVANIE

Zrealizované projekty *Ždaňa-City* a *Ždaňa-City: The Next Generation* podporil Nadačný fond Telekom pri Nadácii Pontis.

Tento článok bol vytvorený v rámci projektu KEGA 012UPJŠ-4/2021 *Vývoj digitálnej knižnice interdisciplinárnych STEAM projektov a jej implementácia do informatického, matematického a prírodovedného vzdelávania na stredných školách* a projektu VVGS-IPEL *Tvorba e-learningovej podpory pre učiteľov informatiky v oblasti programovania SMART CITY s BBC micro:bit*.

BIBLIOGRAFICKÉ ODKAZY

- [1] BAJTOŠ, Ján. *Teória a prax didaktiky*. Žilina : vydavateľstvo žilinskej univerzity – EDIS, 2003. 384 s. ISBN 80-8070-130-X.
- [2] TUREK, Ivan. *Didaktika*. 3. preprac. a dopl. vyd. Bratislava: Wolters Kluwer, 2014, 386 s. Škola, 15. ISBN 978-80-8168-004-5.
- [3] ČAPEK, Robert. *Moderní didaktika: lexikon výukových a hodnoticích metod*. Praha: Grada, 2015. Pedagogika (Grada). ISBN 978-80-247-3450-7.

- [4] TOMKOVÁ, Anna, Jitka KAŠOVÁ a Markéta DVOŘÁKOVÁ. *Učíme v projektech*. Praha: Portál, 2009. ISBN 978-80-7367-527-1.
- [5] TKÁČOVÁ, Zuzana. *Projektové vyučovanie v informatike*. [online] [cit. 2023-02-08] Dostupné na: https://mfi.upol.cz/files/27/2703/mfi_2703_223_230.pdf
- [6] *Charakteristika projektového vyučovania*. [online] [2023-02-08] Dostupné na: https://kekule.science.upjs.sk/chemia/mvp_net/projektove%20Ovyucovanie/menu.html
- [7] ENTER pre školy 2020 – Digitálna generácia. [online] [cit. 2023-02-08] Dostupné na: <https://www.nadaciapontis.sk/wp-content/uploads/2020/09/Grantova-vyzva-microbity-pre-skoly-2020.pdf>
- [8] MANSELL Marek a kol. *BBC micro:bit ako súčasť projektu ENTER*. DIDINFO 2021 Univerzita Mateja Bela, Banská Bystrica. 184 s. ISBN 978-80-557-1823-1 ISSN 2454-051X Dostupné na: http://www.didinfo.net/images/DidInfo/files/DIDINFO_2021_zbornik.pdf
- [9] ENTER [online] [cit. 2023-01-23] Dostupné na: <https://enter.study/>
- [10] Učíme s Hardvérom [online] [cit. 2023-01-23] Dostupné na: <https://www.ucimeshardverom.sk/>
- [11] Rodičovské združenie s ENTERom [online] [cit. 2023-01-23] Dostupné na: https://www.youtube.com/watch?v=aMI_IdR8h_o
- [12] Tretina základných škôl na 2. stupni počas dištančného vzdelávania vôbec nevyučovala informatiku, ukazuje prieskum [online] [cit. 2023-01-23] Dostupné na: <https://www.startup.sk/tretina-zakladnych-skol-na-2-stupni-pocas-distancneho-vzdelavania-vobec-nevyucovala-informatiku-ukazuje-prieskum/>
- [13] PROJEKTY – Ždaňa-City micro:bity | Základná škola, Jarmočná 96, Ždaňa [online] [cit. 2023-02-08] Dostupné na <https://zszdana.edupage.org/a/projekty?eqa=dGV4dD10ZXh0L3RleHQyNCZzdWJwYWdlPTUy%3D>
- [14] ENTER pre školy 2021. *Vyhlasujeme 2. ročník grantovej výzvy ENTER pre školy. Zapojte sa a vyučujte inovatívne!* [online] [cit. 2023-02-08] Dostupné na: <https://www.nadaciapontis.sk/vyzvy/vyhlasujeme-2-rocnik-grantovej-vyzvy-enter-pre-skoly-zapojte-sa-a-vyucujte-inovativne/>

PRÍLOHA

BBC micro:bit SMART CITY

micro:bit

BBC micro:bit je programovateľný mikroprocesor. Stačí vytvoriť jednoduchý program v prostredí Makecode a nahráť ho pomocou USB kábla do micro:bitu. Má niekoľko zabudovaných senzorov, napr.: kompas, teplomer, alebo akcelerometer.

K micro:bitu vieme pripojiť vstavaných výstupných pínov pomocou takmer akýchkoľvek hardvérov a elektroniky - LED pásiky, motorčkové, rôzne senzory, robotické autá, ako je LAMP:bit, ACCESS:bit, tak inteligentné mostičko SMART CITY.

STOP:bit
 podrobnosti od výrobcu: <https://lnk.sk/ta6>

Ukážkový program používa volanie funkcie - odpodávanie na microbite.

ZAMYSLIENIE:
 Ako by vyzeral kód programu využívajúci rádiové spojenie pre semafofy s rovnakým svietením a ako so striedavým svietením?

ACCESS:bit
 podrobnosti od výrobcu: <https://lnk.sk/rv1s>

Závora ACCESS:bit sa otvára a zatvára v určitých intervaloch.

ZAMYSLIENIE:
 Ako by vyzeral program s analógovým zapisovaním - čím menej svetla na fototranzistor dopadá, tým LED dióda jasnejšie svieti?

LAMP:bit
 podrobnosti od výrobcu: <https://lnk.sk/anak>

Fototranzistor na LAMP:bite reaguje na množstvo svetla, ktoré na neho dopadá - ak hodnota klesne pod nastavenú hodnotu, tak LED dióda svieti.

ZAMYSLIENIE:
 Ako by vyzeral program s analógovým zapisovaním - čím menej svetla na fototranzistor dopadá, tým LED dióda jasnejšie svieti?

ZAMYSLIENIE:
 Ako by vyzerali programy, ktoré by využívali rádiové spojenie a to tak, že jedným micro:bit "vysielač" by analyzoval množstvo dopadajúceho svetla a ovládal by svietenie všetkých "prijímačov" LAMP:bit-ov?

SMART CITY bez autíčk? Určite NIE!
 Programovateľné roboty sa môžu pohybovať autonómne, alebo byť rádiovým ovládané iným microbitom poprípade mobilným zariadením.

2,4 GHz anténa rádiová komunikácia

micro:Maqueen
 podrobnosti od výrobcu: <https://lnk.sk/rzvz>

Ak chceme autíčko diaľkovo ovládať iným microbitom odporúčame na programovanie použiť multieditor.

ZAMYSLIENIE:
 Ako by vyzeral program, ktorý by komunikoval s ultrazvukovým senzorom - ak by sa vozidlo priblížilo, tak závora by sa otvorila?

ZIP Tile
 podrobnosti od výrobcu: <https://lnk.sk/fast>

Ovládanie mobilom alebo tabletom? Zladen problém.

Na displeji sa vypisuje text SMART CITY. Displej po stlačení tlačidla A mení farby. Každá LED je samostatne programovateľná.

:MOVE mini buggy
 podrobnosti od výrobcu: <https://lnk.sk/sd4>

ZAMYSLIENIE:
 Ako by vyzeral program, ktorý by navyše zisťoval vzdialenosť od prekážky a včas sa jej vyhol?

Ring:bit Car
 podrobnosti od výrobcu: <https://lnk.sk/rfnp>

Ring:bit car vykonáva náhodný pohyb a LED Light Bar náhodne svieti - efekt tanečníka.

ZAMYSLIENIE:
 Ako by vyzeral program, ktorý by navyše zisťoval vzdialenosť od prekážky a včas sa jej vyhol?

Air:bit dron
 podrobnosti od výrobcu: <https://lnk.sk/egg1>

možnosť pripojiť príslušenstvo: Sonar:bit - na vyhľadanie sa prekážkam, Tracking Module - nasledovanie ciary LED Light Bar - 8 LED diód (svetlomety)

Vytvorené v rámci projektu KEGA 012UPJŠ-4/2021 Vývoj digitálnej knižnice interdisciplinárnych STEAM projektov a jej implementácia do informatického, matematického a prírodovedného vzdelávania na stredných školách.

Súťaž iBobor - analýza obtiažnosti úloh určených pre nevidiacich študentov stredných škôl

Bebras challenge - analysis of the difficulty of tasks designed for secondary blind students

Mária Čujdíková
KDMFI, FMFI, UK
Mlynská dolina
84248 Bratislava
Slovenská Republika
maria.cujdikova@fmph.uniba.sk

Ludmila Jašková
KDMFI, FMFI, UK
Mlynská dolina
84248 Bratislava
Slovenská Republika
ludmila.jaskova@fmph.uniba.sk

Mária Stankovičová
CPŠ, UK
Mlynská dolina
842 48 Bratislava
Slovenská republika
maria.stankovicova@rec.uniba.sk

ABSTRACT

In this paper, we analyse the difficulty of the informatics Bebras tasks for the category of upper secondary blind students. We tried to compare the predicted difficulty determined before the competition with the measured difficulty found according to the achieved results of the participants. Our goal was to find common characteristics of the tasks at each level of difficulty. We focused on the topic area covered by the task, the input information presented in the task, and required skills and knowledge needed to solve the task.

Keywords

Blind students. Informatics contest. Difficulty level. Inclusion. Computing education.

ABSTRAKT

V našom príspevku sa venujeme analýze obtiažnosti úloh zo súťaže iBobor v špeciálnej kategórii Nevidiaci SŠ, určenej pre nevidiacich študentov vyššieho sekundárneho vzdelávania. Pokúsili sme sa porovnať predpokladanú obtiažnosť stanovenú pred súťažou s nameranou náročnosťou, získanou na základe úspešnosti žiakov pri riešení úloh. Naším cieľom bolo nájsť spoločné charakteristiky úloh na jednotlivých úrovniach obtiažnosti. Zamerali sme sa na tematickú oblasť, ktorej sa úloha týkala, na vstupné informácie uvedené v úlohe a na požadované zručnosti a vedomosti potrebné na riešenie úlohy.

Kľúčové slová

Nevidiaci študenti. Informatická súťaž. Úroveň obtiažnosti. Inklúzia. Výučba informatiky.

1 ÚVOD

Hlavným zámerom súťaže iBobor je podnietiť záujem žiakov o spoznávanie informatických konceptov a podporiť rozvoj ich informatického myslenia [1, 2]. Testy a kvízy sú efektívnym nástrojom učenia sa, ak žiaci v relatívne krátkom čase získajú spätnú väzbu a dozvedia sa správne riešenie [3]. Súťaž iBobor spĺňa tieto kritériá. Súťažiaci sa dozvedia svoje skóre bezprostredne po odoslaní riešenia a vzorové riešenia s vysvetlením sú im k dispozícii v priebehu niekoľkých dní. K efektívnej príprave súťažiacich prispieva aj online archív úloh z predchádzajúcich ročníkov súťaže.

Hoci má súťaž ambíciu osloviť všetkých žiakov, úlohy nie sú vo všeobecnosti prístupné pre nevidiacich. Títo žiaci používajú pri práci s počítačom čítač obrazovky a jediné informácie, s ktorými

dokážu pracovať, sú text a zvuk. Na ovládanie počítača nepoužívajú myš, ale vstupy zadávajú výlučne pomocou klávesnice. Pokiaľ ide o súťaž iBobor, nevidiaci žiaci sa jej nemôžu zúčastniť aj napriek tomu, že na hodinách informatiky dosahujú porovnateľné výsledky s intaktnými žiakmi. Úlohy v súťaži iBobor obsahujú viacero prvkov, ktoré sú pre nevidiacich neprístupné. Patria medzi ne obrázky, farby, interaktívne prvky a podobne. Aby sa do súťaže mohli zapojiť aj nevidiaci žiaci, je potrebné urobiť niektoré úpravy úloh a pravidiel [4, 5], ktoré priblížime v časti 2. V časti 3 predstavíme prehľad niekoľkých výskumov zameraných na analýzu obtiažnosti úloh iBobor pre intaktných žiakov.

Súťaž Bebras sa koná v takmer šesťdesiatich krajinách, ale Slovensko je jedinou krajinou, ktorá umožňuje nevidiacim žiakom zúčastniť sa súťaže bez asistencie vidiacej osoby. Je to vďaka špeciálnym kategóriám pre nevidiacich. Pre nevidiacich žiakov ponúkame dve samostatné kategórie: jednu pre nevidiacich žiakov nižšieho sekundárneho vzdelávania (Nevidiaci ZŠ) a druhú pre nevidiacich žiakov vyššieho sekundárneho vzdelávania (Nevidiaci SŠ). V našom príspevku sa zameriame na kategóriu Nevidiaci SŠ, pretože kategórii Nevidiaci ZŠ sme sa venovali v iných príspevkoch [4, 5, 6]. Kategóriu Nevidiaci SŠ sme žiakom prvýkrát ponúkli v roku 2016.

Počas existencie tejto kategórie sme upravili už 63 úloh. Všimli sme si, že nameraná obtiažnosť týchto úloh sa podľa očakávania líšila od odhadovanej obtiažnosti. Aby sme ju v budúcnosti dokázali lepšie odhadnúť, rozhodli sme sa skúmať faktory, ktoré môžu mať na obtiažnosť úloh vplyv.

Všetky úlohy sme podrobne analyzovali. Zamerali sme sa na špecifikáciu príslušnosti k tematickej oblasti a identifikáciu vstupov a požadovaných vedomostí a zručností potrebných na riešenie úloh. Sledovali sme aj nameranú obtiažnosť úloh a porovnávali sme ju s očakávanou obtiažnosťou. Našu výskumnú vzorku a výskumné otázky uvádzame v časti 4. Výsledky našej analýzy sú v časti 5.

2 IBOBOR PRE NEVIDIACICH

Ako sme uviedli v predchádzajúcej časti, aby sa nevidiaci študenti mohli zúčastniť súťaže iBobor, bolo nevyhnutné vykonať niekoľko úprav [7]. Jednou z nich je poskytnutie väčšieho časového intervalu na riešenie úloh [8]. Preto nevidiaci študenti riešia 9 úloh (3 ľahké, 3 stredne ťažké a 3 ťažké) v časovom intervale 40 minút. Majú tak v priemere o 60% viac času ako intaktní, ktorí majú za rovnaký čas vyriešiť 15 úloh (5 ľahkých, 5 stredne ťažkých, 5 ťažkých).

V kategórii Nevidiaci SŠ sa dá súťažiť od roku 2016. V tom roku mali účastníci zobrazené zadania úloh v textovom dokumente. Od roku 2017 mohli súťažiť online a úlohy boli zobrazené vo webovom prehliadači. Môžu však používať aj textový editor na zápis poznámok a jednotlivých krokov riešenia. Okrem toho môžu nevidiaci žiaci používať aj ceruzku a euroobal na kreslenie reliéfnych obrázkov a tiež reliéfnu tabuľku pri riešení úloh zameraných na programovanie pohybu.

Úlohy pre nevidiacich žiakov [9] vznikli adaptáciou úloh určených pre ich intaktných rovesníkov (kategórie Juniori a Seniori [10]). Pri adaptácii úloh sme sa snažili zachovať podstatu problému a vykonať len nevyhnutné zmeny. Niektoré úlohy sa dali použiť v pôvodnom znení, ale väčšina úloh si vyžadovala menšie, či väčšie úpravy nasledujúceho charakteru.

- Nahradiť obrázky obsahujúce relevantnú informáciu textom.
- Nepoužívať farby na odlišenie objektov.
- Používať tabuľky so správnym lineárnym poradím, prípadne ich nahradiť textom alebo zoznamom.
- Nepoužívať interaktívne úlohy, ktoré sa dajú riešiť len pomocou myši.
- Používať kratšie postupnosti objektov, pretože nevidiaci účastníci si musia tieto postupnosti zapamätať.
- Zmeniť úroveň obťažnosti, ak je potrebné si pamätať väčšie množstvo informácií.

3 PREHĽAD PROBLEMATIKY

Ako sme uviedli v predchádzajúcej kapitole, žiaci riešia úlohy rôznej obťažnosti – ľahké, stredne ťažké a ťažké. Úlohy s rozdielnou obťažnosťou sú hodnotené rozdielnym počtom bodov. Je preto dôležité, aby autori úloh dokázali správne odhadnúť úroveň obťažnosti. Analýze obťažnosti úloh určených pre intaktných žiakov sa venovalo vo svojich článkoch množstvo autorov z rôznych krajín, v ktorých dlhoročne prebieha súťaž. Spomenieme niektoré z nich.

Dagiené a Futschek [1] prezentovali zoznam kritérií, ktoré by mali spĺňať dobré úlohy v súťaži iBobor. Ide o všeobecné kritériá, ktoré používa medzinárodný organizačný výbor súťaže pri hodnotení vhodnosti úloh, dodaných z rôznych krajín. Z hľadiska obťažnosti ide o to, aby boli úlohy veku primerané. Kladie sa dôraz aj na to, aby boli úlohy zaujímavé a motivujúce. V niektorých prípadoch sa atraktivita úlohy zvyšuje, ak je problém prezentovaný pomocou príbehu. Táto forma môže prispieť k tomu, že ju ľahšie vyriešia aj mladší účastníci.

Lonati a kolektív autorov [11] skúmali vplyv drobných zmien v zadani algoritmických úloh na úspešnosť účastníkov pri ich riešení. Upozorňujú, že v zadaniach úloh je potrebné veľmi starostlivo zvoliť príklady a obrázky. Tvrdia, že namiesto toho, aby boli užitočné pre pochopenie a riešenie úlohy, môže byť ich účinok rušivý alebo skresľujúci.

Piati autori z piatich univerzít [12] analyzovali dáta o úlohách, ktoré v súťaži riešilo 115 400 študentov vo veku 8–18 rokov v siedmich štátoch. Ich štúdia poskytuje zaujímavý pohľad na celý rad otázok týkajúcich sa možného vplyvu rôznych školských systémov a pohlavia na úspešnosť účastníkov. Uvedomili si, že odhadovanie obťažnosti nových úloh je otvoreným problémom. Podľa autorov je potrebné lepšie pochopiť a charakterizovať rozsah algoritmických stratégií použitých pri súťaži a faktorov, ktoré spôsobili veľké rozdiely v úspešnosti pri riešení úloh medzi

rôznymi krajinami. Autori tvrdia, že koncepčný obsah nemá zásadný vplyv na náročnosť úloh.

Van der Wegt a kolektív analyzovali reálnu obťažnosť úloh v rámci holandskej súťaže Bebras. Výsledky publikovali v niekoľkých štúdiách [13, 14, 15]. Využili rôzne nástroje pri predpovedaní obťažnosti úloh. Pokúsili sa tiež analyzovať vzťah medzi náročnosťou obsahu, náročnosťou stimulov a obťažnosťou úlohy. Dospeli k záveru, že obťažnosť obsahu sa javí ako najmenej jasná položka pri predpovedaní reálnej obťažnosti. Pomocou kvantitatívnych metód dokázali autori potvrdiť tendenciu, že úlohy týkajúce sa reprezentácií údajov a dátových štruktúr, žiaci riešia lepšie, ako úlohy týkajúce sa algoritmov a programovania.

Vaniček [16, 17] skúmal faktory, ktoré preukázateľne zvyšujú obťažnosť úloh. Podľa neho sú takými faktormi formalizovaný zápis, zadania obsahujúce štruktúry, optimalizačné úlohy a úlohy vyžadujúce schopnosť čítať s porozumením. Autor tiež uvádza, že sa nepotvrdilo, že by mali na vyššiu obťažnosť úloh vplyv také faktory, ako je dĺžka zadania, použitie technickej terminológie, algoritmov, diagramov a záporných otázok.

Budinská, Mayerová a Veselovská [18] analyzovali úlohy a výsledky súťaže iBobor v kategórii Bobríci (2. - 4. ročník ZŠ), zozbierané počas šiestich rokov. Kvalitatívnymi a kvantitatívnymi výskumnými metódami autorky na základe analýzy textu vytvorili novú kategorizáciu úloh s cieľom lepšie pochopiť niektoré zákonitosti (a korelácie) medzi úlohami a výsledkami. Vytvorili štyri rôzne kategórie úloh: Úlohy orientované na programovanie, Algoritmické úlohy, Logické úlohy a úlohy s problematikou digitálnej gramotnosti. Pomocou kvalitatívnej analýzy úloh definovali aj niekoľko podkategórií a na základe analýzy výsledkov súťažiacich zistili rôznu koreláciu medzi ročníkom vzdelávania a pohlavím.

4 METODOLÓGIA

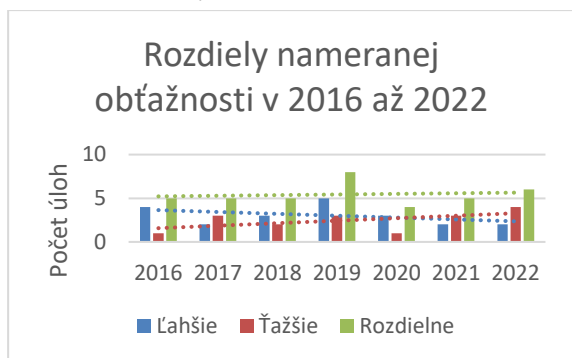
Pri analyzovaní výsledkov siedmich ročníkov súťaže iBobor v kategórii Nevidiaci SŠ sme zistili, že nami nameraná obťažnosť úloh bola iná, ako autori úloh očakávali (Obrázok 1). Primerane zvolená obťažnosť bola len pre 40% úloh. Ťažších bolo 27% úloh a ľahších 33% úloh.



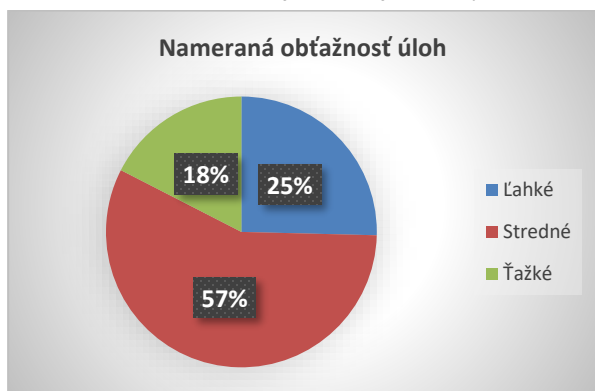
Obrázok 1: Graf percentuálneho podielu úloh s primeranou a odlišnou obťažnosťou od očakávanej

Možno konštatovať, že v priemere viac ako 50% úloh (Obrázok 2, zelená bodkovaná čiara) malo inú obťažnosť ako sme pôvodne očakávali. Môžeme si všimnúť, že počet úloh s nižšou obťažnosťou, ako sme očakávali mal z roka na rok klesajúcu tendenciu (modrá bodkovaná čiara). Naopak, počet úloh s vyššou obťažnosťou ako sme očakávali, mal rastúcu tendenciu (červená bodkovaná čiara).

Súťažiaci každoročne riešia 9 úloh (3 ľahké, 3 stredne ťažké a 3 ťažké). Takže sa dá predpokladať, že ku každej úrovni obťažnosti by mal prislúchať rovnaký počet úloh. Avšak na obrázku 3 vidíme, že iba 25% úloh bolo ľahkých, až 57% úloh bolo stredne ťažkých a iba 18% úloh bolo ťažkých.



Obrázok 2: Graf znázorňuje počet úloh s odlišnou obťažnosťou od očakávanej v rámci jednotlivých ročníkov



Obrázok 3: Graf percentuálneho podielu úloh v jednotlivých úrovniach obťažnosti

Aby sme dokázali v budúcnosti lepšie odhadnúť obťažnosť úloh, rozhodli sme sa preskúmať vplyv niektorých faktorov [19]. Hľadali sme odpovede na nasledujúce výskumné otázky.

O1: Aký je vzťah medzi tematickým zameraním úlohy a jej obťažnosťou?

O2: Aký je vzťah medzi požadovanými vedomosťami a zručnosťami potrebnými na riešenie úlohy a jej obťažnosťou?



Obrázok 4: Počet nevidiacich účastníkov v jednotlivých ročníkoch súťaže iBobor

Použili sme stratégiu prípadovej štúdie, pričom jednotlivými skúmanými prípadmi boli samotné úlohy a ich zadanie (63 úloh).

Pre každú úlohu sme vyjadrili **nameranú obťažnosť Q ako percentuálny podiel súčtu nesprávnych riešení a chýbajúcich riešení ku všetkým riešeniam** [20, 21]. Úlohy s hodnotou Q menšou ako 30 % sme vyhodnotili ako **ľahké**, úlohy s hodnotou Q od 30 % do 70 % ako **stredne ťažké** a úlohy s hodnotou Q od 70 % ako **ťažké**.

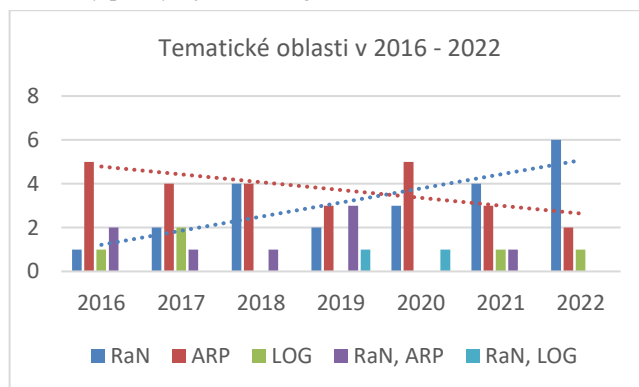
Analyzovali sme úspešnosť všetkých nevidiacich účastníkov vo veku 16 až 20 rokov. Ich počet sa pohyboval medzi 2 až 10 (Obrázok 4).

5 VÝSLEDKY VÝSKUMU

Všetkých 63 úloh sme podľa nameranej obťažnosti Q zaradili do troch skupín (viď predchádzajúca časť). Získali sme tak 16 ľahkých úloh, 36 stredne ťažkých a 11 ťažkých úloh. V rámci každej skupiny sme skúmali príslušnosť úloh k tematickým oblastiam a vedomosti a zručnosti potrebné na ich riešenie.

5.1 Príslušnosť úloh k tematickej oblasti

Počas existencie súťaže iBobor sa viacerí autorov pokúsilo kategorizovať úlohy podľa príslušnosti k tematickej oblasti informatiky. Prvú oficiálnu kategorizáciu navrhli v roku 2008 Dagiené a Futschek [1]. V roku 2017 ju Dagiené a kol. zrevidovali [22]. Na Slovensku sa však ujali kategórie uvedené v Štátnom vzdelávacom programe pre predmet informatika [18, 23]. Ide o nasledujúcich päť tematických okruhov: (1) reprezentácie a nástroje - **RaN**, (2) algoritmické riešenie problémov - **ARP**, (3) komunikácia a spolupráca, (4) softvér a hardvér, (5) informatická spoločnosť. Keď sme sa pokúsili priradiť úlohy určené pre nevidiacich súťažiacich do týchto kategórií, zistili sme, že žiadna úloha sa nedá priradiť do kategórií (3), (4) a (5). Väčšina úloh patrí do kategórií RaN a ARP. No niektoré úlohy sme nedokázali zaradiť ani do týchto dvoch kategórií. Boli to úlohy súvisiace s výrokovou logikou. Vytvorili sme preto ďalšiu kategóriu, ktorú sme nazvali logika - **LOG**. Našli sa úlohy, ktoré sa nedajú jednoznačne zaradiť len do jednej z uvedených troch kategórií, pretože sa v nich tematicky prekrývajú dve kategórie.



Obrázok 5: Príslušnosť úloh k tematickým oblastiam v jednotlivých rokoch

Na obrázku 5 vidíme, že väčšina úloh, ktoré súťažiaci riešili v jednotlivých ročníkoch súťaže, patrili do kategórií (1) a (2), alebo do oboch. Podobne aj väčšina úloh, až 89% z celkového počtu úloh, použitých počas siedmich rokov, patrili do kategórií (1) a (2), alebo do oboch (Obrázok 6).

Pre každú úroveň nameranej obťažnosti úloh sme analyzovali príslušnosť úloh k tematickým oblastiam (obr. 7). Môžeme konštatovať, že až 50% ľahkých úloh možno tematicky priradiť ku

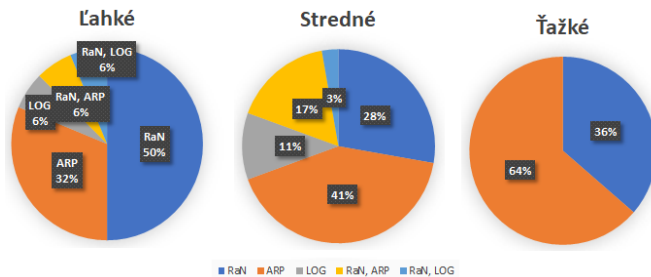
kategórii RaN. Len 32% ľahkých úloh je tematicky zameraných na oblasť ARP.

Medzi stredne ťažkými úlohami je vyššie percento úloh z tematickej oblasti ARP ako v skupine ľahkých úloh. Dokonca najviac úloh je z tejto oblasti. Rozdiel medzi úlohami zameranými na RaN a ARP je viac ako 10 %. V porovnaní s ľahkými úlohami je percentuálny podiel úloh priradených k tematickej oblasti RaN o viac ako 20 % nižší.



Obrázok 6: Kategorizácia úloh podľa tematických oblastí

V skupine ťažkých úloh dominujú úlohy z tematickej oblasti ARP. Len 36 % úloh patrí do tematickej oblasti RaN. Úlohy z iných oblastí sa v skupine ťažkých úloh nenachádzajú. **Je teda zrejmé, že príslušnosť k tematickej oblasti má vplyv na náročnosť úloh (odpoveď na otázku O1).**



Obrázok 7: Priradenie úloh s rôznymi úrovňami obtiažnosti k tematickej oblasti

5.2 Požadované zručnosti a znalosti

Viaceri autori [15, 16] považujú znalosti a zručnosti potrebné na riešenie úlohy za dôležitý faktor ovplyvňujúci jej obtiažnosť. Okrem toho je tiež dôležité, aké informácie sú dané v zadaní a čo je cieľom úlohy. Analyzovaním týchto aspektov sme sa pokúsili nájsť niektoré charakteristiky úloh na jednotlivých stupňoch obtiažnosti.

Na riešenie **ľahkých úloh** sú potrebné nasledujúce vedomosti a zručnosti:

- vykonávanie zadanej postupnosti krokov so znakovým reťazcom (maximálne 5 krokov),
- reprezentovanie postupnosti krokov pomocou podprogramu s parametrami (5 krokov, 3 parametre),

- usporiadanie činností do správnej časovej súslednosti (5 činností),
- dekódovanie zadaného kódu, ak je známy význam jednotlivých symbolov (4 symboly),
- šifrovanie textu, ak je daný vstupný text (maximálne 10 znakov) a postup modifikácie,
- prevod čísla z desiatkovej sústavy do dvojkovej alebo naopak (maximálne 5 cifier v dvojkovej sústave),
- určovanie vzťahov v hierarchickej štruktúre (2 až 3 úrovne),
- výber objektu vyhovujúceho podmienkam, pričom informácie o objektoch sú reprezentované kódom (6 objektov, kód objektu tvoria 2 znaky),
- vytváranie dvojíc objektov tak, aby boli splnené podmienky a platili uvedené výroky (8 objektov, 4 výroky).

Na riešenie **stredne ťažkých úloh** sú potrebné nasledujúce vedomosti a zručnosti:

- určovanie hodnoty prvkov v poli po vykonaní zadanej postupnosti krokov (maximálne 5 prvkov),
- hľadanie postupnosti krokov vedúcich k požadovanému stavu poľa,
- vykonávanie paralelných procesov:
 - určenie času potrebného na vykonanie paralelných procesov,
 - určovanie postupnosti krokov v dvoch paralelných procesoch,
- programovanie pohybu objektu po mriežke (maximálne rozmery 5x7 polí, 4-9 príkazov):
 - vykonávanie postupnosti príkazov,
 - vytváranie postupnosti príkazov,
 - hľadanie cesty spojené s optimalizáciou,
- vykonanie postupnosti krokov podľa pravidiel reprezentovaných v tabuľke,
- vytváranie postupnosti krokov popisujúcich pohyb v cyklickom grafe po definovanej ceste (graf s 8-9 vrcholmi),
- optimalizácia využitia zdrojov na výrobu objektov s maximálnou cenou,
- hľadanie opakujúceho sa vzoru:
 - hľadanie postupnosti opakujúcich sa procesov,
 - zápis opakujúcich sa príkazov pomocou príkazu cyklu,
- reprezentácia postupnosti prvkov:
 - určovanie počtu povolených operácií s dvojhodnotovými prvkami zoznamu na dosiahnutie želaného stavu,
 - určovanie zhodnosti zoznamov spĺňajúcich zadané podmienky,
 - určenie hodnoty prvku zadaného vnoreným formálnym zápisom,
- určenie kódov pre jednotlivé znaky, ak je známe slovo pred a po zakódovaní a podmienky, ktoré musia byť pri vytváraní kódu dodržané,
- šifrovanie textu, ak je známy len text pred zašifrovaním a po zašifrovaní, metóda šifrovania nie je uvedená,

- hľadanie cesty medzi dvoma vrcholmi grafu, vyhovujúcej zadaným podmienkam (graf s orientovanými hranami a 9 vrcholmi),
- usporadúvanie objektov na základe informácií o ich vzájomnom poradí (6 objektov),
- výber podreťazca spĺňajúceho jednoduchú podmienku (16 znakový reťazec),
- vytváranie dvojíc objektov tak, aby boli splnené podmienky a platili uvedené výroky (4 alebo 8 objektov, 3 až 4 výroky),
- výber objektu spĺňajúceho podmienky filtrovania (5 podmienok, 4 kategórie po 3 vlastnosti).

Na riešenie **ťažkých** úloh sú potrebné nasledujúce vedomosti a zručnosti:

- hľadanie stratégie:
 - výherná stratégia pri známych pravidlách hry a vstupnom stave,
 - stratégia delenia skupiny objektov na podskupiny podľa zadaných pravidiel,
- hľadanie postupnosti príkazov s využitím opakovania,
- hľadanie postupnosti krokov vedúcich k želanému koncovému stavu reťazca (3 kroky, 4 reťazce),
- práca s dvoma zásobníkmi vyžadujúca ich zlučovanie a krokovanie programu (6 prvkov v zásobníku),
- usporadúvanie prvkov v poli pomocou vzájomných výmen dvoch prvkov medzi sebou podľa predpísaných pravidiel (6 prvkové pole),
- výber prvkov postupnosti spĺňajúcich zadané podmienky (10 prvkové pole),
- výber vyhovujúceho kódu, ak je známych niekoľko iných kódov a podmienka, ktorá musí byť pri kódovaní dodržaná (nový kód nemôže byť prefixom známych kódov).

Ukázalo sa, že na riešenie úloh s rôznym stupňom obťažnosti boli potrebné podobné zručnosti. Líšili sa však kognitívnu náročnosťou závisiacou od zadaných vstupov a očakávaných výstupov (odpoveď na otázku O2). V nasledujúcich odsekoch uvádzame krátku charakteristiku niektorých takýchto skupín úloh a zistených rozdielov.

Úlohy zamerané na kódovanie slov

Úlohy zamerané na kódovanie slov boli zastúpené na všetkých troch úrovniach obťažnosti. Pri ľahkej úlohe boli známe kódy pre 4 písmená a výsledné zakódované slovo. Úlohou bolo toto slovo dekodovať. Pri stredne ťažkej úlohe bolo známe pôvodné slovo, jeho kód a pravidlá, ktoré musia byť pri kódovaní dodržané. Účastníci mali zistiť kódy jednotlivých písmen v zadanom slove (5 písmen, 2 rovnaké sa opakovali za sebou). Pri ťažkej úlohe boli zadané kódy pre 5 slov (dĺžka kódov 2 až 5) a pravidlá pre kódovanie. Úlohou bolo vybrať jeden kód zo 4 kódov (2 kódy dĺžky 3, 2 kódy dĺžky 4), ktorý môže byť použitý pre ďalšie slovo. Domnievame sa, že pre nevidiacich mohla byť táto úloha náročná aj preto, že si vyžadovala zapamätanie viacerých, len mierne sa líšiacich, kódov.

Úlohy zamerané na šifrovanie textu

Úlohy zamerané na šifrovanie textu sa ukázali ako ľahké a stredne ťažké. Aj tu ich kognitívna náročnosť závisela od zadaných vstupov

a očakávaných výstupov. V ľahkých úlohách mali riešitelia presne zadaný postup na šifrovanie textu a vstupný text, ktorý mali pomocou neho zašifrovať. V stredne ťažkých úlohách mali zadaný len príklad textu pred zašifrovaním a po zašifrovaní. Postup šifrovania nebol uvedený, riešitelia ho mali nájsť a zašifrovať alebo dešifrovať pomocou neho ďalší zadaný text.

Úlohy zamerané na určovanie poradia prvkov

Úlohy zamerané na určovanie poradia prvkov patrili k stredne ťažkým a ťažkým. Ako stredne ťažké sa ukázali úlohy, v ktorých bolo potrebné usporiadať prvky, ak boli známe informácie, ako majú po sebe nasledovať. Na náročnosť nemalo vplyv, či tieto informácie boli o bezprostrednom usporiadaní alebo nie. Pri úlohách, ktoré sa ukázali ako ťažké, nebolo poradie prvkov priamo určené. Riešitelia poznali neusporiadanú postupnosť a pravidlo, na základe ktorého mohli uskutočňovať výmeny medzi dvoma prvkami. V týchto prípadoch sami volili postupnosť výmen, ktorá viedla k želanému stavu. V úlohách bolo potrebné určiť minimálny počet takýchto výmen. Aj pri stredne ťažkých, aj pri ťažkých úlohách sa pracovalo so šiestimi prvkami, takže množstvo prvkov sa v tomto prípade neukázalo ako faktor ovplyvňujúci náročnosť.

6 ZÁVER

V článku sme analyzovali vplyv niektorých faktorov na obťažnosť úloh zo súťaže iBobor v kategórii Nevidiaci SŠ [24]. Analyzovali sme výsledky pri riešení všetkých 63 úloh, ktoré žiaci riešili v rámci súťaže realizovanej v rokoch 2016 - 2022. Úlohy sme rozdelili do troch skupín podľa nameranej obťažnosti Q (ľahké, stredne ťažké, ťažké) a analyzovali sme, či obťažnosť závisí od tematickej oblasti a zručností potrebných na riešenie úlohy.

Zistili sme, že na obťažnosť úlohy má vplyv jej príslušnosť k tematickej oblasti. Úlohy z tematickej oblasti Reprezentácie a nástroje sa vo väčšej miere vyskytovali medzi ľahkými a stredne ťažkými úlohami. Naopak, úlohy z tematickej oblasti Algoritmické riešenie problémov sa vo väčšej miere vyskytovali medzi ťažkými úlohami. Tento výsledok sa zhoduje so zisteniami iných autorov o intaktných žiakoch [21, 25]. Úlohy týkajúce sa výrokovvej logiky sa ukázali ako ľahké alebo stredne ťažké.

Ďalej sme zistili, že je dôležité nielen to, aké kognitívne operácie musí účastník vykonať pri riešení úlohy, ale aj počet potrebných úkonov a počet objektov v sekvencii, s ktorou je potrebné pracovať.

Vykonalí sme aj analýzu úloh, ktoré určité percento účastníkov buď vôbec neriešilo, alebo neuviedlo na tieto úlohy odpovede. Na výsledky tejto analýzy však v tomto článku nie je priestor. Na základe našich zistení sme dospeli k záveru, že na nameranú obťažnosť úlohy má vplyv aj poradie, v akom bola riešená. Účastníci súťaže často riešili úlohy v poradí stanovenom autormi zadaní, teda najprv nimi predikované ľahké úlohy, potom stredne ťažké a nakoniec ťažké. Takže, ak riešenie niektorej skôr zaradenej úlohy trvalo dlhšie, na riešenie zvyšných úloh súťažiaci nemuseli mať dostatok času. V takom prípade sa o riešenie neskôr zaradených úloh možno vôbec nepokúsili.

Vzhľadom na to, že sme nemohli účastníkov pozorovať počas riešenia úloh a hovoriť s nimi a ich učiteľmi po skončení riešenia, sme si vedomí toho, že naše zistenia nemodno zovšeobecniť. Pomohli by nám aj podrobné informácie o študentoch, ako aj o nimi absolvovaných predchádzajúcich hodinách informatiky.

Slabou stránkou nášho výskumu je malý počet zúčastnených nevidiacich účastníkov. Tento aspekt súťaže však nemôžeme ovplyvniť.

POĎAKOVANIE

Tento článok vznikol ako súčasť projektu financovaného z grantu KEGA (Kega 010UK-4/2022). Osobitné poďakovanie patrí učiteľom a zrakovo postihnutým účastníkom súťaže iBobor. Ďakujeme medzinárodnému a slovenskému organizačnému výboru tejto súťaže za databázu originálnych úloh.

BIBLIOGRAFICKÉ ODKAZY

- [1] DAGIENĚ, V., FUTSCHEK, G. Bebras International Contest on Informatics and Computer Literacy: Criteria for Good Tasks. In *Informatics Education - Supporting Computational Thinking (ISSEP 2008)*. Lecture Notes in Computer Science, R. T. Mittermeir, M. M. Syslo (ed.). Berlin, Heidelberg: Springer, 2008, vol 5090.
- [2] DAGIENĚ, V., STUPURIENĚ, G. Exploring Bebras Tasks Content and Performance: A Multinational Study, In *Informatics in Education*, Vilnius University, 2017, Vol. 16, No. 1, p. 39–59.
- [3] CAREY, B. *How we learn*, New York : Random House, 2014.
- [4] JAŠKOVÁ, E., KOVÁČOVÁ, N. Bebras contest for blind pupils. In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*, London, UK, New York, NY, USA : ACM DL, 2015.
- [5] JAŠKOVÁ, E., KOVÁČOVÁ, N. Contest for Blind Pupils – Universal Design of Tasks. In *Proceedings of the Conference Universal Learning Design*, Linz, Masaryk University, Brno, 2016, p. 79–97.
- [6] JAŠKOVÁ, E., KOSTOVÁ N. Difficulty of Bebras Tasks for Lower Secondary Blind Students. In *Informatics in Schools. Engaging Learners in Computational Thinking (ISSEP 2020)*. Lecture Notes in Computer Science, K. Kori, M. Laanpere (ed.). Cham : Springer, 2020, vol 12518. DOI=https://doi.org/10.1007/978-3-030-63212-0_1
- [7] ALLMAN, C. B. *Test Access. Making Tests Accessible for Students with Visual Impairments: A Guide for Test Publishers, Test Developers, and State Assessment Personnel*. 4th Release, Louisville, Kentucky : American Printing House for the Blind, 2009.
- [8] *Overview of exam access arrangements*. Supporting blind and partially sighted people, Effective practice guide, RNIB, 2014, [cit. 2023-02-13]. 12 s. https://staging.rnib.org.uk/sites/default/files/Overview_of_exam_access_arrangements_May_2014.doc
- [9] iBobor pre nevidiacich. <http://vin.edu.fmph.uniba.sk/iBobor.html>
- [10] iBobor. <http://www.ibobor.sk>
- [11] LONATI, V., MONGA, M., MALCHIODI, D., MORPURGO, A. How presentation affects the difficulty of computational thinking tasks, In *Proceedings of the 17th Koli Calling International Conference on Computing Education Research*, 2017.
- [12] IZU, C. et al. Exploring Bebras Tasks Content and Performance: A Multinational Study. In *Informatics in Education*, Vilnius University, 2017, Vol. 16, No. 1, p. 39–59.
- [13] VAN DER VEGT, W. Predicting the difficulty level of a Bebras task. In *Olympiads in Informatics*, Vilnius University, 2013. Dostupné tiež na: <https://ioinformatics.org/journal/INFOL127.pdf>.
- [14] VAN DER VEGT, W. How Hard Will this Task Be? Developments in Analyzing and Predicting Question Difficulty in the Bebras Challenge. In *Olympiads in Informatics*, Vilnius University, 2018. Dostupné tiež na: https://ioinformatics.org/journal/v12_2018_119_132.pdf
- [15] VAN DER VEGT, W. et al. Analysing Task Difficulty in a Bebras Contest Using Cuttle. In *Olympiads in Informatics*, 2019, p. 145-165.
- [16] VANÍČEK J. Bebras Informatics Contest: Criteria for Good Tasks Revised. In *ISSEP 2014*. Lecture Notes in Computer Science, Y. Gülbahar, E. Karataş (ed.). Cham : Springer, 2014, vol 8730. p. 17–28.
- [17] VANÍČEK J. What Makes Situational Informatics Tasks Difficult? In *Informatics in Schools: Improvement of Informatics Knowledge and Perception (ISSEP 2016)*. Lecture Notes in Computer Science, A. Brodnik, F. Tort (ed.), Cham : Springer, 2016, vol 9973.
- [18] BUDINSKÁ, L., MAYEROVÁ, K., VESELOVSKÁ, M. Bebras task analysis in category little beavers in Slovakia. In *Informatics in Schools: Improvement of Informatics Knowledge and Perception (ISSEP 2017)*. V. Dagiene, A. Hellas, (ed.). Cham : Springer, 2017, vol.10696, p. 91–101.
- [19] KOSTOVÁ, N. *Obťažnosť úloh v súťaži Informatický bobor pre nevidiacich* : diplomová práca, Bratislava : FMFI UK, 2020.
- [20] CHRÁSKA, M. *Metody pedagogického výzkumu: Základy kvantitatívneho výzkumu*, Praha : Grada, 2007.
- [21] TOMCSÁNYI, P. Obťažnosť úloh v súťaži iBobor. In *DidInfo 2009*, Banská Bystrica : Univerzita Mateja Bela, 2009.
- [22] DAGIENĚ, V., SENTENCE, S., STUPURIENĚ, G. Developing a Two-Dimensional Categorization System for Educational Tasks in Informatics. In *INFORMATICA*, Vilnius University, 2017, vol. 28, no. 1, p. 23–44.
- [23] KALAŠ, I., TOMCSÁNYIOVÁ, M. Students' attitude to programming in modern informatics. In *Proceedings of the 9th WCCE 2009*, Education and Technology for a Better World, 2009.
- [24] ČUJDÍKOVÁ, M., JAŠKOVÁ, E., STANKOVIČOVÁ, M. Difficulty of Bebras tasks for upper secondary blind students. In *INTED 2023 Proceedinds*, IATED Digital Library. DOI=[10.21125/inted.2023](https://doi.org/10.21125/inted.2023)
- [25] MOČARNÍKOVÁ, K. *Obťažnosť úloh v súťaži iBobor* : diplomová práca, Bratislava : FMFI UK, 2014.

Tabuľky: Ako ich žiaci interpretujú v kontexte prostredia Emil správca

Tables: How pupils interpret them in the context of Emil the Keeper

Mária Čujdíková
KDMFI FMFI

Univerzita Komenského v Bratislave, Slovensko
maria.cujdikova@fmph.uniba.sk

Ivan Kalaš
KDMFI FMFI

Univerzita Komenského v Bratislave, Slovensko
ivan.kalas@fmph.uniba.sk

ABSTRACT

Tables are one of the basic forms of organizing and presenting data. As such, they are important in the development of both computational and mathematical thinking. Nevertheless, they still receive little attention in our field. That is why we have decided to focus on them as a follow-up research to Informatics with Emil, which – among other aims – supports the development of a structural view on data and processes. In our research, we are interested in how pupils think and act when interpreting data in tables in *Emil the Keeper* for Year 3. In this paper, we present the results of an analysis of one of our data sources, namely interviews with six post-lower primary pupils. Our findings show that although the tasks performed in interpreting a table have different cognitive demands for them, pupils perceive it as a structure, recognise different connections in it and can perform different operations with it up to its overall interpretation.

Keywords

Tables and their interpretation. Programming. Computing with Emil. Learning process. Structures and the structural view on data.

ABSTRAKT

Tabuľky patria medzi základné formy organizovania a prezentovania dát. Sú preto dôležité pri rozvoji aj informatického, aj matematického myslenia. Napriek tomu sa im v teórii vyučovania týchto predmetov venuje stále málo pozornosti. Preto sme sa rozhodli zamerať sa práve na ne, ako výskum nadväzujúci na projekt Informatika s Emilom, ktorého cieľom je podporiť u žiakov – okrem iného – aj rozvoj štruktúrného pohľadu na dáta a procesy. V našom projekte nás zaujíma, ako žiaci uvažujú a postupujú pri interpretácii dát v tabuľkách v prostredí *Emil správca* pre 3. ročník ZŠ. V tomto príspevku prezentujeme výsledky analýzy jedného z našich zdrojov dát, a to rozhovorov so šiestimi žiakmi tesne po ukončení 1. stupňa. Naše zistenia ukazujú, že aj keď vykonávané úkony pri interpretácii tabuľky majú pre nich rôznu kognitívnu náročnosť, vnímajú tabuľku ako štruktúru, uvedomujú si v nej rôzne súvislosti a dokážu s ňou vykonávať rôzne operácie, až po jej celkový výklad.

Kľúčové slová

Tabuľky a ich interpretácia. Programovanie. Informatika s Emilom. Poznávaci proces. Štruktúry a štruktúrný pohľad na dáta.

1 ÚVOD

Tabuľky sú jedným zo základných spôsobov organizovania a prezentovania dát. Používajú sa v rôznych formách a oblastiach života. Aj žiaci sa s nimi stretávajú už na prvom stupni základných škôl. Je to reprezentácia, pomocou ktorej môžu objavovať nové skutočnosti a premýšľať o nich. Avšak ako upozornili autori [1, 2], hoci

tabuľky môžu byť pre žiakov výborným kognitívnym prostriedkom, vhodným na riešenie rôznych druhov problémov, je potrebné upustiť od mylnej predstavy, že ide o jednoduchú štruktúru, ktorá si nevyžaduje premyslené učenie sa. Viacero výskumov už potvrdilo, že pre žiakov v tomto veku nie je práca s tabuľkami jednoduchá ani intuitívna [1, 2, 3]. V súlade s ich zisteniami aj my veríme, že je dôležité venovať pozornosť skúmaniu kognitívnej náročnosti tejto štruktúry a návrhu krokov, ktoré môžu pomôcť žiakom prekonať identifikované prekážky.

V našom širšom projekte sa zaoberáme skúmaním významu tabuľiek pri rozvoji informatického myslenia a matematického myslenia. Na tento účel sme pracovali s triedou žiakov 5. ročníka a získali od nich dáta, ktorých analýzu práve dokončujeme. V tomto príspevku sa zameriavame iba na jeden aspekt nášho výskumného problému. Konkrétne, zaujíma nás, ako žiaci vnímajú tabuľky v kontexte prostredia Emil a ako postupujú pri riešení súvisiacich úloh.

2 TABUĽKY V KURIKULE A VO VÝSKUME

Medzi oblasti vzdelávania, v ktorých hrajú tabuľky dôležitú úlohu, patrí aj matematika a informatika. V rámci moderného prístupu k vyučovaniu informatiky, zameraného na rozvoj informatického myslenia, sa majú žiaci zoznámiť s tabuľkami ako jednou z dôležitých dátových štruktúr [4]). Podľa nášho nového štátneho vzdelávacieho programu [5] je jedným zo vzdelávacích cieľov v predmete informatika v 1. cykle (1. – 3. ročník) *Zoznámiť sa s elementárnou lineárnou a tabuľkovou reprezentáciou údajov*, pričom žiak sa má orientovať v jednoduchej štruktúre a má vedieť organizovať informácie do štruktúr – vytvárať ich a podľa konkrétnych jednoduchých pravidiel s nimi manipulovať.

V rámci obsahového štandardu sa majú žiaci v oblasti (nazývanej teraz komponent) *Údajové štruktúry* zoznámiť s pojmami postupnosť, tabuľka, riadok a stĺpec a porozumieť pozícii v postupnosti a tabuľke. Medzi činnosti, ktorými sa majú zaoberať, patria (a) vyhladávanie a získavanie informácie z jednoduchej štruktúry podľa zadaných kritérií, (b) vytváranie jednoduchých štruktúr údajov podľa zadania a (c) manipulácia so štruktúrami údajov podľa konkrétnych jednoduchých pravidiel.

V 2. cykle (4. – 5. ročník) je jedným zo vzdelávacích cieľov *Spoznáť elementárne princípy a koncepty informatiky, vrátane reprezentácie údajov v tabuľkách, stromoch a grafoch*, pričom žiak má okrem predchádzajúcich zručností vedieť aj interpretovať údaje zo štruktúr (vyvodit' existujúce vzťahy zo zadaných údajov v štruktúre, prerozprávať informácie uložené v štruktúre vlastnými slovami) a používať konkrétne nástroje na prácu s rôznymi štruktúrami.

V tomto cykle sa obsahový štandard v oblasti *Údajové štruktúry* zameriava na pojmy: postupnosť, tabuľka (v zmysle frekvenčná, kódovacia, slovník, mriežka), riadok, stĺpec, bunka, adresa bunky. Zahŕňa vzťahy: poradie objektov a ich pozícia v postupnosti, adresa bunky ako pozícia objektov (bunky) v tabuľke, a tiež procesy: práca s grafovými štruktúrami a stromovými štruktúrami, navigácia v štruktúre. Medzi činnosťami, ktorými sa majú žiaci zaoberať, patria:

- Organizovanie informácií do štruktúr, ako sú postupnosti, tabuľka (v zmysle frekvenčná, kódovacia, slovník, mriežka), stromy, grafy... – vytváranie a manipulácia so štruktúrami, ktoré obsahujú údaje a jednoduché vzťahy.
- Orientovanie sa v jednoduchej štruktúre – vyhľadávanie a získavanie informácií zo štruktúry podľa zadaných kritérií.
- Interpretovanie údajov zo štruktúry – vyvodenie existujúcich vzťahov zo zadaných údajov v štruktúre..., prerazprávanie informácie uložené v štruktúre vlastnými slovami.
- Používanie konkrétnych nástrojov na prácu s tabuľkami (práca s riadkom, stĺpcom, bunkou, adresou bunky ako pozíciou objektu v tabuľke), stromami a grafmi...

Aj v modernom vyučovaní matematiky majú tabuľky dôležité postavenie. Kirova [6] v tomto smere uvádza:

„Práca s číselnými údajmi v tabuľkách a schopnosť tvoriť a riešiť slovné úlohy s týmito údajmi sú dôležitými prvkami práce v matematike na 1. stupni základnej školy. V tomto ohľade by sa nové vzdelávacie programy z matematiky mali podstatne líšiť od učebných súborov používaných v období rokov 2002 – 2016. Matematika je všade okolo nás a je potrebné, aby sa žiaci naučili takéto číselné informácie objavovať, rozpoznávať, samostatne vyhľadávať, používať a primerane interpretovať.“

Medzi oblasťami matematiky, v ktorých sa používajú tabuľky, patria aritmetika a algebra [7, 8] a pravdepodobnosť a štatistika [9, 10]. Pri matematickom modelovaní, ktoré v [9] označujú ako *klúčový aspekt matematickej gramotnosti*, môžu tabuľky slúžiť ako zdroj informácií potrebných na preskúmanie určitého problému pomocou matematiky, môžu tiež byť výsledným modelom reprezentujúcim navrhnuté riešenie, pozri [11].

V spomínanom novom vzdelávacom programe [5] patrí medzi ciele matematiky v 1. cykle aj *Orientovať sa v jednoduchej tabuľke a grafe a používať ich pri riešení aplikačných úloh zameraných na vyhľadávanie, zber, zaznamenávanie, triedenie, usporiadanie a interpretáciu údajov*, pričom žiak má vedieť používať jednoduché tabuľky a grafy na zber, triedenie, usporiadanie, zaznamenávanie a správnu interpretáciu údajov v reálnom živote a v aplikačných úlohách. V rámci obsahového štandardu v tomto cykle je práca s tabuľkami súčasťou tematických oblastí (komponentov):

- *Základy práce s jednoduchými závislosťami a vzťahmi;*
- *Základy riešenia jednoduchých kombinatorických situácií;*
- *Základy práce s údajmi.*

V 2. cykle patrí medzi vzdelávacie ciele aj *Zbierať, zaznamenávať, usporiadať a triediť údaje, hľadať vhodnú organizáciu údajov, tvoriť jednoduché frekvenčné tabuľky a grafy a správne interpretovať výsledky*. V rámci obsahového štandardu v tomto cykle je práca s tabuľkami súčasťou tematických oblastí:

- *Rozšírené nástroje na prácu so závislosťami a vzťahmi* – doplnenie alebo rozšírenie tabuľky priamej úmernosti; znázorňovanie údajov z tabuľky priamej úmernosti ako množiny bodov alebo priamky v súradnicovej sústave;
- *Rozšírené kombinatorické postupy a nástroje* – určovanie počtu

možností kombinatorickej situácie manipulačnou činnosťou s predmetmi, vypisovaním všetkých možností alebo použitím tabuľky, či grafu;

- *Rozšírené nástroje na prácu s údajmi* – návrh a tvorba jednoduchých tabuľky a grafu zo súboru kvantitatívnych alebo kvalitatívnych údajov.

Napriek dôležitosti tabuliek vo vyučovaní informatiky a matematiky už od 1. stupňa základnej školy je podľa [1, 2] úloha zostaviť tabuľku pre žiakov v tomto veku náročná preto, že vyžaduje viacero kognitívnych a grafických zručností. Pri skúmaní kognitívnej náročnosti pri tvorbe dvojrozmernej frekvenčnej tabuľky vo svojej štúdiu autori [2] identifikovali, že žiaci môžu mať problém s procesom kategorizácie (usporiadaním údajov do kategórií na základe dvoch kritérií), grafickým zobrazením údajov (priestorovým usporiadaním údajov do buniek v riadkoch a stĺpcoch), procesom abstrahovania potrebných informácií (odfiltrovaním nadbytočných informácií), a tiež s vyjadrením frekvencie (spočítaním položiek v jednotlivých kategóriách a ich zaznamenaním).

Ani interpretovanie údajov z tabuľky nie je pre žiakov jednoduché. Gabucio et al. [12] skúmali kognitívnu náročnosť jednotlivých úkonov pri práci žiakov s údajmi uvedenými v tabuľke, ktorá reprezentovala počet chlapcov a počet dievčat zodpovedajúcich určitému váhovému intervalu, pozri Tabuľku 1. Predložili ju žiakom 5. a 6. ročníka ZŠ a 1. a 2. ročníka SŠ.

Tabuľka 1: Tabuľka, ktorú využili vo výskume [12]

	menej ako 25	medzi 25 až 34	medzi 35 až 44	nad 44
chlapci	1	10	12	2
dievčatá	0	6	13	6
spolu	1	16	25	8

K tabuľke im položili niekoľko otázok, vopred zaradených do štyroch kategórií podľa úkonov, ktoré v nich bolo potrebné vykonať. Na základe správnosti jednotlivých odpovedí potom vyhodnotili stupne rastúcej obťažnosti jednotlivých úkonov takto:

1. priame čítanie údajov (*Kolko dievčat váži 25 až 34 kg?*),
2. pochopenie tabuľkovej štruktúry (*Čo označujú čísla vo vnútri tabuľky?*),
3. odvodzovanie údajov (*Kolko bolo celkovo odvážených dievčat? Kolko detí váži menej ako 35 kg?*),
4. celkový výklad (*Vážia viac chlapci alebo dievčatá?*).

Pre lepšiu predstavu uvádzame v zátvorkách príklady otázok, ktoré autori použili v rámci jednotlivých kategórií.

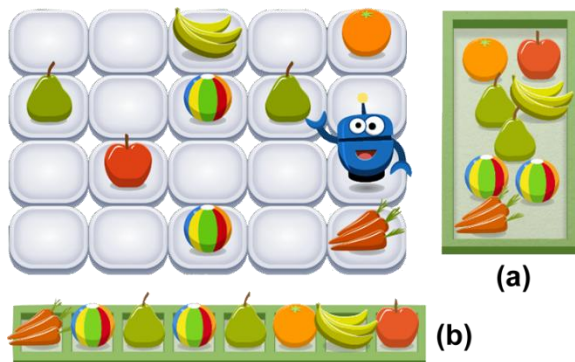
V našom príspevku analyzujeme, ako premýšľajú a postupujú žiaci pri interpretovaní tabuliek v našom aktuálne prebiehajúcom projekte Informatika s Emilom. V rámci tejto analýzy uvažujeme aj o uvedených stupňoch obťažnosti jednotlivých úkonov, ktoré s tabuľkami vykonávajú.

3 INFORMATIKA S EMILOM

Informatika s Emilom je výskumno-vývojový projekt, ktorému sa venujeme od roku 2017. Bližšie sme ho prezentovali v zborníku z konferencie DidInfo 2021 [13] a v ďalších publikáciách [14, 15, 16]. Ide o metódu vyučovania informatiky založenú na premyslenom a systematickom poznávacom procese žiakov. Doteraz sme ukončili vývoj vzdelávacích obsahov zameraných na programovanie v 3. a 4. ročníku ZŠ (ďalej len Emil 3 a Emil 4), Robotiku s Emou, ktorá zahŕňa oblasť robotiky pre celý 1. stupeň,

a prostredí určených pre materské školy. Momentálne dokončujeme prostredie s názvom Živý zošit, určené na rozvoj informatických základov digitálnej gramotnosti na 1. stupni. Toto prostredie bližšie predstavujeme v inom príspevku na tejto konferencii [17]. Emil 3 a 4, Robotika s Emou a Živý zošit spolu úplne pokrývajú požiadavky kurikula na informatiku pre 1. stupeň (a aj pre 1. a 2. cyklus).

Kľúčovým princípom nášho vývoja je *informatika pre každého žiaka v triede* [18, 19]. Stotožňujeme sa s názorom Sentance [19], že je potrebné spochybniť stereotypy o tom, kto sa môže a má učiť informatiku a hľadať cesty k pochopeniu, ako ju sprístupniť všetkým žiakom. V našom prístupe vychádzame z konštrukcionistickej pedagogiky, osvojili sme si Papertov postoj k programovaniu ako k príležitosti zažiť a skúmať *silné myšlienky* (powerful ideas) [20]. V našom vývoji sa preto snažíme porozumieť, s akými *silnými myšlienkami informatiky* sa žiaci v danom veku môžu zaoberať pri vlastnom objavovaní základných programovacích konceptov a operácií. Aby sme umožnili takéto objavovanie informatiky skutočne všetkým žiakom, zaoberáme sa tiež kognitívnou náročnosťou daných konceptov a operácií identifikovaním primerane malých krokov v gradácii aktivít pre žiakov, ktoré vedú k ich hlbokému porozumeniu [15, 16].



Obrázok 1: Krabice (a) a police (b) v prostredí Emil zberač

Medzi kľúčové princípy Informatiky s Emilom patrí spoločné učenie sa. Žiaci najskôr riešia úlohy v dvojiciach na jednom tablete alebo počítači, pričom každý z nich pracuje s vlastným pracovným zošitom. Následne o svojich postupoch diskutujú v rámci celej triedy, tieto diskusie usmerňuje učiteľ.

Okrem rozvoja informatického myslenia chceme u žiakov podporiť aj vnímanie programovania ako nástroja na skúmanie rôznych oblastí a riešenie rozmanitých úloh. Preto považujeme za dôležité vytvárať bohaté prepojenia smerom k ostatným predmetom. Pri našom vývoji sa opierame o pedagogický rámec 5E z projektu ScratchMaths [21], čiže vytvárame aktivity, ktoré motivujú žiakov, aby skúmali problémy, vysvetľovali si navzájom postupy a objavy, uvažovali, zdieľali a prepájali poznávanie s inými областami.

V štúdiu, ktorú tu prezentujeme, využívame úlohy z intervencie Emil 3. Tá pozostáva zo sekvencie gradovaných úloh, rozdelených do troch svetov – *Emil zberač*, *Emil správca* a *Emil umelec*, s rastúcou úrovňou riadenia a reprezentácie [15]. Skupiny gradovaných úloh sú v jednotlivých svetoch označené písmenami **A**, **B**, **C**... Väčšina z nich je navrhnutých tak, aby žiaci pri ich riešení využívali súčasne aj softvérové prostredie, aj pracovný zošit. Za niektorými skupinami úloh v pracovnom zošite však nasleduje aj ďalšia skupina s označením **Bez počítača**. Tieto úlohy žiaci riešia len na papieri.

3.1 Emil a práca s tabuľkami

V našej štúdiu sa snažíme porozumieť, ako žiaci uvažujú a postupujú, keď pracujú s tabuľkami v kontexte prostredia *Emil správca*. V príspevku chceme predovšetkým preskúmať ich interpretáciu dvojrozmernej frekvenčnej tabuľky v tomto prostredí. Budovanie porozumenia tabuliek však v intervencii Emil 3 prebieha postupne. Radí by sme preto v krátkosti predstavili, čo predchádza zoznámeniu sa s dvojrozmernou frekvenčnou tabuľkou.



Obrázok 2: Panel s príkazmi v prostredí Emil správca

Žiaci sa už v prvých sériách úloh vo svete *Emil zberač* zoznámia s jednoduchými dátovými štruktúrami, ktorými sú krabice a police (pozri Obrázok 1). Tieto dve štruktúry sa líšia v tom, ako sú v nich predmety, ktoré Emil zbiera (a ktoré predstavujú dáta) reprezentované. Do krabice sa ukladajú bez usporiadania, v polici sú zobrazené zľava doprava v poradí, v akom ich žiaci s Emilom zbierajú.

Na začiatku sveta *Emil správca* pribudne ďalšia štruktúra, a to panel s príkazmi (Obrázok 2, hore). Ten sa stane súčasťou všetkých ďalších úloh a v závislosti od úrovne reprezentácie [15] predstavuje buď *externý záznam* vykonaných krokov alebo *externý plán* (program). V oboch prípadoch predstavuje táto štruktúra pre žiakov možnosť, ako premýšľať o postupnosti príkazov a analyzovať svoje riešenia.



Obrázok 3: Jednorozmerná frekvenčná tabuľka v prostredí Emil správca

Vo svete *Emil správca* sa žiaci dostávajú už aj k frekvenčným tabuľkám. Najskôr pracujú s jednorozmernými frekvenčnými tabuľkami, pozri Obrázok 3. Tie sú súčasťou úloh, v ktorých analyzujú

záznam svojho riešenia a sústreďujú sa na zistenie, koľkokrát sa v ňom vyskytujú jednotlivé príkazy. Tieto počty zapisujú do buniek tabuľky, nad ktorými je zobrazený daný príkaz. Tabuľka im tak pomáha premýšľať o štruktúre záznamu a zároveň reprezentovať výsledky svojich úvah. V týchto úlohách môžeme tiež identifikovať rôzne kategórie náročnosti, aké sme popísali vyššie podľa [12]. Žiaci sa najskôr zameriavajú iba na počty identických príkazov, neskôr zlučujú do jednej kategórie všetky príkazy na farbenie, bez ohľadu na konkrétnu použitú farbu. Ďalej nasledujú úlohy s dvojrozmernými frekvenčnými tabuľkami, ktoré riešili aj žiaci v rámci našej štúdie, predstavíme ich preto podrobnejšie. Ide o skupiny úloh C. V prvej z nich (Obrázok 4) majú žiaci riadiť Emila na obrazovke tak, aby vyfarbil domy podľa predlohy na obrázku v pracovnom zošite.

Dva domy (na ploche v softvéri) sú na ploche vyfarbené už na začiatku, takže stačí vyfarbiť päť zvyšných. Vykonané príkazy sa zobrazujú v hornom paneli. Ich počet je obmedzený, a teda Emilov pohyb a farbenie domov treba dobre premyslieť. Pri pohybe jedným smerom sa príkazy reprezentujúce daný smer spájajú a zapisujú do jediného políčka, spolu s počtom vyjadrujúcim, koľkokrát sa tento príkaz vykonal. Rovnaký princíp programovania sa využíva aj v ostatných úlohách tejto série. Po dokončení farbenia úloha vyzýva žiakov, aby odpovedali na otázky ohľadom počtu domov určitej farby, a to buď na celej ploche alebo v jednotlivých radoch. Následne majú vyplniť jednoduchú dvojrozmernú frekvenčnú tabuľku, ktorá má dva riadky a dva stĺpce. Do tabuľky zapisujú počty domov zodpovedajúcej farby.

Vymaľuj s Emilom na ploche domy podľa tohto vzoru a odpovedz na otázky:

- Koľko je na ploche červených domov? 4
- Koľko je na ploche modrých domov? _____
- Koľko červených domov je v hornom rade? _____
- Koľko modrých domov je v dolnom rade? _____

Vyplň tabuľku:

v hornom rade	3	
v dolnom rade		

Obrázok 4: Emil správca, úloha C1 podľa pracovného zošita

Druhá úloha, pozri Obrázok 5, okrem programovania pred žiakom predkladá aj kombinatorickú výzvu. Ich úlohou je s Emilom vyfarbiť domy tak, aby bol rovnaký počet červených v hornom aj dolnom rade, pričom aj tu sú niektoré domy už vyfarbené. Žiakom navrhujeme, aby našli viacero riešení. Táto úloha nevyužíva prácu s tabuľkou, ale je prípravou na spôsob uvažovania, ktorý bude dôležitý pri ďalších úlohách.



Obrázok 5: Úloha C2:

Vymaľuj s Emilom všetky domy tak, aby bol v hornom aj dolnom rade rovnaký počet červených. Nájdi viac riešení

V tretej úlohe (Obrázok 6) majú žiaci vymaľovať s Emilom domy tak, aby bol počet červených rovnaký v hornom aj dolnom rade. Aj zelených má byť v oboch radoch rovnaký počet. Aj tu sú niektoré domy vyfarbené už vopred. Žiaci sa tu opäť okrem programovania

zaoberajú aj kombinatorickými úvahami. Svoje riešenie majú potom zaznamenať vymaľovaním domov v zošite a vyplniť zodpovedajúcu dvojrozmernú frekvenčnú tabuľku. Zapisujú do nej počty domov danej farby v danom rade. V porovnaní s prvou úlohou pribudla ešte jedna farba, takže tabuľka má teraz tri stĺpce.



v hornom rade

--	--	--

v dolnom rade

--	--	--

Obrázok 6: Úloha C3

V štvrtej úlohe, pozri Obrázok 7, žiaci prvýkrát do tabuľky nezapisujú, ale interpretujú jej hodnoty a na základe toho vymaľovávajú na obrazovke s Emilom domy. Aj v tomto prípade sú už niektoré domy na začiatku vymaľované a je potrebné ich v uvažovaní zohľadniť. Svoje výsledné riešenie žiaci potom zaznamenávajú do pracovného zošita.

v hornom rade

2	1	1
---	---	---

v dolnom rade

1	1	3
---	---	---

Obrázok 7: Úloha C4:

Vymaľuj s Emilom domy podľa tabuľky

Okrem Emila 3 sa žiaci tabuľkám vo väčšej miere venujú aj v Živom zošite a Robotike s Emou.

4 METÓDA

V tomto príspevku zameriavame našu pozornosť na výskumnú otázku: *Ako žiaci uvažujú a postupujú pri interpretovaní tabuľky v kontexte prostredia Emil správca?* V snahe hľadať odpoveď na ňu sme zvolili kvalitatívny prístup, keďže sme nechceli overovať vopred dané hypotézy, ale chceli sme čo najviac porozumieť danému fenoménu a následne ho na základe toho opísať. Konkrétne sme sa rozhodli pre metódu prípadovej štúdie.

Výskum prebiehal na základnej škole. Participantmi boli žiaci piateho ročníka, ktorí v treťom a štvrtom ročníku absolvovali celú intervenciu Emil 3 a 4. V rámci výskumu znova riešili úlohy súvisiace s dvojrozmernými frekvenčnými tabuľkami zo sveta *Emil správca*, a to na jednej vyučovacej hodine, ktorú nám na tento účel škola vyhradila. Na tejto hodine najprv všetci pracovali na už pre

nich známej sérii úloh C, teda na úlohách C1 až C4 (prezentovali sme ich vyššie). Práca prebiehala v dvojiciach, pričom každá dvojica mala k dispozícii tablet s Emilom. Po tejto sérii nasledovala diskusia s celou triedou, v ktorej žiaci komentovali svoje postrehy a vyriešené úlohy.

Po vyriešení úloh C1 až C4 sme v rámci tej istej vyučovacej hodiny v našom výskume zadali žiakom aj tri ďalšie papierové úlohy C5, C6 a C7. S malou skupinou sme potom realizovali rozhovor a použili pri ňom mierne odlišné varianty spomínaných papierových úloh, pozri nižšie.

V úlohe C5 (Obrázok 8) mali žiaci vyplniť na základe obrázka s farebnými domami prázdnu tabuľku. Mali do nej zapísať počty domov podľa radu a zodpovedajúcej farby. Z hľadiska využitia tabuliek sa venovali podobným úvahám ako pri úlohách C1 a C3 s tým rozdielom, že tu už nevyužívali softvérové prostredie.



Obrázok 8: Úloha C5 bez počítača:
Vyplň tabuľku podľa obrázka

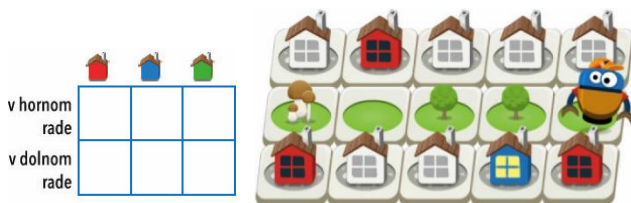
V úlohe C6 (Obrázok 9) bola uvedená tabuľka, ktorá vyjadrovala počty domov danej farby v danom rade. Žiaci mali podľa nej najskôr vyfarbiť domy a potom odpovedať na tri otázky. Otázky sa týkali počtu domov danej farby v niektorom rade plochy alebo na celej ploche, a tiež počtu všetkých domov v niektorom rade. Úloha má, prirodzene, veľa rôznych riešení.



- Koľko je na ploche červených domov? _____
- Koľko modrých domov je v hornom rade? _____
- Koľko domov je v hornom rade? _____

Obrázok 9: Úloha C6 bez počítača:
Vyfarbi domy podľa tabuľky. Potom odpovedz na otázky

V úlohe C7 (Obrázok 10) mali žiaci vymyslieť úlohu pre svojich spolužiakov. K dispozícii bola predpripravená prázdna tabuľka, obrázok s domčekmi, z ktorých niektoré boli vyfarbené a niektoré nie a miesto pre text zadania. Bolo na žiakoch, čo z toho využijú a akým spôsobom.



Obrázok 10: Úloha C7 bez počítača:
Vymysli vlastnú úlohu pre spolužiakov

Po skončení hodiny sme so šiestimi vybranými žiakmi viedli individuálne pološtruktúrované rozhovory, ktoré sme zaznamenávali pomocou audiozáznamu. Išlo o tri dievčatá a troch chlapcov, ktorí podľa výberu pedagóga dobre reprezentujú rôzne úrovne učebných výsledkov v triede. V úvode rozhovoru sme najskôr zisťovali, ako sa žiakom páčila hodina a ako hodnotia úlohy, ktoré riešili. V ďalšej časti rozhovoru sme ich vyzvali, aby pred nami riešili mierne modifikované verzie papierových úloh C5, C6 a C7 (s inými hodnotami v tabuľkách a pod.) a nahlas komentovali, ako pri tom uvažujú. Po vyriešení každej úlohy sme ďalej rozvíjali rozhovor.

Napr. v úlohe C6 nás zaujímalo, ako žiaci uvažujú a postupujú, keď interpretujú zadané údaje. V porovnaní s variantom úlohy, ktorý riešili na hodine, teraz nemali k dispozícii nevyfarbené domy na obrázku, ale vyzvali sme ich, aby na základe dát z tabuľky priamo odpovedali na otázky. Mali teda vynechať medzikrok, v ktorom by si najprv vyfarbili domy.

Dáta získané z časti rozhovoru k úlohe C6 sme analyzovali s cieľom odpovedať na našu výskumnú otázku. Zistenia uvádzame v nasledujúcej časti. V tomto príspevku sa nebudeme zaoberať analýzou úloh, ktoré žiaci riešili na hodine, ale ich opis v predošlej časti môže čitateľovi pomôcť lepšie si predstaviť, s akými výzvami sa stretli bezprostredne pred riešením tejto úlohy. Nebudeme sa tu venovať ani analýzám úloh C5 a C7 riešených počas rozhovorov. O nich budeme referovať pri inej príležitosti.

5 ZISTENIA

Na základe analýzy úvodnej časti rozhovorov sme zistili, že žiakom sa hodina páčila. Úlohy, ktoré počas nej plnili, považovali za ľahké a zaujímavé. Obzvlášť ich bavili úlohy, v ktorých pracovali na tablete s Emilom. Tieto úlohy považovali nielen za zábavnejšie, ako tie, ktoré riešili len na papieri, ale vyjadrili tiež presvedčenie, že im Emil pomáhal s riešením zadaných problémov: „Keby som niečo nevedela, tak si to môžem s tým Emilom vyskúšať“. A ďalší žiak pri úvahe, v čom bolo výhodné používať Emila, konštatoval: „No, aby mi pomohol s tými úlohami a... Aby mi pomohol s tými úlohami,“ aj keď si zároveň uvedomoval, že on je ten, kto ho ovláda: „No akože, no skôr ja som ho posúval, ale akože pomáhal mi dakedy, že vedel nejak kam ísť a keď som mu prikázal, kam má ísť, tak tam išiel.“

V porovnaní s úlohami, ktoré žiaci riešili len na papier, úlohy so softvérom prinášali aj ďalšiu výzvu. Na riadenie Emila totiž mohli použiť len obmedzený počet príkazov. Dve žiačky reflektovali, že sa im niektoré úlohy najskôr nedarilo vyriešiť práve preto, že im chýbali *polička na ďalšie príkazy*, avšak potom, keď sa nad svojím postupom znova zamysleli, riešenie našli. Napriek tejto skutočnosti obe žiačky hodnotili, že sa im práca s Emilom páčila. Jedna dokonca vyslovila, že sa jej s ním „*Veľmi dobre pracuje*“.

Žiaci tiež oceňovali možnosť spolupracovať vo dvojici. Jedna žiačka vyhodnotila spoluprácu ako najzaujímavejšiu zložku hodiny. Iná žiačka zasa uviedla, ako si spolu so svojou spolusediacou rozšírili zadanie druhej úlohy, ktorú robili s Emilom na tablete tak, že namiesto dvoch rôznych riešení našli až štyri „... sme išli vlastne spolu, že sme mali napr. nájsť tie dve riešenia, tak sme dali, že dve riešenia ja a dve riešenia ona. Tak sme mali tak viacej.“

5.1 Interpretácia tabuľky

V snahe odpovedať na našu výskumnú otázku sme analyzovali, ako žiaci postupujú a uvažujú, keď pracujú s údajmi vo vopred vyplnenej tabuľke v rámci úlohy C6 riešenej v priebehu rozhovoru. Všetci žiaci pri jej riešení postupovali tak, že si najskôr prečítali

zadanie. Dvaja navyiac nahlas prečítali aj hodnoty v tabuľke. Jedna žiačka čítala hodnoty v jednotlivých riadkoch, bez priradenia k farbám: „*V hornom rade tri, jedna, jedna, v dolnom dva, dva, jedna*“, druhý žiak čítal hodnoty aj s prislusnosťou k farbám: „*Tuna v hornom rade sú tri modré, jeden zelený, jeden červený, v dolnom rade dva modré, dva zelené a jeden červený.*“ Obaja tým demonštrovali svoje zoznámenie sa s tabuľkou predtým, ako sa pustili do odpovedania na jednotlivé otázky. Ďalší dvaja len skonštatovali, že v tabuľke vidia počty jednotlivých domov a z toho majú teraz vychádzať. Jeden z nich sa vyjadril: „*Tak ako tu máme tú tabuľku a z nej musíme nejako odčítať, že koľko je tam tých domov.*“ Zo slovo druhej žiačky môžeme zároveň usúdiť, že si všimla, že v tejto úlohe (v porovnaní s úlohou z hodiny) nie je obrázok s nevyfarbenými domčekmi: „*No teraz tu nemáme akože ten vzor, ale vieme, že ako to vyfarbil.*“ Zvyšní dvaja žiaci začali rovno po prečítaní zadania hľadať odpovede na otázky.

Prvá otázka: „*Koľko je na ploche červených domov?*“ vyžadovala odvodenie údajov. Podľa stupňov obťažnosti úkonov pri práci s tabuľkami, ako ich definovali v [12], išlo teda o otázku s vyššou kognitívnu náročnosťou (úroveň 3). Žiaci nemali žiadny problém identifikovať, na ktoré políčka tabuľky sa majú zamerať, ani porozumieť, čo s nimi majú urobiť. Viacerí z nich na tieto políčka priamo ukázali a zhodnotili, že majú spočítať ich hodnoty. Väčšina dokonca explicitne argumentovala, že keď sa jedná o červené domy na celej ploche, musia si všimnúť oba rady: „*Tu je otázka, že koľko je na ploche červených domov, tak vieme ako že na celkovej ploche. Tak si sčítame ako tie z horného radu, ten červený dom a z toho dolného radu červený.*“

Druhá otázka: „*Koľko modrých domov je v hornom rade?*“ mala pre žiakov, opäť podľa [12], predstavovať menšiu kognitívnu výzvu ako zvyšné dve. Išlo o otázku, ktorá vyžadovala priame čítanie údajov z tabuľky, čiže o otázku úrovne obťažnosti 1. Na základe reakcií našich účastníkov usudzujeme, že túto otázku skutočne považovali za jednoduchú. Jedna žiačka napr. okamžite vyslovila: „*To je jasné, tri.*“ a následne argumentovala tak, že ukázala najskôr na riadok: „*Lebo v hornom rade,*“ a potom na stĺpec „*modrých.*“ Ďalší žiak zhodnotil: „*To, koľko modrých domov je v hornom rade, iba opíšem tri.*“, čím zároveň zdôraznil, že jeho úlohou je tentokrát len nájsť daný údaj. Podobne reagovali aj ostatní žiaci. Pri riešení zväčša ukazovali na políčko zobrazujúce počet modrých domov v hornom rade a slovné komentovali, že to je to, čo nás teraz zaujíma. Jeden žiak navyiac explicitne komentoval, že pri tejto úlohe nás nezaujíma, čo je v dolnom rade: „*Teraz musíme modrých, ale nemôžeme aj v dolnom, lebo tu píšú, že v hornom rade, takže iba tri.*“

Tretia otázka: „*Koľko domov je v hornom rade?*“ vyžadovala, rovnako ako prvá, odvodenie údajov, čiže podľa [12] išlo znova o úlohu tretej úrovne obťažnosti. Ani v tomto prípade nemali žiaci problém zistiť, na ktoré políčka sa majú zamerať a čo potrebujú urobiť, aby získali odpoveď. Všetci poukázali na to, že majú sčítať hodnoty v hornom rade. Niektorí priamo hovorili, ako počítajú: „*Spočítam $3 + 1$ je 4 , $4 + 1$ je 5* “, ostatní povedali rovno výsledok. Jeden žiak navyiac explicitne vyjadril, že teraz nám nezáleží na farbe domov: „*Koľko domov je v hornom rade? Tak tu není farba, tak máme spočítať všetky domy. Takže 5.*“

5.1.1 Tvorba vlastných otázok

Zaujímalo nás aj aké ďalšie údaje dokážu žiaci z tabuľky odvodiť, preto sme ich požiadali, aby skúsili vymyslieť vlastné otázky. Zadania, ktoré navrhli, môžeme rozdeliť do dvoch základných typov:

- 1. typ – otázky, ktoré boli variáciou na tie naše,

- 2. typ – nové originálne otázky.

Pri prvom type otázok žiaci v porovnaní s našimi otázkami obmedzili napr. farbu alebo riadok, na ktorý sa pýtali. Príkladom takýchto otázok sú: „*Koľko je na ploche modrých domov spolu?*“, „*Koľko je zelených domov?*“, „*Koľko domov je v dolnom rade?*“, „*Koľko červených domov je v dolnom rade?*“

Druhým typom boli nové originálne myšlienky, ktorými sa nezaoberali v žiadnej z predchádzajúcich úloh. Niektoré z týchto otázok vyžadovali náročnejšiu prácu s dátami – uvedenie si viacerých súvislostí a vykonanie niekoľkých rozličných operácií. Žiaci, ktorí otázky navrhli, ich však zároveň dokázali bez problémov aj vyriešiť. Vedeli, na ktoré časti tabuľky sa potrebujú zamerať a predstaviť si a vykonať operácie, ktoré ich privedú k výsledku. V tomto smere sme identifikovali nasledujúce skupiny otázok:

Určenie počtu domov viacerých farieb v jednom rade. Išlo o otázky, ktoré vyžadovali určenie dvoch políčok tabuľky a následné zistenie súčtu ich hodnôt. Napr. jeden žiak uviedol otázku: „*Koľko je dokopy červených a modrých [domov] v hornom rade?*“ Pri odpovedaní na ňu postupoval tak, že našiel v prvom riadku tabuľky hodnoty zodpovedajúce počtu červených domov v hornom rade a modrých domov v hornom rade a tieto hodnoty sčítal.

Určenie počtu domov viacerých farieb celkovo na ploche. Tieto otázky boli o niečo komplikovanejšie ako predvážajúce. Pri odpovedaní na ne bolo potrebné identifikovať štyri políčka tabuľky a sčítať ich hodnoty. Medzi takéto otázky patrili: „*Sčítaj počet zelených a červených domov.*“, „*Koľko je červených a zelených [domov] dokopy?*“ Pri určovaní potrebného súčtu mohli žiaci postupovať viacerými spôsobmi. Autorka druhej z uvedených otázok postupovala napr. tak, že si najskôr spočítala, koľko je červených a zelených domov spolu v hornom rade, potom v dolnom a následne tieto medzisúčty sčítala: „*To sú dve v hornej ploche a dole sú tri. Takže to spoločné je... päť.*“

Určenie počtu všetkých domov spolu. Niektorým žiakom napadlo pýtať sa aj na počet všetkých domov spolu. Napr. „*Koľko je všetkých domov na ploche?*“, „*Koľko je celkovo domov?*“, „*Koľko je domov v dolnom aj hornom rade.*“ Prístupmi k riešeniu tejto otázky sa budeme zaoberať nižšie.

Určenie, o koľko viac je domov jednej farby ako inej – v určenom rade. V týchto otázkach bolo potrebné identifikovať dve políčka a určiť rozdiel ich hodnôt. Jedna žiačka napr. navrhla pýtať sa „*O koľko viac je modrých domčekov v hornom rade ako v dolnom?*“ Na základe štylizácie „*O koľko viac je ...*“ usudzujeme, že tieto otázky vyžadovali, aby si žiaci už pred ich položením uvedomovali, ktorých domov je viac. Ďalším krokom v porovnaní s predošlými otázkami tu teda bolo porovnanie hodnôt daných políčok.

Určenie, o koľko viac domov je jednej farby ako inej – celkovo na ploche. Tieto otázky boli ešte zložitejšie a komplexnejšie ako predchádzajúce. Žiaci v nich potrebovali identifikovať dve dvojice políčok tabuľky, sčítať hodnoty políčok v rámci dvojice a následne získané súčty vhodne odčítať. Aj tu si potrebovali už pred formulovaním otázok uvedomiť, ktorých domov bude viac. Príkladom je otázka: „*O koľko viac je modrých domov ako červených?*“ Žiak, ktorý ju navrhol, komentoval svoj postup riešenia slovami: „*Som si sčítal, tie modré domy. Tých je ako 5. Potom som si sčítal aj tie červené. To je ako 2. Dal som si že $5 - 2$. Tak to mi vyšlo 3.*“

Porozumenie štruktúre tabuľky. Jedna žiačka navrhla otázku zameranú na porozumenie štruktúry tabuľky: „*Aké majú farby tie domy... koľko tých farieb majú akože?*“ Následne vysvetlila, ako

môžeme zistiť odpoveď. Ukázala v tabuľke na vrchný riadok, kde boli nakreslené jednotlivé domčeky a povedala: „*Tu hore, tu sú tie farby.*” a ďalej vyvodila: „*Tak by sme napísali, že modrá, zelená, červená. Alebo číslom, že 3 tam sú.*”

Rozšírenie tabuľky. Zaznel aj návrh na otázku, ktorá by vyžadovala rozšírenie uvedenej tabuľky. Jeden žiak uvažoval: „*Keby sú tam aj rozsvietené domy, tak by som mohol dať, že koľko ich je rozsvietených.*” Keď sme sa ho spýtali, či by sa to dalo vidieť v tabuľke, ktorú má pred sebou alebo by musela byť iná, zhodnotil, že by musela byť iná. Postup, ako by takúto tabuľku vytvoril, predstavil slovami: „*Dal by som to na, no na šesť tých políčok v jednom riadku a dal by som tam, že všetky tie isté farby a dal by som ich rozsvietené.*” A ukazujúc na okienka na domčeku spresnil: „*Akože s tým svetlom v tomto [okne].*”

5.1.2 Naše rozširujúce otázky

Okrem otázok, ktoré žiaci navrhli sami, sme niektorým tiež následne položili naše rozširujúce otázky, zamerané na celkový výklad tabuľky. Podľa [12] išlo o úlohy na štvrtom stupni kognitívnej náročnosti. V tomto smere sme sa pýtali otázky ako: „*Ktoej farby je najviac domčekov v tej dedinke?*”, „*Domčekov ktorej farby je najmenej?*”. Aj na tieto otázky dokázali žiaci bez zaváhania odpovedať a po vyzvaní aj zdôvodniť svoje rozhodnutie: „*[Modrej], lebo tá farba má 5 domčekov vyfarbených a ostatné majú akože menej.*” Žiakov, ktorí nenavrhli zistiť celkový počet domčekov na ploche, sme sa tiež pýtali, či by vedeli, aký je tento počet. Pri riešení sme mohli pozorovať dva základné prístupy. Jedným bolo postupné sčítanie všetkých hodnôt jednotlivých políčok, ako napr. uviedla jedna žiačka: „*3 + 1 je 4, 4 + 1 je 5, 5 + 1 je 6, 6 + 2 je 8 a 8 + 2 je 10*” a druhým bolo sčítanie súčtov riadkov. Napr. jedna žiačka uvažovala: „*No, na celej dedine? Tak keď si spočítam horný a dolný rad, tak je to 10.*” Alebo iný žiak počítal ukazujúc si na jednotlivé riadky tabuľky: „*Päť, desať.*”

6 DISKUSIA A ZÁVER

Na základe analýzy rozhovorov s našimi participantmi sme zistili, že ich práca s tabuľkami v prostredí Emil správca bavila. Úlohy, ktoré riešili na hodine, vnímali ako ľahké a zaujímavé. Predovšetkým sa im páčili úlohy, v ktorých programovali Emila, pričom zároveň vyjadrili presvedčenie, že v týchto úlohách im Emil pomáha s riešením. Niektorí participantmi tiež ocenili možnosť spolupracovať vo dvojici.

Aby sme mohli preskúmať, ako postupujú a uvažujú pri interpretovaní dát z tabuľky v kontexte tohto prostredia, vyzvali sme ich, aby pred nami riešili úlohu, v ktorej potrebovali využívať dáta z pripravenej tabuľky. Podľa ich reakcií pri jednotlivých otázkach, ktoré boli súčasťou tejto úlohy usudzujeme, že ich vnímanie náročnosti požadovaných úkonov bolo v súlade so stupňami obťažnosti, ako ich definovali Gabucio a kol. [12]. Otázka, ktorá vyžadovala priame čítanie údajov predstavovala najmenšiu kognitívnu výzvu. Viacerí participantmi poukázali na to, že pri nej stačí len opísať danú hodnotu políčka. Pri otázkach zameraných na *odvodenie údajov* viedli komplexnejšie úvahy. Identifikovali, na ktoré políčka sa majú zamerať a často aj argumentovali prečo. Ďalej pokračovali konštatovaním, že tieto hodnoty potrebujú sčítať. Keďže nás zaujímalo aj aké ďalšie údaje dokážu z tabuľky odvodiť, vyzvali sme ich, aby skúsili vymyslieť vlastné otázky. Niektoré otázky, ktoré navrhli, boli variácie na tie naše, avšak navrhli aj veľa otázok, ktoré sa zaoberali inými problémami. Boli to otázky s rôznym stupňom obťažnosti, niektoré z nich vyžadovali komplikovanejšie myšlienky ako tie, ktoré riešili predtým. Bolo v nich potrebné

uvvedenie si viacerých súvislostí a vykonanie niekoľkých rozličných operácií. Participantmi, ktorí ich navrhli, ich však zároveň dokázali bez problémov riešiť. V rámci časti rozhovoru venujúcej sa tejto úlohe sme tiež položili naše rozširujúce otázky zamerané na celkový výklad tabuľky, ktoré autori [12] vyhodnotili ako úkony s najvyšším stupňom obťažnosti. Aj na tieto otázky dokázali participantmi bez problémov odpovedať a aj zdôvodniť svoje rozhodnutie.

Za hlavnú limitáciu našej štúdie považujeme to, že išlo o žiakov z jednej školy. Zapojenie viacerých škôl a ďalších tried by mohlo potvrdiť alebo obohatiť naše zistenia. V budúcnosti preto plánujeme pokračovať v takejto výskumnej aktivite aj na ďalších školách.

Tento príspevok je súčasťou nášho širšieho výskumu, v ktorom sa zaoberáme skúmaním významu tabuliek pri rozvoji informatického myslenia a matematického myslenia. Ďalej plánujeme skúmať napr. aj to, ako žiaci vnímajú užitočnosť tabuliek, ako premýšľajú a postupujú pri vkladaní údajov do tabuľky, ale tiež aj to, ako význam tabuliek pri rozvoji matematického myslenia a informatického myslenia vnímajú odborní didaktici týchto dvoch oblastí.

POĎAKOVANIE

Ďakujeme všetkým žiakom, ktorí sa zúčastnili nášho výskumu, za ich ochotu spolupracovať a podeliť sa s nami o svoje komentáre a podnetné myšlienky. Ďakujeme aj vedeniu a pedagógom ZŠ Ľudovíta Štúra v Modre za ich ústretový prístup a mnohoročnú spoluprácu.

BIBLIOGRAFICKÉ ODKAZY

- [1] MARTI, E. Tables as Cognitive Tools in Primary Education. In *Representational Systems and Practices as Learning Tools*, 2009, pp. 133–148. DOI: https://doi.org/10.1163/9789087905286_009
- [2] MARTI, E. GARCIA-MILA, M., GABUCIO, F. The construction of a double-entry table: a study of primary and secondary school students' difficulties. In *European Journal of Psychology of Education*, 2011, vol. 26, no. 2, pp. 215–234, DOI: <https://doi.org/10.1007/sl>
- [3] SABITZER, B., ANTONITSCH, P. K., PASTERK, S. Informatics Concepts for Primary Education. In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education on - WiPSCe '14*, 2014, DOI: <https://doi.org/10.1145/2670757.2670778>
- [4] BLAHO, A., SALANCI, L. Informatics in Primary School Principles and Experience. In *Kalaš, I., Mittermeir, R.T. (eds) Informatics in Schools. Contributing to 21st Century Education. ISSEP 2011. Lecture Notes in Computer Science 2011*, vol. 7013, pp. 129–142, DOI: https://doi.org/10.1007/978-3-642-24722-4_12
- [5] Národný inštitút vzdelávania a mládeže (NIVAM). Projekt nového kurikula, <https://vzdelavanie21.sk/>
- [6] KIROVA, G. Options for Compiling Word Problems with Tabulated Numerical Data. In *KNOWLEDGE - International Journal*, 2018, vol. 22, no. 2, pp. 391–394
- [7] TANIŞLI, D. Functional Thinking Ways in Relation to Linear Function Tables of Elementary School Students. In *The Journal of Mathematical Behavior*, 2011, vol. 30, no. 3, pp. 206–223, DOI: <https://doi.org/10.1016/j.jmathb.2011.08.001>
- [8] CARRAHER, D.; SCHLIEMANN, A.; BRIZUELA, B.; EARNEST, D. Arithmetic and Algebra in Early Mathematics

- Education. In *Journal for Research in Mathematics Education*. 2006, vol. 37, no. 2, pp. 87–115.
- [9] OECD. PISA 2021 Mathematics Framework (draft). Stockholm, Sweden, 2018.
- [10] PAPANCHEVA, R. Working with Data Tables and Graphs at Primary School. In *Journal of Process Management. New Technologies*. 2017, vol. 5, no. 4, pp. 8–12. DOI: <https://doi.org/10.5937/jouproman5-15373>
- [11] MOUSOULIDES, N. et al., C. Modeling with technology in elementary classrooms. In *CERME 6–WORKING GROUP 11*, 2010.
- [12] GABUCIO, F., MARTÍ, E., ENFEDAQUE, J., SANDRA GILABERT, S., Konstantinidou, A. Levels of graph comprehension in primary and secondary school students, In *Culture and Education*, 2010, vol. 22, no. 2, pp. 183-197, DOI:[10.1174/113564010791304528](https://doi.org/10.1174/113564010791304528)
- [13] BLAHO, A., KALAŠ, I., MORAVČÍK, M. Programovací jazyk v prostredí Emil pre 4. ročník. In: *Proc of DidInfo 2021*, s.39-49. Banská Bystrica: Univerzita Mateja Bela, Fakulta prírodných vied, 2021.
- [14] KALAŠ, I., Powerful Ideas in Lower Primary Programming. Abstract. In: *Proc. of Constructionism*, Vilnius, 2018, p. 952, <http://www.constructionism2018.fsf.vu.lt/book-of-abstracts>
- [15] KALAS, I., BLAHO, A., MORAVCIK, M. Exploring Control in Early Computing Education. In: Sergei N. Pozdniakov and Dagienė, V. (eds.) *Informatics in Schools. Fundamentals of Computer Science and Software Engineering*. ISSEP 2018. LNCS, vol 11169, Springer 2018, pp. 3-16.
- [16] KALAŠ, I., HORVÁTHOVÁ, K. Programming Concepts in Lower Primary Years and Their Cognitive Demands. In: *Passey, D. et al. (eds) Digital Transformation of Education and Learning – Past, Present and Future. OCCE 2021*. IFIP Advances in ICT, vol 642. Springer, Cham 2022. DOI: [10.1007/978-3-030-97986-7_3](https://doi.org/10.1007/978-3-030-97986-7_3)
- [17] BLAHO, A., KALAŠ, I., MORAVČÍK, M. Živý zošit: informatika a informatické základy digitálnej gramotnosti na 1. stupni. Submitted to *DidInfo 2023*, Banská Bystrica: Univerzita Mateja Bela, Fakulta prírodných vied, 2023.
- [18] CASPERSEN, M.E. et al. Informatics as a Fundamental Discipline for the 21st Century, *Communications of the ACM*, Vol. 62, No. 4, April 2019, pp. 58-63, DOI: <https://doi.org/10.1145/3310330>
- [19] SENTANCE, S. Moving to mainstream: Developing computing for all. In: *Proc. of the 14th Workshop in Primary and Secondary Computing Education*, 2019, pp: 1-2, DOI: doi.org/10.1145/3361721.3362117
- [20] PAPER, S. *Mindstorms: Children, computers, and powerful ideas*. Basic Books, 230 p., 1980.
- [21] BENTON, L., HOYLES, C., KALAS, I., NOSS, R. Bridging Primary Programming and Mathematics: Some Findings of Design Research in England. *Digit. Exp. in Math. Educ* (2017) Vol 3: 2017, Springer, pp. 115-138.

Rozvoj informatických kompetencií u žiakov so zrakovým postihnutím formou Letnej počítačovej školy

Development of IT competencies of students with visual impairments in the form of a Summer Computer School

Mária Čujdíková
KDMFI, FMFI, UK
Mlynská dolina
84248 Bratislava
Slovenská Republika
maria.cujdikova@fmph.uniba.sk

Mária Stankovičová
Centrum podpory študentov, UK
Šafárikovo námestie 6, P.O.BOX 440
814 99 Bratislava
Slovenská republika
maria.stankovicova@rec.uniba.sk

ABSTRACT

In this paper we share our experience from the Summer Computer School for students with visual impairment. We decided to target upper secondary pupils who are planning to continue their studies at university. From our previous experience with university students, we know that skills in using various technologies are crucial for their studies. In the educational process, ICT opens the door to independence, self-confidence and, finally, broader opportunities in their future career choices. The programme consisted of activities such as using assistive technology, scanning documents, text editing, programming, using mobile applications and digital resources accessible to users with visual impairments. The agenda also included leisure activities involving navigating, self-care activities, board games and watching films with audio description. A valuable experience for the participants was the presence of lecturers with visual impairments who, in addition to the professional programme, shared their own experiences from their student days in informal conversations.

Keywords

Students with visual impairment. ICT. Assistive technology. Summer school.

ABSTRAKT

V našom príspevku prezentujeme skúsenosti nadobudnuté počas realizácie Letnej počítačovej školy pre žiakov so zrakovým postihnutím. Ako cieľovú skupinu sme zvolili žiakov vyššieho sekundárneho vzdelávania, ktorí plánujú pokračovať v štúdiu na vysokej škole. Z našich doterajších skúseností s vysokoškolskými študentmi vieme, že pre ich štúdium sú zručnosti v používaní rôznych technológií kľúčové. Vo vzdelávacom procese im otvárajú dvere k samostatnosti, sebadôvere a v neposlednom rade aj širšie možnosti pri výbere budúceho povolania. Program pozostával z aktivít ako používanie asistenčných technológií, skenovanie dokumentov, editácia textu, programovanie, využívanie mobilných aplikácií a digitálnych zdrojov prístupných pre používateľov so zrakovým postihnutím. Súčasťou programu boli aj voľnočasové aktivity zahŕňajúce orientáciu v priestore, seba-obslužné činnosti, spoločenské hry a sledovanie komentovaných filmov. Cennou skúsenosťou pre účastníkov bola prítomnosť lektorov so zrakovým postihnutím, ktorí sa okrem odborného programu v rámci neformálnych rozhovorov podelili o vlastné zážitky zo študentských čias.

Kľúčové slová

Žiaci so zrakovým postihnutím. IKT. Asistenčné technológie. Letná škola.

1 ÚVOD

V kontexte vzdelávania žiakov v predmete informatika sa informatické myslenie a kompetencie rozvíjajú pomocou základných súčastí informatiky (algoritmy a programovanie), ktoré sú doplnené digitálnymi kompetenciami a digitálnou gramotnosťou [1]. Zrakové postihnutie ovplyvňuje spôsob získavania informácií a vytváranie predstáv o okolitom svete. Používanie digitálnych technológií uľahčuje prístup k informáciám. Technológie tak predstavujú pre ľudí so zrakovým postihnutím dôležitý nástroj na získavanie informácií v osobnom aj profesijnom živote. Žiaci so zrakovým postihnutím plnia vo výchovno-vzdelávacom procese rovnaké ciele ako ich intaktní spolužiaci. Jedným z dôležitých predpokladov je používanie kompenzačných pomôcok a využívanie asistenčných technológií [2, s. 31] Centrum podpory študentov so špecifickými potrebami na Univerzite Komenského v Bratislave (ďalej CPŠ) každoročne organizuje informačné stretnutie pre stredoškolákov s ťažkým zdravotným postihnutím, ktorí majú záujem o štúdium na univerzite [3]. Keďže študenti so zrakovým postihnutím sú najviac odkázaní na používanie technológií, tak považujeme za dôležité venovať sa tejto téme dôkladnejšie. CPŠ má bohaté skúsenosti s rôznymi formami podpory študentov so zrakovým postihnutím a disponuje širokým výberom technológií vhodných práve pre túto cieľovú skupinu [4].

Oblasti vzdelávania študentov so zrakovým postihnutím sa vo svojich výskumoch venujú aj pedagógovia na Katedre didaktiky matematiky, fyziky a informatiky na FMFI Univerzity Komenského. Medzi hodnotné prínosy patria projekty zamerané na vývoj programovacích prostredí pre žiakov so zrakovým postihnutím [5, 6]. Preto sme sa rozhodli pre vzájomnú spoluprácu. Letnú počítačovú školu (ďalej LPŠ) sme využili na výskum zameraný na aktuálne potreby mladých ľudí so zrakovým postihnutím, ktorí sa pripravujú na vysokoškolské štúdium, respektíve sa chcú uplatniť na trhu práce. Zamerali sme sa na používanie technológií užitočných v študijnom aj pracovnom živote.

Na LPŠ sa prihlásilo 8 účastníkov, z toho dvaja nevidiaci a 6 slabozrakých žiakov.

2 ZOSTAVENIE PROGRAMU PODĽA ZÁUJMOV ÚČASTNÍKOV

Súčasťou prihlasovacieho formulára pre záujemcov o LPŠ bol úvodný prieskum, v ktorom žiaci uviedli, o ktoré technológie majú záujem a považujú ich za užitočné. Ďalej sme v prieskume zisťovali, aké skúsenosti s technológiami žiaci už majú. Na základe

odpovedí, ktoré sme získali z prihlasovacieho formulára, sme si utvorili predstavu o používateľských zručnostiach a záujmoch účastníkov.

Súčasťou prieskumu v prihlasovacom formulári bola otázka, ktorou sme zisťovali, ako účastníci hodnotia úroveň svojich počítačových zručností. Uvedomovali sme si, že nesie v sebe riziko, že sa účastníci nadhodnotia alebo podcenia. Žiaci sa vnímali ako bežní používatelia, čo sa aj potvrdilo počas ich činnosti na LPŠ.

Ďalej sme zisťovali, aké asistenčné technológie žiaci aktívne používajú. Zistili sme, že nevidiaci žiaci používajú čítač obrazovky JAWS [7] a slabozrakí študenti používajú aplikáciu *Zväčšovač obrazovky*, ktorá je súčasťou operačného systému. Jeden slabozraký študent používal aj čítač obrazovky NVDA [8]. Dvaja slabozrakí študenti používali notebook s dotykovou obrazovkou.

Tiež sme zisťovali záujem žiakov o aktivity zamerané na využívanie rôznych technológií. V ponuke boli spracovanie zvuku, softvér na prevod reči do textu, spracovanie videa, 3D modelovanie, programovanie, tvorba blogov a osobných webových stránok, robotika, umelá inteligencia, aplikácie pre smartfóny, iné. Žiakovi sme požiadali, aby vybrali najviac tri možnosti.

Najväčší záujem prejavili účastníci o aplikácie pre smartfóny. Ukázalo sa, že k tomu budeme potrebovať viac informácií o operačných systémoch na zariadeniach, ktoré účastníci používajú. Pre ten účel sme ešte dodatočne robili medzi prihlásenými prieskum. V okruhu záujmov ďalej bolo programovanie, tvorba blogov a osobných webových stránok. Účastníci prejavili záujem aj o softvér na prevod reči do textu a spracovanie zvuku a videa. Ojedinele zaujalo 3D modelovanie, robotika a umelá inteligencia. Možnosť „iné“ nikto nezvolil.

Tento prehľad záujmov nám pomohol pri konkretizovaní návrhov workshopov. Oslovili sme vybraných pedagógov z katedry a bývalých absolventov so zrakovým postihnutím, ktorí sa venujú technológiám. Osloveným lektorom sme predstavili preferencie žiakov a prediskutovali s nimi možné smerovanie a náplň workshopov. Obsah workshopov bol vytvorený podľa toho, čo považovali pozvaní lektori za zaujímavé a dôležité. Vychádzali pri tom zo svojich odborných skúseností. Výsledný program tvorili 4 okruhy – **Asistenčné technológie, Základy programovania, Prístupnosť beletrie a študijnej literatúry, Užitočný softvér pre osobnú potrebu**. Okrem odborného programu sme sa zamerali aj na sociálne a voľnočasové aktivity.

3 PRIEBEH WORKSHOPOV

Žiaci sa do jednotlivých aktivít zapájali s nadšením a s ich plnením nemali väčšie problémy. V prípade, že potrebovali s niečím poradiť, obrátili sa na lektorov, ktorí im ochotne pomohli. Na workshopoch panovala priateľská, uvoľnená atmosféra. Účastníci používali vlastné notebooky a smartfóny, ktoré boli rôznorodé z hľadiska hardvéru, ako aj nainštalovaného softvéru. V súvislosti s tým sa vyskytlo niekoľko technických problémov. V konečnom dôsledku žiaci aj lektori považovali túto skúsenosť za pozitívnu. Ocenili, že sa s danými problémami nestretli osamote a mohli ich vyriešiť spoločne s lektormi.

Počas každého workshopu žiaci dostali od lektorov podkladové materiály so stručnými návodmi pre činnosti, ktorým sa venovali. Materiály obsahovali aj internetové odkazy, na ktorých mohli nájsť viac informácií k danej téme workshopu.

3.1 Asistenčné technológie

V časti programu venovanej asistenčným technológiám si žiaci mohli vyskúšať obsluhu rôznych skenerov, snímacie zariadenie do diaľky a brailové riadky. Dozvedeli sme sa, že s uvedenými asistenčnými technológiami sa žiaci oboznámili prvý raz.

Ako najužitočnejšie zariadenie pre nevidiacich aj slabozrakých žiakov sa ukázal skener s hlasovým výstupom **EasyReader** (Obrázok 1). Pozorovali sme, že obsluha skenera bola pre žiakov ľahko pochopiteľná. Ovládacie tlačidlá dokázali jednoducho rozlíšiť hmatom. Ako vhodné sa ukázalo aj kontrastné podsvietenie tlačidiel. „Hovoriaci skener“ žiaci považovali za praktické zariadenie pre podporu samostatnosti pri sprístupňovaní študijných materiálov, ale aj voľnočasovej beletrie. Snímky, ktoré študenti naskenovali, uložili vo formátoch JPG, TXT, MP3 a neskôr použili pri iných workshopoch.



Obrázok 1: Skenovacie zariadenie s hlasovým výstupom EasyReader

Veľmi praktickým zariadením pre slabozrakých žiakov sa ukázalo snímacie zariadenie do diaľky **Readit Scholar**. Pomocou tohto zariadenia pripojeného k notebooku žiaci zameriavali tabuľu a jej obsah sledovali na obrazovke. Všimli sme si, že žiaci sa v obslužnej aplikácii rýchlo zorientovali a využili možnosť modifikovať veľkosť tlačidiel. Ako dôležité sa ukázalo vhodné osvetlenie snímanej plochy, výrazný farebný kontrast pozadia a textu. Snímaný povrch by sa nemal lesknúť.

Na **ručnom skeneri** boli slabo hmatateľné tlačidlá a malý displej s informáciami o nastavení. Žiaci to považovali za nedostatky, pretože manipulácia so skenerom bola pre nich veľmi komplikovaná.

Brailové riadky si vyskúšali iba nevidiaci žiaci (Obrázok 2). Čítanie na brailovom riadku považovali za efektívne. V texte sa orientovali oveľa rýchlejšie v porovnaní s tým, keď používali iba hlasový výstup čítača obrazovky. Ľahšie našli aj preklepy a gramatické chyby v texte.



Obrázok 2: Žiak používa brailový riadok

3.2 Základy programovania

Počas workshopov zameraných na programovanie sme mohli pozorovať prevažne zánietenie a radosť z objavovania nového, keďže so všetkými využitými prostrediami sa žiaci stretli prvýkrát. Pri programovaní v prostredí **Code Jumper** [9] sme si všimli, že žiakov baví skladanie komponentov (Obrázok 3). Zároveň sme však pozorovali, že najefektívnejšie sa darilo spolupracovať vo dvojici. Obaja žiaci spoločne uvažovali a vhodne si asistovali pri realizácii svojich nápadov. Pri trojčlenných skupinách nastávali situácie, kedy niektorý zo žiakov len čakal. Pri jednej trojici sme si všimli, že pre nich bolo náročné postupovať na splnení spoločného cieľa. Nakoniec aktivitu prebral len jeden žiak a zvyšní dvaja sa iba pozerali. Prácu v prostredí Code Jumper komplikovali tiež technické problémy s pripojením Bluetooth. To spôsobilo, že žiaci museli niekedy čakať na opätovné pripojenie a nemohli okamžite prehrať novú verziu svojho kódu. Boli tak pripravení o radosť z okamžitej spätnej väzby, čo sme mohli pozorovať aj na zmene nálady.



Obrázok 3: Žiaci skladajú komponenty programovacieho prostredia Code Jumper

Programovanie v prostredí **Coshi** [9] študentom nerobilo žiadne väčšie problémy. Pre študentov sme pripravili reliéfne vytlačené mriežky v rozmeroch 3x3, 5x5, 7x7 pre rôzne typy úloh. Nevidiaci študenti ocenili možnosť „pozrieť“ si vytlačenú mriežku. S nadšením po nej prechádzali prstami a dodatočne vyhodnotili, že im táto pomôcka pomohla s predstavením si prostredia (Obrázok 4). Samostatná práca podľa pripravených pracovných listov sa ukázala ako vhodná. Študenti mohli postupovať svojím tempom a tešiť sa zo svojich úspechov po vyriešení každej úlohy. V riešení pripraveného zadania sa dostali rôzne ďaleko.



Obrázok 4: Žiak používa reliéfne vytlačenú mriežku pri programovaní v prostredí Coshi

V rámci programovania si žiaci vyskúšali aj tvorbu mobilných aplikácií. Workshopy prebiehali zvlášť pre slabozrakých žiakov a zvlášť pre nevidiacich žiakov. Programovanie aplikácií v prostredí **MIT App Inventor** [10] predstavovalo pre slabozrakých študentov veľkú výzvu. Pozorovali sme, že pre nich bolo pomerne náročné zorientovať sa v prostredí. Pri práci

používali zväčšenie obrazovky a strácali prehľad, kde sa nachádzajú potrebné komponenty. Ako problematické sa pre túto skupinu žiakov ukázalo aj používanie programovacích blokov, ktoré museli uchopiť myšou a ťahať. Mierne problémy v orientácii spôsoboval aj slabý farebný kontrast prostredia. S pomocou lektorov sa však všetkým žiakom nakoniec podarilo úspešne naprogramovať celú aplikáciu na určovanie azimutov. Pri testovaní vytvorenej aplikácie sme u nich mohli vidieť radosť z vlastného fungujúceho produktu. Jednej žiačke sa pri testovaní stalo, že jej mobil kvôli technickým parametrom nechcel spustiť aplikáciu, ktorú vytvorila. Bola z toho sklamaná, no neskôr aplikáciu otestovala na inom zariadení a mohla tiež zažiť pocit úspechu. Aj napriek ťažkostiam pri programovaní v tomto prostredí ho žiaci hodnotili pozitívne. Niektorí dokonca začali uvažovať o ďalších aplikáciách, ktoré by si v ňom chceli neskôr vytvoriť.

Pre nevidiacich žiakov bolo pripravené programovanie v prostredí **Macroroid** [11]. Potrebovali chvíľu čas, aby sa zorientovali v ovládacích prvkoch. Programovanie aplikácií prebiehalo bez komplikácií. Workshop ich zaujal. Žiaci obzvlášť ocenili, že si mohli vyskúšať vytvorenie malých aplikácií užitočných priamo pre nevidiacich. Páčilo sa im napríklad to, že mobil po zatrasení povedal čas, alebo sa po otočení mobilu zapol Messenger.

3.3 Prístupnosť beletrie a študijnej literatúry

Počas workshopov zameraných na prístupnosť informácií sa žiaci dozvedeli o rôznych formátoch spracovania tlačených kníh do prístupných elektronických dokumentov. Cez online hovor sme sa spojili so zástupcami Slovenskej knižnice pre nevidiacich, ktorí predstavili svoju činnosť a pozvali účastníkov stať sa aktívnymi používateľmi knižnice [12]. Okrem spracovaných kníh ponúka knižnica aj zvukové nahrávky komentovaných filmov bez obrazu.

Ďalšou z možností, ako získať prístupný elektronický dokument, je služba **Robobraille** [13]. Výhodou tejto služby je lokalizácia do slovenského jazyka. Žiaci poslali cez webové rozhranie vopred naskenovaný dokument, ktorý nebol rozpoznávaný (neprešiel OCR) a nechali ho spracovať do rôznych formátov (DOCX, MP3 aj PDF). Počas tejto zdanlivo jednoduchšej aktivity nastali neočakávané komplikácie, ktoré súviseli s rôznymi verziami čítačov obrazovky a rôznymi webovými prehliadačmi. U niektorých používateľov zväčšovača obrazovky sa webová stránka nenačítala celá. U jednej žiačky na stránke nefungoval čítač obrazovky. Nevedeli sme si tieto chyby vysvetliť, lebo u iných stránka fungovala bez problémov. Postupne sme u každého účastníka našli vhodnú kombináciu webového prehliadača a ním používanej technológie. Len u jednej nevidiacej žiačky sa to nepodarilo a našli sme pre ňu alternatívne riešenie použitím webu <https://www.onlineocr.net/>. Neplánovane sme tak otestovali ďalšiu webovú službu na získanie prístupného dokumentu.

3.4 Užitočný softvér pre osobnú potrebu

Najväčší úspech mal workshop o aplikáciách pre iOS a Android prístupných pre ľudí so zrakovým postihnutím a práca s aplikačným balíkom **Corvus** [14]. Keďže inštalácia aplikačného balíka si vyžaduje vážnejší zásah do nastavení zariadenia, tak lektor priniesol niekoľko predvádzacích smartfónov. Ušetril sa tak aj čas, ktorý by si vyžiadala inštalácia a následné odinštalovanie zo žiackych zariadení. Aplikácie balíka Corvus ponúkali rozpoznávanie snímaného textu (OCR), snímanie čiarových kódov, rozpoznávanie bankoviek, opis predmetov v okolí, opis obrázkov. Užitočné bolo vyhľadávanie autobusových a vlakových spojov. Žiaci považovali za praktickú aplikáciu na snímanie QR kódov z pokladničných blokov. Aplikačný balík Corvus teda prijali

s veľkým nadšením (Obrázok 5). Ten sa ukázal ako užitočný aj pre nevidiacich aj pre slabozrakých žiakov. Pozorovali sme, že nevidiaci v ňom využívali možnosti hlasového výstupu, zatiaľ čo slabozrakí žiaci skúšali možnosti na zväčšovanie obrazovky. Žiaci uviedli, že tento aplikačný set nepoznali skôr. Zhodli sa na tom, že obsahuje užitočné aplikácie.



Obrázok 5: Žiak používa aplikačný balík Corvus

Okrem aplikačného balíka Corvus si žiaci mohli otestovať aplikácie dostupné na Google Play, respektíve v App Store, ktoré ponúkali podobné funkcie. Lektori im dávali tipy na aplikácie, ktoré sú prístupné pre používateľov so zrakovým postihnutím. Mnohé z týchto aplikácií boli pre žiakov dovtedy neznáme a s nadšením sa púšťali do ich inštalovania a skúmania. Pri práci s aplikáciami sme nezaznamenali žiadne väčšie problémy. Žiaci si niektoré aplikácie ponechali v zariadeniach nainštalované aj po ukončení workshopu (rozpoznávanie textu, skenovanie QR kódov).

Už počas prípravy Letnej školy sme za veľmi dôležitý považovali workshop venovaný **editovaniu textového dokumentu v programe MS Word**. Žiaci so zrakovým postihnutím pracujú s učebnými materiálmi v elektronickej forme. Preto má význam tieto kompetencie u nich rozvíjať [15]. Pre účastníkov sme pripravili jednoduché textové dokumenty s neštruktúrovaným textom. Cieľom workshopu bolo naučiť žiakov používať pri formátovaní textu štýly a tiež ich oboznámiť s rôznymi klávesovými skratkami pre efektívnu prácu s elektronickým dokumentom. V rámci úpravy textu bol zaradený aj prevod textu na tabuľku, vloženie obrázku a priradenie alternatívneho textu k obrázku. Výsledný dokument sa žiaci naučili exportovať do formátu PDF. Okrem praktického formátovania textu sa žiaci dozvedeli aj o význame úpravy v kontexte tvorby prístupného elektronického dokumentu. Workshop viedla slabozraká lektorka, ktorá aktívne používa čítač obrazovky. Všimli sme si, že študenti majú veľmi odlišné skúsenosti s úpravou textu. Po každom kroku niektorej úpravy sme kontrolovali, či sa každému podarila. Ak nastali komplikácie, tak sme ich najprv vyriešili a až potom pokračovali. Napríklad sme si všimli, že účastníci majú rôzne verzie programového balíka MS Office, čo sa prejavilo rôznymi klávesovými skratkami pre niektoré funkcie. Výrazne sa to prejavilo v textovom editore z balíka Libre Office a v MS Word pre Mac. V oboch prípadoch bolo potrebné nájsť spôsob, ako dosiahnuť požadované úpravy a žiaci neboli ukrátení o užitočné skúsenosti.

Prezentovaný hudobný softvér **NoteWorthy Composer** [16] používali účastníci ako demoverziu. Nevidiaci žiaci ocenili jeho plnú prístupnosť pre čítač obrazovky a možnosť ovládania klávesovými skratkami. Žiaci nemuseli mať hudobné vzdelanie. Skúsená nevidiaca lektorka im postupne vysvetlila ovládanie, čiže

vkladanie nôt, nastavenie tónov a dĺžky, prehratie výslednej melódie.

4 Voľnočasové aktivity

Čas medzi jednotlivými workshopmi bol príležitosťou na rôzne spoločenské a voľnočasové aktivity. Účastníci mali možnosť rozvíjať svoje zručnosti aj v seba-obslužných činnostiach. Každý deň niektorá dvojica pripravovala pre ostatných raňajky. Precvičili si krájanie pečiva a zeleniny, natieranie masla, prípravu čaju. V rámci voľnočasových aktivít boli pripravené rôzne spoločenské hry určené pre nevidiacich, ozvučená lopta, zvukové pexeso, šach a piškvorky. Jeden večer sme si spoločne pozreli komentovaný film. Veľa času sme trávili aj v okolitej prírode na výletoch. V rámci jednej prechádzky mali účastníci možnosť naučiť sa rozpoznávať rôzne listy drevín (Obrázok 6).



Obrázok 6: Rozpoznávanie listov

5 SPÄTNÁ VÄZBA OD ÚČASTNÍKOV A LEKTOROV

V závere LPŠ sme sa rozprávali s účastníkmi o ich názoroch na témy a priebeh workshopov. Tiež sme ich požiadali, aby vyplnili krátke hodnotiace dotazníky. Na základe analýzy odpovedí z výstupných dotazníkov a rozhovorov usudzujeme, že študenti hodnotia letnú školu pozitívne a jej program ako užitočný pre ich osobný aj študijný život. Pri otázke: „*Ktoré z tvojich očakávaní LPŠ nesplnila?*“ traja uviedli, že ich očakávania boli naplnené a traja ďalší zhodnotili, že ich očakávania boli dokonca prekonané. Jedna študentka napísala: „*Všetko dopadlo nad moje očakávania. Počas týždňa som sa veľa dozvedela, no i veľa zažila.*“ Niektorí študenti však nemali na začiatku veľké očakávania, ako poukazuje odpoveď: „*Nemala som vysoké očakávania, veľmi ma to tu prekvapilo.*“

Žiaci medzi najužitejšie aktivity zaradili zoznámenie sa s aplikáciami pre smartfóny, prácu s textom, informácie o prístupných dokumentoch v SKN, komponovanie melódii v notačnom programe, programovanie a spoznanie nových klávesových skratiek. Predpokladáme, že táto rôznosť vo vyhodnotení vyplývala z individuálnych záujmov a preferencií jednotlivých študentov.

Veľká rozdielnosť názorov odznela pri hodnotení programovacích workshopov. Žiaci vnímali programovanie v prostrediach Code Jumper ako „programovanie pre deti“, ale aj napriek tomu vyhodnotili, že sa im aktivita páčila. Poukazujú na to slová jednej študentky: „*Spoznali sme rôzne druhy programovania, na ktoré by človek povedal, že sú skôr pre malé deti, no zabavili sme sa s nimi aj my veľké deti. Ved' nie každý deň človek naprogramuje bombu zo zvukov hodín.*“ Iná študentka uviedla: „*Mňa osobne najviac bavilo hmatové programovanie, keď sme si naprogramovali bombu.*“ Podobný názor mali žiaci na programovanie v prostredí Coshi.

Vnímali ho ako prostredie určené skôr pre deti, no aj napriek tomu vyjadřili, že ich to bavilo.

Konkrétne predstavy žiakov, čomu presne by sa v rámci programovania radi venovali, boli rozdielne. Dvaja žiaci by pred programovaním v detských prostrediach uprednostnili programovanie v jazyku Python. Jeden z nich by ocenil zameranie na zložitejšie programovacie koncepty, ako sú funkcie a podmienky.

Nevidiaci lektori vyhodnotili ako najdôležitejšiu oblasť, v ktorej by sa mali študenti zlepšiť, používanie klávesových skratiek. Počas aktivít na workshopoch si všimli, že slabozrakí žiaci nevyužívajú klávesové skratky v dostatočnej miere. Lektori považujú klávesové skratky za kľúčové, keď chcú byť efektívni pri svojej práci. Jedna lektorka ako príklad uviedla: „Už aj v obyčajných mailoch sa bez skratiek ťažko orientuje. Bez nich človek ide, že fajn, ale 10 minút zbytočne stratených.“

Žiaci boli spokojní aj s voľnočasovými aktivitami. Medzi najobľúbenejšie patrili výlety a spoločenské hry. V budúcnosti by uvítali viac výletov a aktivít zameraných na sebaobsluhu, vhodný by bol aj športový deň. Pre žiakov bolo prínosné aj nadobudnutie nových priateľstiev. Jedna študentka to vyjadřila slovami: „Odchod pre veľa z nás bol priam drastický, pretože sme sa stretli taká skvelá priatelia ľudí. Všetci by sme si priali ešte ostať.“

6 ZÁVER

Vďaka organizovaniu Letnej počítačovej školy sa nám podarilo lepšie porozumieť potrebám žiakov vyššieho sekundárneho vzdelávania so zrakovým postihnutím v oblasti používania technológií. Na základe pozorovaní sme zistili, s akými technickými problémami sa stretávajú. Niektoré komplikácie súviseli s tým, že žiaci používali rôzne zariadenia a verzie aplikácií. Ako pozitívnu skúsenosť hodnotíme to, že sa nám podarilo väčšinu problémov okamžite vyriešiť. Vďaka tomu aj žiaci získali nové zručnosti.

Podarilo sa nám vyhodnotiť, čo považovali účastníci v ponúknutých oblastiach za najhodnotnejšie. Ako najužitejšie pre osobný a študijný život považovali aplikácie pre smartfóny a aplikačný balík Corvus. Pozitívny ohlas na užitočné aplikácie pre smartfóny, ktoré zároveň spĺňajú požiadavky prístupnosti pre používateľov so zrakovým postihnutím nám poukazujú na záujem mladých ľudí o efektívne narábanie s informáciami a ich ochotu a schopnosť používať k tomu moderné technológie. Považujeme to za dôležitý predpoklad pre rozvoj informatických kompetencií.

Zisťovali sme aj to, ktoré oblasti by žiaci potrebovali rozvíjať do väčšej hĺbky. Predovšetkým by uvítali workshop zameraný na programovanie v Pythone a tiež zoznámenie sa s náročnejšími programátorskými konceptami.

V budúcnosti by sme preto chceli doplniť workshopy o tie témy, ktoré sa ukázali ako potrebné. Nepodarilo sa nám pokryť všetky okruhy záujmov, o ktorých sa účastníci vyjadřili vo vstupnom prieskume. Takže aj tieto oblasti pre nás ostávajú v pozornosti do ďalších ročníkov.

POĎAKOVANIE

Ďakujeme zamestnancom a spolupracovníkom Centra podpory študentov so špecifickými potrebami UK za trpezlivú a obetavú prácu pri pripravách a tiež aj počas priebehu Letnej počítačovej školy. Vážime si aj spoluprácu s pedagógmi z KDMFI FMFI UK a ďakujeme za zapožičanie programovacieho prostredia Code Jumper a aktívny príspevok formou workshopu. Tento článok

prezentuje časť výstupov naplánovaných a spolufinancovaných z grantu KEGA 010UK-4/2022. Naša vďaka patrí aj lektorom so zrakovým postihnutím z radov bývalých absolventov (nielen z UK), ktorí venovali svoj voľný čas a poskytli okrem odborných informácií aj osobné skúsenosti.

BIBLIOGRAFICKÉ ODKAZY

- [1] BLAHOVÁ, Viera. *Informatické myslenie na Slovensku a v Európe*. In: *Studia Scientifica Facultatis Paedagogicae Universitas Catholica Ružomberok* [online]. Ružomberok: VERBUM – vydavateľstvo Katolíckej univerzity v Ružomberku, 2023, 22 (1), s. 18-25. ISSN 1336-2232. Dostupné na: <https://doi.org/10.54937/ssf.2023.22.1.18-25>
- [2] *Vzdelávací program pre deti a žiakov so zrakovým postihnutím*. [online] Bratislava: Ministerstvo školstva slovenskej republiky, Štátny pedagogický ústav 2009 [cit. 2023-02-10]. s. 31. Dostupné na: https://www.statpedu.sk/files/sk/svp/statny-vzdelavaci-program/vp-deti-ziakov-so-zdravotnym-znevychodnenim/vp-deti-ziakov-so-zrakovym-postihnutim/vp_zp-isced_1_2_3_5.pdf
- [3] MENDELOVÁ, E. *Supporting students with disabilities, training future teachers*. Visions and Strategies for the New Century: Proceedings European Conference. Grave: ICEVI, 2000. - 5 s., ISBN 90-76898.02.2.
- [4] MORAVČÍKOVÁ, E. *Support Centre for Visually impaired Students*. In: Zagler W.L., Busby G., Wagner R.R. (eds) *Computers for Handicapped Persons*. ICCHP 1994. Lecture Notes in Computer Science, vol 860. Springer, Berlin, Heidelberg, 229 – 235 s., ISBN 3-540-58476-5.
- [5] JAŠKOVÁ, E. *Three programming environments friendly to blind students in lower secondary education*, EDULEARN22 Proceedings, 2022. pp. 8099-8108, ISSN 2340-1117.
- [6] JAŠKOVÁ, E., KARASOVÁ, M., KOVÁČOVÁ, N. *Dva programovacie jazyky priateľské k nevidiacim žiakom nižšieho sekundárneho vzdelávania*, DidInfo 2019, Banská Bystrica: Univerzita Mateja Bela, Fakulta Prírodných vied, 2019, s. 82-87, ISBN 978-80-557-1533-9. Dostupné na: https://io.fpv.umb.sk/didinfo/Zbornik_Didinfo_2019.pdf
- [7] JAWS – čítač obrazovky. <https://www.freedomscientific.com/products/software/jaws/>
- [8] NVDA – čítač obrazovky. <https://www.nvaccess.org/download/>
- [9] Programovacie prostredia pre nevidiacich. <http://vin.edu.fmph.uniba.sk/prostredia.html>
- [10] Programovacie prostredie MIT App Inventor. <https://appinventor.mit.edu/>
- [11] Programovacie prostredie Macrodroid. <https://www.macrodroid.com/>
- [12] SKN – Slovenská knižnica pre nevidiacich Mateja Hrebandu v Levoči. <https://www.skn.sk/>
- [13] Robobraille – online webová služba na transformáciu elektronických dokumentov do prístupnej formy pre ľudí so zrakovým postihnutím. <https://www.robobraille.org/sk/>
- [14] Corvus – aplikačný balík na prístupnenie mobilných telefónov s operačným systémom Android. <http://corvuskit.com/>
- [15] KARUNOVÁ, H., HERZINGER, M. *Digitalizace textů pro studenty se zrakovým postižením na vysoké škole*. Olomouc: Univerzita Palackého, 2021, ISBN 978-80-244-5834-0.
- [16] Hudobný softvér NoteWorthy Composer. <https://noteworthycomposer.com/>

Výsledky sociálně znevýhodněných žáků v soutěži Bobřík informatiky 2022

Results of socially disadvantaged pupils in the Bebras challenge 2022

Václav Dobiáš
Katedra informatiky PF JČU
Braníšovská 1645/31a
České Budějovice
Česká republika
dobias@pf.jcu.cz

ABSTRACT

According to the 2030+ strategy, one of the biggest problems in the Czech education system is inequality in access to quality education [1]. Our main research aim is thus to ascertain divides in computational thinking. We were also interested in time development of this inequality. Therefore, we selected schools with a higher proportion of socially disadvantaged pupils (hereafter referred to as ZPSZŽ) and the rest schools participating in the Bebras Challenge. We compared this two groups. The results show statistically significant differences in computational thinking between pupils from mainstream schools and ZPSZŽ schools, these differences widening as pupils age. Furthermore, we found statistically significant differences in all components of computational thinking between pupils from mainstream schools and ZPSZŽ schools. We found the largest differences were in the components of abstraction and finding repeating patterns.

Keywords

Computational thinking, digital divide, socially disadvantaged students.

ABSTRAKT

Podle Strategie 2030+ je jedním z největších problémů českého školství nerovnost v přístupu ke kvalitnímu vzdělávání [1]. Zajímalo nás, jak se tato nerovnost projevuje v soutěži Bobřík informatiky a jak v jednotlivých složkách informatického myšlení. Dále nás zajímalo, zda s věkem dochází k vývoji nerovnosti. Ze škol zúčastněných v soutěži jsme proto vybrali školy se zvýšeným podílem sociálně znevýhodněných žáků (dále ZPSZŽ). Jejich výsledky jsme pak porovnali s výsledky ostatních škol zúčastněných v soutěži Bobřík informatiky. Výsledky ukazují na statisticky významné rozdíly v informatickém myšlení mezi žáky z běžných škol a ZPSZŽ škol, přičemž tyto rozdíly se s věkem žáků postupně prohlubují. Dále jsme našli statisticky významné rozdíly ve všech složkách informatického myšlení mezi žáky z běžných škol a ZPSZŽ škol. Největší rozdíly jsme našli u složek abstrakce a hledání opakujících se vzorů.

Klíčové slová

Informatické myšlení, digitální rozdělení, sociálně znevýhodnění žáci.

1 ÚVOD

Svět kolem nás se stále více digitalizuje. Čerstvým příkladem může být nástup digitálních technologií v zemědělství, který je běžně nazýván čtvrtá zemědělská revoluce [2, 3, 4, 5]. V kontextu

zemědělství tak zaznívají pojmy spadající do počítačových věd jako například: umělá inteligence, automatizace, datová analýza, nebo robotika [2, 4]. Stejná situace panuje i v oblasti Průmyslu 4.0 [6, 7]. Práce v digitalizovaných oborech ovšem vyžaduje jistou míru porozumění těmto technologiím informatického myšlení. Pro podporu tohoto trendu mnohé státy (například: Velká Británie, Slovensko, Česká republika, Estonsko) inovují výuku a zavádějí do výuky rozvoj informatického myšlení.

Zásadní otázkou je, jak se tomuto trendu dokáží přizpůsobit sociálně znevýhodnění. V literatuře je již velmi dobře popsán koncept různých forem digitálního rozdělení. První forma digitálního rozdělení byla popsána jako rozdíl v přístupu k digitálním technologiím [8]. Druhá forma digitálního rozdělení je popsána jako nerovnost ve schopnostech využívání digitálních technologií [9]. Třetí forma digitálního rozdělení je popsána jako nerovnost ve využívání digitálních technologií ve svůj prospěch [10]. V kontextu zemědělství například mnozí autoři mluví o digital divide in farming [2, 3, 11]. Někteří další autoři si začínají všímat nerovnosti přesahující klasické digital divide směrem k informatickému myšlení. Jako například nerovnost v chápání, jak rozhodují algoritmy ve světě kolem nás [12], případně se snaží integrovat pojmy z počítačových věd do stávajícího systému digitálního rozdělení [13]. Ovšem jednotlivé formy digitálního rozdělení se především týkají uživatelského přístupu k digitálním technologiím, nedomníváme se proto, že je vhodné je spojovat se schopnostmi informatického myšlení.

Tímto článkem navazujeme na náš předchozí článek [14], ve kterém jsme identifikovali rozdíly v jednotlivých kategoriích informatického myšlení mezi žáky ze sociálně znevýhodněných a ostatních škol.

1.1 Informatické myšlení

Informatické myšlení je nově jedním z hlavních směrů výuky informatiky. Wing definuje informatické myšlení jako myšlenkové postupy zapojené při takovém formulování problémů a jejich řešení, které umožní tato řešení efektivně provést agentem zpracovávajícím informace [15]. Informatické myšlení je také popisováno jako schopnost „myslet, jako informatik při řešení problémů“ [16]. Existují různá dělení informatického myšlení do jednotlivých složek. Častým je například dělení Selbyho a Woollarda [17] na složky: abstrakce, algoritmizace, dekompozice, evaluace, generalizace a automatizace. V současné době se začínají objevovat snahy využít přístupů informatického myšlení i v jiných předmětech, než je informatika [18, 19, 20].

2 Metodologie

2.1 Výzkumné otázky

Jak bylo zmíněno v úvodu, svět se vlivem digitálních technologií výrazně mění, na což reagují vzdělávací systémy různých zemí zaváděním výuky informatického myšlení. Obáváme se, že tato změna dále rozevře pomyslné nůžky mezi běžnou společností sociálně znevýhodněnou částí společnosti. Z tohoto důvodu jsme se rozhodli navázat na námi realizovaný výzkum [14]. Oproti předchozímu článku využíváme o 68 % větší výzkumný vzorek, díky kterému máme větší skupinu ZPSZŠ škol. Dále používáme vylepšené dělení složek informatického myšlení, tak aby více odpovídal běžné klasifikaci podle Selbyho a Woollarda [17]. Dále jsme navíc sledovali rozdíly v interaktivních programovacích úlohách v české verzi soutěže Bobřík informatiky. Na základě výzkumných cílů jsme formulovali následující výzkumné otázky:

Výzkumná otázka č. 1: Existují rozdíly v informatickém myšlení mezi žáky z běžných škol a žáky ze ZPSZŠ škol?

Výzkumná otázka č. 2: Dochází k vývoji rozdílů v informatickém myšlení v závislosti na věku žáků mezi žáky z běžných škol a žáky ze ZPSZŠ škol?

Výzkumná otázka č. 3: Ve kterých složkách informatického myšlení existují rozdíly mezi žáky z běžných škol a žáky ze ZPSZŠ škol?

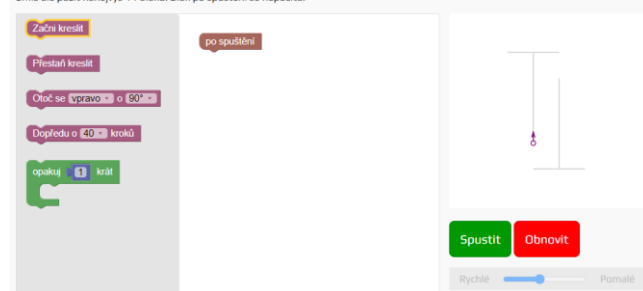
Výzkumná otázka č. 4: Existují rozdíly v interaktivních programovacích úlohách mezi žáky z běžných škol a žáky ze ZPSZŠ škol?

2.2 Sběr dat

K zodpovězení výzkumných otázek byla analyzována data z české verze soutěže Bobřík informatiky v roce 2022. Žáci v rámci soutěže vždy vypracovávali test s 12 otázkami. Na vypracování testu bylo 40 minut. V tomto roce se soutěže zúčastnilo celkem 183 564 soutěžících žáků. Žáci byli rozděleni podle věku do jednotlivých věkových kategorií. Z toho u každé věkové kategorie byly 2 interaktivní programovací úlohy. V těchto úlohách žáci pomocí blokového programovacího prostředí programovali postavu tak, aby dosáhli vyřešení předloženého problému [21]. Příklad jedné takové úlohy pro žáky 8. a 9. tříd je na obrázku č. 1.

Značka firmy

Sestav program, který nakreslí logo firmy Thomas Taylor podle obrázku. Smis ale požit nanejvýš 14 bloků. Blok po spuštění se nepočítá.



Obrázek 1: Příklad interaktivní programovací úlohy pro žáky 8. a 9. tříd

Pro potřeby výzkumu byly využity tři nejmladší kategorie spadající do základní školy. Konkrétně se jednalo o kategorie: Mini (4.–5. třída ZŠ), Benjamin (6.–7. třída ZŠ) a Kadet (8.–9. třída ZŠ). Počty jednotlivých žáků jsou uvedeny v tabulce 1. Z dat jsme vyloučili ty žáky, kteří nestandardním způsobem ukončili test. V tabulce 1 je

dále uveden procentní podíl žáků, kteří se účastnili soutěže ze všech žáků daných ročníků v České republice.

Tabulka 1: Počty účastníků Českého kola soutěže Bobřík informatiky 2022

	Mini	Benjamin	Kadet
Počet účastníků	51 174	60 853	43 259
Ze všech žáků v ČR	23,9 %	28,9 %	22,3 %

2.3 Výběr škol se zvýšeným podílem sociálně znevýhodněných žáků (ZPSZŠ škol)

Vzhledem k cílům výzkumu jsme vzorek účastníků rozdělili na 2 části. Na ZPSZŠ školy a na ostatní školy. Bohužel v současné době nejsou dostupné oficiální statistiky identifikující školy se zvýšeným podílem sociálně znevýhodněných žáků. Panem Nebřenským z MŠMT nám ale byla poskytnuta výkonová data regionálního školství v České republice. Jde o statistiky, které české školy každoročně vykazují. Ve výkazech školy reportují i počty sociálně znevýhodněných žáků. Ke školám vybraným v našem předešlém výzkumu [14] jsme tak přidali celkem 5 škol vybraných z výše uvedených výkonových dat regionálního školství. Tyto školy deklarovaly nejvyšší poměry sociálně znevýhodněných žáků ze všech svých žáků a zároveň se účastnily soutěže Bobřík informatiky. Podle informací získaných z telefonických rozhovorů se školními koordinátory Bobříka informatiky a podle statistik MŠMT mají vybrané školy mezi 18,5 % až 66 % sociálně znevýhodněných žáků. Další škola zúčastněná v soutěži bobřík informatiky se zvýšeným podílem znevýhodněných žáků pak měla podíl 16,6 % znevýhodněných žáků. Tuto školu jsme již do kategorie ZPSZŠ škol nezahrnovali. Počty jednotlivých žáků jsou v tabulce 2.

Tabulka 2: Počty žáků zúčastněných na výzkumu

	Mini	Benjamin	Kadet
Počet účastníků ze ZPSZŠ škol	352	546	371
Počty ostatních účastníků	46241	59067	41794

2.4 Klasifikace úloh podle složek informatického myšlení

Pro zodpovězení výzkumné otázky 3 jsme využili klasifikace jednotlivých úloh do složek informatického myšlení podle Selbyho a Woollarda [17]. Použité složky informatického myšlení tedy jsou: Abstrakce, Algoritmizace, Dekompozice, Evaluace, Generalizace a Hledání opakujících se vzorů – tato složka nahrazuje složku Automatizace u dělení Selbyho a Woollarda [17]. Reflektujeme tak tím krátkodobý charakter soutěže, který neumožňuje významnější využití automatizace. Rozdělení úloh do jednotlivých složek informatického myšlení podle věkových kategorií je vyobrazeno v tabulce 3. Některé otázky zároveň ověřovaly více složek informatického myšlení. Protože jsme u jednotlivých otázek vyhodnocovali pouze dvě možné varianty (žák vyřešil/neyřešil úlohu), mohla nastat situace, kdy žák nevyřešil úlohu z důvodu nedostatku dovedností z pouze v jedné složky informatického myšlení, přesto mu byl započítán neúspěch u obou

složek inforatického myšlení. Vzhledem k velikosti výzkumného vzorku, ale předpokládáme, že takováto situace výrazně neovlivnila výsledky.

Tabuľka 3: Klasifikace úloh podle složek inforatického myšlení a věkových kategorií

	Mini	Benjamin	Kadet	Celkem
Abstrakce	2	3	4	9
Algoritmizace	10	4	7	21
Dekompozice	1	2	4	7
Evaluace	0	3	2	5
Generalizace	0	0	0	0
Hledání opakujících se vzorů	2	6	3	11
Interaktivní programovací úlohy	2	2	2	6

2.5 Analýza dat

Pro analýzu dat byly využity programy MS Excel a R. V těchto programech jsme vytvářeli kontingenční tabulky a statisticky testovali hypotézy. Vzhledem k velikosti našeho výzkumného vzorku ($n = 148\ 371$) jsme veškeré statistické testy realizovali na hladině významnosti $\alpha = 0,01$.

Pro zodpovězení výzkumné otázky č. 1 jsme testovali normalitu dat pomocí Anderson-Darling testu. Protože data nebyla normálně rozložena, využili jsme dvouvýběrový Wilcoxonův test. Pro zodpovězení výzkumné otázky č. 3 a č. 4 jsme pak využili Chí-kvadrát testu.

3 Výsledky

3.1 Výzkumná otázka č. 1: Existují rozdíly v inforatickém myšlení mezi žáky z běžných škol a žáky ze ZPSZŽ škol?

Nejvyšší dosaženou p-hodnotou byla hodnota $p = 0,0000018$ v kategorii Mini. Ve všech ostatních kategoriích jsme zjistili ještě nižší p-hodnoty. Na základě výsledků můžeme odpovědět na výzkumnou otázku č. 1.: Tedy mezi žáky ze ZPSZŽ škol a žáky z ostatních škol existují statisticky významné rozdíly ve všech sledovaných věkových kategoriích.

3.2 Výzkumná otázka č. 2: Dochází k vývoji rozdílů v inforatickém myšlení v závislosti na věku žáků?

Námi zjištěné výsledky ukazují, že ve všech sledovaných věkových kategoriích existují statisticky významné rozdíly mezi žáky ze ZPSZŽ škol a žáky ostatních škol. Pokud porovnáme procentní rozdíly v průměrném zisku bodů (tabulka 5) uvidíme výrazný nárůst z 10,8 p. b. u věkové kategorie Mini až ke 22 p. b. u věkové kategorie Kadet. Rozdíly v úrovni inforatického myšlení se tak pravděpodobně v závislosti na věku žáků postupně zvětšují. Dochází zde tedy k postupnému rozevírání nůžek v úrovni inforatického myšlení.

Tabuľka 5: Rozdíly v bodovém zisku mezi jednotlivými věkovými kategoriemi

	Mini	Benjamin	Kadet
Průměrné bodové zisky žáků ze ZPSZŽ škol	83	77	53
Průměrné bodové zisky žáků z ostatních škol	93	91	68
Procentní rozdíl	10,8 p. b.	15,4 p. b.	22 p. b.

3.3 Výzkumná otázka č. 3: Rozdíly v jednotlivých složkách inforatického myšlení.

Pomocí Chí-kvadrát testu jsme našli statisticky významné rozdíly ve všech věkových kategoriích u všech složek inforatického myšlení. Jedinou výjimkou byla dekompozice v kategorii Mini. V kategorii Mini ale byla na dekompozici zaměřena pouze jedna úloha. Proto tento výsledek nemusí mít vypovídací hodnotu. Konkrétní výsledky jsou vyobrazeny v tabulce č. 6. Celkově ale docházelo v kategorii Mini k menším rozdílům než u ostatních dvou kategoriích, což odpovídá výsledkům z kapitoly 3.2 Pokud nebudeme rozlišovat jednotlivé věkové kategorie, pak ve všech složkách inforatického myšlení docházelo ke statisticky významným rozdílům.

Tabuľka 6: Výsledky testování hypotézy: Žáci ze ZPSZŽ škol dosahují stejných výsledků v úlohách zaměřených na jistou složku inforatického myšlení. Statisticky významné rozdíly na hladině $\alpha = 0,01$ jsou označeny hvězdičkou

	Mini	Benjamin	Kadet	Celkem
Abstrakce	0,0003*	0,0000*	0,0000*	0,0000*
Algoritmizace	0,0000*	0,0000*	0,0000*	0,0000*
Dekompozice	0,8973	0,0000*	0,0000*	0,0000*
Evaluace		0,0000*	0,0000*	0,0000*
Hledání opakujících se vzorů	0,0009*	0,0000*	0,0000*	0,0000*

V tabulce 7 jsou dále uvedeny pravděpodobnosti vyřešení úloh rozdělených podle jednotlivých složek inforatického myšlení. Podle těchto výsledků zaostávali žáci ze ZPSZŽ škol především v oblastech Abstrakce a Hledání opakujících se vzorů.

Tabuľka 7: Rozdíly v pravděpodobnosti vyřešení úloh zaměřených na jistou složku inforatického myšlení

	ZPSZŽ Školy	Ostatní školy	Rozdíl
Abstrakce	25,1 %	33,7 %	8,6 p. b.
Algoritmizace	35,7 %	42,6 %	6,9 p. b.
Dekompozice	26,3 %	32,8 %	6,5 p. b.
Evaluace	37,9 %	44,8 %	6,9 p. b.
Hledání opakujících se vzorů	34,7 %	42,9 %	8,2 p. b.

Tyto výsledky odpovídají našim předchozím zjištěním [14], kdy jsme krom dvou výše zmíněných složek ještě identifikovali jako nejvíce rozdílovou složku: Algoritmizace.

3.4 Výzkumná otázka č. 4: Rozdíly v interaktivních programovacích úlohách

Pro zodpovězení výzkumné otázky č. 4 jsme pomocí Chí-kvadrát testu sledovali, zda existují rozdíly mezi interaktivními programovacími úlohami a ostatními úlohami. Výsledky ukazují na statisticky významné rozdíly mezi ZPSZŽ školami a ostatními školami u věkových kategorií Benjamin a Kadet. V nejmladší věkové kategorii jsme tyto rozdíly nezaznamenali. Výsledky jsou vyobrazeny v tabulce č. 8.

Tabuľka 8: Výsledky testování hypotézy: Žáci ze ZPSZŽ škol dosahují stejných výsledků v interaktivních programovacích úlohách.

	Míni	Benjamin	Kadet	Celkem
p-hodnoty	0,9723	0,0025*	0,0059*	0,0001*

Stejně jako v předchozí kapitole jsme spočítali pravděpodobnosti vyřešení interaktivních programovacích úloh. Rozdíl v pravděpodobnosti vyřešení takových úloh mezi ZPSZŽ a ostatními školami činil pouze 3,8 %. V porovnání s tabulkou 7, kde byla nejnižší hodnota rozdílu 6,5 %, tak konstatujeme, že interaktivní programovací úlohy činily žákům ze ZPSZŽ škol méně obtížné než ostatní typy úloh.

Tabuľka 9: Rozdíly v pravděpodobnosti vyřešení interaktivních programovacích úloh

	ZPSZŽ Školy	Ostatní školy	Rozdíl
Interaktivní programovací aktivity	43,2 %	39,4 %	3,8 %

4 Závěr

Naš výzkum zjišťoval rozdíly v úrovni informatického myšlení mezi žáky ze škol se zvýšeným podílem sociálně znevýhodněných žáků (ZPSZŽ) a žáky ostatních škol. Výsledky byly zjišťovány na datech české verze soutěže Bobřík informatiky 2022.

Zjistili jsme, že ve všech věkových kategoriích existují statisticky významné rozdíly v informatickém myšlení. Tyto rozdíly se postupem času zvyšují. Rozevírají se tak nůžky v informatickém myšlení mezi sociálně znevýhodněnými a běžnými žáky. Pokud tento trend bude dále pokračovat, pak v dospělosti budou rozdíly v informatickém myšlení mezi sociálně znevýhodněnou částí populace a zbytkem společnosti ještě významně větší. Obdobných výsledků jsme dosáhli i v našem předchozím výzkumu [14].

Dále jsme zjistili, že ve všech složkách informatického myšlení existují statisticky významné rozdíly mezi žáky ze ZPSZŽ škol a žáky ostatních škol. Žáci ze ZPSZŽ škol nejvíce zaostávali u úloh zaměřených na abstrakci a hledání opakujících se vzorů. Další výzkumná je proto vhodně zaměřit u sociálně znevýhodněných žáků na způsoby řešení úloh zaměřených na abstrakci a hledání opakujících se vzorů.

Sledovali jsme dále rozdíly mezi žáky ze ZPSZŽ škol a žáky ostatních škol u interaktivních programovacích úloh. Zde jsme v nejmladší věkové kategorii nezaznamenali rozdíly. Ve starších věkových kategoriích jsme ale již zaznamenali statisticky významné rozdíly. Výsledky naznačují, že u interaktivních

programovacích úloh nebyl tak významný rozdíl mezi žáky ze ZPSZŽ a žáky z ostatních škol, jako byl u ostatních úloh.

Nevýhodou námi aplikovaného přístupu testování po školách. V ZPSZŽ školách byla velká část běžných žáků. Vzhledem k povaze dat, které máme k dispozici, není možné na současných datech rozlišit sociálně znevýhodněné žáky a běžné žáky na ZPSZŽ školách. Jako další výzkum je možné realizovat výzkum, který by se zaměřil přímo na digitální gramotnost sociálně znevýhodněných žáků.

POĎAKOVANIE

Děkuji doc. Vaničkovy a dr. Šimandlovi za organizaci soutěže Bobřík informatiky a poskytnutí dat z ní.

BIBLIOGRAFICKÉ ODKAZY

- FRYČ, J., MATUŠKOVÁ, Z., KATZOVÁ, P. et al. *Strategie vzdělávací politiky České republiky do roku 2030* [online]. Praha: Ministerstvo školství, mládeže a tělovýchovy, 2020 [cit. 2023-01-12]. ISBN 978-80-87601-46-4. Dostupné z: https://www.msmt.cz/uploads/Brozura_S2030_online_CZ.pdf
- BARRETT, H., ROSE D, Ch. Perceptions of the Fourth Agricultural Revolution: What's In, What's Out, and What Consequences are Anticipated?. In *Sociologia Ruralis*. 2022, vol. 62, no. 2, p. 162-189. ISSN 0038-0199. DOI=<https://doi.org/10.1111/soru.12324>
- RIJSWIJK, K., KLERKX, L., BACCO, M. et al. Digital transformation of agriculture and rural areas: A socio-cyber-physical system framework to support responsabilisation. In *Journal of Rural Studies*. 2021, vol. 85, p. 79-90. ISSN 07430167. DOI=<https://doi.org/10.1016/j.jrurstud.2021.05.003>
- KLERKX, L., JAKKU, E., LABARTHE, P. A review of social science on digital agriculture, smart farming and agriculture 4.0: New contributions and a future research agenda. In *Wageningen Journal of Life Sciences*. 2019, vol. 90-91, no. 1, p. 1-16. ISSN 1573-5214. DOI=<https://doi.org/10.1016/j.njas.2019.100315>
- BRONSON, K. Looking through a responsible innovation lens at uneven engagements with digital farming. In *Wageningen Journal of Life Sciences*. 2019, vol. 90-91, no. 1, p. 1-16. ISSN 1573-5214. DOI=<https://doi.org/10.1016/j.njas.2019.03.001>
- LASI, H., FETTKE, P., KEMPER, H.-G., FELD T., HOFFMANN, M. Industry 4.0. In *Business & Information Systems Engineering*. 2014, vol. 6, no. 4, p. 239-242. ISSN 1867-0202. DOI=<https://doi.org/10.1007/s12599-014-0334-4>
- ZHONG, R. Y., XU, X., KLOTZ, E., NEWMAN, S. T. Intelligent Manufacturing in the Context of Industry 4.0: A Review. In *Engineering*. 2017, vol. 3, no. 5, p. 616-630. ISSN 20958099. DOI=<https://doi.org/10.1016/J.ENG.2017.05.015>
- NTIA, Falling Through the Net: Defining the Digital Divide. Washington: NTIA, 1999. Dostupné z: <https://www.ntia.doc.gov/report/1999/falling-through-net-defining-digital-divide>
- DEWAN, S., RIGGINS, F. The digital divide: Current and future research directions. In *Journal of the Association for Information Systems*. 2005, vol. 12, no. 6, p. 299-337. DOI=<https://doi.org/10.17705/1jais.00074>

- [10] SCHEERDER, A., VAN DEURSEN, A., VAN DIJK, J. Determinants of Internet skills, uses and outcomes. A systematic review of the second- and third-level digital divide. In *Telematics and Informatics*. 2017, vol. 34, no. 8, p. 1607-1624. ISSN 07365853. DOI=
<http://dx.doi.org/10.1016/j.tele.2017.07.007>
- [11] HENNESSY, T., LÄPPLE, D., MORAN, B. The Digital Divide in Farming: A Problem of Access or Engagement? In *Applied Economic Perspectives and Policy*. 2016, vol. 38, no. 3, p. 474-491. ISSN 2040-5790. DOI=
<http://dx.doi.org/10.1093/aep/ppw015>
- [12] GRAN, A. B., BOOTH, P., BUCHER, T. To be or not to be algorithm aware: a question of a new digital divide? In *Information, Communication & Society*, 2021, vol. 24, no. 12, p. 1779-1796. ISSN 1369-118X. DOI=
<https://doi.org/10.1080/1369118X.2020.1736124>
- [13] LUTZ, Ch. Digital inequalities in the age of artificial intelligence and big data. In *Human Behavior and Emerging Technologies*. 2019, vol. 1, no. 2, p. 141-148. ISSN 2578-1863. DOI= <https://doi.org/10.1002/hbe2.140>
- [14] Dobiáš, V., Šimandl, V. Socially disadvantaged pupils and computational thinking: Is there a new form of digital divide? In *INTED2022 Proceedings*. Valencia: IATED, 2022, s. 6542-6551. ISBN 978-84-09-37758-9, ISSN: 2340-1079. DOI= <https://doi.org/10.21125/inted.2022.1659>
- [15] WING, J. M. *Computational thinking—What and why*. 2010. p. 1-6. Dostupné z:
<https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>
- [16] LESSNER, D. Analýza významu pojmu Computational thinking. In *Journal of Technology and Information*. 2014, vol. 6, no. 1, p. 71-88. ISSN 1803537X. DOI=
<https://doi.org/10.5507/jtie.2014.006>
- [17] SELBY, C., WOOLLARD, J. *Refining an understanding of computational thinking*. University of Southampton, 2014. Dostupné z: <https://eprints.soton.ac.uk/372410/>
- [18] WEINTROP, D., BEHESHTI, E., HORN, M., ORTON, K., JONA, K., TROUILLE L., WILENSKY, U. Defining Computational Thinking for Mathematics and Science Classrooms. In *Journal of Science Education and Technology*, 2016, vol. 25, no. 1, p. 127-147. ISSN 1059-0145. DOI= <https://doi.org/10.1007/s10956-015-9581-5>
- [19] WANG, Ch., SHEN, J., CHAO, J. Integrating Computational Thinking in STEM Education: A Literature Review. In *International Journal of Science and Mathematics Education*, 2021. ISSN 1571-0068. DOI=
<https://doi.org/10.1007/s10763-021-10227-5>
- [20] ISRAEL-FISHELSON, R., HERSHKOVITZ, A. Studying interrelations of computational thinking and creativity: A scoping review (2011–2020). In *COMPUTERS & EDUCATION*. 2022, vol. 176, p. 1-22. ISSN 03601315. DOI= <https://doi.org/10.1016/j.compedu.2021.104353>
- [21] VANÍČEK, J., ŠIMANDL, V., DOBIÁŠ, V. Bebras Tasks Based on Assembling Programming Code. In: *Informatics in Schools. A Step Beyond Digital Education. ISSEP 2022. Lecture Notes in Computer Science*, Bollin, A., Futschek, G. (eds). Videň, vol 13488. Cham: Springer, 2022, p. 113-124. DOI= https://doi.org/10.1007/978-3-031-15851-3_10

Kurz Data, informace a modelování pro učitele informatiky na gymnáziu

Data, information and modelling course for computer science teachers at grammar school

Jindra Drábková
KGD FP TUL
Univerzitní náměstí 1410
461 17 Liberec
Česká republika
jindra.drabkova@tul.cz

ABSTRACT

From 2025, the new informatics should be taught at grammar schools according to the new framework educational program. The program is focused on informatics, not information and communication technology, as is the case so far. Currently, the teaching of informatics focuses primarily on application topics, thus on the use of text editors, graphic editors, and spreadsheets. The student has to acquire these digital skills in other subjects of secondary school studies.

The new programs will primarily focus on acquiring competence in four areas, namely algorithmization and programming, digital technology, information systems and data, information, modelling. The article describes a presentation created for a course in the area of Data, information, modelling. Furthermore, the course participants evaluation is summarized here, on the basis of which the original presentation was modified into its current form.

Keywords

framework educational program, informatics, data, information, modelling

ABSTRAKT

Od roku 2025 by se měla vyučovat nová informatika na gymnáziích podle nového rámcově vzdělávacího programu. Ten je zaměřen na informatiku, nikoli informační a komunikační technologie, jak je tomu dosud. V současné době se totiž výuka informatiky zaměřuje především na témata aplikační, tedy na použití textového editoru, grafického editoru, tabulek. Tyto digitální schopnosti má student získat v jiných předmětech středoškolského studia.

Nové programy se budou zaměřovat především na získání kompetence ve čtyřech oblastech, a to algoritmicizace a programování, digitální technologie, informační systémy a data, informace, modelování.

V článku je popsána prezentace, která byla vytvořena pro kurz z oblasti Data, informace, modelování. Dále je zde shrnuto zhodnocení účastníků kurzu, na základě kterého byla původní prezentace upravena do stávající podoby.

Klíčová slova

rámcově vzdělávací program, informatika, data, informace, modelování

1 ÚVOD

Vzhledem k zásadní změně pojetí výuky informatiky na českých gymnáziích [6] je nezbytné zasáhnout co nejširší skupinu učitelů

informatiky, aby bylo nové pojetí efektivně nasazeno do výuky v plánovaném termínu, tj. od září 2025.

Z iniciativy Národního pedagogického institutu vzniklo několik kurzů, které byly ověřovány a na základě zpětné vazby účastníků byly upraveny. Kurzy byly určeny pro učitele gymnázií. Témata kurzů byla vybrána podle oblastí nového rámcově vzdělávacího programu (RVP). Vzdělávací obsah nového RVP je rozdělen do čtyř oblastí:

- Data, informace a modelování
- Algoritmizace a programování
- Informační systémy
- Digitální technologie

Kurzů vzniklo více, a to na téma data, informace, modelování, informační systémy, umělá inteligence, programování v jazyce Python, robotika s BBC Micro:bit, robotika s Arduino, algoritmicizace a robotika.

Od června 2022 probíhají odborné kurzy pro učitele informatiky na gymnáziích, většinou jsou čtyřhodinové a každý kurz byl uskutečněn minimálně jednou, většina dvakrát a víckrát. Kde a kdy proběhne jaký seminář, koordinuje Národní pedagogický institut, na jejichž webových stránkách je možné se na kurzy přihlásit.

2 KURZ DATA, INFORMACE, MODELOVÁNÍ

Na naší katedře jsme vytvořili kurz pro učitele gymnázií pro oblast Data, informace a modelování, který by měl pomoci se zorientovat v této vzdělávací oblasti předmětu Informatika v Rámcovém vzdělávacím programu [6]. Měl by učitele inspirovat a poskytnout jim tipy na konkrétní příklady, kterými výuku obohatí a usnadní studentům pochopení látky a ověření získání požadovaných znalostí.

První běh kurzů měl za cíl materiály ověřit a upravit na základě zpětné vazby od účastníků. Po absolvování každého kurzu účastníci vyplňovali dotazník, který byl vytvořen pracovníky Národního pedagogického ústavu. Tvůrci kurzů měli odpovědi jednotlivých účastníků k dispozici. V kapitole Hodnocení kurzu jsou uvedeny některé z odpovědí účastníků kurzu Data, informace a modelování. Na základě připomínek účastníků prvního běhu došlo k úpravě prezentace, a to zejména v přesunu a rozšíření některých jejích částí. Zde představíme již upravenou prezentaci.

Pilotního ověřování se zúčastnilo 12 frekventantů, většinou učitelů informatiky na gymnáziích.

2.1 Osnova vytvořené prezentace

Podle RVP je vzdělávací obsah oblasti Data, informace a modelování rozdělen na čtyři části (v RVP učivo). První část seznamuje s daty a informacemi, druhá s kódováním a šifrováním, třetí s modelováním a poslední s interpretací dat. V prezentaci jsou uvedeny jen první tři části (bez interpretace dat), a to zejména z časového důvodu.

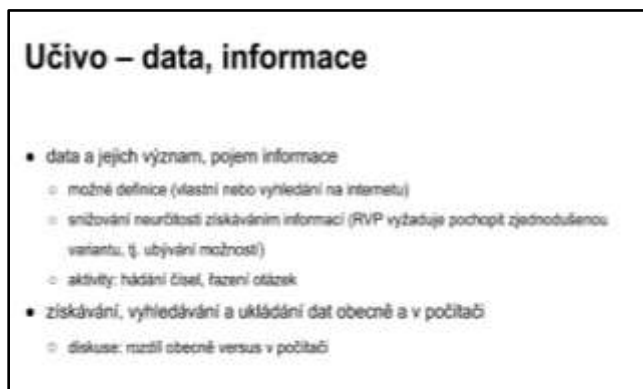
V původní prezentaci byly umístěny očekávané výstupy i učivo na úplném konci prezentace. Na tyto důležité informace bylo v rámci závěrečné diskuse velmi málo času. Proto jsme se rozhodli k rozdělení učiva na tři části podle RVP a k přesunu na začátek každé oblasti. K jednotlivým položkám v učivu byly přidány odkazy na různé příklady, na kterých účastníci lépe pochopí podstatu učiva, a aktivity, z nichž se některé s účastníky dělali v rámci kurzu.

Ke každé oblasti je za učivem z RVP úvod do tématu a pak několik aktivit, které si učitelé na školení vyzkoušeli a o kterých se následně diskutovalo. U všech témat jsou uvedeny odkazy a další příklady, které je možné se studenty k danému tématu dělat. Je dobré, když některé příklady řeší studenti ve dvojicích, aby spolu museli o problematice diskutovat.

Na závěr celého semináře proběhla diskuse.

2.2 Data a informace

Materiál seznamuje na začátku s učivem z RVP týkajícím se části data, informace. V prezentaci je ke každému bodu z učiva v RVP uvedeno vysvětlení pojmů, případně příklady, z nichž je možné učivo lépe pochopit (viz obrázek 1).



Obrázek 1: Snímek prezentace s učivem oblasti data, informace

Cílem by mělo být zamyslet se a pochopit rozdíly mezi pojmy data, informace, případně komunikace, a pochopit jak získáváním informací dojde ke snížení neurčitosti.

Výsledkem diskuse nad tím, co jsou to data, informace a co je komunikace, by mělo být uvědomění si rozdílu mezi jednotlivými pojmy. Odpovědi je možné najít na internetu a vzájemně diskutovat o rozdílech, příkladech apod. Stejným způsobem by učitelé měli postupovat při výuce se svými žáky.

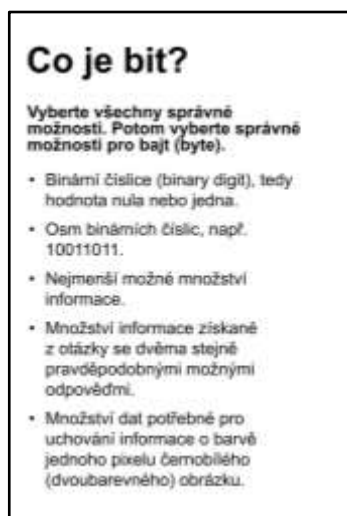
Dále jsou v kurzu tipy na příklady. První příklad je hádání jakéhokoli přirozeného čísla. Na začátku by si učitelé měli uvědomit, že je nutné stanovit interval, ze kterého budou studenti

hádat. Tuto aktivitu je vhodné dělat ve dvojicích a následně diskutovat, jaký algoritmus kdo použil pro hledání čísla. Mělo by se dojít k závěru, že neefektivnější metoda je půlení intervalu. Stejnou metodu je možné použít na jakékoli vyhledávání v seřazené posloupnosti (jak číselně, tak abecedně).

Další příklad je zaměřen na množství informace, které zaručeně poskytnou dané otázky při tom, když se snažíte doptat na zvolené číslo od 0 do 255. Otázek je v tomto příkladu sedm, například zda je to prvočíslo nebo zda je číslo trojčíferné nebo zda je to číslo 12 nebo zda je to číslo mezi 64 a 191.

Další úlohou může být, kolik otázek a odpovědí ano/ne je třeba, abychom zjistili konkrétního českého poslance, kterých je 200.

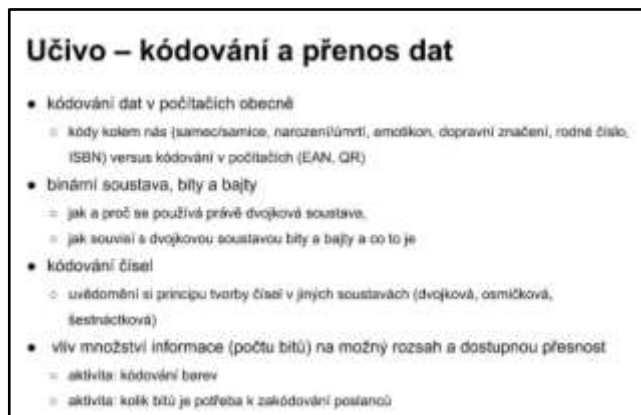
Poslední příklad v této oblasti už souvisí s kódováním. Mají se vybrat správné odpovědi na otázku, co je to bit, případně byte (viz obrázek 2).



Obrázek 2: Snímek prezentace s úlohou co je bit a byte

2.3 Kódování a přenos dat

Část kódování a přenos dat je ze všech tří nejobsáhlejší. Téma opět začíná seznamem učiva z RVP týkající se kódování a přenosu dat.



Obrázek 3: Snímek prezentace s učivem oblasti kódování a přenos dat

Podobně jako v první části je i zde uveden výčet učiva s odkazy na různé aktivity, ze kterých je možné daný pojem lépe pochopit (kódování obrazu, barevné modely, komprese textu, přenos, šifrování apod.). Na obrázku 3 je jen část učiva z oblasti Kódování a přenos dat.

Cílem je vyzkoušet si kódování a přenos dat, případně i šifrování na takových příkladech nebo aktivitách, které by vedly k pochopení principů.

Vzhledem k tomu, že tato oblast je velmi rozsáhlá, pro kurz jsme vybrali jen aktivity, které pracují s kódováním obrazu, textu, barevným modelem RGB, kompresí textu, přenosem dat a šifrováním.

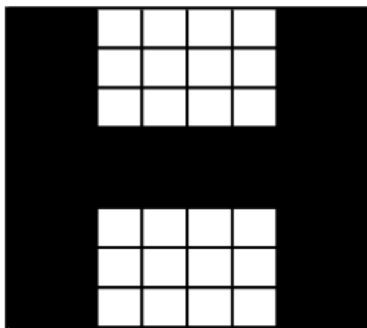
Je vhodné vysvětlit, z jakého důvodu je v RVP binární (dvojková) soustava a pojmy bity a bajty a jak bity a bajty souvisí s dvojkovou soustavou. Není podstatné umět ve dvojkové soustavě počítat, ale je podstatné umět pochopit, jak dvojkové číslo vzniká a proč se používá zrovna dvojková soustava.

Do této oblasti je možné zařadit také aktivity, které seznamují s kódy kolem nás, např. ISBN, rodné číslo, EAN, QR.

Další aktivity mohou být například: Kolik bitů je potřeba k zakódování písmen anglické abecedy, dvou barev, osmi barev, 200 poslanců.

Pro aktivitu s kódováním textu účastníci kódují text, a to pomocí dvou různých znakových sad (Windows-1250, Latin-2), kódování ukazujeme na slovu TRÍŠŤ (s diakritikou), aby se ukázaly rozdíly v různých národních znakových sadách.

K ukázce kódování obrazu používáme mřížku 8×8 . Je to aktivita pro dva účastníky, kteří si navzájem kódují a dekódují obrázek, který do mřížky zakreslili. Je na účastnících, jaký způsob přenosu vyberou. V rámci kurzů, které již proběhly, se vyskytlo několik způsobů zakódování obrázků.



Obrázek 4: Zakódovaný obrázek v mřížce 8×8

Uvádíme zde některé způsoby zakódování obrázku 4, které používali účastníci kurzu.

- Označení sloupců a řádků čísly a písmeny podobně jako u šachů (A1, B1, G1, H1, A2, B2, G2, H2, A3, B3, G3, H3, A4, B4, C4, D4, E4, F4, G4, H4, A5, B5, C5, D5, E5, F5, G5, H5, A6, B6, G6, H6, A7, B7, G7, H7, A8, B8, G8, H8)
- Označení sloupců a řádků jen čísly (11, 12, 17, 18, 21, 22, 27, 28, 31, 32, 37, 38, 41, 42, 43, 44, 45, 46, 47, 48, 51, 52, 53, 54, 55, 56, 57, 58, 61, 62, 67, 68, 71, 72, 76, 77, 81, 82, 87, 88)

- Černá a bílá políčka v řádcích (1 1 0 0 0 0 1 1, 1 1 0 0 0 0 1 1, 1 1 0 0 0 0 1 1, 1 1 1 1 1 1 1 1, 1 1 1 1 1 1 1 1, 1 1 0 0 0 0 1 1, 1 1 0 0 0 0 1 1, 1 1 0 0 0 0 1 1)
- Počty černých a bílých políček v řádcích (2Č-4B-2Č, 2Č-4B-2Č, 2Č-4B-2Č, 8Č, 8Č, 2Č-4B-2Č, 2Č-4B-2Č, 2Č-4B-2Č)

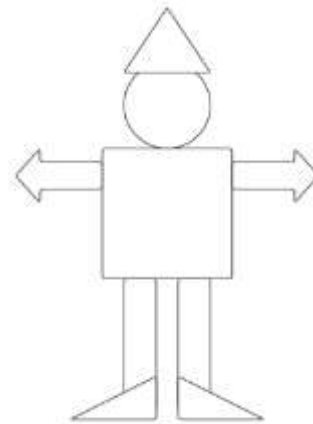
Účastníci představili způsoby zakódování a vysvětlili, co je potřeba říci před přenosem obrázku. Je dobré upozornit na to, že je potřeba např. říci, zda jsou označené písmenem sloupce nebo řádky, jaké číslo představuje jakou barvu apod. Přestože jde o velmi jednoduchou úlohu pro studenty gymnázií (jen zakódování), je dobré studentům nechat prostor pro samotný přenos dat a podrobně probrat i možnost případně komprese.

Dalším vhodným příkladem, který si účastníci kurzu mohli vyzkoušet, je kódování barev, na kterém lze ukázat, jak vypadá obrázek zakódovaný v 8bitovém nebo 24bitovém kódování RGB (www.csfieldguide.org.nz/en/interactives/colour-matcher/).

Studenti si ověřili, že použití vícebitového zakódování umožní dosáhnout přesněji požadované barvy.

Další aktivita, kterou mohou učitelé použít ve výuce a na kterou jsme účastníky kurzu upozornili, je ukázka komprese textu. Z věty se vyberou slovní spojení, která se v textu vyskytují často, a nahradí se nějakým symbolem. V tabulce se pak ukáže, z kolika procent byl text zkomprimován, je možné použít přednastavený i vlastní text (studio.code.org/s/text-compression/lessons/1/levels/2).

Další možnou aktivitou je přenos obrázku vytvořeného z geometrických tvarů druhému studentovi, který obrázek nevidí a má ho reprodukovat. První student obrázek popisuje prostřednictvím specifikace polohy jednotlivých geometrických útvarů (nad, pod, vlevo, vpravo, trojúhelník, kružnice atd.). Příklad obrázku vytvořeného z geometrických tvarů je na obrázku 5. Obrázek byl vytvořen v online aplikaci Lucidchart [5].



Obrázek 5: Příklad obrázku pro aktivitu přenos

Aktivitu, kterou jsme měli připravenou na šifrování, jsme ani v jednom z kurzů nestihli. Jde o jednoduchý příklad šifrování zprávy pomocí nějaké známé šifry, např. posouvání v abecedě, šifrování s klíčem.

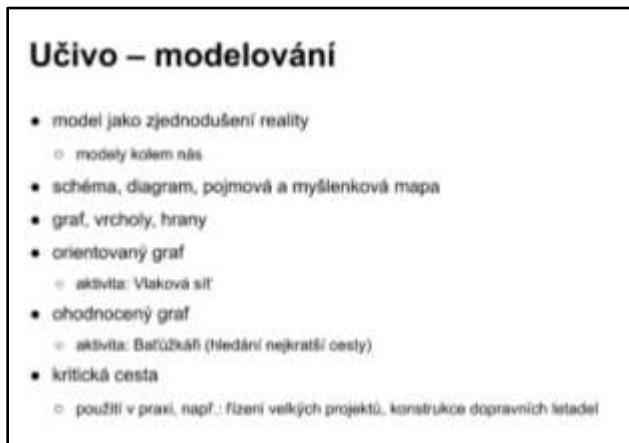
2.4 Modelování

Modelování je oblast, se kterou mají učitelé nejmenší zkušenosti a o které mají nejméně znalostí.

Třetí část opět začíná učivem z RVP s vysvětlením a ukázáním na příkladech (viz obrázek 6). I v této části jsou příklady, které učitelé nejprve sami řešili a pak o postupu řešení diskutovali. Stejným způsobem by to pak měli dělat i oni se svými studenty.

Cílem je zjistit, kde všude se modely používají a dojít k závěru, že v některých případech je nutné si model nakreslit, protože bez nakreslení některé souvislosti není možné zjistit.

Na začátku této části je vhodné si modely přiblížit na reálných příkladech, například koloběh vody v přírodě, jízdní řády, stanice metra.



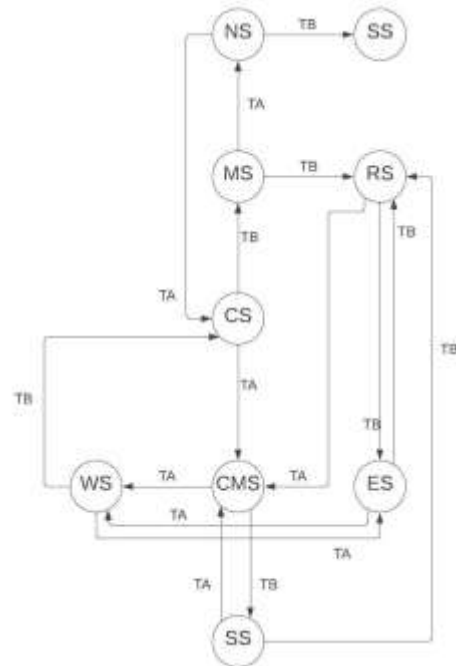
Obrázek 6: Snímek prezentace s učivem oblasti modelování

Dále je dobré probrat různé pojmy, například schéma, diagram, pojmová a myšlenková mapa, model jako zjednodušení reality. V další části se zaměříme na grafy, takže se soustředíme na pojmy s tím související, např. vrchol (uzel), hrana (spoj). Je důležité si uvědomit, že graf v informatice je něco jiného než graf v matematice nebo fyzice, tedy že slouží k modelování situací, kde záleží na vzájemných vztazích.

Pro vysvětlení pojmu orientovaný graf je možné udělat aktivitu vlaková síť. Pro vysvětlení pojmu ohodnocený graf je vhodné použít aktivitu baťůžkáři, tedy hledání nejkratší cesty. Obě úlohy jsou popsány níže.

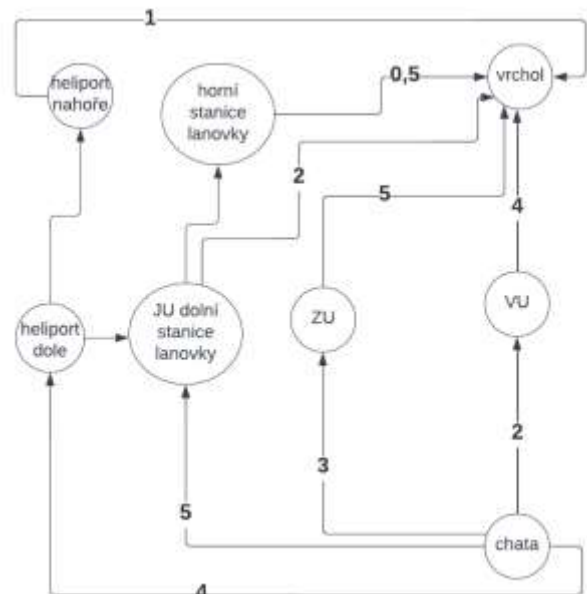
Příklad vlaková síť je on-line aktivita, kde se mezi jednotlivými stanicemi lze pohybovat vlakem A nebo vlakem B. Cílem je dostat se z počáteční do cílové stanice, vytvořit model celé vlakové sítě, případně zjistit, zda je nalezená cesta nejkratší, zda je to jediná možná cesta apod. Aktivitu je možné najít na stránkách www.csfieldguide.org.nz/en/interactives/trainsylvania/. Příklad grafu vlakové sítě je uveden na obrázku 7. Tento model byl vytvořen v prostředí Lucidchart [5].

Tato aktivita byla pro účastníky hodně složitá z důvodu velkého množství stanic (uzlů) a cest (hran). Málodko dokázal nakreslit přehledný graf.



Obrázek 7: Graf vlakové sítě

Podle mého názoru je jednodušší úloha baťůžkáři, která je převzata z učebnice Základy informatiky pro střední školy [4]. V úloze je popsána cesta z chaty na vrchol několika různými způsoby. Cílem je nakreslit graf možných cest z chaty na vrchol a nalezení nejkratší cesty. V zadání je schválně jedna informace redundantní. Je dobré na to upozornit, že ne vždy jsou všechny informace, které jsou v zadání, k řešení potřeba. To, jak může vypadat graf, je vidět na obrázku 8. Graf byl opět vytvořen v prostředí Lucidchart [5].



Obrázek 8: Graf úlohy baťůžkáři

Poslední aktivita je vlk, koza, zelí. Přes řeku je potřeba převézt vlka, kozu a zelí a na přivoz se vejde s převozníkem jen jeden objekt. Ani na břehu ani v loďce nesmí být najednou vlk a koza nebo koza a zelí. Úkolem je zorganizovat převoz všech objektů na druhý břeh tak, aby vlk nesežral kozu a koza nesežrala zelí. Zde je možné se studenty vytvořit graf se všemi možnými stavy a vyřadit ty stavy, které nemohou nastat. Pak je vhodné mezi jednotlivými stavy vytvořit přechody.

2.5 Diskuse

Na konci celého kurzu je prostor pro diskusi, který u prvního běhu nebyl dostatečný, což účastníkům chybělo. Diskuse se týkala zejména změn v nové informatice a toho, kde je možné získat informace z dalších oblastí.

3 HODNOCENÍ KURZU

Ze zpětného hodnocení účastníků, kteří se zúčastnili prvního běhu, vyplývá, že přínos kurzu hodnotí pozitivně. Účastníci kladně hodnotili množství odkazů a použité aktivity, které mnohým pomohly pochopit (hlavně prakticky) témata nového RVP.

Jeden z účastníků navrhol rozdělit kurz na dvě části (modelování zvlášť) a díky tomu se dostat do větší hloubky, zkusit více aktivit a podrobněji je rozebrat.

Někteří by přivítali více aktivit v ucelenější formě nebo celý plán některé z aktivit včetně toho, co předchází, jak se žáky pracovat potom, více doplňujících informací o typickém průběhu (typických řešeních, chybách).

Účastníci kladně hodnotili množství odkazů na webové stránky s již hotovými on-line cvičeními.

Kurz učitelům pomohl udělat si představu o nové informatice v RVP a přinesl mnohým nápady do tvorby školního vzdělávacího programu.

4 ZÁVĚR

Kurz Data, informace a modelování, který byl v článku představen, slouží k seznámení, porozumění, rozšíření jedné z oblastí informatiky. Měl by pomoci učitelům gymnázií uchopit výuku informatiky tak, aby byl kladen důraz na rozvoj inforatického myšlení žáků a porozumění principům informatiky.

Informatika se podle nového RVP má začít učit na základních školách nejpozději od 1. září 2024, takže je zřejmé, že to bude trvat ještě minimálně 4 roky, než na gymnázia nastoupí studenti, kteří už absolvovali celou informatiku na základní škole v novém pojetí. Proto je vhodné, aby učitelé gymnázií do té doby využívali materiály a aktivity z učebnic pro 2. stupeň, které vznikly v rámci projektu Inforatické myšlení [3], neboť jsou v nich jednodušší aktivity, které studenti bez předchozí přípravy zvládnou.

Je vhodné, aby byla výuka informatiky založená zejména na aktivitách. Důležitá je i následná diskuse se studenty. Je potřeba nezapomínat na to, že by diskuse měla mít nějaký závěr, i když tím závěrem může být, že není jen jedno správné řešení.

PODĚKOVÁNÍ

Děkuji účastníkům kurzu za hodnocení a Mgr. Janu Berkimu, Ph.D., a Mgr. Danielu Lessnerovi, Ph.D., za nápady, podněty a připomínky.

BIBLIOGRAFICKÉ ODKAZY

- [1] *Computer Science Field Guide* [online]. Computer Science Education Research Group at the University of Canterbury. [vid. 24. 2. 2023] Dostupné z: www.csfieldguide.org.nz/en/
- [2] DRÁBKOVÁ, J. *Data, modelování, informace* [prezentace online]. 2023. Dostupné z: <https://1url.cz/JrPKF>
- [3] *Inforatické myšlení* [online]. Jihočeská univerzita v Českých Budějovicích. © 2018. [vid. 24. 2. 2023] Dostupné z imysleni.cz.
- [4] LESSNER, D., LÁNA, M., PODRÁZKÁ TOMKOVÁ, M. a HAUT, J. *Základy informatiky pro střední školy*. 1. vyd. 2020. ISBN 978-80-7394-785-9.
- [5] *Lucid* [online]. © 2023 Lucid Software Inc. [vid. 24. 2. 2023]. Dostupné z <https://www.lucidchart.com/pages/>
- [6] *Rámcový vzdělávací program pro gymnázia: RVP G* [online]. Praha: MŠMT, 2021 [vid. 24. 2. 2023]. Dostupné z: <https://revize.edu.cz/navrh-zmen-v-rvp-pro-gymnazia>
- [7] *Text Compression* [online], © Code.org, 2023. [vid. 24. 2. 2023]. Dostupné z: <https://studio.code.org/s/text-compression/>

Transformace úloh z blokového prostředí Lego Lab EV3 do blokového prostředí Scratch

Transforming tasks from the Lego Lab EV3 block environment to the Scratch block environment

Filip Frank

Katedra výpočetní a didaktické techniky Fakulta pedagogická
Západočeská univerzita v Plzni

Klatovská tř. 51
30614

Česká republika
frankf@kv.d.zcu.cz

Tomáš Jakeš

Katedra výpočetní a didaktické techniky Fakulta pedagogická
Západočeská univerzita v Plzni

Klatovská tř. 51
30614

Česká republika
tjakes@kv.d.zcu.cz

ABSTRACT

The article deals with the problem of transforming a test set of tasks from the Lego Mindstorms EV3 environment to the Scratch environment. We determine how the modification will be done. The number of parameters to be modified and the overall complexity of the tasks became key in our case. For the reasons mentioned, the way in which the object movement occurs changed, originally it was a robot, and two tasks even had to be omitted. The transformed set of tasks will be used in the future as part of a dissertation investigating the development of the algorithmic component of computational thinking using the Scratch block programming environment. Once the tasks have been deployed, the results obtained in the tasks will be compared with the results from the original version of the test set.

Keywords

Scratch. Lego Mindstorms EV3. Test set of tasks. Computational thinking. Algorithmic thinking.

ABSTRAKT

Článek se zabývá problematikou transformace testovací sady úloh z prostředí Lego Mindstorms education EV3 Lab do prostředí Scratch. Stanovujeme, jakým způsobem bude docházet k úpravě. Klíčovým se v našem případě stal počet upravovaných parametrů a celková náročnost úloh. Ze zmíněných důvodů se změnil způsob, jakým dochází k pohybu objektu, původně se jednalo o robota, dvě úlohy bylo dokonce potřeba vypustit. Transformovaná sada úloh bude v budoucnu použita jako součást disertační práce, která se zabývá zkoumáním rozvoje algoritmičké složky informatického myšlení za pomoci blokového programovacího prostředí Scratch. Po nasazení úloh budou dosažené výsledky v úlohách porovnány s výsledky z původní verze testovací sady.

Klíčové slová

Scratch. Lego Mindstorms EV3. Testovací sada úloh. Informatické myšlení. Algoritmičké myšlení.

1 ÚVOD

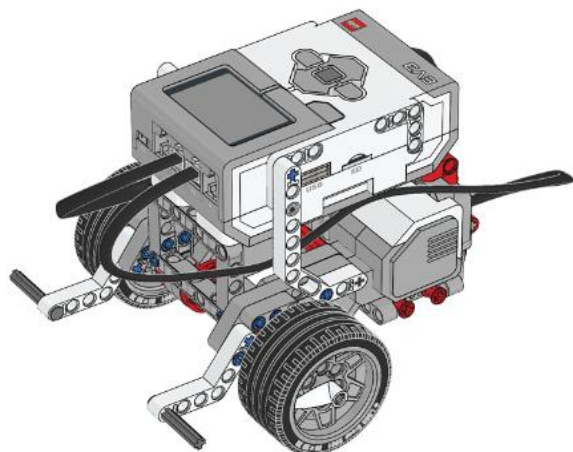
V české republice došlo k revizi rámcových vzdělávacích programů (RVP), které definují státní úroveň vzdělávání v České republice. Mezi jednu ze zásadních změn patří právě výuka informatiky, jejíž obsah se významně posunul od výuky ICT k učení opravdové informatiky, jako vědního oboru. Zároveň se objevuje pojem informatické myšlení. [1] Obsahem informatiky nyní jsou: „Data, informace a modelování, algoritmizace a

programování, informační systémy, digitální technologie [1].“

V příspěvku se zaměřujeme právě na míru rozvoje informatického myšlení, respektive na testování jedné konkrétní složky. Námí zkoumanou složkou je algoritmičká složka informatického myšlení. Pro toto testování je nutné vytvořit, nebo převzít testovací sadu úloh. Záměrem budoucího testování je zachytit míru rozvoje v době, kdy ještě není nové RVP zcela nasazeno. Školy budou muset podle nového RVP upravit své školní vzdělávací programy (ŠVP). Školní vzdělávací programy jsou vytvářeny na základě RVP samotnými školami. Školy v nich mohou zohlednit vlastní zaměření a zvolenou oblast posílit, ale vždy musí naplnit minimální požadavky RVP. Podle dokumentu „Postupné zahájení vzdělávání se ŠVP upraveným podle RVP ZV s novou vzdělávací oblastí informatika s účinností od 1. září 2021“ jsou školy povinny upravit ŠVP podle nového RVP na prvním stupni nejdříve do září roku 2023 a na druhém stupni do září roku 2024. Naskýtá se tedy poslední příležitost získat informace z doby, kdy nové RVP nemusí být na všech školách ještě nasazeno. [2] Pro otestování míry rozvoje algoritmičké složky informatického myšlení a navázání na předchozí výzkumy jsme zvolili existující sadu úloh, která se zabývala testováním rozvoje abstraktního myšlení s využitím Lego Mindstorms EV3. Úlohy jsme analyzovali a pokusili se převést je do prostředí Scratch. Při převodu se snažíme respektovat množství parametrů, které daná úloha využívá. Zároveň se snažíme zachovat jejich náročnost ve změněném prostředí. V průběhu práce se dvě úlohy ukázaly jako nepřevaditelné a dvě úlohy významně změnily svou náročnost a cíl. Z toho důvodu byly přidány další dvě úlohy, které jsou svým řešením zaměřené čistě na Scratch a jeho prostředí. Tyto poslední úlohy nelze zpětně převést při zachování obtížnosti zpět pro lego. Úlohy, které nešlo převést vhodným způsobem do opačného prostředí nebudou v budoucnu objektem porovnávání úspěšnosti žáků při jejich řešení.

2 PŮVODNÍ SADA ÚLOH

Původní sada úloh byla aplikována v disertační práci Dr. Bařka v roce 2020. Disertace měla téma „Robotická stavebnice jako prostředek pro rozvoj abstraktního myšlení žáků základní školy“. Žáci měli za úkol rozpožhybovat pojízdného robota tak, jak bylo určeno v zadání. [3] Sada úloh byla zvolena záměrně z důvodu možné spolupráce a porovnání výsledků s původním autorem.



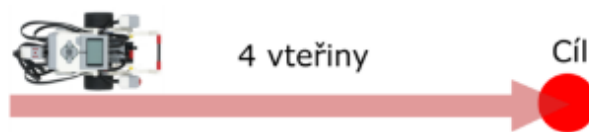
Obrázek 1 Pojízdný robot (Zdroj: Building Instructions for Robot Educator. Lego Education. [Online] 2015. [Citace: 17. únor 2023.] <https://education.lego.com/en-us/product-resources/mindstorms-ev3/downloads/building-instructions#building-core>.)

Baťko se ve své práci vyhnul práci na konstrukci robota. Což je dalším důvodem pro volbu právě této sady úloh. Žáci dostali pojízdného robota, kterého vidíme na obrázku 1. Jejich úkolem nebylo vytvářet robota, který by byl schopen trasu projíždět po konstrukční stránce. Úkolem bylo projet zadanou trasu pouze s vytvořením programu pro robota, který takový úkol dokáže splnit. Programování robota probíhalo v prostředí Lego MINDSTORMS Education EV3 Lab. [3] V dnešním kontextu je vhodné dodat, že prostředí Lab již není společností Lego plně podporováno. Stále je však možné jej ze stránek výrobce stáhnout v zastaralých produktech. [4]



Obrázek 2 Prostředí Lego MINDSTORMS EV3 (Zdroj: Vlastní)

Baťkova sada úloh se skládá z 15 úloh, jejichž náročnost se postupně stupňuje. Náročnost je možné stanovit zejména podle počtu parametrů, které je potřeba upravit. Zároveň můžeme vypočítat, jaké množství bloků bylo potřeba pro realizaci řešení úlohy. Baťkovi úlohy se opírají převážně o pojezd robota. Žákům je zadán tvar, který musí robot objet. První úloha se zabývá zdánlivě jednoduchým popojetím vpřed po dobu 4 vteřin. [3]



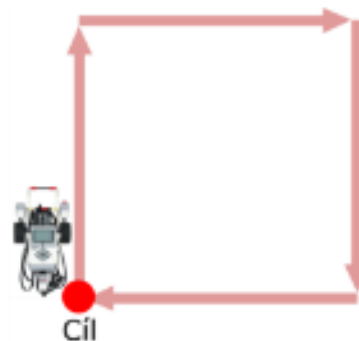
Obrázek 3 První Baťkova úloha (Zdroj: Baťko, Jan. Robotická stavebnice jako prostředek pro rozvoj abstraktního myšlení žáků základní školy. Plzeň: Západočeská univerzita v Plzni, 2020.)

Pro vyřešení této úlohy musí žáci zvolit správný blok, který bude obsahovat dva porty pro motory. Tyto dva parametry je nutné dále nastavit podle toho, kam jsou motory připojeny. Na výběr mají žáci ze 4 možností (A, B, C, D). Dále je potřeba přenastavit funkci bloku na časové nastavení. Ve výchozím nastavení mají motorové bloky nastavení na počet otáček motoru. Na závěr je potřeba nastavit bloku aktivitu po dobu 4 sekund. Výsledné řešení může vypadat, jako na obrázku 4.



Obrázek 4 Možné řešení první úlohy (Zdroj: Vlastní)

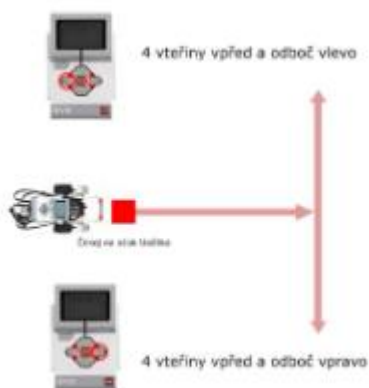
Další úlohy pak dále pracují s jízdou vpřed až do páté úlohy. V mezích žáci mění rychlost robota, případně přidávají blok pro pauzu. V páté úloze je zařazena zatáčka o 90°. Robot nejprve jede vpřed 4 vteřiny, zatočí a jede další 4 vteřiny vpřed. Žáci si zde budou muset uvědomit, že je potřeba zabrat buď pouze jedním motorem, nebo oběma, ale v opačném směru. Obě řešení jsou přípustná. Úloha připravuje základní pochopení zatočení pro úlohu číslo šest. V úloze číslo šest musí žáci zajistit, aby robot jízdou vytvořil čtverec. Při prvním pokusu žáci vytvoří čtverec libovolným způsobem. V následující úloze jsou pobídnuti k tomu, zda by bylo možné použít blok pro opakování při vytváření čtverce. [3]



Obrázek 5 Pátá a šestá Baťkova úloha (Zdroj: Baťko, Jan. Robotická stavebnice jako prostředek pro rozvoj abstraktního myšlení žáků základní školy. Plzeň: Západočeská univerzita v Plzni, 2020.)

Stejný přístup se opakuje až do deváté úlohy. Nutno zdůraznit, že do této chvíle si žáci vystačili se zatočením pouze do jednoho směru. Desátou úlohou se toto mění a žáci musí jednou zatočit doleva a jednou doprava. Až do jedenácté úlohy si však stále vystačí se zatočením o 90°. Dvanáctá a třináctá úloha pak počítá s využitím zatočení o ostrý úhel. Konkrétně třináctá pak pracuje se zatočením o 60° a tím vytvoření rovnostranného trojúhelníku. [3]

Poslední dvě úlohy pak zcela mění používané strategie. Program se sice stále spouští z počítače, ale nově by měl obsahovat nekonečný cyklus. Do nekonečného cyklu pak žáci umístí podmínku, která reaguje na tlačítka řídicí kostce. Pokud tedy žák stiskne tlačítko vpřed, pojedou robot dopředu. Pokud stiskne vzad, pojedou robot dozadu. Podobným způsobem funguje i poslední úloha patnáct. Zde žáci při stisku tlačítka zařídí, aby robot jel 4 vteřiny dopředu a následně, podle strany stisknutého tlačítka, zatočil.



Obrázek 6 Patnáctá úloha (Zdroj: Bařko, Jan. Robotická stavebnice jako prostředek pro rozvoj abstraktního myšlení žáků základní školy. Plzeň: Západočeská univerzita v Plzni, 2020.)

3 MODIFIKOVANÁ SADA ÚLOH PRO PROSTŘEDÍ SCRATCH

V následujících podkapitolách popisujeme způsob a podmínky za jakých byly úlohy převedeny. V některých případech nebylo možné úlohy převést při zachování náročnosti. Toto je řešeno buď vynecháním úlohy, nebo jejím nahrazením úlohou, která sleduje stejný cíl při použití technik, které jsou ve Scratchi přirozenější.

3.1 Řešené problémy

Při modifikaci úloh jsme narazili na několik problémů. Základní otázkou bylo, jaký použít objekt pro pohyb, aby dával smysl. Pohybujícím objektem se tedy stal autobus a místem pro zastavení se staly různě vzdálené a orientované autobusové zastávky.

Prvním problémem se stal fakt, že Bařko počítá ve své sadě úloh s tím, že se robot pohybuje danou dobu. Ve Scratchi je sice možné nastavit, aby objekt klouzal po stanovenou dobu. Při použití zmíněného bloku však musí žák nastavovat nejen čas, po jaký se bude objekt pohybovat, ale i souřadnice X i Y. Tím by došlo k významnému nárůstu obtížnosti. Z toho důvodu v modifikovaných úlohách používáme blok „Dopředu o N kroků“. Tento blok sice neanimuje pohyb objektu vpřed, ale chová se srovnatelně, jako když žáci nastavují, aby jel robot vpřed po dobu 4 sekund. U tohoto bloku nastaví, o kolik skočí vpřed bez animace.

Druhým zásadním problémem se stala snaha zachovat krom náročnosti i myšlenku úloh. Tím, že objekt po ploše skáče se ztratila

možnost sledovat dráhu, po které se objekt pohyboval. Tento problém jsme vyřešili umístěním a spuštěním bloku „pero“. Požadovaný objekt tedy žáci dostanou na požadované místo a objekt zároveň zanechá stopu. Tuto stopu žáci nenastavují, nezapínají, ani nevypínají. Je to vlastnost, kterou jsme připravili my.

Po stanovení bloku pro pohyb a zanechávání stopy bylo potřeba vyřešit poslední problém, který spočívá v umístění autobusu na původní pozici. V případě výchozí sady úloh stačilo fyzicky vzít robota a toho umístit tam, kam žáci potřebovali. Typicky šlo jen o to, aby měl robot dostatek prostoru. V našem případě se jedná o virtuální autobus v počítači. Bylo by sice možné, aby si žáci autobus vrátili zhruba do výchozí pozice myši, tím by si však mohli úkol udělat těžší, protože by jej mohli umístit příliš nízko a autobus by pouhým pohybem vpřed už netrefil zatáčku. Zároveň se částečně vymstilo využití bloku pera pro zanechání stopy. Kdyby žáci vraceli autobus na své místo přetažením myši, stopa by zůstávala na ploše. Ze zmíněných důvodů se na obrazovce krom zastávek a autobusu nachází také velké červené tlačítko reset. Stiskem tlačítka reset smažou žáci zanechanou stopu a vrátí vše do výchozího stavu.

Poslední otázkou zůstalo, jakým způsobem budou žáci spouštět program. Rozhodli jsme se, že žáci budou pro finální spuštění programu používat klasickou zelenou vlaječku, která ve Scratchi slouží pro spuštění programu, který je napojen na blok „Po kliknutí na zelenou vlajku“. Zde se nabízí otázka, proč jsme vytvářeli tlačítko reset, když bylo možné vše mazat a uvádět do výchozího stavu stiskem vlaječky. Tento přístup jsme zvolili proto, abychom žákům dali možnost experimentovat a měli lepší představu o tom, jaká vzdálenost jejich autobus čeká. Žáci takto mohou klikat na jednotlivé bloky a testovat. Mohou N-krát stisknout blok „Dopředu o 10 kroků“ a počítat počet kliků. Následně jen nastaví celkový počet a úloha je vyřešena. Díky tlačítku reset si mohou nezávisle na spuštění programu uvést vše do výchozího stavu.

3.2 Vybrané modifikované úlohy

Následující podkapitoly popisují vybrané úlohy, které byly převedeny z původního prostředí Lego Mindstorms education Lab EV3 do prostředí Scratch. Z důvodu rozsahu článku nebylo možné uvést všechny převedené úlohy. Úlohy byly zvoleny tak, aby ilustrovaly zvyšující se náročnost.

3.2.1 Modifikovaná první úloha

V prostředí Lego byl použit plynulý pojezd robota, kdy robot urazí vzdálenost 4 vteřin a následně zastaví. Vzdálenost ve Scratchi se touto cestou měřit nedá. Žáci budou muset tedy zařídít, aby se s použitím vhodného bloku dostal autobus do zastávky.



Obrázek 7 První modifikovaná úloha (Zdroj: Vlastní)

3.2.2 Modifikovaná čtvrtá úloha

Čtvrtá úloha se v originální pojetí věnuje pojezdu robota 5 vteřin 30% rychlostí, následně robot zastaví na 3 vteřiny a bude pokračovat další 3 vteřiny 50% rychlostí. [3]

I za předpokladu, že bychom použili blok pro klouzání, stále není možné ve Scratchi nastavit rychlost klouzání. Z toho důvodu je úloha převedena na pohyb mezi dvěma zastávkami, které jsou různě daleko. Autobus dojede do první zastávky, kde počká 3 vteřiny. Pak skočí do další zastávky.

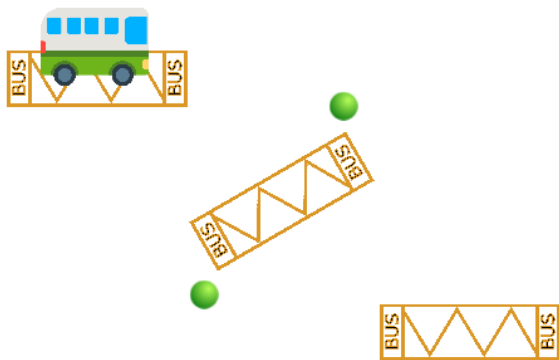
RESET



Obrázek 8 Modifikovaná čtvrtá úloha (Zdroj: Vlastní)

3.2.3 Modifikovaná dvanáctá úloha

V původní verzi dvanácté úlohy měli žáci za úkol projet dráhu ve tvaru písmene Z. V tom případě bylo zapotřebí zatočit v ostrém úhlu postupně na obě strany. Při převodu této úlohy jsme narazili na problém, kdy je zapotřebí projet autobusem ještě o kus dál, než je zastávka. Abychom žákům tento problém přiblížili, použili jsme na trase zelené kužely, které žáci musí objet. Kužely jim zároveň pomohou s odhadem vzdálenosti, která je zapotřebí pro správné najetí do „šikmé“ zastávky.



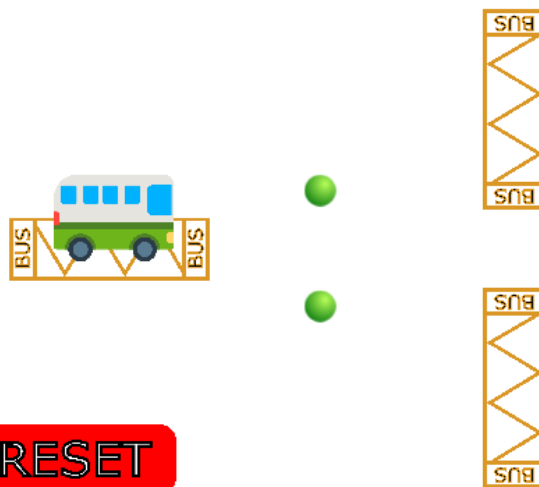
RESET

Obrázek 9 Modifikovaná dvanáctá úloha (Zdroj: Vlastní)

3.3 Vybrané úlohy s problematickou modifikací

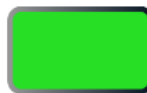
Problémově převeditelnými úlohami se ukázaly úlohy čtrnáct a patnáct. Žáci zde potřebovali využít nekonečný cyklus, který reaguje na stisk tlačítka na kostce. Ve Scratchi nedává smysl používat stejný přístup, navíc by se úloha stala velmi náročnou. Místo toho lze pohodlně použít blok z kategorie událostí. Spouštěcím blokem se tak stane libovolná klávesa na klávesnici.

Na její popud se provede požadovaná akce. Tím je úloha velmi zjednodušená.



Obrázek 10 Modifikovaná čtrnáctá úloha (Zdroj: Vlastní)

Aby žáci prošli požadovanými vlastnostmi úlohy, byly přidány dvě zcela nové úlohy. Tyto úlohy pracují s vnímáním autobusu. Autobus stojí na červené značce. Červená značka se v náhodný čas změni v zelenou. V tu chvíli by autobus měl vyrazit a dojet do zastávky. Žáci zde budou muset využít cyklus, který skončí ve chvíli, kdy se autobus dotkne zelené barvy.



RESET

Obrázek 11 První přidaná úloha (Zdroj: Vlastní)

Druhou přidanou úlohou je zadání, kdy žáci nastaví autobus tak, aby se pohyboval vpřed a zastavil ve chvíli, kdy se dotkne černé čáry. Žáci zde použijí cyklus, který bude autobusem pohybovat vpřed o 1 krok do té doby, dokud se autobus nedotkne černé čáry.

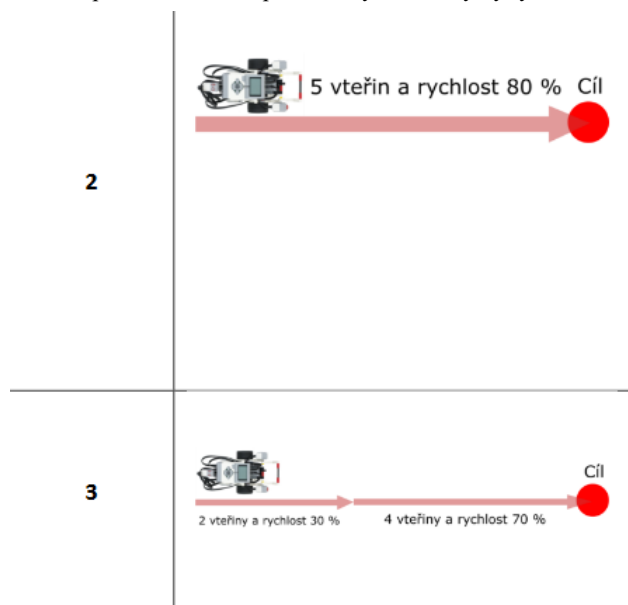


RESET

Obrázek 12 Druhá přidaná úloha (Zdroj: Vlastní)

3.4 Úlohy, které nelze převést

Úlohy, které nelze převést jsou takové, které není možné při stejné obtížnosti provést v prostředí Lego i v prostředí Scratch. Je paradoxní, že takovými úlohami se staly právě úlohy číslo dvě a tři. Důvodem je nemožnost ovládat rychlost pohybu objektu ve Scratchi, pokud nebudeme používat cyklus, který by rychlost řídil.



Obrázek 13 Úlohy dvě a tři (Zdroj: Bařko, Jan. *Robotická stavebnice jako prostředek pro rozvoj abstraktního myšlení žáků základní školy*. Plzeň: Západočeská univerzita v Plzni, 2020.)

To že jsme úlohy nebyli schopni převést při zachování obtížnosti neznamená, že efektu nelze dosáhnout. Rychlost můžeme ovládat zmíněným cyklem, nebo počtem kroků. V případě, že bychom do cyklu dali blok „skoč dopředu o 1 krok“, stalo by se určujícím pro vzdálenost počet iterací. Pokud bychom chtěli pohyb zpomalit, stačilo by pracovat s „délkou“ kroku. Taková práce s cyklem je však mnohem náročnější, než nastavit v bloku Lego čas ve vteřinách a rychlost v procentech.

4 ZÁVĚR

V roce 2023 je jedna z posledních příležitostí zaznamenat, jaký byl stav na školách před zavedením inovovaného RVP. Bez těchto informací nebude možné v budoucnu porovnávat, jak se revize RVP skutečně projevila. Pro získání takových dat jsme vytvořili sadu úloh, která původně vychází z dnes již nepodporované stavebnice. Transformovaná sada úloh je postavena na Scratchi. Převod jednotlivých úloh předcházela analýza každé úlohy v prostředí Lego. Byly identifikovány počty parametrů, bloků a celková náročnost úloh. Na základě této analýzy byly úlohy transformovány do prostředí Scratch. Zásadním problémem, který bylo potřeba při transformaci vyřešit byla absence plynulého posunu objektu ve Scratchi. Plynulý pohyb by ve Scratchi představoval výrazné zvýšení náročnosti. Proto bylo přistoupeno k využití bloku „Dopředu o N kroků“. Zmíněný blok sice s objektem skočí, místo toho, aby s ním plynule pohyboval, umožňuje však zachování počtu měněných parametrů a tím náročnosti úkolu. Aby byl zachován požadavek na projetí konkrétní trasy a úlohy dávaly smysl, žáci programují autobus, který projíždí zastávkami, ve kterých zastavuje. Autobus zároveň zanechává stopu, aby bylo možné zkontrolovat konkrétní trasu. Některé úlohy nebylo možné převést vůbec, a to zejména z důvodu plynulé jízdy. Závěrem byly přidané dvě úlohy, které se v původní sadě nevyskytovaly. Bylo to proto, že dvě poslední úlohy v původní sadě ztratily při převodu na náročnosti a jejich smyslu. Proto byly vytvořeny dvě zcela nové úlohy, které pracují s cykly a podmínkami, což bylo náplní původních dvou posledních úloh. Přestože tedy nebylo možné převést poslední dvě úlohy při zachování náročnosti, jejich obsah bylo možné převést. Proto vznikly dvě nové úlohy na závěr, které mají jiný obsah, ale stejný princip řešení, jako měly původní. Úlohy zatím nasazeny nebyly, ale budou použity při experimentálním ověření míry rozvoje algoritmické složky informatického myšlení v rámci disertace.

BIBLIOGRAFICKÉ ODKAZY

- [1] Rámcový vzdělávací program pro základní vzdělávání. [Online] Leden 2021. [Citace: 16. Únor 2023.] <https://www.edu.cz/wp-content/uploads/2021/07/RVP-ZV-2021-zmeny.pdf>.
- [2] Postupné zahájení vzdělávání se ŠVP upraveným podle RVP ZV s novou vzdělávací. [Online] Leden 2021. [Citace: 16. Únor 2023.] <https://revize.edu.cz/files/nabeh-rvpzv-2021-informatika.pdf>.
- [3] Bařko, Jan. *Robotická stavebnice jako prostředek pro rozvoj abstraktního myšlení žáků základní školy*. Plzeň : Západočeská univerzita v Plzni, 2020.
- [4] Lego. Retired products. *Lego*. [Online] Lego, 2023. [Citace: 17. únor 2023.] <https://education.lego.com/en-us/downloads/retiredproducts>.
- [5] Building Instructions for Robot Educator. *Lego Education*. [Online] 2015. [Citace: 17. Únor 2023.] <https://education.lego.com/en-us/product-resources/mindstorms-ev3/downloads/building-instructions#building-core>.

Kompetencie učiteľov informatiky v oblasti programovania

Competences of informatics teachers in the field of programming

Ján Guniš
Prírodovedecká fakulta UPJŠ
Jesenná 5
041 54 Košice
Slovenská republika
jan.gunis@upjs.sk

Lubomír Šnajder
Prírodovedecká fakulta UPJŠ
Jesenná 5
041 54 Košice
Slovenská republika
lubomir.snajder@upjs.sk

Csaba Török
Prírodovedecká fakulta UPJŠ
Jesenná 5
041 54 Košice
Slovenská republika
csaba.torok@upjs.sk

Lubomír Antoni
Prírodovedecká fakulta UPJŠ
Jesenná 5
041 54 Košice
Slovenská republika
lubomir.antoni@upjs.sk

ABSTRACT

Teaching programming is an important part of teaching informatics in our elementary and secondary schools. What experience do informatics teachers have with programming? Which programming languages do they teach? Which topics do they teach? Which topics do they know and which ones do not? In this article, we present the answers to these questions obtained from a survey involving 2865 out of 9184 informatics teachers from 1456 of a total 3531 elementary and secondary schools. We conclude the article with suggestions for improving this situation.

Keywords

Informatics teachers. Competencies. Programming.

ABSTRAKT

Vyučovanie programovania je významnou súčasťou vyučovania informatiky na našich základných a stredných školách. Aké majú skúsenosti s programovaním učitelia informatiky? Ktoré programovacie jazyky vyučujú? Ktoré témy vyučujú? Ktoré témy ovládajú, ktoré nie? V tomto článku prinášame odpovede na tieto otázky získané z prieskumu, do ktorého sa zapojilo 2865 z celkového počtu 9184 učiteľov informatiky z 1456 z celkového počtu 3531 základných a stredných škôl. V závere článku uvádzame návrhy na zlepšenie tohto stavu.

Kľúčové slová

Učitelia informatiky. Kompetencie. Programovanie.

1 ÚVOD

Oblasť Algoritmické riešenie problémov (ako súčasť predmetu informatika) sa na Slovensku podľa Inovovaného štátneho vzdelávacieho programu (ďalej IŠVP) vyučuje od 3. ročníka základnej školy [1]. Táto oblasť je súčasťou všeobecného vzdelávania aj na druhom stupni základnej školy a na gymnáziách. Podľa našich skúseností a na základe rozhovorov s učiteľmi informatiky základných a stredných škôl [2] konštatujeme, že napriek proklamovaným vzdelávacím cieľom pre jednotlivé vzdelávacie stupne, sa pri výučbe programovania na následných stupňoch začína prakticky od začiatku, na čo poukazujú aj iní autori

[3]. Rovnaký poznatok máme aj z úvodných kurzov programovania na slovenských vysokých školách [4, 5, 6], ktoré nepredpokladajú nejaké poznatky študentov z tejto oblasti.

Tieto zistenia sú znepokojujúce aj preto, lebo aktuálne požiadavky na kompetencie a zručnosti pre úspešné uplatnenie sa človeka v spoločnosti predpokladajú aspoň základné znalosti z oblasti riešenia problémov a programovania [7, 8, 9]. Informatické myslenie ako komplexná schopnosť riešiť problémy aj pomocou digitálnych technológií sa najprirodzenejšie a najkomplexnejšie rozvíja prostredníctvom programovania, ktoré umožňuje vytvárať nielen artefakty, ale aj vlastné nástroje pre skúmanie, modelovanie a riešenie problémov [7].

Vzdelávacie ciele, ktoré žiaci na jednotlivých stupňoch škôl reálne dosahujú, sú vo veľkej miere závislé na ich učiteľoch. Rozhodli sme sa preto zistiť, kto vyučuje informatiku na základných a stredných školách, aké je vzdelanie a aké sú kompetencie učiteľov informatiky. Zrealizovali sme prieskum medzi učiteľmi informatiky a niektoré naše zistenia uvádzame v nasledujúcom texte.

2 PRIESKUM KOMPETENCIÍ UČITEĽOV INFORMATIKY V OBLASTI PROGRAMOVANIA

V rámci Národného projektu IT Akadémia – vzdelávanie pre 21. storočie a projektu KEGA 012UPJŠ-4/2021 [10] sme realizovali prieskum medzi učiteľmi informatiky v oblasti ich programátorských kompetencií a postojov.

2.1 Výskumné otázky

Cieľom prieskumu bolo získať odpovede na nasledovné výskumné otázky:

1. Aká je kvalifikačná štruktúra učiteľov informatiky?
2. Akú dlhú prax majú učitelia informatiky vo vyučovaní programovania?
3. Ktoré témy z programovania vyučujú učitelia informatiky?
4. Ktoré programovacie jazyky a koľko jazykov vyučujú učitelia informatiky?

5. Aké sú postoje učiteľov informatiky k programovaniu (náročnosť, zaujímavosť, dôležitosť)?
6. V ktorých oblastiach informatiky a vyučovania informatiky sa chcú zdokonaľiť učители informatiky?

2.2 Realizácia prieskumu

Primárne sme e-mailom oslovili riaditeľov všetkých slovenských základných škôl (ďalej ZŠ), gymnázií (ďalej G) a stredných odborných škôl okrem umeleckých škôl a konzervatórií (ďalej SOŠ) [11] so žiadosťou o distribúciu dotazníka učiteľom informatiky (bez ohľadu na ich formálnu kvalifikáciu).

Samotný zber dát sme realizovali v termíne 8. 2. 2022 – 27. 2. 2022 formou elektronických dotazníkov. Zloženie výskumnej vzorky uvádzame v Tabuľka 1.

Tabuľka 1 Zloženie výskumnej vzorky

	ZŠ	SOŠ	G	spolu
počet respondentov	1943	628	294	2865
počet škôl	1037	273	146	1456

2.3 Štatistická analýza

Štatistické analýzy boli vykonané v programovacom prostredí R verzia 4.2.1. Na analýzu asociácie riadkových a stĺpcových premenných tabuliek sme použili buď Chí-kvadrát test nezávislosti (X2TI) alebo Fisherov test (FT). Chí-kvadrát test dobrej zhody (X2GFT) sa použil na analýzu štatistických rozdielov. P-hodnoty menšie ako 0,05 sa považovali za štatisticky významné.

2.4 Výsledky prieskumu

V tomto článku uvádzame vybrané výsledky týkajúce sa prvých 4 výskumných otázok.

2.4.1 Kvalifikačná štruktúra učiteľov informatiky

Kvalitu učiteľa informatiky primárne určuje jeho formálna kvalifikácia a dĺžka jeho pedagogickej praxe.

Učiteľom sme v tejto súvislosti položili dve otázky:

1. Uveďte, akým spôsobom ste získali kvalifikáciu na vyučovanie informatiky.
2. Koľko rokov učíte programovanie?

Z pohľadu formálnej kvalifikácie učiteľov informatiky rozlišujeme nasledovné spôsoby získania ich kvalifikácie:

- VŠ – magisterské učiteľské, rozširujúce alebo doplnujúce pedagogické štúdium na vysokej škole,
- MPC – kvalifikačné štúdium na Metodicko-pedagogickom centre,
- INÉ – iný spôsob získania kvalifikácie,
- NEK – učiteľ informatiky bez formálnej kvalifikácie.

Rozdelenie učiteľov podľa spôsobu získania formálnej kvalifikácie uvádzame v Tabuľka 2.

Tabuľka 2: Kvalifikačná štruktúra učiteľov informatiky (zaokrúhlené na celé percentá)

kvalifikácia \ typ školy	ZŠ	SOŠ	G	Spolu
VŠ	45 %	53 %	77 %	50 %
MPC	12 %	7 %	7 %	10 %
INÉ	10 %	17 %	6 %	11 %
NEK	34 %	23 %	11 %	29 %

Zaujalo nás zistenie, že až 34 % učiteľov informatiky na ZŠ nemá formálnu kvalifikáciu na vyučovanie predmetu informatika. Na základe informácií z Ministerstva školstva, vedy, výskumu

a športu SR (ďalej MŠVVaŠ) sú učители na 1. stupni kvalifikovaní pre výučbu predmetu informatika. Túto informáciu sme učiteľom v dotazníku sprostredkovali, takže ich evidujeme ako kvalifikovaných. Primárne ide teda o učiteľov na 2. stupni ZŠ.

Vyššie uvedené údaje sme ako jediné v našom prieskume dokázali porovnať s oficiálnymi štatistikami MŠVVaŠ [12]. Porovnanie uvádzame v Tabuľka 3.

Tabuľka 3: Kvalifikovanosť učiteľov informatiky – porovnanie s dátami z MŠVVaŠ (zaokrúhlené na desiatiny percenta)

typ školy	prieskum		MŠVVaŠ	
	počet učiteľov	% kvalifikovaných	počet učiteľov	% kvalifikovaných
ZŠ	1943	66,1	7923	66,0
SOŠ	628	77,2	729	81,1
G	294	89,5	532	90,6
celkom	2865	70,9	9184	68,6

Vzhľadom na vysoký počet respondentov, považujeme vzorku učiteľov informatiky z pohľadu kvalifikovanosti za reprezentatívnu (Wilcoxonov test rozdelenia vzorky a populácie, p-hodnota = 0,5).

Na otázku dĺžky praxe vyučovania programovania odpovedali učители nasledovne Tabuľka 4:

Tabuľka 4: Dĺžka praxe vyučovania programovania (zaokrúhlené na celé percentá)

dĺžka praxe typ školy	menej ako 5 rokov	5 až 10 rokov	10 až 20 rokov	viac ako 20 rokov
ZŠ	54 %	26 %	15 %	5 %
SOŠ	57 %	17 %	18 %	9 %
G	21 %	19 %	34 %	26 %
spolu za všetky typy škôl	51 %	23 %	18 %	8 %

Zaujímavým zistením je, že viac ako polovica učiteľov informatiky ZŠ a SOŠ má menej ako 5-ročné skúsenosti s výučbou programovania. Sú to v podstate začínajúci a menej skúsení učители programovania, ktorí by možno ocenili pomoc skúsenejších kolegov. Nerovnomerné rozdelenie učiteľov podľa dĺžky praxe vyučovania programovania na ZŠ a SOŠ môže naznačovať aj skutočnosť, že väčšina málo skúsených učiteľov opúšťa svoje pôsobisko do 5 rokov. Do skupiny mierne pokročilých učiteľov (5 až 10 rokov praxe) tak prichádza menej učiteľov, než by sme očakávali. Tento trend je badateľný aj pri prechode z kategórie 5 – 10 do kategórie 10 – 20, aj keď už nie v takej miere.

Na druhej strane rozdelenie učiteľov informatiky podľa dĺžky praxe vyučovania programovania je na G rovnomernejšie, čo naznačuje väčšiu stabilitu ich pôsobenia na G.

Celkovo len 26 % učiteľov informatiky môžeme považovať za skúsených učiteľov programovania (prax s vyučovaním viac ako 10 rokov).

Rozdiely v dĺžke praxe pre jednotlivé typy škôl sú štatisticky významné (X2TI p-hodnota = 6.8e-09, FT p-hodnota = 3.6e-04). Percentuálne zastúpenie učiteľov v jednotlivých kategóriách podľa dĺžky praxe je štatisticky rovnaké pre G (X2GFT p-hodnota= 0.14), pre ZŠ a SOŠ je štatisticky rozdielne.

2.4.2 Učители a programovacie jazyky

Aj keď pri výučbe programovania by nemal byť programovací jazyk cieľom, ale skôr nástrojom, zisťovali sme ktoré

programovacie jazyky a v akých situáciách učiteľia používajú. Domnievame sa, že ak učiteľ pozná viac programovacích jazykov, dokáže niektoré koncepty zovšeobecniť a zbytočne nesústrediť pozornosť na prvky, ktoré sú špecifikom konkrétneho programovacieho jazyka. Učiteľom sme položili otázku:

- Uveďte ako využívate nasledovné programovacie jazyky. Pre každý programovací jazyk môžete vybrať aj viac možností (pre potreby tohto článku sme formuláciu tejto otázky mierne modifikovali, ale bez zmeny jej významu).

Uviedli sme niekoľko konkrétnych programovacích jazykov:

- Imagine Logo,
- Scratch,
- Baltík,
- Pascal, Lazarus, Delphi,
- Python,
- C, C++, C#, NXC,
- NXT-G, EV3-G,
- PHP,
- Javascript,
- Java,
- MIT App Inventor,
- Iné

a pri každom mohol učiteľ vybrať niekoľko možností, ako daný jazyk používa:

- Nevie v ňom programovať,
- Viem v ňom programovať, ale nepoužívam ho,
- Vyučujem ho,
- Programujem v ňom pre vlastnú alebo profesijnú potrebu,
- Programujem v ňom profesionálne (žijím sa ako programátor).

V Tabuľka 5 uvádzame vyhodnotenie tých programovacích jazykov, ktoré učiteľia vyučujú.

Tabuľka 5 Zastúpenie programovacích jazykov, ktoré vyučujú učiteľia informatiky (zaokrúhlené na celé percento)

	Imagine Logo	Scratch	Baltík	Pascal, Lazarus, Delphi	Python	C, C++, C#, NXC	NXT-G, EV3-G	PHP	Javascript	Java	MIT App Inventor	Iné
ZŠ	37	33	19	3	6	11	4	1	2	1	1	15
SOŠ	9	4	2	11	21	20	3	11	8	6	4	10
G	36	32	2	25	71	9	10	6	4	4	14	10

Potvrďilo sa, že zastúpenie vyučovaných programovacích jazykov je štatisticky významné a závislé od typu školy (FT p-hodnota = 1.7e-04).

Učiteľia ZŠ učia programovanie dominantne v prostredí Imagine Logo a Scratch. Pri učiteľoch SOŠ dominujú jazyky typu Python a C, aj keď ich zastúpenie nie je také výrazné. Učiteľia G učia programovanie dominantne prostredníctvom programovacích jazykov typu Python, Imagine Logo a Scratch, pričom Python má výraznú dominanciu (71,4 %).

Na ZŠ je štatisticky významné výrazné zastúpenie skupiny jazykov Imagine Logo, Scratch v porovnaní s tretím najpočetnejším jazykom Baltík (X2GFT p-hodnota = 0.048). Na G je štatisticky významné zastúpenie jazyka Python v porovnaní so skupinou jazykov Imagine Logo, Scratch a jazykov typu Pascal (X2GFT p-hodnota = 5.6e-07). Na SOŠ takáto výrazná úzka skupina jazykov nie je, čo sa potvrdilo aj štatisticky. Absenciu takejto dominantnej skupiny programovacích jazykov môžeme interpretovať ako dôsledok rôznych zameraní SOŠ.

Tieto výsledky vypovedajú nie len o samotných učiteľoch, ale sú ovplyvnené aj typom školy, kde učiteľ pôsobí (napr. učiteľ môže ovládať jazyk Java, ale ak pôsobí na ZŠ, zrejme ho do vyučovania nezarádi).

Zaujímalo nás, koľko programovacích jazykov učiteľia vyučujú a v akom vzťahu je táto informácia k typu formálneho vzdelania a dĺžke praxe vyučovania programovania. Naše zistenie uvádzame v Tabuľka 6 a v Tabuľka 7.

Tabuľka 6 Priemerný počet vyučovaných programovacích jazykov v závislosti od typu formálneho vzdelania učiteľa informatiky

typ formálneho vzdelania	počet vyučovaných programovacích jazykov
Mgr. štúdium na VŠ	1,41
RŠI štúdium na VŠ	1,82
doplňujúce pedagog. štúdium na VŠ	1,57
štúdium na MPC	1,29
iný spôsob získania kvalifikácie	1,22
bez formálnej kvalifikácie	0,87

Tabuľka 7 Priemerný počet vyučovaných programovacích jazykov učiteľom informatiky v závislosti od dĺžky praxe vyučovania programovania

dĺžka praxe	počet vyučovaných jazykov
menej ako 5 rokov	0,81
5 – 10 rokov	1,57
10 – 20 rokov	1,93
viac ako 20 rokov	2,02

Z tabuliek vyplýva, že počet vyučovaných programovacích jazykov učiteľom informatiky súvisí s dĺžkou praxe vyučovania programovania a so spôsobom získanej formálnej kvalifikácie.

Ďalšou analýzou získaných dát sme zistili, že nekvalifikovaný učiteľ informatiky začínajúci vyučovať programovanie vyučuje v priemere len 0,72 programovacieho jazyka. Druhý extrém je kvalifikovaný učiteľ informatiky (s kvalifikáciou získanou na VŠ) s viac ako 20-ročnou praxou vyučovania programovania. Takýto učiteľ vyučuje v priemere až 2,4 programovacieho jazyka.

Zistili sme, že nezanedbateľná časť učiteľov informatiky pri každom z ponúknutých programovacích jazykoch (vrátane možnosti Iné) vybrala možnosť „Neviem v ňom programovať“. Na G je to 1,02 %, na SOŠ 12,9 % a na ZŠ až 18,01 % učiteľov informatiky. Výsledok pre ZŠ je znepokojujúci aj z dôvodu,

že programovanie je súčasťou štandardov informatiky už od prvého stupňa ZŠ.

2.4.3 Učítelia a oblasti programovania

Vytipovali sme niekoľko kľúčových oblastí programovania a zisťovali sme, v akom vzťahu je učiteľ k jednotlivým oblastiam programovania. Učiteľom sme položili otázku:

- Uvedte vaše skúsenosti s nasledujúcimi témami z oblasti programovania.

Učiteľom sme ponúkli konkrétne témy:

- cykly,
- podmienené príkazy,
- podprogramy (funkcie, procedúry),
- dátové štruktúry – zoznamy/polia,
- dátové štruktúry – slovníky/asociatívne polia/mapy, množiny, n-tice, záznamy ...,
- dátové štruktúry – súbory,
- programovanie pokročilejších algoritmov (usporadúvania, vyhľadávania, šifrovanie, kódovania, grafové algoritmy ...),
- krokovanie, ladenie a testovanie programov,
- spracovanie výnimiek,
- spracovanie vstupov zo senzorov (intenzity zvuku, osvetlenia, teploty, zrýchlenia, GPS ...),
- rekurzia,
- objektové programovanie),

a pri každej téme niekoľko navzájom sa vylučujúcich možností:

- neovládam,
- ovládam a vyučujem,
- ovládam ale nevyučujem.

Z pohľadu tém programovania, ktoré učítelia informatiky vyučujú, sme zistili nasledovné (Tabuľka 8).

Typ školy je hlavným určujúcim faktorom pre výber vyučovaných tém. Dôvodom je, že pre každý typ školy je určený iný vzdelávací štandard.

Ďalšou analýzou sme zistili, že na vyučované témy v rámci daného typu školy vplývajú spôsob získanej kvalifikácie a dĺžka praxe vyučovania programovania. Učiteľ s kvalifikáciou získanou na VŠ (denné štúdium, RŠI, DPŠ) a učítelia s dlhšou pedagogickou praxou výučby programovania vyučujú väčší počet tém a aj náročnejšie témy.

Percento učiteľov vyučujúcich danú tému je na ZŠ a SOŠ nižší oproti G. Na ZŠ to môže byť spôsobené tým, že učiteľ vyučuje na prvom stupni, kde uvedené témy nie sú uvedené vo vzdelávacom štandarde. Podobne na SOŠ môže byť dôvodom fakt, že SOŠ s učebnými odbormi bez maturity nemajú v štandarde oblasť Algoritmické riešenie problémov.

Výučbu témy „krokovanie, ladenie a testovanie programov“ uviedlo len 21,46 % učiteľov ZŠ, pričom táto téma je uvedená v štandardoch pre ZŠ (aj na prvom stupni). Túto tému vyučuje 60,54 % učiteľov G, čo je vzhľadom na povinné postavenie tejto témy relatívne nízke zastúpenie.

Výučbu tém „cykly“ a „podmienené príkazy“ vyučuje len necelá polovica učiteľov ZŠ. Tieto témy vyučuje viac ako 90 % učiteľov G.

Výučbu témy „spracovanie vstupov zo senzorov“ uviedlo pomerne nízke percento učiteľov na všetkých typoch škôl. Príčinami môžu byť novosť témy, viazanosť na špeciálny hardvér a výučba témy až v sekundárnom kurze programovania.

Výučbu tému „programovanie pokročilejších algoritmov“ uviedlo len 32,65 % učiteľov G. Vzhľadom na to, že žiaci v tomto veku už majú za sebou minimálne 6 rokov programovania na ZŠ, by sme očakávali väčšie zastúpenie tejto témy na G.

Výučbu témy „rekurzia“ uviedlo až 37,41 % učiteľov G. Táto téma nie je uvedená vo vzdelávacích štandardoch (dokonca ani v maturitnom). Učia ju, možno kvôli tradícii, najmä učítelia s dlhšou praxou vyučovania programovania.

Výučbu témy „spracovanie výnimiek“ uviedlo len 26,19 % učiteľov G, pričom je implicitne uvedená vo vzdelávacích štandardoch.

Prekvapilo nás pomerne vysoké percento učiteľov, ktorí v rámci programovania vyučujú tému „dátové štruktúry – súbory“. Táto téma nie je uvedená v IŠVP. Domnievame sa, že učítelia pravdepodobne nesprávne pochopili otázku a problematiku dátovej štruktúry súbor vyhodnotili v kontexte celej informatiky a nie len v kontexte témy programovanie.

Podobne prekvapivé je naopak nízke percento učiteľov, ktorí vyučujú tému „objektové programovanie“. Dominantné jazyky, Imagine Logo a Scratch na ZŠ a Python na G, sú objektové jazyky. Predpokladáme, že učítelia si túto skutočnosť buď nevedomujú alebo programovanie vyučujú primárne ako procedurálne.

Keďže v tejto chvíli nemáme k dispozícii ďalšie informácie, obidve interpretácie predchádzajúcich výsledkov ostávajú len v rovine hypotéz.

V záujme korektnosti musíme uviesť, že ak učiteľ nevyučuje nejakú tému, nemusí to byť nutne negatívne zistenie. Učiteľ môže dlhodobo učiť len v tých ročníkoch, kde sa daná téma nevyučuje. Jeho kolega v inom ročníku tak môže túto tému žiakom sprostredkovať. Z dát, ktoré sme získali v prieskume, to ale nevieme zistiť.

Z pohľadu tém programovania, ktoré učítelia neovládajú, sme zistili nasledovné (Tabuľka 8).

Znepokojujúcim zistením je pomerne veľké zastúpenie učiteľov informatiky, ktorí neovládajú základné témy z programovania (cykly, podmienené príkazy, podprogramy) na ZŠ.

Tému „spracovanie výnimiek“ na G neovláda 40,48 % učiteľov informatiky.

Aj tu sa potvrdzuje, že učítelia G sú pomerne dobre odborne vybavení a „základné“ témy z programovania ovládajú takmer všetci.

Samotný fakt, že učítelia informatiky neovládajú základné programátorské koncepty sa podľa nás prejaví nie len v oblasti vyučovania programovania. Títo učítelia nemusia mať celkom jasnú predstavu o automatickom spracovaní údajov a o postupoch pri hľadaní chýb, čo sa následne prenesie aj do iných oblastí školskej informatiky. Vo výučbe tak môžu uprednostňovať postupy s prevládajúcou manuálnou činnosťou pred automatizáciou a strojovou prácou.

Tabuľka 8 Pomer učiteľov, ktorí jednotlivé témy programovania vyučujú a ktorí jednotlivé témy programovanie neovládajú (záhlavia sú skrátené a údaje zaokrúhlené na celé percento)

Typ školy	cykly	podmienené príkazy	podprogramy	dátové štruktúry – zoznamy/polia	dátové štruktúry – slovníky, množiny, n-tice, záznamy	dátové štruktúry – súbory	programovanie pokročilejších algoritmov	krokovanie, ladenie a testovanie programov	spracovanie výnimiek	spracovanie vstupov zo senzorov	rekurzia	objektové programovanie
témy programovania, ktoré učitelia informatiky VYUČUJÚ												
ZŠ	48	44	31	11	5	12	3	21	4	14	5	8
SOŠ	52	53	45	37	18	28	12	30	12	17	14	18
G	92	93	86	75	39	68	33	61	26	22	37	23
témy programovania, ktoré učitelia informatiky NEOVLÁDAJÚ												
ZŠ	38	37	48	57	69	57	81	60	79	67	79	75
SOŠ	18	16	24	29	51	36	61	40	64	62	52	53
G	2	2	4	9	26	13	32	11	40	48	16	35

2.5 Vyhodnotenie prieskumu a diskusia

Výsledky prieskumu nám potvrdili viaceré naše predpoklady a zároveň pomohli vysvetliť prečo tomu tak je. Naše zistenia poukazujú na to, že problémom nie sú samotní učitelia informatiky, ale systém vzdelávania na Slovensku. Tento systém umožňuje, aby žiakov v takejto miere vzdelávali nekvalifikovaní učitelia informatiky a zároveň v systéme chýbajú kontrolné mechanizmy pre zistenie úrovne plnenia cieľov vzdelávacieho štandardu v informatike. Podľa štatistik CVTI SR [13] je problém s kvalifikovanosťou učiteľov informatiky dlhodobý. Podľa nás si však vyžaduje razantnejšie riešenie.

Za obdobie 5 rokov (2017 – 2021) VŠ na Slovensku vyprodukovali priemerne ročne 92 učiteľov informatiky pre 2. stupeň ZŠ a SŠ¹. Podľa RIS portálu [12] bolo k 15. 9. 2021 zamestnaných na 2. stupni ZŠ a SŠ spolu 4733 učiteľov informatiky (kvalifikovaných aj nekvalifikovaných). Predpokladajme, že priemerný učiteľ odchádza do dôchodku vo veku 63 rokov a jeho pracovné obdobie je 39 rokov. Zjednodušene, každý rok do dôchodku odchádza 1/39 učiteľov, t. j. 121 učiteľov. To znamená, že každý rok nám v priemere narastá deficit o 29 učiteľov informatiky. Toto číslo je v skutočnosti väčšie, lebo nie každý absolvent učiteľstva informatiky sa zamestná v školstve, nie každý učiteľ odpracuje v školstve až 39 rokov a absolventi RŠI a DPŠ sú často z radov nekvalifikovaných učiteľov informatiky.

Ukázalo sa, že na našich školách pôsobí neprimerane veľa začínajúcich učiteľov v oblasti programovania na úkor skúsených

učiteľov. Navyše sa ukazuje, že začínajúci učitelia neprechádzajú do skupiny skúsených učiteľov v takej miere ako by sa očakávalo. Tieto problémy sú najvypuklejšie na základných školách. Situácia na gymnáziách je neporovnateľne lepšia, ale aj tu je veľký priestor na zlepšenie. Navyše hodinová dotácia vyučovania informatiky na gymnáziách neumožňuje eliminovať vedomostný sklz žiakov z oblasti programovania zo základných škôl.

Už v predchádzajúcej časti sme naznačili, že počet programovacích jazykov, ktoré učiteľ ovláda mu umožňuje zamerať sa na koncepčné, podstatné veci z oblasti programovania. Aj tu sa ukazuje, že spôsob získania formálnej kvalifikácie a dĺžka praxe vyučovania programovania majú vplyv na počet jazykov, ktoré učiteľ vyučuje. Znepokojivé je zistenie, že existuje pomerne veľká skupina učiteľov informatiky na ZŠ, ktorá neovláda žiaden programovací jazyk.

Témy, ktoré učiteľ vyučuje sú primárne určené typom školy na ktorej pôsobí. Zistili sme však, že mnohí učitelia neovládajú ani základné témy programovania. Tento problém sa najviac prejavuje na základných školách. Prejavuje sa aj negatívny dopad nízkeho počtu učiteľov s dlhšou praxou. Nielen základné témy, ale aj náročnejšie témy z programovania, sú schopní učiť najmä títo skúsení učitelia.

Aj keď sme primárne zisťovali informácie o učiteľoch informatiky, tieto údaje môžu nepriamo naznačovať, čo sa na školách učí a čo nie. Tento predpoklad potvrdzuje aj fakt, že na každom stupni vzdelávania sa výučba programovania realizuje prakticky od začiatku.

3 ZÁVER

Kvalita výsledkov vzdelávania žiakov je primárne závislá na kvalite učiteľov, ktorí žiakov vzdelávajú. Kvalifikačná štruktúra učiteľov informatiky je taká, akú samotný vzdelávací systém umožňuje, resp. toleruje. Pre zlepšenie tohto nelichotivého stavu je potrebné razantné a systémové riešenie s dlhodobou víziou.

Na Slovensku sa realizovali dva veľké národné projekty [10, 14], ktoré prispeli k zlepšeniu kvalifikačnej štruktúry učiteľov informatiky (kvalifikačné a inovačné vzdelávania, metodické a odborné materiály pre učiteľov). Aj napriek tomu sa situácia v oblasti kvalifikačnej štruktúry učiteľov informatiky výrazne nezlepšila.

Uvedené problémy a ich dôsledky už „kypia“ cez našu informatickú pokrievku. Verejne sa prezentujú informácie ako „buď si na informatike púšťajú nejaké filmy, YouTube, hrajú hry alebo sa nudia“, „[pani učiteľka] učí slovenčinu a dejepis a pani riaditeľka jej oznámila, že bude učiť informatiku“, „zhruba 50 % učiteľov na školách je nekvalifikovaných alebo nemajú dostatočnú kvalifikáciu pre výučbu informatiky“ alebo „Len málo z nich [detí s talentom pre technológie] však má vytvorené podmienky na to, aby mohli svoj talent ďalej rozvíjať.“ [15], „Nejaké základné zručnosti im to dá [výučba informatiky deťom], aby si vedeli zapnúť počítač a pracovať s myšou, ale nemá to žiadny hlbší obsah, čo vidíme ako problém“ [16], „Informatiku na každej druhej základnej škole vyučuje telocvikár, slovenčinárka či učiteľka zemepisu“ [17]. Takéto vnímanie vyučovania informatiky, aj keď z časti oprávnené, dehonestuje našu prácu a ak včas nezasiahne, bude nás stáť nemalo úsilia tento trend zvrátiť.

všetky VŠ na Slovensku pripravujúce učiteľov informatiky na základe informačného zákona.

¹ Informácie o počte absolventov (denného, rozširujúceho a dopĺňajúceho pedagogického štúdia informatiky) nám poskytli

Riešenie vidíme v zatriktní povolaní učiteľa informatiky, aby sa vysoké školy pripravujúce budúcich učiteľov informatiky mohli z režimu náboru uchádzačov presunúť do režimu výberu uchádzačov. Súčasne s tým je potrebné podniknúť také kroky, aby absolventi učiteľstva informatiky mali záujem pracovať v školstve a videli v tejto práci perspektívu a zmysel a zo zamestnania v školstve predčasne neodchádzali. Aktivity firiem (aj keď často nekomerčné) a tretieho sektora majú svoj význam, ale riešenie tohto problému nie je primárne v ich kompetencii.

Zlepšenie tohto stavu môže začať hlbšou diskusiou v odborných kruhoch (didaktici informatiky a informatici) a aj s učiteľmi informatiky. Výsledkom diskusie by mohla byť výzva pre kompetentných podložená relevantnými dátami.

POĎAKOVANIE

Tento článok bol vytvorený v rámci Národného projektu IT Akadémia – vzdelávanie pre 21. storočie, ktorý sa realizuje vďaka podpore z Európskeho sociálneho fondu a Európskeho fondu regionálneho rozvoja v rámci Operačného programu Ľudské zdroje a v rámci projektu KEGA 012UPJŠ-4/2021 Vývoj digitálnej knižnice interdisciplinárnych STEAM projektov a jej implementácia do informatického, matematického a prírodovedného vzdelávania na stredných školách.

BIBLIOGRAFICKÉ ODKAZY

- [1] NÁRODNÝ INŠTITÚT VZDELÁVANIA A MLÁDEŽE. Inovovaný Štátny vzdelávací program. In: *Inovovaný Štátny vzdelávací program – ŠPÚ, 2022* [online] [cit. 11.01.2023]. Dostupné na: <https://www.statpedu.sk/svp/inovovany-statny-vzdelavaci-program/>
- [2] ODDELENIE ZNALOSTNÝCH VIED A DIDAKTIKY INFORMATIKY, ÚSTAV INFORMATIKY, PRÍRODOVEDECKÁ FAKULTA, UPJŠ V KOŠICIACH. Klub učiteľov informatiky. In: *Klub učiteľov informatiky* [online] [cit. 12.01.2023]. Dostupné na: <https://lms.science.upjs.sk/course/view.php?id=3>
- [3] KALAŠ, Ivan. *Informatika na križovatke. Mój názor Informatics education at the crossroad. Position paper* [online]. Banská Bystrica: Univerzita Mateja Bela, Fakulta prírodných vied, Katedra informatiky a Technická univerzita v Liberci, Fakulta prírodovedne-humanitní a pedagogická, 2021 [cit. 10.01.2023]. ISBN 978-80-557-1823-1. Dostupné na: http://www.didinfo.net/images/DidInfo/files/DIDINFO_2021_zbornik.pdf
- [4] UNIVERZITA P. J. ŠAFÁRIKA V KOŠICIACH. *ÚINF/PAZ1a/15 – Programovanie, algoritmy, zložitost'* [online]. 2023 [cit. 11.01.2023]. Dostupné na <https://studijne-programy.upjs.sk/predmet/%C3%9AINF%2FPAZ1a%2F15>
- [5] UNIVERZITA KOMENSKÉHO V BRATISLAVE. *INFORMAČNÝ LIST PREDMETU – Programovanie (1)* [online]. 2022 [cit. 11.01.2023]. Dostupné na: https://sluzby.fmph.uniba.sk/infolist/sk/1-UIN-140_22.html
- [6] FAKULTA PRÍRODNÝCH VIED, UMB BANSKÁ BYSTRICA. Informačné listy predmetov zoradené v slede povinných a povinne voliteľných predmetov podľa odporúčaného študijného plánu dennej formy štúdia [online]. 2022 [cit. 15.01.2023]. Dostupné na: <https://www.fpv.umb.sk/app/cmsFile.php?disposition=i&ID=21160>
- [7] BAREFOOT. Computational Thinking Concepts and Approaches | Barefoot. In: [cit. 21.11.2022]. Dostupné na: <https://www.barefootcomputing.org/concept-approaches/computational-thinking-concepts-and-approaches>
- [8] COMMISSION, European et al. *DigComp 2.2, The Digital Competence framework for citizens : with new examples of knowledge, skills and attitudes*. Publications Office of the European Union, 2022. DOI: [doi/10.2760/115376](https://doi.org/10.2760/115376)
- [9] ECDL FOUNDATION, SLOVENSKÁ INFORMATICKÁ SPOLOČNOSŤ. Moduly v skupine Standard. In: *Moduly v skupine Standard* [online]. 2023 [cit. 10.01.2023]. Dostupné na: <https://www.ecdl.sk/moduly-v-skupine-standard>
- [10] CVTI SR. IT Akadémia – vzdelávanie pre 21. storočie. In: *Hlavná stránka – IT akadémia* [online]. s.d. [cit. 26.08.2022]. Dostupné na: <https://itakademia.sk/>
- [11] MINISTERSTVO ŠKOLSTVA, VEDY, VÝSKUMU A ŠPORTU. RIS – Portál. In: *Registre regionálneho školstva* [online]. 2023 [cit. 10.01.2022]. Dostupné na: <https://crinfo.iedu.sk/RISPortal/register/>
- [12] MINISTERSTVO ŠKOLSTVA, VEDY, VÝSKUMU A ŠPORTU SLOVENSKEJ REPUBLIKY. RIS – Portál. In: *RIS - Portál* [online]. 2022 [cit. 10.01.2023]. Dostupné na: <https://crinfo.iedu.sk/RISPortal>
- [13] CENTRUM VEDECKO-TECHNICKÝCH INFORMÁCIÍ SR. IKT(MŠVVŠ SR) 1 – 01 – CVTI SR. In: *IKT(MŠVVŠ SR) 1 – 01 Ročný výkaz o informačných technológiách v škole* [online]. 30. 5. 2022 [cit. 10.01.2023]. Dostupné na: https://www.cvtisr.sk/cvti-sr-vedecka-kniznica/informacie-o-skolstve/zber-udajov/vykazy-typu-skol-msvvs-sr/iktmvvs-sr-1-01.html?page_id=10019
- [14] KALAŠ, Ivan et al. Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika – Záverečná správa projektu DVUi (2008 – 2011) [online]. 2018 [cit. 15.01.2023]. Dostupné na: https://www.statpedu.sk/files/sk/o-organizacii/projekty/projekt-dvui/zaverecna_sprava_final.pdf
- [15] © ZOZNAM/. Stav vzdelania učiteľov je žalostný! Treba urobiť zmenu, digitálna doba ide dopredu. In: *Stav vzdelania učiteľov je žalostný! Treba urobiť zmenu, digitálna doba ide dopredu | Topky.sk* [online]. 10. 2. 2023 [cit. 10.02.2023]. Dostupné na: https://www.topky.sk/cl/1005135/2461365/?itm_source=topky&itm_medium=right-column
- [16] FOLENTOVÁ, Veronika. Snažia sa zlepšiť vyučovanie informatiky: Takmer polovica učiteľov informatiku neštudovala, niektorí minulý rok učili výtvarnú. In: *Snažia sa zlepšiť vyučovanie informatiky: O bezpečnosti, hoaxoch a sociálnych sieťach hovoríme so žiakmi neskoro* [online]. 16. 2. 2019 [cit. 10.02.2023]. Dostupné na: <https://dennikn.sk/1385660/snazia-sa-zlepsit-vyucovanie-informatiky-o-bezpecnosti-hoaxoch-a-socialnych-sietach-hovorime-so-ziakmi-neskoro/>
- [17] NEJEDLÝ, Tomáš. Vyše polovica škôl vyučujú deti informatiku neodborne. In: *Vyššie polovica škôl vyučujú deti informatiku neodborne | TREND* [online]. 2. 3. 2018 [cit. 10.02.2023]. Dostupné na: <https://www.trend.sk/spravy/vyse-polovica-skol-vyucujuj-deti-informatiku-neodborne>

Subjektivní vnímání obtížnosti programovacích úloh v závislosti na výuce programování

Subjective perception of the difficulty of programming tasks depending on the teaching of programming

Petr Hanzal

Jihočeská univerzita v Českých
Budějovicích, Pedagogická fakulta
Jeronýmova 10
37115 České Budějovice
Česká republika
hanzalp@jcu.cz

ABSTRACT

The paper deals with the subjective difficulty of algorithmic and programming tasks in the Bebras Challenge 2022 (senior category), as experienced by third-year and fourth-year students at secondary schools.

A questionnaire survey was conducted to explore the subjective task difficulty experienced by individual contestants. The survey also asked if the contestants had experience with programming in informatics classes. The contestants were then divided into groups according to the existence or non-existence of previous programming experience.

The research findings have shown that programming tasks are easier for contestants with previous programming experience from informatics classes, compared to contestants with no previous programming experience. The research findings also include the popularity of specific tasks and the programming languages the contestants used when they were learning programming in secondary school classes.

Keywords

Bebras challenge, programing, programing tasks, difficulty

ABSTRAKT

Článek se zabývá problematikou subjektivně vnímané obtížnosti studenty třetích a čtvrtých ročníků středních škol u algoritmičtých a programovacích úloh zařazených do informatické soutěže Bobřík informatiky v kategorii senior, ročník 2022.

Bylo provedeno dotazníkové šetření, ve kterém bylo zkoumáno subjektivní vnímání obtížnosti soutěžních úloh jednotlivými soutěžícími. Dále bylo zjišťováno, jestli se soutěžící setkali při výuce informatiky s programováním. Soutěžící byli rozdělení do skupin podle toho, jestli se s programováním setkali nebo nesetkali.

Při výzkumu bylo zjištěno, že programovací úlohy jsou jednodušší pro soutěžící, kteří se s programováním setkali při výuce informatiky, než pro soutěžící, kteří se s programováním nesetkali. Během výzkumu byla zjištěna i obliba jednotlivých úloh a programovacích jazyků, ve kterých se soutěžící učili programovat při výuce na střední škole.

Klíčové slová

Bobřík informatiky, programování, programovací úlohy, obtížnost

1 ÚVOD

Informatické soutěže Bobřík informatiky se každý rok účastní tisíce soutěžících z celé České republiky. Součástí této soutěže jsou úlohy, které jsou zaměřené na algoritmizaci a programování. Novým typem úloh jsou programovací úlohy, ve kterých soutěžící sestavují vlastní programy pomocí blokového programovacího jazyka.

Algotmizace a programování se díky revizím rámcových vzdělávacích programů pomalu, ale jistě stává povinnou součástí výuky informatiky na všech typech škol.

V tomto příspěvku dojde k porovnání subjektivně vnímané obtížnosti soutěžních úloh z oblasti algoritmizace a programování mezi respondenty, kteří se s programováním setkali ve výuce informatiky, a respondenty, kteří se s programováním nesetkali.

2 Bobřík informatiky

Informatická soutěž Bobřík informatiky probíhá na českých školách od roku 2008 [1]. Soutěž seznamuje žáky a učitele s informatickými úlohami a problémy. Soutěž dokazuje, že informatika není o ovládání počítače. Soutěžící vyplňují online test v počítačové učebně. Soutěží se v pěti věkových kategoriích od 4. ročníku základní školy po maturitu. Soutěžící vybírají správnou odpověď z několika odpovědí, nebo přemísťují objekty na obrazovce [2]. V novém typu soutěžních úloh soutěžící sestavují programy pomocí blokového programovacího jazyka Blockly.

Soutěžní úlohy se liší svou obtížností, zaměřením a způsobem řešení. Tematicky je možné úlohy rozdělit do čtyř oblastí.

1. Algotmizace a programování.
2. Porozumění informacím a jejich prezentacím (kódování, šifrování) a strukturám (grafy).
3. Řešení problémů (hledání strategií).
4. Digitální gramotnost [3].

Tato témata jsou součástí revidovaného Rámcového vzdělávacího plánu pro gymnázia [4].

2.1 Programovací úlohy

Programovací úlohy jsou interaktivní situační úlohy z programování. Do soutěže byl implementován modul Blockly (<https://developers.google.com/blockly>) s omezenou sadou programovacích příkazů [5].

V těchto úlohách soutěžící tvoří program z bloků. Program lze testovat (opakovaně spouštět) a ladit. Při každém spuštění získávají od systému okamžitou zpětnou vazbu [5].

Programovací úlohy vycházejí ze tří šablon tzv. mikrosvětů. Tyto mikrosvěty obsahují odlišené sady příkazů a ovládané postavy se chovají jinak. Ke každému mikrosvětu lze vytvořit sadu úloh s odlišnou grafikou i tématem [5].

Soutěžní test kategorie Senior ročníku 2022 obsahoval jednu úlohu z mikrosvěta Film, dvě úlohy z mikrosvěta Želva a čtyři algoritmičké úlohy.

Mikrosvět Film

V tomto mikrosvětě je programován objekt (postava), která v čase mění svou polohu a velikost (soutěžící vytváří animaci). Parametry objektu jsou:

- pozice x;
- pozice y;
- velikost;
- otočení.

Parametr otočení udává otočení postavy oproti původnímu směru [5].

Základním příkazem programovacího jazyka je příkaz postava. Tento příkaz vykreslí postavu na určené pozici, s velikostí danou parametrem velikost a otočenou o přesný počet stupňů. Jazyk obsahuje blok pro realizaci základních početních úkonů a blok Když (podmínka), který kontroluje podmínku času [5].

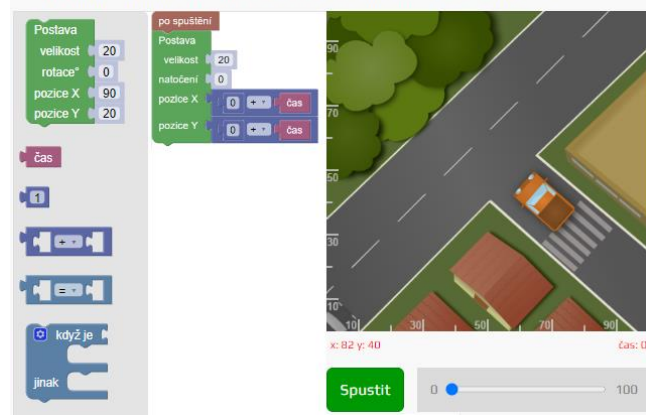
Soutěžící mají v těchto úlohách za úkol naprogramovat postavu tak, aby se po celou dobu trvání animace chovala stejně jako její stín, který soutěžící vidí po spuštění animace (Obrázek 1) [5].

Dvě rychlá auta

Dvě rychlá auta projela po hlavní silnici.

Nejprve stiskni tlačítko *Spustit*.

Sestav program, podle něhož se po spuštění bude barevný obrázek chovat stejně jako jeho stín.



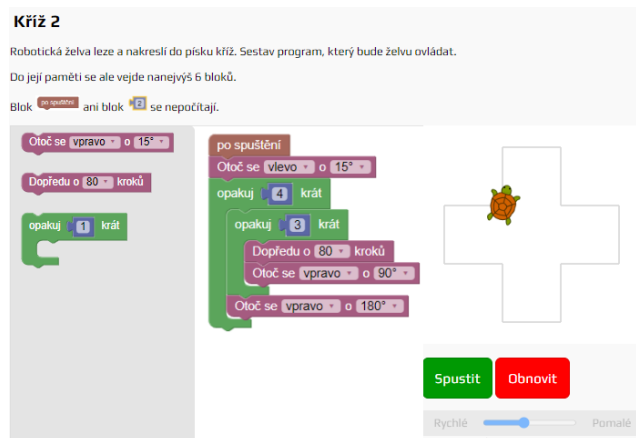
Obrázek 1: Ukázka prostředí mikrosvěta Film

Prostředí tohoto mikrosvěta využívá koncept parametr, pracuje s proměnnou a výrazy. Z tohoto důvodu jsou tyto úlohy určené především pro soutěžící ze středních škol a posledních ročníků škol základních [5].

Mikrosvět Želva

V mikrosvětě Želva soutěžící ovládá pomocí programového kódu želvu, která kreslí obrázek. Cílem těchto úloh je vytvořit program,

podle kterého želva nakreslí obrázek podle zadání. Zadaný obrázek soutěžící vidí (Obrázek 2).



Obrázek 2: Ukázka mikrosvěta Želva

Součástí mikrosvěta jsou bloky pro posun a otočení želvy. Želva se posouvá pouze dopředu. Otáčí se o zadaný počet stupňů. Soutěžící má k dispozici také blok Opakuji představující cyklus s pevným počtem opakování. U složitějších obrazců s přerušovanou čarou jsou k dispozici bloky pro zahájení a ukončení kreslení.

3 Výzkumná otázka

Pro účely výzkumu byla stanovena následující výzkumná otázka: „Jsou programovací úlohy jednodušší pro soutěžící, kteří se při výuce informatiky setkali s programováním, než pro soutěžící, kteří se s programováním nesetkali?“

Na základě výzkumné otázky byla formulována hypotéza: programovací úlohy jsou jednodušší pro soutěžící, kteří se s programováním setkali. Tuto hypotézu bylo potřeba ověřit pomocí vhodné statistické metody.

4 Výzkumná metoda

Výzkumnou metodou byla metoda kvantitativního výzkumu. Sběr dat byl realizován dotazníkovým šetřením. Dotazník byl vytvořený pomocí nástroje na vytváření online dotazníků, který je součástí testové aplikace.

Každému soutěžícímu v kategorii senior byla po dokončení soutěžního testu nabídnuta možnost vyplnit dotazník. Výhodou tohoto dotazníku je možnost propojit odpovědi v dotazníku s konkrétním soutěžícím pomocí jeho soutěžního kódu.

Soutěžící v dotazníku hodnotili obtížnost jednotlivých soutěžních úloh. Ze všech soutěžních úloh mohli vybrat dvě, které se jim nejvíce líbily. Odpovídali i na otázky, které zjišťovaly, s jakými tematickými celky se setkali a nesetkali při výuce informatiky (jedním z těchto celků bylo programování). Soutěžící dále sdělili, v jakém programovacím jazyce se učili programovat, pokud se s programováním setkali.

Dvouvýběrový t-test

Tento test slouží k porovnání průměrné hodnoty určité vlastnosti u dvou různých populací. Hodnotí se rozdíl průměrů dané náhodné proměnné ve dvou populacích. Výběr z první populace se označí x_1, \dots, x_n , předpokládá se rozdělení $N(\mu_1, \sigma^2)$. Výběr z druhé populace y_1, \dots, y_n , s rozdělením $N(\mu_2, \sigma^2)$. Předpokládá se, že rozptyly v obou populacích jsou stejné [6].

Testuje se nulová hypotéza $H_0 : \mu_1 = \mu_2$ proti alternativní hypotéze $H_1 : \mu_1 \neq \mu_2$ (případně $H_0 : \mu_1 = \mu_2$ proti $H_1 : \mu_1 < \mu_2$ resp. $H_1 : \mu_1 > \mu_2$) [6].

Dvouvýběrová testovací statistika pro t-test má následující tvar:

$$t = \frac{\bar{x} - \bar{y}}{S.E_{\bar{x}-\bar{y}}}$$

\bar{x} a \bar{y} jsou výběrové průměry z obou populací. $S.E_{\bar{x}-\bar{y}}$ je směrodatná odchylka rozdílu průměrů [6].

5 Výsledky

5.1 Práce s daty

V roce 2022 se soutěže zúčastnilo 6 880 soutěžících v kategorii senior [7]. Dotazník vyplnilo celkem 534 respondentů (7,76 %). Z tohoto počtu bylo vyřazeno 55 respondentů. Důvody vyřazení byly: testovací aplikace uživatele určila k vyřazení (devět respondentů), respondent nedal souhlas s propojením svých odpovědí se soutěžním výsledkem (23 respondentů), a protirečení si v otázkách dotazníku, které zjišťovaly, se kterými tématy se soutěžící setkali a nesetkali při výuce informatiky (23 respondentů). Vyřazení byli pouze ti respondenti, kteří v obou otázkách vybrali všechny stejné možnosti. Po vyřazení těchto respondentů zbylo 479 odpovědí.

Tyto odpovědi bylo nutné rozdělit na dvě skupiny, podle toho, jestli se respondenti setkali s programováním ve výuce informatiky. K tomuto účelu posloužily dvě otázky v dotazníku. První otázka zněla: „Z nabízených témat vyberte ty, se kterými jste se setkal/a při výuce informatiky“. V druhé otázce respondenti vybírali témata, se kterými se ve výuce nesetkali. V obou otázkách se rozhodovali mezi stejnými tematickými celky, které vycházejí z revidovaného Rámcového vzdělávacího programu [4]. Jedním z nabízených témat bylo programování. Respondenti, kteří v dotazníku vybrali možnost setkal/a jsem se s programováním a zároveň nevybrali možnost, že se s programováním nesetkali tvoří jednu skupinu. Respondenti, kteří se s programováním nesetkali a zároveň v dotazníku nevybrali možnost, že se s programováním setkali tvoří skupinu druhou.

Další otázkou použitou k filtrování nevhodných respondentů byla otázka, která zněla: „Pokud vás učí programovat, napište, v jakém jazyce“. Odpovědi respondentů, kteří se s programováním setkali a na tuto otázku neodpověděli, byly vyřazeny. Odpovědi respondentů z druhé skupiny, kteří na tuto otázku odpověděli názvem konkrétního jazyka, byly také vyřazeny.

Otázka, která zjišťovala subjektivní vnímání obtížnosti jednotlivých soutěžních otázek zněla: „Jak obtížné ti přišly soutěžní otázky?“. Respondenti hodnotili obtížnost všech soutěžních otázek na škále 1-6. Číslo jedna označovala lehké, číslo šest těžké. Pokud si respondent nepamatoval soutěžní úlohy podle názvu, mohl si zobrazit náhledový obrázek úlohy. Pokud si úlohu nepamatoval ani podle obrázku, byla mu nabídnuta možnost Nepamatuji se. Ze všech odpovědí byly vybrány pouze odpovědi zaměřené na algoritmické a programovací úlohy (sedm úloh). U obou skupin respondentů byly ponechány odpovědi respondentů, kteří si pamatovali minimálně čtyři úlohy (maximálně u tří úloh vybrali možnost Nepamatuji se).

Po odfiltrování všech nevhodných odpovědí zůstalo ve skupině respondentů, kteří se s programováním setkali, 205 odpovědí. Ve skupině respondentů, kteří se s programováním nesetkali, zůstalo 68 odpovědí.

U obou skupin byly dopočítány průměry subjektivně vnímané obtížnosti soutěžních úloh z oblasti Algoritmizace a programování. Průměrnou subjektivně vnímanou obtížnost úloh pro obě skupiny respondentů zobrazuje

Tabulka 1. Názvy úloh, ve kterých soutěžící sestavují vlastní program pomocí blokového programovacího jazyka jsou v tabulce zvýrazněny tučně.

Tabulka 1: Subjektivně vnímaná obtížnost soutěžních úloh podle respondentů

Název úlohy	Subjektivní vnímání obtížnosti úloh	
	Soutěžící, kteří se s programováním setkali	Soutěžící, kteří se s programováním nesetkali
Seznamy	2,58	4,17
Dveře a klíče	2,70	3,67
Který program špatně obarví žábu?	3,45	4,55
Kříž 2 (mikrosvět želva)	3,79	4,93
Myšová hra	3,82	4,58
Fotorámeček (mikrosvět želva)	4,46	5,20
Dvě rychlá auta (mikrosvět film)	4,76	5,47

Z tabulky vyplývá, že nejobtížnější úlohou pro obě skupiny byla úloha Dvě rychlá auta. Tato úloha byla úlohou programovací z mikrosvěta Film. Soutěžící v této úloze sestavovali vlastní program pomocí blokové programovacího jazyka. Nejjednodušší úlohou pro skupinu respondentů, kteří se s programováním setkali, byla úloha Seznamy. Jde o algoritmickou úlohu, kde soutěžící píše odpověď do textového pole. Pro skupinu respondentů, kteří se s programováním nesetkali, byla nejjednodušší úloha s názvem Dveře a klíče. Tato úloha byla úlohou algoritmickou, ve které soutěžící vybírali správnou odpověď ze čtyř možností.

Ověření hypotézy

Z tabulky dále vyplývá, že programovací úlohy jsou jednodušší pro respondenty, kteří se s programováním setkali, než pro respondenty, kteří se s programováním nesetkali.

Toto tvrzení bylo nutné ověřit vhodnou statistickou metodou. Z dostupných statistických metod byl vybrán dvouvýběrový jednostranný t-test.

Byla formulována nulová hypotéza H_0 a hypotéza alternativní H_1 . Hypotézy byly otestovány pomocí statistického programu R.

H_0 : Programovací úlohy jsou pro obě skupiny soutěžících stejně obtížné ($\mu_1 = \mu_2$).

H_1 : Programovací úlohy jsou jednodušší pro soutěžící, kteří se s programováním setkali ($\mu_1 < \mu_2$).

μ_1 μ_2 označuje střední hodnoty subjektivně vnímané obtížnosti programovacích úloh u jednotlivých skupin respondentů.

Před ověřením hypotézy bylo nutné ověřit normální rozdělení dat. Pro ověření normálního rozdělení dat byl použit Shapiro-Wilkův test na hladině významnosti $\alpha = 5\%$. Vypočtená p-hodnota pro skupinu respondentů, kteří se s programováním setkali byla 0,63. Pro skupinu respondentů, jež se s programováním nesetkali, byla: 0,96. Obě vypočtené p-hodnoty jsou vyšší než hladina významnosti α . Normální rozdělení dat tedy nelze zamítnout.

Dále bylo nutné otestovat rovnost rozptylů obou veličin. K otestování shodnosti rozptylů byl použit f-test. Byly stanoveny nulová hypotéza o rovnosti rozptylů $H_0: \sigma_1^2 = \sigma_2^2$ a hypotéza alternativní $H_1: \sigma_1^2 \neq \sigma_2^2$ o nerovnosti rozptylů. Hladina významnosti α byla stanovena na 5%. Vypočtená p-hodnota f-testu byla 0,50. Z výsledku testu lze usuzovat, že rovnost rozptylů nelze zamítnout.

Po ověření předpokladů t-testu byl provedený samotný test. Hladina významnosti α byla stanovena na 5%. Vypočtená p-hodnota testu byla 0,02. Tato hodnota je nižší než hladina významnosti α . Lze tedy zamítnout nulovou hypotézu H_0 a přijmout hypotézu alternativní H_1 , která zní: **programovací úlohy jsou jednodušší pro soutěžící, kteří se s programováním setkali.**

6 Diskuse výsledků

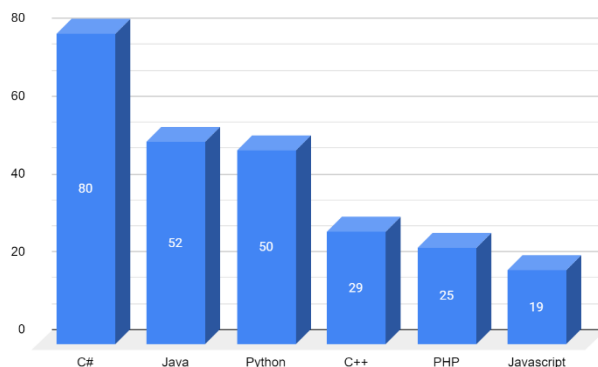
Subjektivní vnímání obtížnosti soutěžních úloh může být ovlivněno řadou faktorů. Jedním takovým faktorem může být míra interaktivity úlohy.

Nejobtížnější úloha pro obě skupiny respondentů (Dvě rychlá auta) je vysoce interaktivní. Soutěžící si může své řešení průběžně ověřovat a program ladit. Tvůrci soutěže tuto úlohu označili jako středně těžkou. V této úloze mohlo být pro soutěžící obtížné uvědomit si, že se auto pohybuje uhlupříčně a obě souřadnice se musejí měnit stejně. Pro soutěžící mohla být obtížná i samotná práce se souřadnicemi.

Úloha Seznamy, která byla nejjednodušší pro respondenty, kteří se s programováním setkali, není interaktivní. Soutěžící píše svou odpověď do textového pole. Autoři soutěže tuto úlohu označili jako těžkou. Pro respondenty, kteří s programováním setkali, může být jednoduchá, protože umějí pracovat s datovými strukturami.

Dvěře a klíče, úloha, která byla nejjednodušší podle respondentů, kteří se s programováním nesetkali, byla autory soutěže označena jako lehká. Úloha není interaktivní, soutěžící vybírá správnou odpověď ze čtyř možností. Cílem úlohy je nalézt algoritmus pro efektivní průchod bludištěm. Správné řešení se dá logicky odvodit a vypočítat.

U skupiny respondentů, kteří se s programováním setkali, může subjektivní vnímání obtížnosti úloh ovlivnit i programovací jazyk, ve kterém se učili programovat. Obrázek 3 zobrazuje graf, na kterém jsou zobrazeny aktuálně vyučované programovací jazyky. Někteří respondenti uvedli více než jeden programovací jazyk.

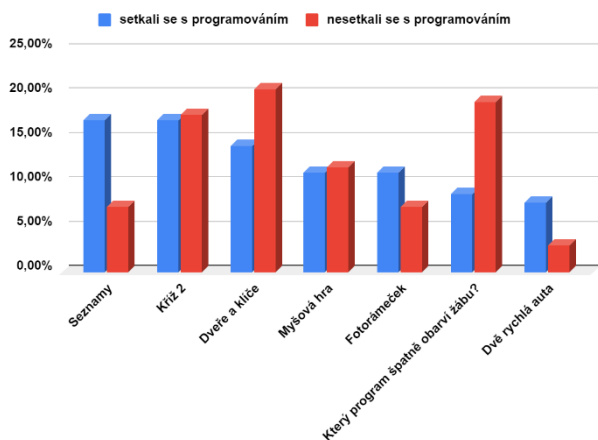


Obrázek 3: Aktuálně vyučované programovací jazyky na středních školách

Z grafu je patrné, že převládají jazyky, které nejsou blokové. Programovací jazyk Scratch zmínili pouze čtyři respondenti. Řada respondentů uvedla jako programovací jazyky HTML a CSS. Nejedná se však o jazyky programovací, ale značkovací.

Soutěžící, kteří se nikdy nesetkali s blokovým programovacím jazykem a jsou zvyklí na vyšší programovací jazyky typu C#, mohou mít s programovacími úlohami problém, protože nejsou zvyklí tvořit program z bloků.

Zajímavým ukazatelem je i obliba soutěžních úloh podle respondentů. Součástí dotazníku byla otázka, ve které měli respondenti označit dvě úlohy, které se jim nejvíce líbily. Oblibu soutěžních úloh u jednotlivých skupin zobrazuje následující graf (Obrázek 4).



Obrázek 4: Obliba soutěžních úloh podle respondentů

Úloha Dvě rychlá auta (mikrosvět film), která byla pro obě skupiny nejobtížnější, je zároveň i nejméně oblíbená u obou skupin. Úloha Kříž 2 zasazená do mikrosvěta Želva, je hodnocena jako druhá nejoblíbenější pro obě skupiny s minimálním rozdílem.

Největší rozdíl v oblíbě úlohy mezi jednotlivými skupinami můžeme pozorovat u úlohy Který program špatně obarví žábu? Pro respondenty, kteří se s programováním nesetkali, jde o druhou nejoblíbenější úlohu. Pro respondenty, kteří se s programováním setkali, je to druhá nejméně oblíbená úloha. V této úloze soutěžící nastavovali kód z bloků, ale vybírali jednu ze čtyř možností. Úloha

byla zařazena mezi programovací úlohy, protože soutěžící při jejím řešení pracovali s podmínkami. Podmínky byly zapsány slovně. Z tohoto důvodu byla úloha pravděpodobně oblíbenější pro respondenty, kteří se s programováním nesetkali. Úloha se dala vyřešit logickou úvahou.

7 Závěr

Programovací úlohy jsou jednodušší pro soutěžící, kteří se s programováním setkali při výuce informatiky. Tvrzení, které je předvídatelné, nyní je ovšem i ověřené pomocí statistické metody. V budoucnu by bylo vhodné provést obdobný výzkum na více soutěžních kategoriích a sledovat, jestli toto tvrzení platí i pro soutěžící ze základních škol.

BIBLIOGRAFICKÉ ODKAZY

- [1] VANÍČEK, Jiří. Rozvoj informatických kompetencí žáků pomocí soutěže: kauza Bobřík informatiky. Sborník konference Počítač ve škole [online]. Nové Město na Moravě: Gymnázium Vincence Makovského, 2009, 2009 [cit. 2022-11-20]. Dostupné z: <https://www.bebas.org/sites/default/files/documents/publications/Vanicek%20-2009.pdf>
- [2] Obecné informace. In: Bobřík informatiky: Informatická soutěž pro žáky základních a středních škol [online]. České Budějovice: Katedra informatiky PF JČU, c2008-2022 [cit. 2022-12-03]. Dostupné z: <https://www.ibobr.cz/o-soutezi/obecne-informace>
- [3] Soutěžní otázky. In: Bobřík informatiky: Informatická soutěž pro žáky základních a středních škol [online]. České Budějovice: Katedra informatiky PF JČU, c2008-2022 [cit. 2022-11-20]. Dostupné z: <https://www.ibobr.cz/o-soutezi/soutezni-otazky>
- [4] Rámcový vzdělávací program pro gymnázia: RVP G. In: . Praha: MŠMT, 2021, ročník 2021, MSMT-24248/2021-2. Dostupné také z: <https://www.edu.cz/rvp-ramcove-vzdelavaci-programy/ramcove-vzdelavaci-programy-pro-gymnazia-rvp-g/>
- [5] VANÍČEK, Jiří, Václav ŠIMANDL a Václav DOBIÁŠ. Bebras Tasks Based on Assembling Programming Code. In: BOLLIN, Andreas a Gerald FUTSCHEK. *Informatics in Schools: A Step Beyond Digital Education*. Vienna: Springer, 2022, s. 113-124. ISBN 978-3-031-15851-3. ISSN 1611-3349.
- [6] HENDL, Jan. Základy matematiky, logiky a statistiky pro sociologii a ostatní společenské vědy v příkladech. Druhé, rozšířené vydání. Praha: Univerzita Karlova, nakladatelství Karolinum, 2021. ISBN 978-80-246-4869-9.
- [7] Statistika 2022. In: Bobřík informatiky: Informatická soutěž pro žáky základních a středních škol [online]. České Budějovice: Katedra informatiky PF JČU, c2008-2022 [cit. 2022-12-27]. Dostupné z: <https://www.ibobr.cz/vysledky-a-statistiky/statistiky>

Bezpečnosť v digitálnom veku – koncepcia vzdelávania na ZŠ a SŠ

Safety in the digital age – a conception of education in primary and secondary schools

PaedDr. Roman Hruščeký, PhD.
Katedra didaktiky matematiky, fyziky
a informatiky, FMFI UK
Mlynská dolina
842 48 Bratislava
Slovensko
roman.hrusecky@fmph.uniba.sk

prof. RNDr. Ivan Kalaš, PhD.
Katedra didaktiky matematiky, fyziky
a informatiky, FMFI UK
Mlynská dolina
842 48 Bratislava
Slovensko
ivan.kalas@fmph.uniba.sk

Mgr. Klára Dvorská
Slovensko
klar.dvorska@gmail.com

ABSTRACT

In this paper we describe the created conception of education in primary and secondary schools for the area of safety in the digital world. We present our own analysis of some foreign curriculum and educational programs, as well as an analysis of the current curriculum in the subject of computer science, as well as in other subjects. We have created a map of different aspects of security in the digital age, with four perspectives at its core - Society, People, Data, Technology. The conception includes recommendations for changes and measures that the Department of Education should urgently implement.

Keywords

Digital security. Conception. Curriculum.

ABSTRAKT

V článku popisujeme vytvorenú koncepciu vzdelávania na ZŠ a SŠ pre oblasť bezpečnosti v digitálnom svete. Uvádame vlastné analýzy niektorých zahraničných kurikul a vzdelávacích programov, a tiež analýzu súčasného ŠVP v predmete informatika, ale aj v iných predmetoch. Vytvorili sme mapu rôznych aspektov bezpečnosti v digitálnom veku, ktorej jadrom sú štyri pohľady – Spoločnosť, Človek, Dáta, Technológia. Súčasťou koncepcie sú odporúčania zmien a opatrení, ktoré by mal rezort školstva naliehavo realizovať.

Kľúčové slová

Digitálna bezpečnosť. Koncepcia. Kurikulum.

1 ÚVOD

Používanie digitálnych technológií mení náš život – naše vzdelávanie, prácu i zábavu. Prináša ale celkom nové riziká, na ktoré musíme nevyhnutne reagovať.

Ak chceme využívať výhody digitálnych technológií – ale v skutočnosti to už vlastne inak ani nejde – musíme poznávať aj s nimi spojené riziká a reagovať na ne. Zrejme najúčinnjším nástrojom prevencie nežiaducich dôsledkov života v digitálnom veku (ktoré zahŕňajú aj počítačovú kriminalitu) je kvalitné, komplexné a efektívne vzdelávanie, a to od začiatku povinnej školskej dochádzky (resp. už od materskej školy) až po jej ukončenie.

V spolupráci s Indícia, n.o. sme vytvorili koncepciu vzdelávania na ZŠ a SŠ pre oblasť bezpečnosti. Táto koncepcia bola súčasťou projektu **Prevencia kriminality v digitálnom veku**, ktorý bol

realizovaný s podporou Odboru prevencie kriminality, Ministerstva vnútra SR.

Jedným z východísk našej koncepcie boli vlastné analýzy zahraničných kurikul a vzdelávacích programov z krajín, ktoré ich nedávno reformovali. Ďalej sme realizovali niekoľko rozhovorov s významnými učiteľmi na Slovensku. Uvedomujeme si, že problematika bezpečnosti sa neustále vyvíja a bude potrebné koncepciu revidovať. Ale snažili sme sa ju vytvoriť tak, aby v nej boli zahrnuté aj oblasti a hrozby, ktoré ešte nepoznáme. Z týchto zdrojov sme vytvorili tabuľku rôznych tém v oblasti digitálnej bezpečnosti, ktoré sme sa následne snažili identifikovať v ŠVP v predmete informatika a aj ostatných predmetoch.

Na základe všetkých získaných a analyzovaných dát sme vytvorili mapu rôznych aspektov **Bezpečnosť v digitálnom veku**. Jadrom sú štyri pohľady na bezpečnosť, ktoré sú ďalej rozvinuté do druhej aj tretej úrovne. Táto podrobná mapa by mohla pomôcť didaktikom **rôznych predmetov** pri tvorbe budúcej verzie ŠVP. Z jednej strany posúdiť, či okruh, či téma ŠVP v danom predmete prispieva k rozvoju poznania žiakov v niektorej oblasti mapy bezpečnosti, alebo z druhej strany, či naopak niektorá oblasť bezpečnosti sa nedá zmysluplne a efektívne implementovať v nejakom tematickom okruhu daného predmetu.

Pripravili sme tiež sériu odporúčaní, ktoré by mal štát a predovšetkým rezort školstva implementovať.

Na Slovensku existuje viacero iniciatív zo strany rôznych občianskych združení a neziskových či súkromných organizácií v oblasti bezpečnosti detí a mladých ľudí v digitálnom veku, napr. zodpovedne.sk, kybersikanovanie.sk, ovce.sk, nezavislost.sk, stopline.sk, nehejtuj.sk, preventista.sk a ďalšie. Tieto iniciatívy sú mimoriadne úspešné, významné a pre inovatívne školy predstavujú výborný zdroj materiálov, námetov na aktivity, kreslených filmov a videí a pod. Samozrejme, trpia tým, že ich financovanie závisí na rôznych projektoch, resp. na partneroch z súkromného a neziskového sveta, a len výnimočne sú spolufinancované štátom. Čo však považujeme za najväčší problém týchto aktivít: skoro nikdy ich nepodporuje trvalá reakcia zo strany štátu, napr. inovácia vzdelávacích programov, ktorá by predmety zaviazala k tomu, aby témy bezpečnosti integrovali do svojich obsahov.

Práve tento nedostatok chceme adresovať našou koncepciou. Považujeme za nevyhnutné, aby aj u nás rezort školstva venoval tejto dramaticky sa vyvíjajúcej situácii podstatne väčšiu pozornosť.

2 Ciele koncepcie vzdelávania v oblasti bezpečnosti

V našej koncepcii sa zameriavame výlučne na preskúmanie a pomenovanie problému bezpečnosti žiakov a mládeže v digitálnom svete a rámcový návrh koncepcie, ako má na tento fenomén reagovať formálne vzdelávanie, a to od úrovne materských škôl, cez celú základnú školu, až po tretí stupeň všeobecného a odborného vzdelávania – na informatike a na ďalších predmetoch a v prierezových témach.

Za ciele tejto koncepcie sme si stanovili ([1]):

- upriamiť pozornosť odbornej verejnosti na problematiku bezpečnosti v digitálnom veku, poukázať na najohrozenejšie vekové skupiny, teda deti a mládež, a sprostredkované aj seniori,
- vzhľadom na všeobecne známu efektívnosť preventívnych prístupov pri riešení väčšiny spoločenských problémov zamerať sa – celkom prirodzene – na sektor vzdelávania, a to formálneho vzdelávania, v období predškolského a povinného školského vzdelávania žiakov,
- v kontexte vzdelávania preskúmať, ako na tento fenomén súčasného života reagujú rôzne zahraničné vzdelávacie systémy,
- preskúmať tiež, ako na fenomén bezpečnosti reaguje náš inovovaný Štátny vzdelávací program, a to (a) v predmete informatika, (b) v ďalších predmetoch, a napokon (c) z pohľadu rôznych prierezových tém,
- realizovať pilotný prieskum medzi niekoľkými učiteľmi informatiky s cieľom spoznať ich pohľad na to, aké oblasti bezpečnosti v kontexte vzdelávania považujú za dôležité a ako ich oni sami implementujú do vzdelávacieho procesu,
- zo všetkých takto získaných vstupov vytvoriť konsolidovanú mapu rôznych pohľadov na bezpečnosť v digitálnom veku a jej aspektov a tém, a to v prostredí formálneho vzdelávania,
- navrhnúť prvú (východiskovú) iteráciu toho, akú úlohu by v implementácii týchto pohľadov na bezpečnosť a jej rôznych tém mala hrať školská informatika, a ktoré z týchto tém by sa mali premietnuť do vzdelávacích obsahov iných predmetov, resp. by sa mali produktívne implementovať ako prierezové témy. V tejto analýze budeme venovať špeciálnu pozornosť aj argumentácii, prečo a ako sa otázky bezpečnosti v digitálnom veku majú realizovať ako nástroj prevencie už v materskej škole a v primárnom vzdelávaní,
- sformulovať sériu odporúčaní ako východisko pre udržateľnú a efektívnu implementáciu rôznych komponentov bezpečnosti do formálneho vzdelávania,
- medzi tieto odporúčania zahrnúť aj také, ktoré majú potenciál sekundárneho dopadu na postoje a správanie rodičov a starých rodičov,
- poukázať na riziká, ktoré môžu ohroziť implementáciu nami navrhovaných zmien vo vzdelávaní zameraných na zvýšenie bezpečnosti mladých ľudí v digitálnom veku,
- stručne prezentovať súbor vybraných aktivít z rôznych krajín sveta, ktoré sa týkajú tém bezpečnosti žiakov v digitálnom veku.

3 Téma bezpečnosti vo vzdelávacích systémoch

Jedným zo vstupov bolo získať prehľad o tom, ako na tému bezpečnosti žiakov v digitálnom veku reagujú niektoré vzdelávacie systémy v zahraničí, a tiež súčasný ŠVP (Štátny vzdelávací program) v predmete informatika, ale aj v iných predmetoch.

Zo zahraničných zdrojov sme sa venovali národnému kurikulu Anglicka, Austrálie a Nového Zélandu, ďalej spoločnej iniciatíve niekoľkých amerických inštitúcií (ACM, CSTA, code.org,...), významných pri formovaní vzdelávacích politik jednotlivých štátov v USA, a napokon reforme Rámcového vzdelávacieho programu v Českej republike.

Výstupom našej analýzy uvedených zahraničných zdrojov je tabuľka (Obrázok 1), ktorá stručne pomenúva oblasti spomínaných vzdelávacích obsahov a stratégií, s ohľadom na vzdelávacie stupne.

AnalYZovali sme tiež inovovaný Štátny vzdelávací program od prvého stupňa ZŠ až po koniec gymnázia z pohľadu tém týkajúcich sa bezpečnosti žiakov v digitálnom prostredí – najprv v informatike, potom aj v ďalších predmetoch.

AnalYZovali sme viaceré predmety (matematika, etická výchova, náboženská výchova, prvouka, prírodoveda, vlastiveda, občianska náuka, pracovné vyučovanie, technika, fyzika), a taktiež niektoré prierezové témy (výchova k manželstvu a rodičovstvu, mediálna výchova). Zameriavali sme sa na predmety a témy, ktoré by mohli súvisieť s bezpečnosťou žiakov v kontexte digitálnych technológií.

Témy súvisiace s bezpečnosťou žiakov v digitálnom veku sme priamo našli len v predmete etická výchova (4. a 9. ročník ZŠ) [*rozlíšiť na príkladoch pozitívne i negatívne vplyvy televízie a iných komunikačných prostriedkov; diskutovať o výhodách a nevýhodách používania internetu*]. V žiadnom z ďalších predmetov, ktoré sme analYZovali, sa témy priamo súvisiace s bezpečnosťou neuvádzajú. Možno ich však niektorí učitelia preberajú v rámci tém, ktoré majú všeobecnejší charakter (napr. šikanovanie alebo výhody a nevýhody digitálnych technológií).

Téma šikanovania sa spomína v predmete *občianska náuka* pre 7. ročník základných škôl a pre gymnázia, aj keď z textu vzdelávacieho programu nie je zrejmé, či má byť súčasťou tejto témy aj kyberšikanovanie.

Pri analýze inovovaného Štátneho vzdelávacieho programu pre 4-ročné gymnázia môžeme v *etickej výchove* a *náboženskej výchove* nájsť témy médií, pornografia, drogy, gamblerstvo, avšak šikanovanie (v bežnom i virtuálnom svete), ani bezpečnosť v digitálnom svete sa nespomína.

To, že niektoré témy súvisiace s bezpečnosťou v rámci iných predmetov neuvádzame, neznamená, že sa v nich neučia. Znamená to len toľko, že nie sú explicitne uvedené v našom inovovanom Štátnom vzdelávacom programe.

ročník	Veľká Británia	Austrália	Nový Zéland	Česká Republika	K12 ACM, ISTE et al.
1	používať DT bezpečne a s rešpektom; chrániť si osobné informácie a súkromie	aplikovať bezpečné a etické praktiky, aby sme chránili seba a ostatných pri online interakcii pri učení sa a komunikácii		poznať pozitívne a negatívne vplyvy DT, bezpečnosť seba a ostatných; vedieť povedať nie	poznať organizáciu siete, používať silné heslá, vedieť o zbieraní a ukladaní dát, používať online komunikáciu, zdieľať informácie, dávať pozor na bezpečnosť v digitálnom priestore
2					
3		pochopiť, prečo je dôležité pri zdieľaní názorov a komunikácii v online priestore zvažovať emócie ich publika a správať sa s rešpektom	vytvárať, používať, ukladať, vyhľadávať a zdieľať digitálny obsah s cieľom zvládnuť technologické výzvy; poznať digitálne prístroje a ich účel	poznať vlastnú identitu; pracovať s rešpektom, dodržiavať pravidlá používania DT; rozpoznať a nahlásiť nevhodný obsah; chrániť zdravie, rešpektovať autorské práva a pravidlá slušného správania; kriticky zvažovať spoľahlivosť zdrojov	
4	rozumieť počítačovým sieťam; používať DT bezpečne, s rešpektom a zodpovedne; rozpoznať akceptovateľné a neakceptovateľné správanie; poznať rôzne spôsoby, ako informovať o nevhodnom obsahu a kontakte	pri online spolupráci si chrániť svoje zdravie; dodržiavať bezpečnostné a etické pravidlá, týkajúce sa spoločenských rozdielov či súkromia osobných informácií; bezpečne ukladať dáta	rozumieť DT a ich vplyvu na človeka a spoločnosť i tomu, že aj nástroje a ich vplyv sa časom mení a vyvíja		poznať detailnejší popis architektúry siete; vedieť, čo sú vírusy, ako ochraňovať informácie a dáta a ako neporušovať autorské práva
5					
6				rozpoznať rôzne úrovne oprávnenia pri zdieľaní dát, vlastného a cudzieho obsahu; chrániť identitu, reputáciu a zdravie; komunikovať eticky a s rešpektom	poznať rozdiel medzi HTTP a HTTPS; poznať riziká online pirátstva a obsahu na internete; osobné dáta a informácie udržiavať v bezpečí
7	chrániť si vlastnú identitu a súkromie; rozpoznať nevhodný obsah, kontakt a správanie a vedieť, ako o ňom informovať	pri online komunikácii a spolupráci rozumieť rozdielnemu spoločenskému kontextu a právnej zodpovednosti	rozumieť úlohe operačného systému v riadení DT a zaistení bezpečnosti; rozumieť, že pri ukladaní dát vzniká aj zodpovednosť za ich bezpečnosť a súkromie		
8					
9		zamýšľať sa nad súkromím a bezpečnosťou pri práci s dátami, ako sa používajú a spravujú; vedieť navrhnúť pravidlá na zabezpečenie informačných systémov; rešpektovať autorské práva		poznať počítačovú sieť, bezpečnosť a vplyv DT na spoločnosť; vedieť o svojej digitálnej stope; ochraňovať osobné údaje; chrániť si zdravie pri práci s počítačom; dodržiavať bezpečnostné zásady; spoľahlivo vyhľadávať informácie	poznať právne normy v oblasti bezpečnosti; zachovávať súkromie; vedieť ukladať dáta do cloudových systémov
10	rozumieť, ako vývoj nových DT ovplyvňuje bezpečnosť, poznať nové spôsoby ochrany bezpečia a identity				
11					
12			v reálnom kontexte vedieť samostatne preskúmať danú problematiku a navrhnúť možné riešenia problémov, ktoré identifikujeme		
13					

Obrázok 1 - Bezpečnosť v digitálnom veku podľa rôznych vzdelávacích systémov

4 Mapa bezpečnosti v digitálnom veku

Zdroje, ktoré sme preskúmali a prezentovali v predchádzajúcej kapitole, sa – prirodzene – navzájom líšia dôrazom, ktorý vo svojich vzdelávacích systémoch kladú na rôzne oblasti bezpečnosti. Preto sme sa snažili vytvoriť určitú syntézu všetkých tém a aspektov.

Tému bezpečnosti v digitálnom veku sme sa napokon rozhodli rozdeliť do štyroch hlavných pohľadov, a to:

- spoločnosť,
- človek,
- dáta a
- technológie.

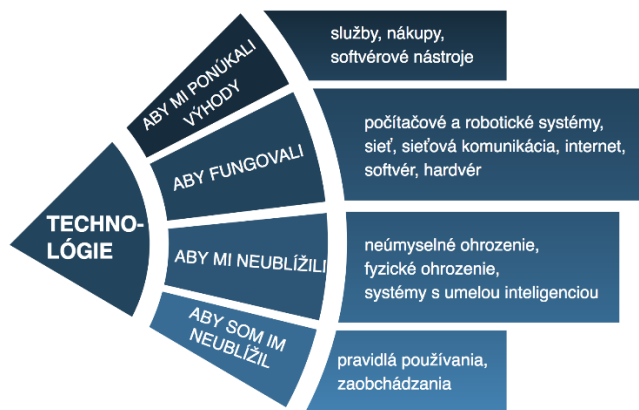
Každý z pohľadov sa na celú oblasť bezpečnosti pozerá z určitého uhla a môže byť prepojený s niektorými ďalšími. V nasledujúcich odsekoch detailnejšie popisujeme jednotlivé pohľady. Spoločná mapa prvej a druhej úrovne pohľadov je na Obrázok 2.



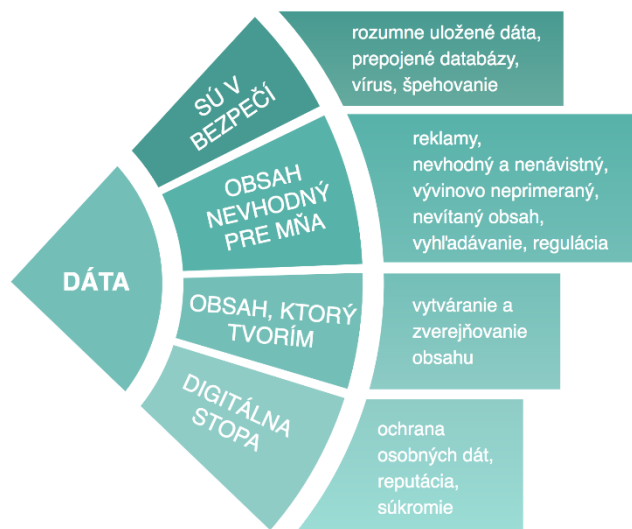
Obrázok 2 - Mapa rôznych pohľadov na otázky bezpečnosti v digitálnom veku

4.1 Technológia

Nosnou kostrou digitálneho sveta sú technológie. Máme záujem, aby nám poskytovali svoje možnosti, aby nám umožňovali využívať rôzne služby (ako elektronický obchod) a softvérové nástroje. Často sme do veľkej miery alebo úplne odkázaní na ich fungovanie (riadenie leteckej dopravy počítačovými systémami, sieťová komunikácia, robotmi realizované chirurgické zákroky, skenery a iné vstupné zariadenia a pod.), preto im potrebujeme na vývinovo primeranej úrovni rozumieť. Chceme ich správne používať, aby sme zachovali ich funkčnosť (deti, ktoré samy používajú v materskej škole digitálny fotoaparát; žiaci, ktorí budujú a programujú robota a pod.). Chceme, aby nám neublížili (bezpečnosť pri práci v počítačovej učebni, bezpečnosť pri používaní projektoru, bezpečnosť v doprave, ktorej súčasťou sú autonómne autá a iné systémy riadené UI a pod.).



Obrázok 3 - Rozvinutý pohľad Technológie



Obrázok 4 - Rozvinutý pohľad Dáta

4.2 Dáta

Život, ktorý je v digitálnom veku intenzívne popretkávaný technológiami, produkuje, vyžaduje a komunikuje veľké objemy dát – niekedy z rôznych dôvodov veľmi citlivých. Neustále za sebou zanechávame digitálnu stopu, ktorá pri nevhodnej manipulácii či nedostatočnom zabezpečení môže ohroziť naše

súkromie, reputáciu a bezpečie. Chceme deti a žiakov chrániť pred nevhodným, nenávisťným, nepravdivým či nevitánym obsahom na internete, chceme, aby oni sami takéto obsah neprodukovali. Chceme, aby naše dáta boli neustále v bezpečí – ale zároveň neustále poruke. Potrebujeme na primeranej úrovni poznať a rozumieť riešeniam, ktoré nám to umožňujú.

4.3 Človek

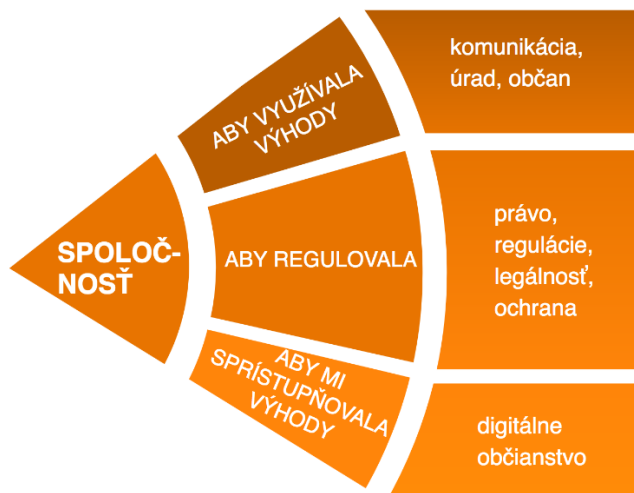
Chceme sa vedieť brániť pred nebezpečným kontaktom a iným nevhodným správaním druhých, potrebujeme vedieť rozpoznať nepravdivý alebo nenávisťný obsah, potrebujeme vedieť, ako sa brániť šikanovaniu v digitálnom prostredí. Sami nesmieme takto postupovať voči iným, musíme rešpektovať ich súkromie. Musíme poznať a rešpektovať pravidlá na reguláciu v digitálnom svete, autorstvo a iné práva a povinnosti. Musíme vedieť, ako správne chrániť svoju vlastnú identitu. Musíme si chrániť vlastné zdravie. Aby sme dokázali efektívne a bezpečne využívať možnosti digitálneho prostredia, uchovať si svoje bezpečie a neohrozovať bezpečie iných, musíme byť digitálne gramotní, mať rozvinuté kritické myslenie a rozvíjať naše porozumenie digitálneho sveta zo všetkých možných pohľadov.



Obrázok 5 - Rozvinutý pohľad Človek

4.4 Spoločnosť

Úlohou spoločnosti je regulovať digitálny priestor legislatívou a inými pravidlami, vytvárať podmienky, za ktorých prekvitá digitálne občianstvo a ktoré spríjemňujú a zjednodušujú život občanom. Legislatívu a pravidlá prijíma a aplikuje spoločnosť preto, aby chránila občanov pred neželanými a nebezpečnými zásahmi prostredníctvom digitálnych technológií. Jednou formou takejto preventívnej ochrany je aj priebežná podpora rôznych mimovládnych aktivít a iniciatív, ktoré si kladú za cieľ zvyšovať povedomie občanov v tomto smere. Za kľúčovú úlohu tu však považujeme prijatie série opatrení a konkrétnych krokov – včítane revízie Štátneho vzdelávacieho programu – ktoré efektívne implementujú vzdelávanie žiakov v oblasti bezpečnosti v digitálnom svete do informatiky a do skoro všetkých ostatných predmetov (avšak nie iba ako ďalšej prierezovej témy popri desiatkach iných).



Obrázok 6 - Rozvinutý pohľad Spoločnosť

5 Záver

V našej koncepcii sme vyjadrili názor, že najlepšou formou prevencie proti rôznym rizikám života žiakov v digitálnom veku je kvalitné, komplexné a efektívne vzdelávanie. Navrhli sme niekoľko zmien a opatrení, ktoré by mal rezort školstva naliehavo realizovať. Navrhujeme:

- v rámci rezortu školstva vytvoriť permanentné inštitucionálne a odborné personálne zabezpečenie efektívnej implementácie problematiky bezpečnosti detí a žiakov v digitálnom veku vo formálnom vzdelávaní,
- revidovať súčasný ŠVP a identifikovať v ňom témy a okruhy, ktoré sú pri dnešnej európskej a slovenskej legislatíve rizikové alebo neakceptovateľné (napr. regulácia GDPR významne obmedzuje možnosti vytvárať školské e-mailové účty pre mladších žiakov a pod., jej výklad v tomto kontexte je školám nejasný),
- upraviť vzdelávacie programy predmetov tak, aby sa aktuálne, efektívne a komplexne venovali téme bezpečnosti žiakov v digitálnom veku, a to už od úrovne predprimárneho vzdelávania,
- prostredníctvom procesu akreditácie študijných programov týmto zmenám prispôsobiť aj prípravu budúcich učiteľov. Nie je mysliteľné, aby sa rezort školstva rôznymi krokmi snažil o zlepšenie stavu pripravenosti žiakov v oblasti bezpečnosti priamo na školách, ale akceptoval, že fakulty, ktoré vychovávajú budúcich učiteľov, sa tejto téme naďalej nevenujú. Rovnakú chybu robí štát aj v oblasti rozvoja digitálnej gramotnosti – žiakov a ich učiteľov,
- pripraviť, odborne garantovať a kontinuálne poskytovať programy ďalšieho vzdelávania učiteľov z praxe na tému bezpečnosti a jej implementácie na všetky stupne formálneho vzdelávania. S tým úzko súvisí nasledujúci bod:
- zaistiť, aby vznikli série moderných metodických materiálov, ktoré pomôžu učiteľom implementovať tieto nové témy do vyučovania. Ak sa štát nepostará o kontinuálnu podporu učiteľov zo škôl v tejto problematike, nepodarí sa zvýšiť pripravenosť žiakov na život a prácu v digitálnom veku,

- kontinuálne podporovať rôzne kampane, súťaže a iniciatívy (zrejme najmä cez neziskové organizácie), ktoré efektívnym spôsobom zvyšujú úroveň vzdelávania žiakov a ich učiteľov v oblasti bezpečnosti a digitálnej gramotnosti, napr. rôzne vyššie spomínané portály, súťaž iBobor a pod. Dať dôraz na také kampane, ktoré môžu sprostredkované ovplyvniť aj postoje a správanie rodičov, resp. starých rodičov žiakov,
- podporovať, resp. iniciovať kampane, ktoré by zvyšovali povedomie verejnosti o vážnosti problematiky bezpečnosti žiakov, seniorov a všetkých ľudí, teda nielen v škole, ale i v spoločnosti.

POĎAKOVANIE

Táto práca vznikla za podpory Indícia, n.o., dotácie z Ministerstva vnútra SR, projektu VEGA 1/0621/22 ako aj z APVV projektu APVV-20-0353.

BIBLIOGRAFICKÉ ODKAZY

- [1] Indícia, n.o. *Prevenencia kriminality v digitálnom veku*. [online] <https://www.indicia.sk/assets/Uploads/ebook-prevencia-final.pdf>
- [2] Australian curriculum, assessment and reporting authority (2015). *Digital Technologies: Sequence of content F-10*. [online]. [cit. 12. 3. 2019] http://docs.acara.edu.au/resources/Digital_Technologies_-_Sequence_of_content.pdf
- [3] Department for Education (2013). *Computing programmes of study: key stages 1 and 2 framework documents*. [online]. [cit. 12.3.2019] https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239033/PRIMARY_national_curriculum_-_Computing.pdf
- [4] Department for Education (2013). *Computing programmes of study: key stages 3 and 4 framework documents*. [online]. [cit. 12.3.2019] https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239067/SECONDARY_national_curriculum_-_Computing.pdf
- [5] K12CS. 2016. *K–12 Computer Science Framework*. [online]. ACM, Code.org, CSTA, CIC, National Math and Science Initiative 2016. <http://www.k12cs.org>
- [6] Ministry of Education (2017). *Technology in the New Zealand Curriculum*. [online]. [cit. 12.3.2019] <http://nzcurriculum.tki.org.nz/content/download/167461/1235900/file/Technology%20in%20the%20New%20Zealand%20Curriculum%202017.pdf>
- [7] Štátny pedagogický ústav (2014). *Inovovaný štátny vzdelávací program*. [online]. <http://www.statpedu.sk/sk/svp/inovovany-statny-vzdelavaci-program/>

Hodnotenie žiakov v edukačnej robotike – výsledky prvej etapy výskumu

Assessment of pupils in educational robotics - results of the first phase of research

Mgr. Jakub Krcho

Fakulta matematiky, fyziky a informatiky
Univerzita Komenského v Bratislave

Mlynská dolina F1
842 48 Bratislava
Slovenská republika

krcho@fmph.uniba.sk

Mgr. Karolína Miková, PhD.

Fakulta matematiky, fyziky a informatiky
Univerzita Komenského v Bratislave

Mlynská dolina F1
842 48 Bratislava
Slovenská republika

mikova@fmph.uniba.sk

ABSTRACT

Educational robotics has been finding a place in the classroom for years, and we try to implement it as much as possible. The goal of these efforts is to develop students' computer science skills and other competencies such as manipulation, communication, collaboration, critical thinking, and problem-solving. To make the most of the educational potential of these robotic kits and robots, the teachers must set appropriate specific learning objectives for the lesson that they will want to achieve with their students. But how can we (as teachers) know if all the objectives we have created have been achieved by the pupils? We can monitor the extent to which these objectives have been met, for example, through assessment. With educational robotics being a relatively young part of the teaching process, are the assessment methods we have developed sufficient for educational robotics? In this paper, we present the results of the first phase of our dissertation research, which takes up the topic of learner assessment in educational robotics.

Keywords

Educational Robotics. Assessment. Pupils. Learning Objectives.

ABSTRAKT

Edukačná robotika si roky hľadá miesto vo vyučovacom procese a snažíme sa ju v čo najväčšej možnej miere implementovať. Cieľom týchto snáh je rozvíjať zručnosti žiakov v oblasti informatiky a ďalších kompetencií, ako je manipulácia, komunikácia, spolupráca, kritické myslenie, či riešenie problémov. Aby sa v čo najväčšej miere využil edukačný potenciál týchto robotických stavebníc a robotov, je potrebné, aby vyučujúci stanovil vhodné špecifické vzdelávacie ciele hodiny, ktoré bude chcieť so svojimi žiakmi dosiahnuť. Ale ako môžeme (my učitelia) zistiť, či všetky nami vytvorené ciele žiaci dosiahli? Mieru naplnenia týchto cieľov môžeme sledovať napríklad pomocou hodnotenia. Tým, že je edukačná robotika relatívne mladou súčasťou vyučovacieho procesu, sú vytvorené metódy hodnotenia dostatočné aj pre edukačnú robotiku? V rámci tohto článku prinášame výsledky prvej etapy nášho dizertačného výskumu, ktorý sa zoberá témou hodnotenia žiakov v edukačnej robotike.

Kľúčové slová

Edukačná Robotika. Hodnotenie. Žiaci. Vzdelávacie Ciele.

1 ÚVOD

Robotika si roky hľadá a nachádza svoje miesto vo vyučovacom procese na všetkých stupňoch vzdelávania. V rámci hodín informatiky na slovenských školách sa učitelia čoraz viac snažia

implementovať edukačnú robotiku do svojho vyučovacieho procesu [1,2]. Cieľom týchto snáh je rozvíjať zručnosti žiakov v oblasti informatiky a ďalších kompetencií, ako je komunikácia, spolupráca, kritické myslenie, či riešenie problémov [3,4,5]. Pod pojmom edukačná robotika rozumieme využívanie robotov, či robotických stavebníc na vyučovaní s cieľom naplniť výchovno-vzdelávacie ciele.

Aby sa v čo najväčšej miere využil edukačný potenciál týchto robotických stavebníc a robotov, je potrebné, aby vyučujúci stanovil vhodné špecifické vzdelávacie ciele hodiny, ktoré bude chcieť so svojimi žiakmi dosiahnuť [6]. Avšak na to, aby učiteľ zistil, či a do akej miery naplnil stanovené vzdelávacie ciele, potrebuje svojich žiakov hodnotiť [7]. Nemáme pritom na mysli klasické udeľovanie známok. Preto sa vynára otázka – „Ako hodnotiť napredovanie žiakov pri práci s robotickými stavebnicami počas vyučovacieho procesu?“

Každý učiteľ má svoje stratégie, ktoré používa na hodnotenie svojich žiakov. Sú tieto stratégie vhodné aj pre edukačnú robotiku? Keď chceme pracovať s robotmi na vyučovaní, musíme pamätať na rôzne aspekty, ktoré v sebe edukačná robotika zjednocuje. Aspekt konštrukcie – žiaci musia postaviť robotický model, ktorý je schopný svojou konštrukciou vyriešiť zadanú úlohu. V čom sa skrýva aj istá miera dizajnu. Aspekt programovania – žiaci musia naprogramovať správanie robotického modelu, aby dokázal svojim programom vyriešiť zadanú úlohu. Teda robotika nie je len o programovaní alebo konštruovaní. Je to kombinácia týchto dvoch hlavných aspektov a jeden aspekt výrazne ovplyvňuje ten druhý a naopak.

Robotike a edukačnej robotike sa venujeme už viac, ako desiatku rokov. Z vlastnej skúsenosti vieme teda povedať, že používanie robotických stavebníc vo vzdelávaní umožňuje žiakom efektívne rozvíjať veľké množstvo kompetencií a zručností. Hravým a zábavným spôsobom sa žiaci naučia programovať, porozumejú jednotlivým programátorským konceptom, či si zlepšia kvalitu jemnej motoriky (konštruovanie robota). Všetky naše skúsenosti pochádzajú z nášho aktívneho učenia prostredníctvom robotických stavebníc na všetkých stupňoch povinného i voľnočasového vzdelávania.

V rámci predošlého výskumu [8] sme sa venovali téme taxonómii pre stanovenie vzdelávacích cieľov v oblasti edukačnej robotiky. V tejto téme chceme pokračovať a to využitím v rámci hodnotenia, resp. overovania získaných vedomostí študentmi.

2 Edukačná robotika a hodnotenie

Edukačná robotika (používanie robotov a robotických stavebníc na vyučovaní) pokrýva niekoľko kľúčových oblastí, v ktorých možno rozvíjať rôzne kompetencie žiakov. Často vedie učiteľov k využívaniu netradičných vyučovacích metód, ktoré prispievajú k budovaniu kompetencií užitočných pre život - napr. tímová práca, riešenie problémov a kreativita [9], kreativita a spolupráca [3], komunikácia a riešenie problémov [4,5], či dokonca technické a sociálne zručnosti [10]. Edukačná robotika je vhodný pedagogický nástroj, vďaka ktorému môže učiteľ realizovať vzdelávanie atraktívnym spôsobom [11]. Štúdie ukazujú [12,13], že roboty a robotické stavebnice motivujú študentov k intenzívnejšej práci a štúdiu. Zaradenie robotiky do výučby má pozitívny vplyv na zručnosti mladých študentov, o čom svedčí rastúci počet štúdií skúmajúcich vplyv edukačnej robotiky [14]. Okrem motivujúceho charakteru učenia, edukačná robotika dáva študentom príležitosť získať značné množstvo kompetencií a zručností.

Pri integrácii edukačnej robotiky do bežného vzdelávania by učitelia mali zvážiť, ktoré konkrétne vzdelávacie ciele budú posudzovať. Akonáhle má učiteľ záujem implementovať edukačnú robotiku do svojho vyučovacieho procesu a záujem poskytnúť svojim žiakom možnosť učiť sa prostredníctvom robotov a robotických stavebníc, musí špecifikovať, aké vzdelávacie ciele chce dosiahnuť prostredníctvom edukačnej robotiky [15,16].

Akákolvek ľudská činnosť, ktorá je vykonávaná s určitým zámerom, cieľom, či účelom, úzko súvisí s prirodzenou potrebou hodnotenia. Taktiež vzdelávanie chápeme ako jednu z ľudských činností smerujúcu k vopred určenému vzdelávaciemu cieľu, preto musíme hľadať adekvátne postupy na zistenie stavu dosiahnutia tohto cieľa. Zistenie stavu a posúdenie úrovne dosiahnutých cieľov vo vyučovacom procese definujeme, ako pedagogické diagnostikovanie [17]. Vo vyučovacom procese je predmetom diagnostikovania učebná činnosť žiaka a jej výsledky. „Zisťovanie výsledkov vyučovacieho procesu považujeme za pedagogické diagnostikovanie a posúdenie získaných výsledkov považujeme za hodnotenie“ [18]. V zahraničnej odbornej literatúre pojem hodnotenie nájdeme pod anglickým slovom „assessment“. Aj keď sa zo zahraničia musíme veľa inšpirovať, budeme sa snažiť ostať pri domácich zdrojoch, prípadne kultúrne blízkom prostredí, napr. ČR, či Poľsko. V týchto krajinách existuje podobný vzdelávací systém ako u nás a preto aj literatúra zaoberajúca sa hodnotením je pomerne prenositeľná. Kdežto v iných európskych krajinách je viac zaužívané medzipredmetové vyučovanie (STEAM) a to nespomíname americký vzdelávací systém.

Hodnotenie je problematika, ktorá zaujíma všetky oblasti didaktického výskumu napríklad v matematike, fyzike a informatike [7]. Nás predovšetkým zaujíma hodnotenie v edukačnej robotike, ktorá najčastejšie spadá do informatiky, prípadne voľnočasových aktivít.

Vyučovanie a učenie s robotickými stavebnicami zahŕňa mnoho aspektov, medzi ktoré patrí aj hodnotenie. Hodnotenie je významným prvkom učenia založeného na robotoch a môže pridať na celkovom zážitku z takéhoto učenia sa [18]. Avšak hodnotenie práce žiakov s robotickými stavebnicami je pre mnohých učiteľov stále výzvou. Štúdiom dostupných článkov kníh a publikácií, sme objavili prípady, keď učitelia použili vzájomné hodnotenie [19] pri používaní edukačnej robotiky, či hodnotenie žiakov zamerané na používanie rubrik pri učení sa prostredníctvom robotov a robotických stavebníc [20].

3 Metodológia

Témou nášho výskumu je hodnotenie žiakov v edukačnej robotike na vyučovacom procese. Pre povahu výskumu (ako aj na možný počet participantov v našom výskume) sme zvolili kvalitatívny prístup, konkrétne výskum vývojom [21], ktorý sa realizuje iteratívne. V tejto publikácii opisujeme prvé zistenia a výsledky z prvej etapy nášho dizertačného výskumu.

3.1 Ciele a výskumné otázky

Hlavným cieľom nášho výskumu je zmapovať súčasný stav hodnotenia robotiky na vyučovaní a sformulovať teóriu o tom, ako v súčasnosti vnímajú učitelia hodnotenie v edukačnej robotike a ako ho realizujú. Aké typy hodnotenia používajú učitelia pracujúci s robotickými stavebnicami a aké sú kladné a záporné stránky týchto stratégií. Ďalším cieľom je na základe výsledkov vytvoriť vlastnú formu hodnotiaceho postupu, ktorý iteratívne overíme na rôznych školách priamo vo vyučovacom procese s vyučujúcimi a žiakmi. Táto forma hodnotiaceho postupu bude zohľadňovať vytvorenú teóriu o učiteľovom vnímaní a realizácii hodnotenia žiakov v edukačnej robotike na vyučovaní.

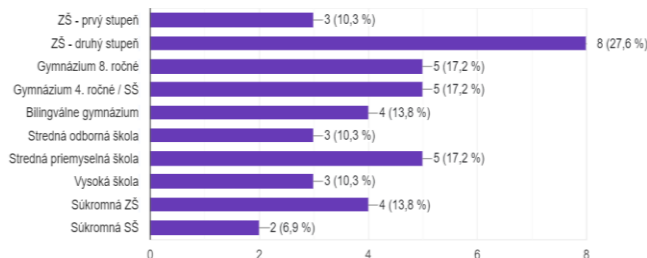
Aby sme naplnili naše ciele, stanovili sme si pre potreby tohto príspevku nasledujúce výskumné otázky:

1. Ako hodnotiť progres žiakov pri práci s robotickými stavebnicami počas vyučovacieho procesu?
2. Ktoré aspekty hodnotenia progresu žiaka sú pre nás kľúčové?

3.2 Zber dát

Prvé dáta nášho výskumu pochádzajú z dotazníkového prieskumu [21], ktorý pozostával z 8 otázok a bol realizovaný pomocou online formulára. Pýtali sme sa v ňom učiteľov informatiky, či pracujú s edukačnou robotikou a ak áno, aký typ hodnotenia žiakov využívajú. Taktiež sme sa učiteľov opýtali, aké sú ich skúsenosti, prípadne odporúčania k hodnoteniu žiakov pri používaní edukačnej robotiky. Dotazníkový prieskum sme sa rozhodli použiť aj vzhľadom na veľkosť našej krajiny a množstvo učiteľov, ktorí vyučujú programovanie, či programovanie prostredníctvom robotických stavebníc.

Dotazníkový prieskum je stále aktívny a zber dát pre náš výskum stále prebieha. Na druhej strane sme získali dostatočné množstvo dát potrebných pre prvotnú analýzu. Na základe týchto výsledkov vieme usmerniť smerovanie nášho ďalšieho výskumu a kontakty na učiteľov, ktorí sú ochotní s nami ďalej spolupracovať. K dnešnému dňu sme získali 42 odpovedí od učiteľov využívajúcich edukačnú robotiku na vyučovaní naprieč všetkými stupňami vzdelávania (ISCED1, ISCED2, ISCED3, ISCED 5A) a všetkými typmi škôl (súkromné, odborné, bilingválne, všeobecno-vzdelávacie), ktoré sú zriadené na Slovensku.



Obrázok 1: Percentuálne rozdelenie účastníkov podľa typu školy

Na Slovensku máme vysoký nedostatok odborných učiteľov pre predmet informatika / výpočtové myslenie a školy túto situáciu riešia tým, že na tieto hodiny umiestnia neodborného učiteľa [22]. Na Slovensku máme 1186 základných škôl, 424 stredných odborných škôl a 234 všeobecnovzdelávacích stredných škôl. Podľa prieskumu centra vedecko technických informácií sa zistilo, že až 42% odučených hodín na predmete informatika odučili neodborní učitelia [23]. Týmto údajmi chceme ukázať, že 42 respondentov v rámci dotazníkového prieskumu je pomerne hodnotná vzorka učiteľov v našej krajine.

Dotazníkový prieskum slúžil aj na oslovenie učiteľov, ktorí by boli ochotní s nami uskutočniť riadený osobný rozhovor, vďaka ktorému sme mohli bližšie porozumieť ich výberu spôsobu hodnotenia žiakov pracujúcich s robotikou. Riadeného rozhovoru sa zúčastnilo (doteraz) šesť učiteľov z rôznych typov škôl a rôznych stupňoch vzdelávania. Išlo o dvoch učiteľov na strednej priemyselnej škole, učiteľku na druhom stupni základnej školy, učiteľku na prvom stupni súkromnej ZŠ, učiteľa súkromného gymnázia a jedného učiteľa z bilingválneho gymnázia.

V súčasnosti prebiehajú aj ďalšie osobné rozhovory s inými učiteľmi, avšak tieto rozhovory ešte nie sú súčasťou tejto publikácie.

3.2.1 Individuálne rozhovory s učiteľmi

Individuálny rozhovor s učiteľmi prebiehal na základe dopredu vytvorených konkrétnych otázok. Otázky, ktoré sme im kladli, boli začlenené do troch základných okruhov.

1. Dôvod používania edukačnej robotiky na vyučovanom procese
2. Stanovenie vzdelávacích cieľov a ich používanie
3. Sledovanie naplnenia vzdelávacích cieľov, hodnotenie žiakov

Otázky z prvého okruhu boli zamerané na zistenie **motivácie** používať edukačnú robotiku na hodinách, ako aj aké typy robotov používajú so svojimi žiakmi.

Nastavovanie vzdelávacích cieľov, bola téma druhého okruhu pokladaných otázok [24]. Ako bolo uvedené v predchádzajúcej kapitole, vzdelávacie ciele zohrávajú vo vyučovanom procese významnú úlohu. Pomáhajú viesť premyslené vyučovacie hodiny a plánovať vzdelávacie aktivity tak, aby pomohli žiakovi osvojiť si potrebné a konkrétne vedomosti, zručnosti, schopnosti a kompetencie pre prácu s edukačnou robotikou, ako aj výpočtové a algoritmické myslenie, umožňuje stanovenie vzdelávacích cieľov. Zaujímalo nás, či si učitelia dokážu stanoviť adekvátne vzdelávacie ciele a akým spôsobom to robia (či používajú taxonómie vzdelávacích cieľov, či vytvárajú vzdelávacie ciele pre danú hodinu a pod.)

Sledovanie **naplnenia** vzdelávacích cieľov bol tretí okruh otázok. Pokiaľ si učiteľ stanoví vzdelávacie ciele, mal by kontrolovať mieru naplnenia týchto cieľov, napr. ako žiaci zvládli prácu na hodine. V podstate ide o hodnotenie práce žiakov na základe určitých kritérií. Nás zaujímalo, či učitelia zúčastnení na osobnom rozhovore hodnotia svojich žiakov, keď pracujú s robotickou stavebnicou. Ak áno, tak aký spôsob hodnotenia využívajú a prečo. Taktiež, či vnímajú výhody alebo nevýhody zvoleného spôsobu hodnotenia a či uvažovali aj and inými formami / spôsobmi hodnotenia.

Po absolvovaní rozhovor s učiteľmi sme zozbierané dáta, ktoré sme zaznamenali na digitálny hlasový záznamník, doplnili našimi osobnými poznámkami, ktoré sme si urobili počas rozhovoru. Taktok pripravené a zozbierané dáta boli pripravené na následnú analýzu.

3.3 Analýza dát

Pri analýze individuálnych rozhovorov sme pracovali s doslovnými prepismi nahrávok pomocou **induktívneho prístupu**. [25] Pred samotnou analýzou prepisov boli pre každú zo skúmaných oblastí, (ktoré sme si určili na základe výskumných otázok) vytvorené hlavné kategórie, ktoré odrážali štruktúru riadeného rozhovoru. Kategórie boli vytvorené na základe otázok, ktoré sme sa pýtali učiteľov a ktoré sme popisali v predchádzajúcej kapitole. Pri ich definovaní boli využité taktiež existujúce sekundárne dáta k danej téme, informácie, získané predchádzajúcou analýzou problematiky a osobnými poznámkami vytvorenými počas rozhovorov.

Počas analýzy sme využili proces **otvoreného kódovania**, ktorý sme použili na všetky prepisy vytvorených nahrávok. Po otvorenom kódovaní sme pristúpili k procesu **axiálneho kódovania**, čím sme dokázali identifikovať vzťahy medzi kategóriami, ktoré nám vznikli.

Pri analyzovaní prepisov rozhovorov bola snaha identifikovať čo najširšie tematické spektrum týkajúce sa konkrétnej oblasti, a na tomto základe vytvárať podkategórie, ktoré sa daného problému týkajú, vysvetľujú jeho príčiny či súvislosti. Prepisy rozhovorov boli segmentované a jednotlivé časti priradované zodpovedajúcim kategóriám. Využitím **konštantnej komparácie** sme údaje priebežne a neustále porovnávali a hľadali medzi nimi spoločné a rozdielne vlastnosti [25].

4 Výsledky prvej etapy výskumu

Na základe analýzy zozbieraných dát sme zistili, že učitelia sa snažia implementovať edukačnú robotiku do svojho vyučovacieho procesu v čo najväčšej možnej miere z dôvodu, že je pre žiakov motivujúca. Žiaci, keď pracujú s robotickou stavebnicou sú nadšení a šťastní, a potom sa oveľa viac snažia pracovať.

Napriek tomu, že sa žiaci viac tešia, keď pracujú s robotmi, učiteľia nám v rámci našich zistení ukázali, že vyučovať pomocou edukačnej robotiky je oveľa náročnejšie. Nejde o to, že by bolo málo materiálov a aktivít, ale problém, s ktorým sa učitelia stretávajú je **hodnotenie**. Každý z našich respondentov uviedol, že najnáročnejšie pri učení s robotickou stavebnicou je hodnotenie žiakov, pričom otázku, ktorú si pokladajú zakaždým, keď idú hodnotiť žiaka je – “Čo je pre mňa dôležité hodnotiť?”. Hodnotenie nie len v zmysle, čo sa žiaci naučili, čo vedia, ale predovšetkým ohodnotiť žiakov stanoveným spôsobom (známky, body, percentá, ...) a ktorá časť práce s robotom je dôležitejšia (programovanie, konštrukcia, celková funkčnosť, snaha pracovať,...).

Každý z našich respondentov uviedol mnoho faktorov a problémov, ktoré ovplyvňujú prácu s edukačnou robotikou na vyučovaní, ale aj prečo je edukačná robotika stvorená pre žiaka na získavanie nových znalostí a zručností. V nasledujúcich bodoch prinášame pár doslovných komentárov našich respondentov. Pre zachovanie anonymity našich respondentov nebudeme uvádzať mená.

- **Učiteľ 1** – „Skúsenosti len zbieram, ale viac-menej sa snažím v známke zohľadniť nielen úspešný "produkt", ale aj snahu riešiť vzniknuté problémy, samostatne hľadať nové alebo iné riešenia, schopnosť pracovať v tíme. Práve preto známku najmä pri projektových úlohách dopĺňam slovným hodnotením.“
- **Učiteľ 2** – „Nechápu, že ak chcú dobre programovať robota, musia zvládnuť aj také veci ako prevody, prevodový pomer a iné.“
- **Učiteľ 3** – „Je dôležité hodnotiť funkčnosť zapojenia a porozumenie ako dve zložky. Na mnou učeních skupinách som vyzozorovala, že známka za aktivitu (na rozdiel od napríklad jazykových predmetov) by mala zohrávať úplne minimálnu úlohu. Šikovný žiak s introvertnou povahou je ňou domotivovaný, naopak výkonovo slabší žiak často považuje takú známku za ľútosť a len ho frustruje.“
- **Učiteľ 4** – „Nedá sa hodnotiť len výsledok. Dôležité je ich sledovať a zistiť, či rozumejú, čo robia, a ako spolupracujú, ako komunikujú s členmi tímu. Je dôležité ich povzbudiť a hodnotiť aj proces, cestu k cieľu.“

Aj na základe reakcií respondentov sme si teda potvrdili, že je potrebné venovať sa tejto téme a zistiť, či existuje spôsob hodnotenia, ktorý je vhodný pre edukačnú robotiku, ale aj vhodný na použitie vo vyučovacom procese.

4.1 Akým spôsobom hodnotiť?

Analýzou osobných rozhovorov s vybranými učiteľmi sme zistili, že si nie sú istí, ktorá forma hodnotenia práce žiaka je najvhodnejšia. Pod pojmom forma hodnotenia žiaka učители mysleli, či žiaka hodnotiť formou známkovania, slovného hodnotenia, bodov, percent, rubrik, portfólií a podobne.

Rozhovory s učiteľmi nám ukázali, že najčastejšie používajú **hodnotenie prostredníctvom známkov** a to najmä preto, že táto forma je preferovaná na ich školách. Avšak pri hodnotení edukačnej robotiky pomocou známkov, objavili niekoľko nedostatkov. Jedným z nedostatkov bola nedostatočná informovanosť pre žiaka, prečo dostal príslušnú známku (napr. čo znamená, že dostal napr. dvojku za svoj robotický model). Nedostatočná informovanosť vplýva aj na rodičov, či samotné vedenie školy, čo znamená, že žiak dostal dvojku pri práci s robotickou stavebnicou.

Ďalším problémom, s ktorým sa stretávajú je, **čo je** pre nich **dôležité** hodnotiť, keďže robotický model je potrebné naprogramovať, ale aj skonštruovať. Výsledkom týchto ťažkostí bolo pri vybraných učiteľoch, že mnohokrát sklzli do formy – ak má žiak funkčný model dostane 1, keď model funkčný nie je, tak známku nedostane prípadne dostane hodnotenie 5. Taktiež sme sa snažili zistiť, prečo hodnotia žiakov práve týmto spôsobom, a či uvažovali aj and inými formami hodnotenia. Respondenti nám povedali, že tento typ hodnotenia je najpoužívanejší na školách na

Slovensku a vôbec ich nenapadlo hľadať / skúmať iné formy hodnotenia, ktoré by mohli byť vhodnejšie na hodnotenia žiaka pri práci s edukačnou robotikou.

Ďalšou zaujímavou informáciou, ktorú sme u našich respondentov zistili je, že občas **nemajú čas** hodnotiť žiakov z dôvodu problémov s nefunkčnosťou robotického modelu z dôvodu softvérového vylepšenia (angl. *update*), s veľkým množstvom individuálnych problémov u každého žiaka (každému žiakovi niečo iné nefunguje a netuší prečo), prípadne sa žiaci zamotajú pri práci s robotom a nestíhajú úlohy dokončiť včas, čím sa preberané látky presúvajú na ďalšie hodiny. Výsledkom je, že učители prestávajú hodnotiť spôsobom, ktorý by reflektoval skutočnú prácu a napredovanie žiaka a sklznu do hodnotenia typu – „pracoval si celú hodinu, máš jednotku“.

Učители sa nám zdôverili, že sa snažia čo najlepšie a najobjektívnejšie hodnotiť prácu žiakov. Avšak každým školským rokom pri boji s toľkými problémami a technickými komplikáciami sklzli do hodnotenia, kde hodnotili, či žiak spravil to, čo mal a dostal známku 1 alebo nemal hotový produkt a učiteľ dal zo solidarity známku 2. Hodnotenie teda prešlo do spôsobu – „vieš s robotom robiť, tak máš jednotku a keď nevieš s robotom robiť, tak máš dvojku“.

Takto určené problémy síce nie sú odpoveďami na naše otázky, avšak zistili sme, že problémom hodnotenia progresu žiaka pri práci s edukačnou robotikou môže byť aj nedostatočná informovanosť učiteľov o ďalších formách hodnotenia. A z tohto bodu môžeme presnejšie nasmerovať ďalšie kroky pre ďalšie etapy nášho výskumu, ktoré popisujeme v kapitole 5.

4.2 Čo by sme mali hodnotiť v edukačnej robotike?

Pri rozhovoroch s učiteľmi sme zisťovali aj to, čo všetko na žiakoch počas vyučovania hodnotia. Zistili sme pri učiteľoch (ktorí absolvovali osobný rozhovor), že sa snažia hodnotiť predovšetkým **programátorskú časť**, teda ako kvalitne bol robotický model naprogramovaný. Či použili správne programovacie koncepty podľa úlohy a či robotický model funguje.

Robotické modely žiaci zvyčajne stavajú podľa návodov a teda konštrukcia robota je postavená zväčša funkčne, a preto učители hodnotia len programovaciu časť. V prípade, že žiaci vytvárajú model na základe ich tvorivej činnosti a nie na základe návodu, tak ani v tomto prípade učители **nehodnotili konštrukciu**. Pri rozhovore s nimi sme sa snažili zistiť, prečo nehodnotia aj konštrukčnú časť. Všetci respondenti nám povedali, že konštrukčnú časť nehodnotia z dôvodu, že **nevedia**, ako by ju mali ohodnotiť. Jeden z našich respondentov uviedol: „*Bojím sa, že by som to zle ohodnotil pretože sa do konštrukcie a prevodov veľmi nerozumiem a hodnotil by som len podľa toho, či sa mi to páči a či sa mi zdá, že je to dobre postavené – teda len na základe mojich domniek a môjho názoru, ktorý môže byť v skutočnosti chybný...*“.

Hľadaním odpovede na druhú výskumnú otázku sme zistili, že učители si uvedomujú, čo všetko vplýva na robotický model (správne pripravená konštrukcia, správne použité programovacie koncepty), ale nie sú si istí, čo je dôležitejšie pri hodnotení, ako aj si nie sú istí, ako hodnotiť konštrukčnú časť stavebnice.

Častokrát si učители pokladajú otázky typu: „Ako hodnotiť, že konštrukcia je dostatočne stabilná?“, „Ako hodnotiť žiaka, ktorého robot funguje, no kód je “škaredo” napísaný?“, „Ako ohodnotiť žiakov, ktorých roboty fungujú, ale jeden žiak má lepšiu konštrukciu a druhý žiak lepší program?“ a mnohé podobné otázky.

Môže to byť spôsobené aj tým, že sa programovanie najčastejšie vyučuje na informatike a teda prirodzene sa zaujímajú o programovanie.

Pri analýze kódov z prepisov rozhovorov sme identifikovali aj ďalšiu zásadnú vec, ktorá sťažuje učiteľovo hodnotenie žiaka. Učitelia, ktorí používajú aktivity z metodických materiálov určených pre prácu s konkrétnou robotickou stavebnicou, pociťujú neprítomnosť hodnotenia - "čo je dôležité v danej aktivite, aby sa žiak naučil z konštrukcie / z programovania?". Učitelia zdôraznili, že keby v každej aktivite metodických materiálov boli doplnené poznámky, ako najlepšie hodnotiť danú aktivitu, resp. čo sa majú žiaci v danej aktivite naučiť (stanovenie vzdelávacích cieľov) a ako to overiť, tak by učiteľom výrazne zjednodušilo prácu a pomohlo.

V neposlednom rade sa nám počas rozhovorov s učiteľmi viackrát do popredia vynárala otázka hodnotenia skupinovej práce žiakov. Je potrebné žiakov hodnotiť individuálne alebo skupinu ako celok? Učitelia z uskutočnených rozhovorov sa častokrát stretávajú so situáciou, kedy v skupine jeden žiak robí takmer všetko a druhý nič. V inej situácii jeden žiak robí konštrukciu a druhý programuje. V tretej situácii obidvaja sa zapájajú do konštrukcie aj do programovania. Ako teda týchto žiakov hodnotiť?

Respondenti niekoľkokrát zdôraznili, čo všetko okrem programovania a konštruovania robotika žiakov naučí. Teda potvrdili výsledky výskumov o benefitoch edukačnej robotiky - prispieva k budovaniu kompetencií užitočných pre život - napr. tímová práca, riešenie problémov a kreativita [8], kreativita a spolupráca [2], komunikácia a riešenie problémov [3,4], či dokonca technické a sociálne zručnosti [9]. Keďže budujeme u našich žiakov aj tieto kompetencie, mali by sme ich tiež brať do úvahy v rámci hodnotenia.

5 Diskusia a smerovanie výskumu

Počas doterajšej analýzy dát sme si uvedomili, že hodnotenie je proces, v ktorom nezohráva úlohu len učiteľ, ale aj samotný žiak. Z toho dôvodu sa budeme v nasledujúcich etapách nášho výskumu zameriavať na žiakov, ako účastipantov výskumu a pokúsime sa získať ich pohľad na hodnotenie edukačnej robotiky (ako žiaci vnímajú, keď sú hodnotení konkrétnym spôsobom hodnotenia).

Učitelia, ktorí absolvovali osobný rozhovor, majú povedomie o ďalších spôsoboch hodnotenia, ktoré môžu využívať na svojich hodinách. Taktiež sú otvorení vyskúšať aj netradičné / iné spôsoby hodnotenia žiakov. Z toho dôvodu plánujeme s týmito učiteľmi ďalšiu spoluprácu v druhej etape nášho výskumu, vďaka ktorej uskutočnime hlbší prieskum pre jednotlivé spôsoby hodnotenia, kde sa pokúsime zistiť silné a slabé stránky konkrétneho typu hodnotenia pri použití v edukačnej robotike.

Učitelia počas osobného rozhovoru načrtli zaujímavú myšlienku. Majú pocit, že žiaci majú oveľa lepšie známky, keď pracujú s edukačnou robotikou, čo môže byť následkom toho, že učitelia nevedia presne, ako hodnotiť žiakov v edukačnej robotike a sklúzu do hodnotenia typu - máš to spravené, tak máš jednotku. Možno aj preto sa žiaci tešia na takúto hodinu, lebo vedia, že dostanú dobrú známku, aj keď nezískajú zásadné vedomosti, či zručnosti, pretože im to spraví spolužiak a podobne. Túto myšlienku by bolo potrebné preskúmať výskumom a overiť, či motivácia žiakov ku práci s edukačnou robotikou súvisí aj s formami hodnotenia žiakov.

Jedným z výsledkov výskumu je záver, že hodnotenie edukačnej robotiky bude potrebné rozdeliť na dva hlavné smery. Prvým smerom bude hodnotenie žiaka pri individuálnej práci s robotickou stavebnicou. Druhým smerom bude hodnotenie žiaka / žiakov pri

skupinovej práci, ktorá býva práve pri robotike veľmi často využívaná. Túto informáciu plánujeme využiť pri tvorení vlastnej formy hodnotenia pre edukačnú robotiku, ktorú budeme tvoriť po druhej etape výskumu (overovanie silných a slabých stránok jednotlivých spôsobov hodnotenia žiaka na vyučovaní).

6 ZÁVER

Počas nášho výskumu sme sa snažili nájsť odpovede na položené výskumné otázky:

1. Ako hodnotiť progres žiakov pri práci s robotickými stavebnicami počas vyučovacieho procesu?
2. Ktoré aspekty hodnotenia progresu žiaka sú pre nás kľúčové?

Pri skúmaní prvej výskumnej otázky sme nenašli úplne exaktnú odpoveď, ale skôr nasmerovanie do ďalších etáp výskumu. Objavila sa otázka typu - "Aké vedomosti majú učitelia o možných spôsoboch hodnotenia?", "Ako zmeniť zaužívaný trend používania známok, ako hodnotenie žiaka?". Ukázalo sa, že učitelia sa snažia za každú cenu aplikovať hodnotenie známkami, ako spôsob hodnotenia napriek tomu, že si uvedomujú, že toto hodnotenie nie je najvhodnejšie. Najmä z dôvodu nedostatočnej výpovednej hodnoty pre žiaka. Okrem toho učitelia pociťujú ťažkosť s hodnotením konštrukcie robotických modelov. Nevedia ako vyhodnotiť, či žiak postavil konštrukciu správne, dostatočne stabilne, spoľahlivo a podobne. Ide o to, že každý žiak pri svojej vlastnej tvorivej činnosti môže prísť k rôznym spôsobom, ako postaviť spoľahlivý robotický model.

Napriek tomu, že učitelia videli tieto problémy, nehľadali iné formy hodnotenia, ktoré by mohli použiť.

Ďalším zaujímavým zistením bolo, že učitelia strácajú motiváciu k využívaniu rôznych spôsobov hodnotenia z dôvodu, že robotické stavebnice v sebe skrývajú technické problémy, ktoré nedokáže vyriešiť ani učiteľ a ani žiak (problém s pripojením robota na tablet cez Bluetooth, nefunkčnosť robota z dôvodu nového systémového softvéru robota, ...). Učitelia tak sklúzu do hodnotenia snahy a nie vedomosti. Také hodnotenie potom vyzerá nasledovne – „ak si dnes pracoval/a s robotom tak máš jednotku, ak si dnes nepracoval/a tak máš dvojku/prípadne päťku“.

Hľadaním odpovede na druhú výskumnú otázku sme zistili, že učitelia si uvedomujú, čo všetko súvisí s kvalitnou stavbou robotického modelu (správne pripravená konštrukcia, správne použité programovacie koncepty), ale nie sú si istí, čo je dôležitejšie pri hodnotení. Taktiež sme zistili, že učiteľom by výrazne pomohlo, keby aktivity v metodických materiáloch mali určené, čo sa majú žiaci naučiť a ako hodnotiť vedomostný progres žiakov pri danej aktivite.

Jedným zo záverov tohto výskumu je, aby **hodnotenie** v edukačnej robotike bolo **rozdelené na dve oblasti**. Hodnotenie progresu žiaka, keď pracuje s robotickou stavebnicou sám a hodnotenie progresu žiaka, keď pracuje s robotickou stavebnicou v rámci skupiny / tímu.

V rámci nášho výskumu máme v pláne zrealizovať aj ďalšie osobné rozhovory s učiteľmi na rôznych typoch škôl, aby sme získali tak širší prehľad o danej problematike a mali sme komplexne pokryté odpovede na naše výskumné otázky. Po rozhovoroch s učiteľmi plánujeme s vybranými učiteľmi v spolupráci pokračovať a s ich pomocou otestovať najvhodnejšie typy hodnotenia pre edukačnú robotiku na vyučovaní. Analýzou kladných a záporných stránok tohto overovania sa pokúsime priniesť overenú a kvalitnú formu

hodnotenia pre edukačnú robotiku. Okrem rozhovorov s učiteľmi máme v pláne uskutočniť aj rozhovory so žiakmi. Keďže hodnotenie je proces, v ktorom nezohráva úlohu len učiteľ, ale aj samotný žiak. Chceme tak získať pohľad žiakov na spôsoby hodnotenia, ktoré využívajú ich učitelia a obohatiť tak našu perspektívu na hodnotenie.

Získané výsledky nám dali prvotný pohľad na problematiku hodnotenia žiakov pri práci s edukačnou robotikou a utvrdili sme sa v názore, že je potrebné sa tejto téme venovať, pretože učitelia informatiky si nie sú istí, ako vhodne hodnotiť svojich žiakov na vyučovaní s edukačnou robotikou, čoho dôsledkom je nekvalitná až nedostatočná spätná väzba žiakom.

POĎAKOVANIE

Ďakujeme recenzentom za ich pripomienky k tomuto článku. Radi by sme poďakovali projektom VEGA 1/0621/22 ako aj APVV projektu APVV-20-0353 a projektu FERTILE 2021-1-EL01-KA220-HED-000023361 za poskytnuté finančné prostriedky a za uverejnenie našich výsledkov.

BIBLIOGRAFICKÉ ODKAZY

- [1] CVIK, P. Uplatňovanie učebnej pomôcky ROBO LAB v základných školách na Slovensku – výsledky výskumu. In *MIF* 26. ISSN 1335-7794. 2005. s. 143.
- [2] VESELOVSKÁ, M. MAYEROVÁ, K. ROBORIC KITS IN SECONDARY SCHOOL. In *Journal of Technology and Information Education*. 2013. 5.1: 96.
- [3] EGUCHI, A. RoboCupJunior for promoting STEM education, 21st century skills, and technological advancement through robotics competition. *Robotics and Autonomous Systems*. 75, 2016. p. 692 - 699.
- [4] VESELOVSKÁ, M. *Cooperation of children in teams at robotic competitions*: diplomová práca. Comenius University Bratislava. 2012. 90 p.
- [5] USART, M., et al. Are 21st Century Skills Evaluated in Robotics Competitions? The Case of First LEGO League Competition. In *Proceedings of the 11th International Conference on Computer Supported Education (CSEDU 2019)*, 2019. p. 445 – 452.
- [6] KRCHO, J. *Kognitívna a psychomotorická taxonómia v edukačnej robotike*: diplomová práca. Univerzita Komenského v Bratislave. 2022. 55 s.
- [7] GANAJOVÁ, M. et al. *Formatívne hodnotenie vo výučbe prírodných vied, matematiky a informatiky*. Univerzita Pavla Jozefa Šafárika v Košiciach. 2021. ISBN 978-80-8152-973-3.
- [8] MIKOVÁ, K., KRCHO, J. Cognitive Taxonomy and Task Gradation in Educational Robotics–Preliminary Results. In *Iberian Robotics conference*. Springer, Cham, 2023. p. 575-585.
- [9] KABÁTOVÁ, M., PEKÁROVÁ, J. Learning how to teach robotics. In *Constructionism 2010 conference*. 2010.
- [10] KANDLHOFER, M., STEINBAUER, G. Evaluating the impact of educational robotics on pupils' technical and social skills and science related attitudes. In *Robotics and Autonomous Systems*. 75, 2016. p. 679 – 685.
- [11] AFARI, E., KHINE, M. S. Robotics as an educational tool: impact of Lego Mindstorms. In *International Journal of Information and Education Technology* 7(6). 2017. p. 437 – 442.
- [12] ARÍS, N., ORCOS, L.: Educational Robotics in the Stage of Secondary Education: Empirical Study on Motivation and STEM Skills. In *Education Science* 9(2). 2019.
- [13] SKLAR, E., et al.: RoboCupJunior: learning with educational robotics. In: *Robot Soccer World Cup, vol 2752 Springer, Heidelberg*. 2002. p. 238-253.
- [14] ANWAR, S. et al.: A Systematic Review of Studies on Educational Robotics. In *Journal of Pre-College Engineering Education Research (J-PEER)* 9(2). 2019. p. 19 – 42.
- [15] KALAŠ, I., MAYEROVÁ, K., VESELOVSKÁ, M.: Developing a Learning Taxonomy for Educational Robotics. In *Journal of Technology and Information Education* 6(1).2014. p. 30-44.
- [16] PASCH, M. *Teaching as decision making: Successful practices for the elementary teacher*. Longman, 10 Bank Street, White Plains, NY. 1995.
- [17] GAVORA, P.: Akí sú moji žiaci?. Nitra: Enigma. 2019.
- [18] GURA, M. Getting started with LEGO robotics: a guide for K-12 educators. In *International Society for Technology in Education*. 2011.
- [19] CATLIN, D. Using Peer Assessment with Educational Robots, In *ICWL 2014 Workshops*, LNCS vol. 8699, Springer International Publishing, Switzerland. 2014. p. 57-65.
- [20] VESELOVSKÁ, M., MAYEROVÁ, K. Assessment of Lower Secondary School Pupils' Work at Educational Robotics Classes. In *International Conference EduRobotics 2016*. Springer, Cham, 2016. p. 170-179.
- [21] CRESWELL, J. W. *Educational research: Planning, conducting, and evaluating quantitative*. Prentice Hall Upper Saddle River, NJ, 2002. ISBN 978-0-13-136739-5.
- [22] ŠŠI. *Správa o stave a úrovni výchovy a vzdelávania v školách a školských zariadeniach v Slovenskej republike v školskom roku 2016/2017*. [cit. 2023-02-15] Bratislava: ŠŠI, 2018. Dostupné na <https://www.ssiba.sk/Default.aspx?text=g&id=3&lang=sk>
- [23] CVTI SR. *Kvalifikovanosť pedagogických zamestnancov*. [cit. 2023-02-14]. Dostupné na <https://www.minedu.sk/kvalifikovanost-pedagogickych-zamestnancov/>
- [24] PASCH, M. et al. *Od vzdelávacieho programu k vyučovací hodine*. 2. vyd. Praha: Portál, 2005. 416 s. ISBN 80-7367-054-2.
- [25] ŠVAŘÍČEK, R. et al. *Kvalitatívni výzkum v pedagogických vědách*. Praha: Portál, 2007. ISBN 978-80-7367-313-0.

Preferencie pedagógov pre personalizované vyučovanie informatiky – prieskum

Educators' preferences for personalized teaching of informatics – survey

Zuzana Kubincová

Univerzita Komenského v Bratislave
Mlynská dolina
84248 Bratislava
Slovensko
kubincova@fmph.uniba.sk

Karolína Miková

Univerzita Komenského v Bratislave
Mlynská dolina
84248 Bratislava
Slovensko
mikova@fmph.uniba.sk

ABSTRACT

Personalized learning is an approach that aims to adapt learning, its form, content, and pace to the needs, abilities, and interests of each learner. The development of digital technologies and their use in education nowadays greatly affects teaching and learning, but also vice versa - digital technologies used in education are adapting to educational approaches and trends. Personalization and adaptability are therefore becoming important elements of learning environments. In this paper, we present the preliminary results of a survey among university teachers of informatics and informatics education, focusing on the personalization of teaching and the use of personalized learning environments.

Keywords

Personalized teaching. Personalized learning environment. Higher education. Survey.

ABSTRAKT

Personalizované vyučovanie je prístup, ktorého cieľom je prispôbiť učenie sa, jeho formu, obsah, aj tempo potrebám, schopnostiam a záujmom každého učiaceho sa.

Rozvoj digitálnych technológií a ich využívanie vo vzdelávaní v súčasnosti veľmi ovplyvňuje výučbu a učenie sa, ale aj naopak – digitálne technológie využívané vo vzdelávaní sa prispôbujú edukačným prístupom a trendom. Personalizácia a adaptívnosť sa preto stávajú dôležitými prvkami vzdelávacích prostredí. V tomto článku prinášame predbežné výsledky prieskumu medzi vysokoškolskými učiteľmi informatiky a učiteľstva informatiky zameraného na personalizáciu vyučovania a využívanie personalizovaného vzdelávacieho prostredia.

Kľúčové slová

Personalizované vyučovanie. Personalizované vzdelávacie prostredie. Vysokoškolské vzdelávanie. Prieskum.

1 ÚVOD

Rozvoj digitálnych technológií a následne aj webových služieb a ich začleňovanie do všetkých oblastí života viedli k výraznému nárastu množstva informácií dostupných na internete [1]. Toto so sebou prináša nielen riziko presýtenosti informáciami ale aj potrebu správneho orientovania sa v nich. Ako reakcia na túto situáciu začali vznikať tzv. odporúčacie systémy. Ide o softvérové nástroje, ktorých cieľom je pomôcť používateľovi nájsť a vybrať to, čo najlepšie zodpovedá jeho potrebám a požiadavkám [2]. V snahe

reagovať na individuálne potreby používateľov, uplatňujú tieto systémy personalizovaný prístup založený spravidla na priebežne aktualizovanom profile používateľa [3].

Odporúčacie systémy sa stali často používaným nástrojom v oblasti webu, elektronického obchodu a portálových riešení, ale stretneme sa s nimi napr. aj vo forme odporúčaných videí na YouTube, filmov na streamovacích službách, priateľov na sociálnych sieťach a pod. [3, 4].

Odporúčacie a personalizované systémy vzbudzujú veľkú pozornosť aj v oblasti vzdelávania, pretože by mohli ponúknuť každému jednotlivému študentovi obsah, ktorý vyhovuje jeho štýlu učenia, už nadobudnutým vedomostiam/zručnostiam a tiež jeho potrebám [5]. Takýto prístup ku vzdelávaniu zodpovedá filozofii tzv. "personalizovaného vyučovania". Napriek tomu, že tento pojem vznikol už pred šiestimi desaťročiami, jeho význam sa stále vyvíja. Ide o metódu vzdelávania orientovanú na študenta, kedy je tempo výučby i vzdelávací prístup prispôbovaný silným stránkam, schopnostiam a záujmom každého študenta individuálne [6].

Na Fakulte matematiky, fyziky a informatiky UK využívame pri vyučovaní viacerých kurzov predovšetkým v študijných programoch aplikovanej informatiky a učiteľstva informatiky vlastné vzdelávacie prostredie prispôbené potrebám daných kurzov. Naším cieľom je posilniť toto prostredie o prvky personalizácie, aby sme vo svojom vyučovaní mohli lepšie pomáhať našim študentom optimalizovať ich učenie sa napríklad odporúčaním pre nich vhodných študijných materiálov, edukačných aktivít a pod. Takto rozšírené personalizované vzdelávacie prostredie by malo zohľadňovať nielen štýl učenia študenta, jeho vstupné znalosti, zručnosti a kompetencie, či jeho pokrok počas absolvovania kurzu [1], ale malo by tiež viesť k naplneniu vzdelávacích cieľov kurzu a spĺňať predstavy a požiadavky vyučujúcich. Preto sme medzi učiteľmi aj študentmi našej fakulty uskutočnili prieskum zameraný na personalizované vyučovanie vo virtuálnom vzdelávacom prostredí. V tomto príspevku sa zaoberáme len prieskumom medzi učiteľmi a podávame správu o jeho predbežných výsledkoch.

Doterajšie zistenia ukazujú, že aj keď väčšina našich respondentov v súčasnosti využíva vo svojom vyučovaní isté prvky personalizácie, učiteľia zrejme ešte neuvažujú o využívaní celej komplexnej funkcionality takéhoto prostredia.

2 METODIKA

Cieľom prieskumu bolo zistiť postoj učiteľov k personalizovanému vzdelávaniu, ich oboznámenosť s možnosťami, ktoré môže ponúknuť personalizované vzdelávacie prostredie a ich ochotu a pripravenosť takéto prostredie vo svojom vyučovaní používať.

Prieskum bol realizovaný koncom roka 2022 medzi učiteľmi študijných programov učiteľstva informatiky a aplikovanej informatiky. Zber dát sa uskutočnil prostredníctvom anonymného online dotazníka rozdeleného na niekoľko sekcií, z ktorých každá bola zameraná na jednu konkrétnu oblasť: *Základné údaje, Systémy, nástroje a materiály vo vyučovacom procese, Personalizácia vyučovacieho procesu a Personalizácia vzdelávacieho prostredia.*

V sekcii *Základné údaje* sa zbierali základné charakteristiky respondentov, ako sú pohlavie, vek, dĺžka učiteľskej praxe, stupeň, na ktorom respondent vyučuje, oblasť a typ vyučovania.

Druhá sekcia bola zameraná na technológie a materiály, ktoré respondenti využívajú vo svojom vyučovaní, na proces overovania vedomostí a zručností študentov, spôsob doručovania výsledkov a spätnej väzby študentom a tiež na funkcie, ktoré učiteľom v používaných softvérových aplikáciách chýbajú.

Účelom sekcie o personalizácii vyučovacieho procesu bolo zistiť, aké formy personalizácie respondenti využívajú pokiaľ ide o spôsoby hodnotenia študentov, poskytovania spätnej väzby, motivovania k lepším výkonom, či k rozvoju zručnosti autoregulácie pri učení sa.

Posledná sekcia bola orientovaná na možné črty a nástroje personalizovaného vzdelávacieho prostredia s cieľom zistiť, ktoré z nich by respondenti vo svojom vyučovaní chceli využívať.

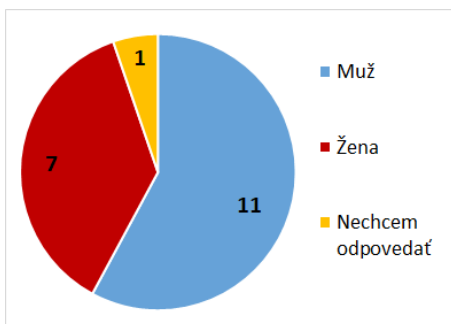
3 VÝSLEDKY

3.1 Analýza dát

Dotazník bol respondentom doručený prostredníctvom Microsoft Forms. Získané dáta boli vyexportované do tabuľkového kalkulátora a ďalej spracované kvalitatívnymi metódami (napr. obsahová analýza textu) ako aj základnými metódami popisnej štatistiky [7, 8].

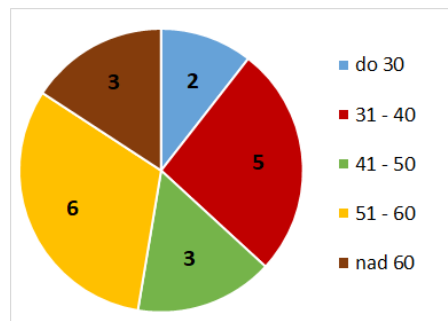
3.2 Vzorka

Dotazník vyplnilo 19 učiteľov vyučujúcich na programoch učiteľstva informatiky a aplikovanej informatiky. Medzi respondentmi bolo 11 mužov, 7 žien a jeden respondent s neuvedeným pohlavím (Obrázok 1).



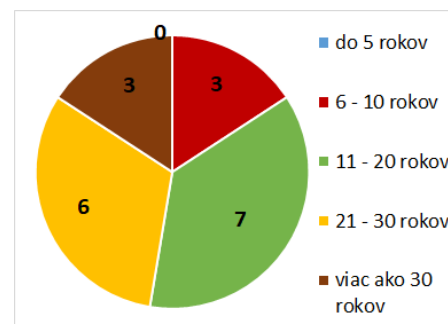
Obrázok 1: Pohlavie respondentov

Boli zastúpené všetky vekové kategórie – od učiteľov do 30 rokov až po učiteľov nad 60 rokov (Obrázok 2)



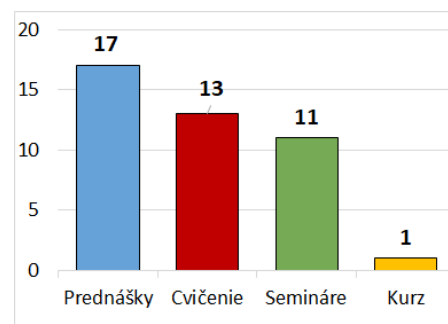
Obrázok 2: Vek respondentov

Z pohľadu dĺžky učiteľskej praxe na vysokej škole patrili respondenti do štyroch rôznych kategórií od 6 rokov vyššie. Chýbalo zastúpenie učiteľov s praxou do 5 rokov (Obrázok 3)



Obrázok 3: Dĺžka praxe respondentov

Všetci respondenti uviedli, že učia na magisterskom stupni, trinásti učia aj na bakalárskom stupni a sedem respondentov vyučuje aj na doktorandskom štúdiu. Pritom väčšina respondentov (17) vedie prednášky, trinásť z nich vedie cvičenia, jedenásti vedú semináre a jeden z respondentov uviedol, že učí kurz (Obrázok 4). Iba štyria respondenti uviedli, že vedú len jeden typ vyučovania, z toho dvaja iba prednášajú. Ostatní vedú prednášky a aj cvičenia alebo semináre.



Obrázok 4: Typ vyučovania respondentov

Keďže respondentmi boli učelia z informatických študijných programov, medzi oblasťami ich výučby dominovalo programovanie, ale časté odpovede boli aj počítačová grafika a počítačové videnie, umelá inteligencia, kognitívna veda a iné.

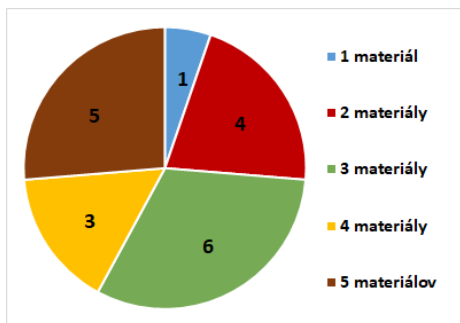
3.3 Zistenia

Výsledky analýzy odpovedí respondentov uvádzame rozdelené do jednotlivých oblastí.

3.3.1 Systémy, nástroje a materiály vo vyučovacom procese

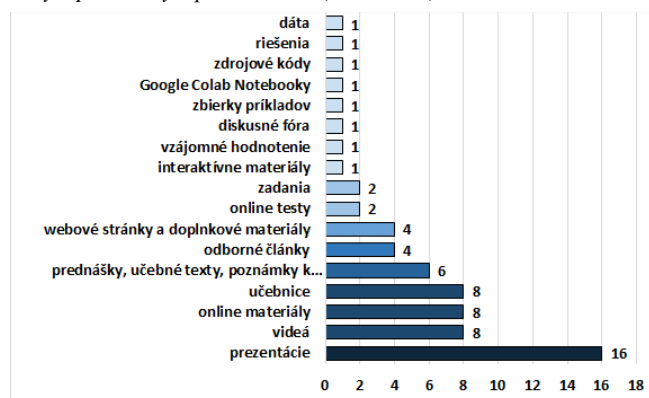
Táto časť dotazníka obsahovala šesť otázok s otvorenou odpoveďou zameraných na materiály a technológie, ktoré učitelia využívajú vo svojom vyučovaní, na proces hodnotenia a poskytovanie spätnej väzby a výsledkov, tiež v kontexte používaných technológií.

Odpovede na otázku, aké typy materiálov učitelia poskytujú študentom, boli veľmi rozmanité. Okrem jedného respondenta, ktorý stručne odpovedal: *online materiály*, všetci ostatní uviedli, že študentom poskytujú viacero typov materiálov (Obrázok 5).



Obrázok 5: Koľkí učitelia poskytujú daný počet materiálov

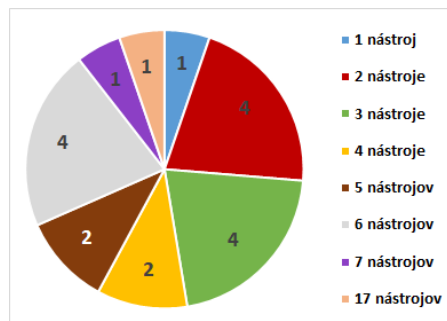
Ako najčastejšie poskytované materiály boli udávané *prezentácie* – uviedlo ich až 16 učiteľov. Veľmi často boli spomenuté *učebnice*, *videá* a *online materiály*, medzi časté patrili *prednášky*, *učebné texty* a *poznámky k prednáškam* (Obrázok 6).



Obrázok 6: Typy poskytovaných materiálov a ich početnosť

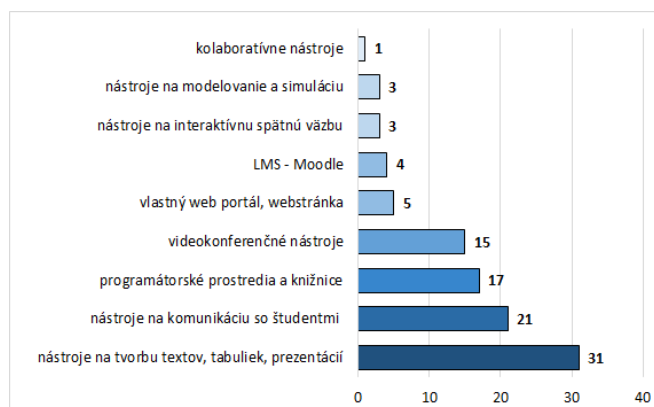
V nasledujúcej otázke sme sa učiteľov pýtali, na technológie, ktoré používajú vo vyučovaní. Požiadali sme ich, aby menovali konkrétne systémy, aplikácie, resp. softvéry a tiež ich funkcie, ktoré najviac využívajú.

Väčšina respondentov vymenovala viacero používaných nástrojov spolu s účelom ich použitia (Obrázok 7). Respondent, ktorý uviedol iba jeden nástroj (PowerPoint), bol iba jeden. Opačný extrém bola respondentka, ktorá popísala použitie až 17 rôznych nástrojov. Išlo o mladú učiteľku do 30 rokov, ktorá sa okrem infromatických predmetov venuje aj pedagogike.



Obrázok 7: Koľkí učitelia používajú dané počty nástrojov

Medzi najčastejšie používané aplikácie patria *nástroje na tvorbu textov, tabuliek a prezentácií* (Google nástroje, MS Office, Latex, OpenOffice), ktoré boli v súhrne spomenuté v odpovediach až 31-krát (Obrázok 8). Ďalej nasledovali *nástroje na komunikáciu so študentmi* najmä prostredníctvom krátkych správ, medzi ktorými respondenti spomenuli MS Teams, Moodle, MS Outlook, AIS a Doodle. Veľmi často sa v odpovediach vyskytli aj rozmanité *programátorské prostredia a knižnice* a tiež rozličné *videokonferenčné nástroje* (najčastejšie MS Teams a Google Meet). V piatich odpovediach bol spomenutý *vlastný systém, web portál alebo webstránka* a v štyroch odpovediach *LMS Moodle*. Okrem toho učitelia využívajú *nástroje na interaktívne prezentácie a spätnú väzbu od študentov* (napr. Padlet, Kahoot, Polleverywhere) a rozličné *nástroje na modelovanie a simuláciu*. Aj keď viaceré z vyššie spomenutých nástrojov môžu slúžiť ako kolaboratívne, ich využitie na *kolaboratívne aktivity* bolo explicitne spomenuté len v jednej odpovedi.



Obrázok 8: Typy využívaných nástrojov a ich početnosť

V tejto sekcii sme ďalej požiadali respondentov, aby stručne popísali nimi najčastejšie používaný proces overovania vedomostí a zručností študentov a zamerali sa pritom na systémy, aplikácie, resp. softvéry, ktoré pritom používajú oni a ich študenti.

Takmer polovica (9) respondentov nezodpovedala túto otázku dostatočne zrozumiteľne – z mnohých odpovedí nebolo jasné, či a aké technológie učitelia využívajú, resp. ako pri overovaní vedomostí postupujú. Zo zvyšných 10 respondentov štyria využívajú k overovaniu vedomostí študentov LMS Moodle a rovnaký počet používa vlastný systém. V týchto systémoch zverejňujú zadania (6), zbierajú riešenia (8), poskytujú hodnotenie (7) aj spätnú väzbu (4). V troch prípadoch učitelia využívajú aj funkcie systému umožňujúce vzájomné hodnotenie študentov.

Niektorí respondenti využívajú na zverejňovanie zadaní, zber riešení aj spätnú väzbu okrem vlastného systému, resp. emailu aj Google dokumenty. Popri využívaní nejakého LMS (Moodle alebo vlastného systému) druhou najčastejšou odpoveďou o overovaní vedomostí boli písomné (papierové) testy – na takejto odpovedi sa zhodli až siedmi respondenti. V troch prípadoch sa priamo alebo nepriamo vyjadrili, že nepoužívajú žiadne systémy, alebo aplikácie.

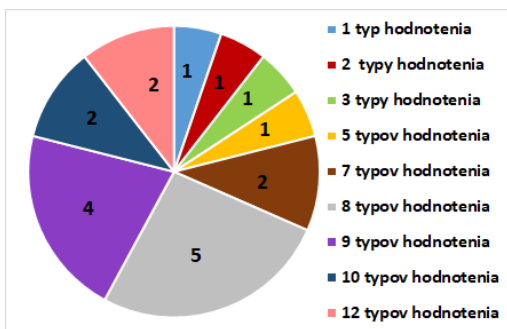
Pokiaľ ide o nástroje, ktoré pri overovaní vedomostí používajú študenti, tak respondenti najčastejšie uvádzali programátorské a modelovacie nástroje (9 respondentov), textové a grafické editory (6 respondentov) a prezentačný softvér (4 respondenti).

O udelenom hodnotení učiteľa najčastejšie informujú študentov prostredníctvom LMS (Moodle alebo vlastného systému), kde k prideleným bodom pridávajú aj komentár (7 odpovedí). Šiesti používajú na zverejňovanie výsledkov Google tabuľky a šiesti zasielajú výsledky emailom. Osem respondentov pridáva k výsledkom sprostredkovaným technológiami aj ústne komentáre na hodine, v prípade online vyučovania cez niektorý video-konferenčný nástroj.

3.3.2 Personalizácia vyučovania

Cieľom tejto sekcie bolo zistiť, nakoľko respondenti vo svojich kurzoch využívajú metódy a postupy zodpovedajúce personalizovanému vyučovaniu.

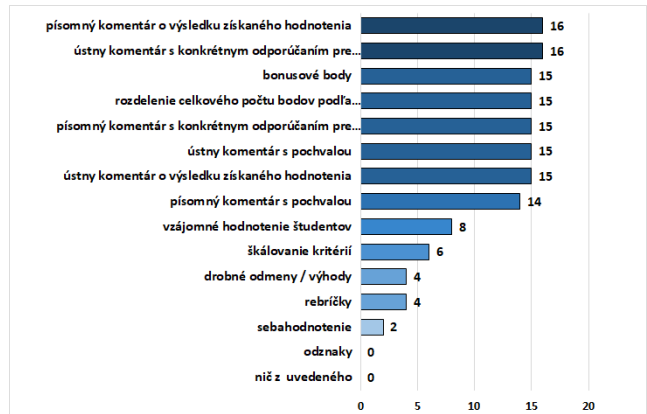
V prvej otázke s výberom viacerých možností odpovede sme sa pýtali, ktoré z uvedených foriem formatívneho hodnotenia respondenti používajú vo výučbe (Obrázok 9). Iba jeden spomedzi respondentov (ten istý, ktorý v predchádzajúcich otázkach odpovedal veľmi stručne, často jednoslovné) zvolil iba jednu možnosť odpovede – *rebričky*. Podobne dve, tri a päť možností zvolil vždy iba jeden respondent. Ostatní respondenti označili aspoň 7 z 15 uvedených možností. Najviac – až 12 možností – využívajú vo svojej praxi dvaja z respondentov. Obaja uviedli aj ďalší spôsob hodnotenia, ktorý využívajú: *ústna aj písomná celková spätná väzba - všeobecná pre celú skupinu a týždenné vzájomné hodnotenie odovzdaných riešení študentov komentármi a hviezdikami, zverejňovanie poradia a vyhlasovanie víťazov*.



Obrázok 9: Koľkí učitelia používajú koľko rôznych typov hodnotenia

Dva najčastejšie využívané typy hodnotenia (16 výskytov) boli *písomný komentár o výsledku získaného hodnotenia* a *ústny komentár s konkrétnym odporúčaním pre zlepšenie* (Obrázok 10). Medzi veľmi často používané (15 výskytov) patrili štyri typy hodnotenia: *písomný komentár s konkrétnym odporúčaním pre zlepšenie*, *ústny komentár o výsledku získaného hodnotenia*, *ústny komentár s pochvalou*, *rozdelenie celkového počtu bodov podľa oblastí/úloh - študent vie, za čo získal koľko bodov a bonusové*

body. Možno k nim zaradiť aj *písomný komentár s pochvalou*, ktorý sa v odpovediach vyskytol 14-krát. Najmenej z uvedených typov hodnotení využívajú učitelia *sebahodnotenie študentov* (2 výskyty) a *odznaky*, ktoré neoznačil žiadny z respondentov.

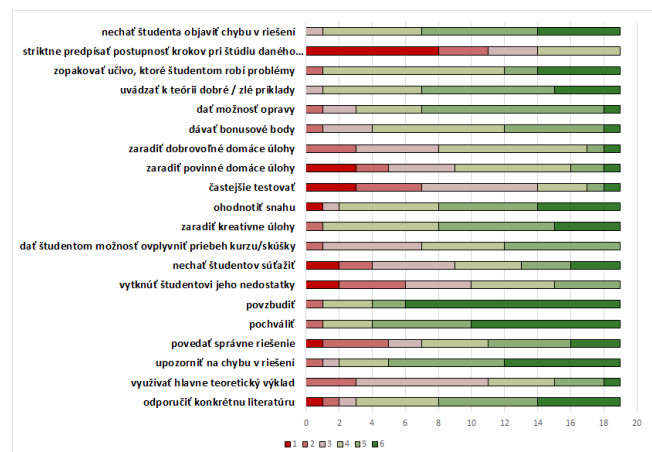


Obrázok 10: Ako učitelia využívajú rôzne typy hodnotenia

Ďalej sme respondentov požiadali ohodnotiť na škále 1-6 (1: 'vôbec', 6: 'veľmi') dvadsať rôznych prístupov k vyučovaniu a hodnoteniu podľa toho, nakoľko ich považujú za prínosné pre zlepšenie výsledkov študentov (Obrázok 11).

Výrazne najlepšie hodnotenie získala možnosť *povzbudiť* – až 13 respondentov jej udelilo najvyššie bodové hodnotenie a nikto jej neudelil najnižšie hodnotenie. Na druhom a treťom mieste sa umiestnili možnosti *pochváliť* (9-krát najvyššie hodnotenie) a *upozorniť na chybu v riešení* (7-krát najvyššie hodnotenie), ktorým rovnako žiadny respondent neudelil najnižšie hodnotenie.

Najmenej sa učiteľom páčila možnosť *striktné predpísať postupnosť krokov pri štúdiu daného predmetu*, ktorej nikto z nich neudelil ani najvyššie ani druhé najvyššie hodnotenie, a ktorej až 8 učiteľov udelilo najnižšie hodnotenie. Napriek tomuto výsledku, ani možnosť *dať študentom možnosť ovplyvniť priebeh kurzu/skúšky*, ktorá je v istom zmysle náprotivkom predchádzajúcej – najhoršie hodnotenej – možnosti, nezískala od respondentov dobré hodnotenie a umiestnila sa až ku koncu tohto rebrička.

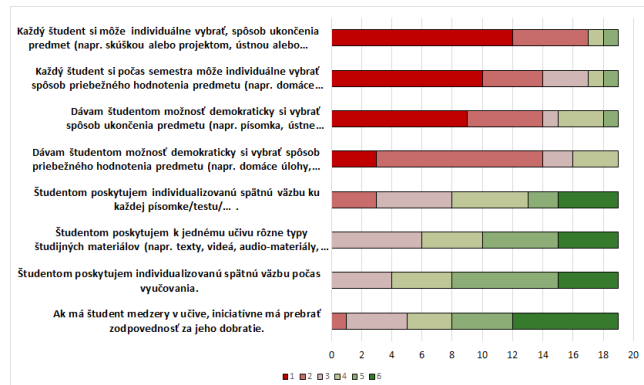


Obrázok 11: Prínos jednotlivých metód pre zlepšenie výsledkov študentov

Následne sme respondentov požiadali, o ohodnotenie série výrokov týkajúcich sa spôsobu vyučovania a hodnotenia. Hodnotili opäť na

škále 1-6 (1: 'striktné nesúhlasím', 6: 'úplne súhlasím') podľa toho, nakoľko sa stotožňovali s daným výrokom.

Najviac z nich sa stotožnilo s výrokom *Ak má študent medzery v učive, iniciatívne má prebrať zodpovednosť za jeho dobratie*, ktorému najvyššie a druhé najvyššie hodnotenie udelilo 11 učiteľov a najnižšie hodnotenie neudelil nikto (Obrázok 12). Podobne dobré hodnotenie získal výrok *Študentom poskytujem individualizovanú spätnú väzbu počas vyučovania*. Najmenej respondentov sa stotožnilo s výrokom: *Každý študent si môže individuálne vybrať, spôsob ukončenia predmetu (napr. skúškou alebo projektom, ústnou alebo písomnou skúškou, bez skúšky, ...)*. Tomuto výroku dalo najnižšie alebo druhé najnižšie hodnotenie až 17 učiteľov a žiaden z nich mu nedal najvyššie hodnotenie.



Obrázok 12: Ako sa učители stotožňujú s výrokmi o vyučovaní a hodnotení

3.3.3 Personalizácia vzdelávacieho prostredia

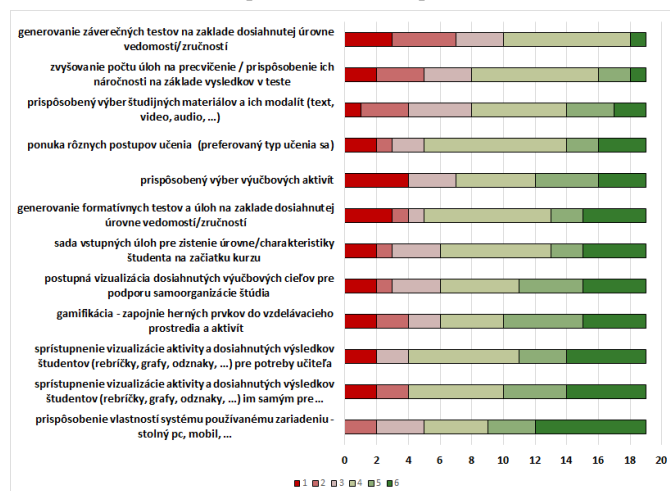
V poslednej sekcii dotazníka sme sa snažili zistiť, ktoré z potenciálnych funkcií personalizovaného a adaptívneho vzdelávacieho prostredia by učители chceli využívať. Za týmto účelom sme im dali opäť na škále 1-6 (1: 'vôbec nie', 6: 'určite áno') ohodnotiť nasledovné možnosti:

- sada vstupných úloh pre zistenie úrovne/charakteristiky študenta na začiatku kurzu
- prispôbený výber študijných materiálov a ich modalít (text, video, audio, ...)
- prispôbený výber výučbových aktivít
- ponuka rôznych postupov učenia (preferovaný typ učenia sa)
- generovanie formatívnych testov a úloh na základe dosiahnutej úrovne vedomostí/zručností
- generovanie záverečných testov na základe dosiahnutej úrovne vedomostí/zručností
- zvyšovanie počtu úloh na precvičenie / prispôbenie ich náročnosti na základe výsledkov v teste
- postupná vizualizácia dosiahnutých výučbových cieľov pre podporu samoorganizácie štúdia
- sprístupnenie vizualizácie aktivity a dosiahnutých výsledkov študentov (napr. pomocou rebríčkov, grafov, odznakov, ...) pre potreby učiteľa
- sprístupnenie vizualizácie aktivity a dosiahnutých výsledkov študentov (napr. pomocou rebríčkov, grafov, odznakov, ...) im samým pre zvýšenie ich motivácie
- gamifikácia - zapojenie herných prvkov do vzdelávacieho prostredia a aktivít
- prispôbenie vlastností systému používanému zariadeniu - stolný pc, mobil, ...

Z uvedených možností by učители najviac chceli, aby *systém prispôboval svoje vlastnosti používanému zariadeniu* (Obrázok 13). Tejto možnosti udelilo 7 respondentov najvyššie hodnotenie a žiadny neudelil najnižšie hodnotenie. Ďalšie dve funkcie, ktoré by učители chceli využívať, by mali zabezpečiť *sprístupnenie vizualizácie aktivity a dosiahnutých výsledkov študentov*, jednak *študentom samotným pre zvýšenie ich motivácie* a tiež *pre potreby učiteľa*. V oboch prípadoch dalo týmto funkciám najvyššie hodnotenie 5 respondentov a najnižšie dvaja.

Najmenej sa respondentom pozdávala možnosť *generovania záverečných testov na základe dosiahnutej úrovne vedomostí/zručností*. Najvyššie hodnotenie jej udelil len jeden z nich, druhé najvyššie hodnotenie nikto a najnižšie hodnotenie 3 respondenti.

Celkovo sa nám hodnotenie potenciálnych funkcií personalizovaného a adaptívneho systému učiteľmi zdalo byť trochu rozpačité. Vlastnosti, ktoré vyزدvihli, patria v súčasnosti medzi bežné funkcie vzdelávacích prostredí a nevyžadujú, aby prostredie bolo personalizované a adaptívne. Naopak, vlastnosti, ktoré by umožňovali prispôbenie prostredia, spôsobu učenia sa, či hodnotenia, neboli respondentmi veľmi preferované.



Obrázok 13: Aké funkcie personalizovaného prostredia by chceli učители využívať

V poslednej otázke sme dali respondentom priestor na uvedenie funkcií, ktoré sme vyššie nespomenuli, a ktoré by mali v personalizovanom vzdelávacom prostredí radi k dispozícii. Odpovedali len štyria respondenti, z toho dvaja sa vyjadrili v zmysle: *Bez dostatočného množstva času navyše by to nemalo zmysel*. Zvyšné dve odpovede navrhovali *Zabezpečenie prístupnosti pre osoby so špeciálnymi vzdelávacími potrebami a zdravotným postihnutím* a *Nástroje podporujúce interaktivitu medzi študentami navzájom aj medzi študentami a učiteľom*.

4 ZÁVER

Cieľom nášho prieskumu bolo získať obraz o tom, ako vysokoškolskí učители vnímajú personalizované vyučovanie študentov učiteľstva informatiky a aplikovanej informatiky, či a v akom rozsahu ho vo svojej praxi využívajú a tiež nakoľko by privítali personalizované vzdelávacie prostredie, a ktoré jeho funkcie považujú za dôležité.

Z analýzy dát z oblasti *Systémy, nástroje a materiály* vyplynulo, že účastníci prieskumu poskytujú svojim študentom pomerne veľa

rozličných materiálov – sumárne uviedli viac ako 15 typov, pričom až 74% z nich poskytuje študentom tri, štyri alebo päť typov materiálov. Pri vyučovaní mnohí pracujú s viacerými druhmi technológií – opäť až 74% respondentov využíva viac ako dva (tri až sedem) typy aplikácií, nástrojov, či systémov.

Prieskum v oblasti *Personalizácie vyučovania* ukázal, že učitelia deklarujú využívanie širokej škály foriem formatívneho hodnotenia od písomných komentárov k výsledku získaného hodnotenia, cez konkrétne odporúčania k zlepšeniu, či vzájomné hodnotenie študentov až k sebahodnoteniu. Podľa získaných odpovedí až 89% našich respondentov poskytuje študentom aspoň tri typy hodnotenia, väčšina z nich dokonca značne viac (až do 12). Pre zlepšenie výsledkov svojich študentov majú učitelia tendenciu využívať skôr pozitívnu motiváciu prostredníctvom povzbudenia a pochvaly, byť študentom nápomocní pri problémoch so zvládnutím učiva (zopakovať učivo, ktoré robí problémy, odporučiť literatúru), ale aj viesť študentov k samostatnosti a k učeniu učiť sa (upozorniť na chybu v riešení, ale nechať samotného študenta chybu odhaliť). Za prínosné v tomto smere veľmi nepovažujú častejšie testovanie, vytykanie nedostatkov, ale ani striktný predpis postupnosti krokov pri štúdiu daného predmetu.

Pri zisťovaní preferencií učiteľov v súvislosti s *funkciami personalizovaného a adaptívneho vzdelávacieho prostredia* sa ukázalo, že väčšina z nich zrejme nie je pripravená využiť potenciál takéhoto prostredia. Funkcie, ktoré najviac označovali za želané, väčšinou nesúviseli s personalizovaným vyučovaním. Dôvodom zrejme nie je neskúsenosť učiteľov v tejto oblasti, ale skôr uvedomenie si značne väčšej náročnosti takéhoto prístupu k vyučovaniu, čo nakoniec vyjadрили aj dvaja z respondentov, keď na otázku *Čo iné by ste chceli mať k dispozícii v personalizovanom vzdelávacom prostredí?* odpovedali: *viac času*.

Dôsledkom týchto zistení je potenciálne prehodnotenie ďalších krokov pri vývoji funkcionality nášho personalizovaného systému. Predtým ale plánujeme vyhodnotiť podobný prieskum, ktorý sme zrealizovali so študentmi. Zaujímavé môže byť napríklad porovnanie odpovedí učiteľov v oblasti personalizovaného vyučovania s odpoveďami študentov. Taktiež výsledky študentského prieskumu ohľadne očakávaných funkcií personalizovaného prostredia môžu nasmerovať jeho vývoj tak, aby lepšie spĺňalo predstavy študentov.

POĎAKOVANIE

Tento výskum bol podporený z projektu VEGA 1/0621/22 ako aj z APVV projektu APVV-20-0353. Radi by sme tiež poďakovali kolegom Lucii Budinskej a Martinovi Homolovi za pomoc pri zostavovaní dotazníka.

BIBLIOGRAFICKÉ ODKAZY

- [1] Urdaneta-Ponte, M.C., Mendez-Zorrilla, A. and Oleagordia-Ruiz, I. Recommendation systems for education: systematic review. In *Electronics*, vol. 10, no. 14, 2021, p. 1611.
- [2] Karakaya, M.Ö., Aytekin, T. Effective methods for increasing aggregate diversity in recommender systems. In *Knowledge and Information Systems*, 2018, vol. 56, p. 355–372.
- [3] Nováková, J. *Virtuální vzdělávací prostředí a jejich využití pro výuku heterogenních skupin studentů*. Dizertačná práca, Karlova univerzita, Praha, 2015
- [4] Rivera, A.C., Tapia-Leon, M. and Lujan-Mora, S. Recommendation systems in education: A systematic mapping study. In *Proceedings of the International Conference on Information Technology & Systems (ICITS 2018)*, Springer International Publishing, 2018, p. 937-947.
- [5] Dominic, M. and Francis, S. An adaptable e-learning architecture based on learners' profiling. In *International Journal of Modern Education and Computer Science*, 2015, vol. 7, no. 3, p.26.
- [6] Pávová, A. and Váňová, D. *Metódy a techniky personalizovaného vyučovania*. Metodicko-pedagogické centrum, Bratislava, 2020, pp. 20, ISBN: 978-80-565-1468-9
- [7] Švaříček, R. *Kvalitativní výzkum v pedagogických vědách*. PORTÁL sro., 2007
- [8] Creswell, J.W. *Educational research: Planning, conducting, and evaluating quantitative and qualitative research*. Pearson, 2012

Informační systémy v českém kurikulu informatiky

Information systems in the Czech Computer Science Curriculum

Daniel Lessner

Katedra geoinformatiky
a didaktiky informatiky
Univerzitní náměstí 1410/1
460 17, Liberec 1
Česko
daniel.lessner@tul.cz

Martin Prade

Katedra geoinformatiky
a didaktiky informatiky
Univerzitní náměstí 1410/1
460 17, Liberec 1
Česko
martin.prade@tul.cz

ABSTRACT

In this paper we present the concept of the Information Systems thematic area in the newly introduced Framework Educational Programmes for Czech primary schools and grammar schools. It is a somewhat unconventional topic, although it is based on databases that Czech school informatics is familiar with. It adds to them mainly the consideration of the needs of the users, or the overall usability of the developed solutions.

In this paper we discuss in more detail the expectations from the inclusion of information systems development in the curriculum. We also provide specific outcomes for teaching in secondary schools. At the end of the paper, we illustrate the expected outcomes and learning opportunities through two more specific examples.

Keywords

Computer Science Education. Information systems. Databases. Usability.

ABSTRAKT

V tomto článku představujeme pojetí tematického okruhu Informační systémy v nově zaváděných rámcových vzdělávacích programech pro české základní školy a gymnázia. Jde o okruh poněkud netradiční, jakkoliv vychází z databází, které česká školská informatika zná. Přidává k nim totiž především ohled na potřeby uživatelů, resp. celkově použitelnost (usability) vyvíjených řešení.

V článku podrobněji diskutujeme očekávání od zařazení tvorby informačních systémů v kurikulu. Uvádíme také konkrétní výstupy pro výuku na gymnáziích. Na konci článku očekávané výstupy a možnosti výuky ilustrujeme prostřednictvím dvou konkrétnějších příkladů.

Klíčové slová

Výuka informatiky. Informační systémy. Databáze. Půžitelnost.

1 ÚVOD

Na českých školách právě probíhá zavádění informatiky zaměřené na rozvoj informatického myšlení [1–3]. Většinu původního obsahu vzdělávací oblasti Informační a komunikační technologie (dále jen IKT) přebírá nově zavedená klíčová kompetence (digitální). Odpovědnost ze její rozvoj už tak neponesou výlučně učitelé informatiky. První stupně začnou nově zavedenou informatiku vyučovat od září 2023, o rok později naváží druhé stupně a o další rok později (více než deset let od veřejného zahájení prací na změnách [4]) budou připravena i gymnázia.

Nově zaváděná informatika se podle revidovaných Rámcových vzdělávacích plánů (dále jen RVP, [2]) člení na čtyři okruhy nazvané následovně:

- Data, informace, modelování
- Algoritmizace a programování
- Informační systémy
- Digitální technologie

V prvním okruhu žáci poznávají různé možnosti kódování dat a reprezentace reálných situací (např. pomocí grafů, [5]). Algoritmizace a programování je okruh pro informatiku patrně nejtradičnější, žáci nejrozličnějšími způsoby pracují s pracovními postupy a jejich zápisy, nejčastěji v podobě programů. Digitální technologie nejtěsněji navazují na zrušenou IKT. Nejde ovšem tolik o použití technologií, jako spíše o porozumění principům jejich fungování.

Okruh informačních systémů ovšem budí otázky, v kontextu výuky informatiky v Česku je nezvyklý. V tomto článku přiblížíme, jak okruh vznikl, jakou hraje v celku školské informatiky roli a jaké od něj očekáváme přínosy.

2 MOTIVACE PRO ZAVEDENÍ INFORMAČNÍCH SYSTÉMŮ DO VÝUKY INFORMATIKY

Konkrétní obsah změn ve výuce informatiky připravovala pracovní skupina spadající pod Národní pedagogický institut České republiky zvaná ICT panel. Hlavní autor tohoto článku je jedním z jejích členů. Zadáním pro nově zaváděnou vzdělávací oblast byl především rozvoj informatického myšlení žáků a cíle, které vedou k vlastnímu zkoumání a osvojování dovedností, nikoliv k pamětnímu učení. V tom se přístup nelišil od řady dalších evropských zemí [6]. Pracovní skupina se podle očekávání soustředila především na otázky algoritmizace, programování, kódování dat a modelování. Stranou si ponechala otázku digitálních technologií, protože dlouhou dobu nebylo vyjasněno, jestli a jak má být v RVP zpracován trénink uživatelských dovedností. Podobně nejprve odložila otázku „dat v tabulkách“. Zatímco pro výuku práce s texty nebo grafikou se přirozeně nabízely jiné vzdělávací oblasti, u tabulek bylo jasné, že aspoň částečně patří i do informatiky. Nebylo také prodiskutováno, které konkrétní

koncepty a dovednosti pro nově zaváděnou informatiku pod označením „data v tabulkách“ rozumět.

2.1 Tvorba databází

Pracovní skupina postupně dospěla k tomu, že pro informatiku jako obor je zásadnější a pro žáky užitečnější přístup databázový, s tabulkami stejnorodých záznamů, oproti tabulkám z nezávislých buněk známých z tabulkových kalkulátorů. Pro menší úlohy se takový přístup žákům může zdát nemotorný, lze ale nepoměrně lépe škálovat na větší množství dat. Zároveň lépe odpovídá tomu, jak data reálně zpracovávají stroje. Pracovali jsme i s předpokladem, že osvojení si pohledu na data optikou záznamů s jednotnými atributy a relacemi mezi tabulkami žákům usnadní dobře uspořádané tabulky i v tabulkových kalkulátorech.

Databáze byly navíc tradiční (byť nikterak významnou) součástí i předchozí verze [7], nabízely tedy na rozdíl od většiny nových vzdělávacích cílů jistou kontinuitu.

Následovalo zvažování a rozhodování, co přesně z databází mají žáci kdy znát. Podobně jako u ostatních oblastí informatiky, kritériem byl přínos pro myšlení žáků, praktická využitelnost, dlouhodobá platnost a možnost stanovené cíle skutečně dosáhnout.

Celou dobu máme samozřejmě na mysli nikoliv použití hotových databází, nýbrž jejich navrhování, realizaci a pokud možno i provoz.

O informačních systémech přitom ve vzdělávacím programu hovoříme proto, že v praxi samotné databáze jako sada provázaných tabulek nestačí. Žáci by měli promyšlet jejich skutečné použití v dalších potřebných souvislostech. Kromě samotných tabulek a dat v nich jde i o uživatelské rozhraní (nikoliv jeho existenci, ale jeho kvalitu) a procesy, které uživateli umožňují řešit potřeby, kvůli kterým byl systém vůbec vyvinut. I ty je potřeba umět navrhnout, implementovat a otestovat. Při této transformaci pojmů souvisejících s informačními systémy pro potřeby žáků jsme se opírali o definice z [8].

2.2 Systematická práce na úplném projektu

Tabulková data byla přebytkem z počátečního mapování oboru. Kromě nich chybělo zařadit také činnosti spojené s realizací většího projektu. Okruh algoritmizace a programování sice z rámcového postupu vychází, ale spíše kvůli samotnému strukturování cílů v textu RVP. Navíc už takhle byl daný okruh poměrně obsáhlý. Výstupy směřující k systematické práci v rozlišitelných fázích, mezi nimiž vývojář přechází a podle potřeby se vrací, proto zatím v návrhu RVP chyběly.

2.3 Zaměření na uživatele a použitelnost vyvinutých produktů

Druhý aspekt práce v informatice, který v RVP dosud chyběl, byli uživatelé výsledků žákovské práce. Řešení problémů v oblasti algoritmizace a programování mohlo probíhat pohodlně odtržené od souvislostí skutečného světa. Důvodem pro takové formulace byla opět snaha pracovní skupiny daný okruh pokud možno nepřetížít a nekomplikovat.

Zároveň by ale bylo chybou ohled na uživatele zcela vynechat. I pro žáky, kteří by snad digitální produkty pro jiné lidi nevytvářeli, je do budoucna užitečné si uvědomit svou roli jako uživatelů či klientů a nastavit odpovídající očekávání vůči vývojářům. Přínos pro žáky, kteří digitální produkty (a přeneseně i jakékoliv jiné, včetně služeb) vytvářet budou, je zřejmý.

Orientace na uživatele a použitelnost produktu (usability) navíc užitečným způsobem posouvá pojetí jinak poměrně logicky založeného předmětu. Žáci, kteří nachází zalíbení právě v abstraktních nebo technických aspektech informatiky, zjistí, že formální správnost výsledku není jediným a už vůbec ne dostatečným kritériem kvality výsledků jejich práce: správně fungující produkt, který ale uživatelé nedovedou používat, je v důsledku stejně zbytečný, jako produkt nefunkční.

Z druhé strany tento pohled může nabídnout vítanou šanci najít v informatice zalíbení pro žáky, kteří technicky zaměření nejsou, ale také se chtějí zapojit a být užiteční. A pro všechny je nakonec obohacující zjištění, že např. povedené uživatelské rozhraní není výsledkem úvah „zdravým rozumem“, ale pokorného uživatelského testování v celém průběhu práce na produktu. Komplexní zpracování tématu na středoškolské úrovni nabízí učebnice [9].

2.4 Dopady technologií

V Česku poněkud nezvyklou, ve světě ovšem stále významnější součástí výuky informatiky jsou otázky dopadu rozvoje a použití technologií na životy jednotlivců i společnosti jako celku. Žáci se učí si dopadů všimnout, jak těch předpokládaných a záměrných, tak i těch nezamýšlených, nepřímých, někdy nepředvídatelných. Samozřejmě na to navazují otázky odpovědnosti tvůrců, kterými se žáci přinejmenším v hodinách informatiky stávají.

Žáci by měli vysledovat i obecnější mechanismy, které se v podobných případech uplatňují opakovaně. Patří mezi ně záměna metriky a cíle, která pokrývá motivace a tím i rozhodování a chování autorů i uživatelů. Obecněji lze hovořit o incentivech, které (nejen) informační systémy pro jednotlivé skupiny vytváří. Příkladem je optimalizace algoritmů sociálních sítí a její důsledky.

Jiným užitečným směrem úvah je identifikace skupin, které naše technologie ovlivňují nepřímo a zprvu vůbec nebyly brány v úvahu. Příkladem jsou obyvatelé jinak klidných čtvrtí, kterými navigační aplikace vedou řidiče, aby se vyhnuli zácpám.

2.5 Nezařazené aspekty informačních systémů

Řada dalších souvislostí informačních systémů je relevantní, ale z různých důvodů je pracovní skupina nezařadila vůbec, nebo jen jako zmínku bez odpovídajícího očekávaného výstupu.

Zásadní vlastností informačních systémů je jejich zabezpečení. To žáci samozřejmě zvažují, základní způsoby zabezpečení znají ze systémů, které sami používají, včetně např. zacházení s hesly a se souvisejícími zásadami (to si osvojují v rámci rozvoje digitálních kompetencí). Nezdá se ale reálné, aby dokázali samostatně implementovat opravdu bezpečné systémy nad rámec hotových řešení typu nastavení přístupových práv ke sdílené tabulce. Zabezpečení tedy hodnotí pouze v tom smyslu, že rozpoznají nejběžnější chyby.

Další důležitou souvislostí jsou Big Data. V souvislosti s nimi pracovní skupina neidentifikovala žádnou specifickou dovednost, která by vyžadovala zařazení do RVP.

Pokročilým tématem jsou normální formy databázových tabulek. Do kurikula zařazené nebyly jako příliš abstraktní. Žáci se nicméně učí zvažovat důsledky různě strukturovaných schémat, jejich výhody a nevýhody.

V souvislosti s informačními systémy pracovní skupina diskutovala také tematiku umělé inteligence a strojového učení (právě nad velkými soubory dat). Neočekává ale, že by měli žáci v této souvislosti získat nějaké konkrétní dovednosti, natož aby

využití takových metod pro vyvíjený informační systém sami navrhovali. Strojové učení proto mezi výstupy zařazeno nebylo.

3 OČEKÁVANÉ VÝSTUPY VÝUKY NA GYMNÁZIÍCH

Na základě výše popsaných potřeb pracovní skupina zformulovala následující očekávané výstupy a učivo v oblasti informačních systémů. Připomeňme, že pro informatiku jako pro novou oblast nemáme sdílenou zkušenost a sdílený jazyk [10]. Ze samotných formulací proto nemusí být zřejmé, jaká úroveň pokročilosti výstupů nebo hloubky porozumění se očekává. Náповědou budiž srovnání s již zavedenými obory: žáci mají informačním systémům rozumět podobně, jako se učí rozumět spalovacím motorům, fotosyntéze nebo legislativnímu procesu. Jinými slovy jde o základní principy fungování a základní souvislosti, nikoliv možnost profesionální práce.

Očekávané výstupy:

žák

- rozpozná informační toky v systémech; analyzuje a hodnotí informační systémy z různých hledisek; zvažuje i nepřímé a nezamýšlené dopady informačního systému na různé skupiny
- nastavuje účelné zobrazení dat, filtruje a řadí data úpravou databázového dotazu
- určí cílovou skupinu, formuluje problém, validuje potřeby, určí a prioritizuje požadavky na řešení
- určí jednotlivé uživatelské role, specifikuje jejich činnosti, navrhne, otestuje a přizpůsobí rozhraní uživatelům
- navrhne a vytvoří strukturu vzájemného propojení tabulek; navrhne procesy zpracování dat
- otestuje správnost a použitelnost svého řešení, navrhne a realizuje potřebná vylepšení; během provozu informačního systému rozpozná funkčně či věcně nesprávný stav, zjistí jeho příčinu a navrhne způsob jeho odstranění

Učivo:

- informační systémy** – informační systém – data, jejich struktura a vazby, definované procesy, role uživatelů, technické řešení informačních systémů; veřejné informační systémy
- hromadné zpracování dat** – tabulka, její struktura – data, hlavička a legenda; řazení a filtrování dat, rozpoznávání vzorů a trendů v datech, vizualizace dat; velká data – zdroje, metody zpracování, využití
- vývoj informačního systému** – postup tvorby informačního systému; návrh uživatelského rozhraní, datového modelu a procesů; návrh databázové tabulky, atributy polí, primární klíč; návrh struktury a propojení více tabulek – cizí klíč, relace

4 OČEKÁVANÉ VÝSTUPY VÝUKY NA ZÁKLADNÍCH ŠKOLÁCH

V této části uvedeme očekávané výstupy a učivo pro deváté třídy základních škol. Níže potom okomentujeme rozdíl vůči gymnáziím. Popíšeme také způsob, jak k nim pracovní skupina došla. Odpovídá totiž zároveň tomu, jak bylo postupováno u ostatních okruhů.

Očekávané výstupy

žák

- vysvětlí účel informačních systémů, které používá, identifikuje jejich jednotlivé prvky a vztahy mezi nimi; zvažuje možná rizika při navrhování i užívání informačních systémů
- nastavuje zobrazení, řazení a filtrování dat v tabulce, aby mohl odpovědět na položenou otázku; využívá funkce pro automatizaci zpracování dat
- vymezí problém a určí, jak při jeho řešení využije evidenci dat; na základě doporučeného i vlastního návrhu sestaví tabulku pro evidenci dat a nastaví pravidla a postupy pro práci se záznamy v evidenci dat
- sám evidenci vyzkouší a následně zhodnotí její funkčnost, případně navrhne její úpravu

Učivo

- informační systémy:** informační systém ve škole; uživatelé, činnosti, práva, struktura dat; ochrana dat a uživatelů, účel informačních systémů a jejich role ve společnosti
- návrh a tvorba evidence dat:** formulace požadavků; struktura tabulky, typy dat; práce se záznamy, pravidla a omezení; kontrola správnosti a použitelnosti struktury, nastavených pravidel; úprava požadavků, tabulky či pravidel
- hromadné zpracování dat:** velké soubory dat; funkce a vzorce, práce s řetězci; řazení, filtrování, vizualizace dat; odhad závislostí

Při specifikaci očekávaných výstupů pro nižší stupně se typicky vycházelo z již hotových stupňů vyšších. Aby se udržela provázanost mezi stupni škol, pracovalo se s kontinuem rozvoje dílčích dovedností. Jednotlivým výstupům (nebo častěji jejich částem) odpovídá okruh dovedností, které žáci rozvíjí a prohlubují po celou dobu základní a střední školy. Tohle také ulehčilo formulaci výstupů. Nebylo potřeba znovu řešit, co znamená umět např. právě informační systémy. Místo toho skupina hledala odpovědi na dílčí otázky, např. na jaké úrovni má žák deváté třídy umět nastavovat účelné zobrazení dat.

Přitom se skupina opírala o několik zásad. Předně má být požadovaná úroveň dosažitelná pro běžného žáka. Zároveň má být přiměřeně relevantní pro jeho (nejen budoucí, ale i současný) život. Zkoumané koncepty a osvojované dovednosti by měly mít přesah i mimo informatiku jako takovou, měly by být rozvíjející i v obecnějším smyslu. Z formulací výstupů musí být také patrný kvalitativní pokrok mezi jednotlivými stupni vzdělávání. Nestačí jen neurčitě zvyšování obtížnosti řešených úloh, o to se učitel při plánování výuky nemá, jak opřít.

Výstupy na nižších stupních vzdělávání zároveň sledují obecnou koncepci. Na prvním stupni se žáci seznamují se základními koncepty, způsobem kladení otázek a uvažování v informatice. Často pracují bez technologií a formou her. Získávají zkušenosti, teorie je naprosté minimum. Příklady takové práce v kontextu tabulkových dat nabízí učebnice [11].

Na druhém stupni se zkušenosti množí, žáci již mají co zobecňovat. Teprve na střední škole přibývají případné pojmy, teorie a abstraktnější koncepty, které ovšem díky bohatství předchozích zkušeností žákům dávají smysl.

V případě informačních systémů a jejich tvorby tak RVP pro základní vzdělávání hovoří pouze o návrhu a vedení „evidenci dat“ – právě proto, aby se učitelé neopírali o doslovné definice a nebyli tlačeni například do práce s aplikacemi, na které by žáci ještě nebyli připraveni. V rámci podpory škol při zavádění informatiky Národní pedagogický institut uvádí, že taková evidence dat může být i papírová. Podstatné je, aby dovolila obdobnou práci jako informační systémy elektronické. To plyne jednoduše z toho, že systematicky evidujeme data za nějakým výslovně deklarovaným účelem, ať už s počítačem, nebo bez.

5 SPECIFIKA INFORMAČNÍCH SYSTÉMŮ V KONTEXTU OSTATNÍCH OKRUHŮ

Při práci na RVP je na místě si opakovaně pokládat otázku, jestli daný vzdělávací obor, oblast nebo výstup přináší něco tak hodnotného a jedinečného, aby bylo potřeba to výslovně uvádět a následně očekávat od všech žáků. Nešlo by obdobných výsledků dosáhnout jiným způsobem. Proto se i v případě informačních systémů se nabízí otázka, jestli bychom nevystačili s ostatními třemi okruhy.

Klíčové koncepty, kterými se tento okruh od ostatních liší (data v tabulkách, použitelnost) jsme již uvedli. Zde okomentujeme ještě několik dalších specifík, která činí informační systémy hodnotnou součástí kurikula informatiky.

Výsledky práce v okruhu informačních systémů jsou na rozdíl od výsledků v programování častěji dotaženy k praktické použitelnosti (už jen proto, že je to snazší). Řeší konkrétní potřebu konkrétních lidí. Díky tomu žáci dostávají lepší šance svou práci ověřovat v reálnějším kontextu.

Hodnota okruhu informační systémy pro žáky tkví i v tom, že se s jejich tvorbou v nějaké podobě (třeba i jen formou sešitů v tabulkových procesorech) běžní žáci setkají mnohem častěji než s programováním jako takovým.

Dále uveďme, že zatímco programy běží typicky krátce a skončí, informační systémy fungují dlouhodobě. To žáky stává před nové, zajímavé výzvy. Relevantní otázkou se stává údržba a průběžné vylepšování systému, množství dat, které se za delší dobu nasbívá, přehlednost rozhraní, rozlišování časových období (např. školních nebo účetních roků) apod.

Uvedené aspekty lze ve výuce otevírat i v kontextu programování, ale není to tak snadné a přímočaré. V RVP je programování zaměřené jinak, na postupy a algoritmicizaci jako takovou – a nechceme, aby se ztratila vedle všeho ostatního. Předpokládáme, že se na uvedené souvislosti učitelé zaměří spíše v souvislosti s informačními systémy, kde je to navíc o něco přirozenější.

Na závěr této části poznamenejme, že RVP výstupy strukturuje do v úvodu zmíněných čtyř okruhů pro přehlednost. Z ničeho neplyne, že by bylo třeba je vyučovat izolovaně. Je naopak je velmi vhodné rozvíjené dovednosti kombinovat podle potřeb žáků a podle problémů, které zrovna řeší. I programy je vhodné tvořit s ohledem na to, jak s nimi budou pracovat uživatelé, i při vývoji informačních systémů a navrhování struktury tabulek modelujeme, při návrhu (ideálně automatizovaných) procesů zpracování dat algoritmicizujeme, celou dobu využíváme digitální technologie atd.

6 PŘÍKLADY

Výše uvedené informace ilustrujeme dvěma příklady z výuky. První je využíván v úvodu k tematickému okruhu, nevyžaduje žádné digitální zařízení. Pomáhá žákům si uvědomit, že principy na

použité technologii tolik nezáleží. Technologie také žáky nerozptyluje, vše se děje ručně, nikde neprobíhá žádná „kouzelná“ automatizace.

Jako druhý příklad jsme naopak zvolili konkrétní technické řešení žákovského projektu. Ukazuje zároveň možnosti aplikace Coda.io.

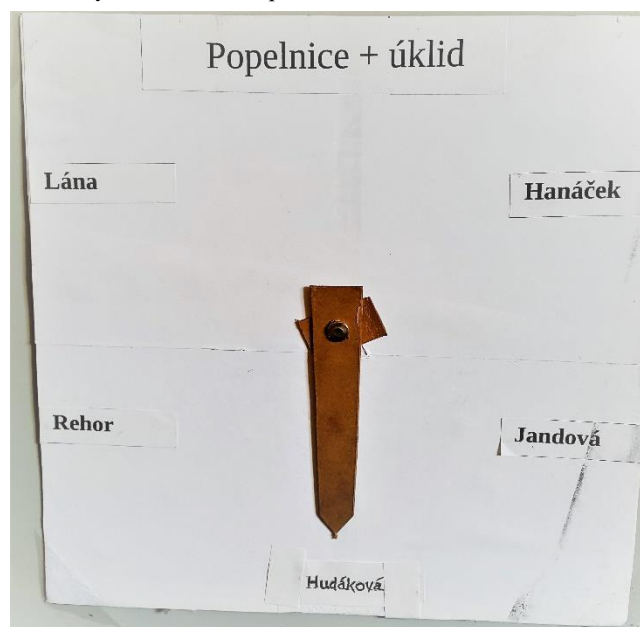
6.1 Úklid na chodbě

První příklad se obejde bez databáze. Volíme jej (zde i ve výuce) právě proto, že dobře ilustruje otázky, které si v souvislosti s informačními systémy klademe. Nejde totiž nutně o otázky technické či abstraktní.

Žáci dostanou za úkol navrhnout řešení pro férové uklizení chodby s několika bytovými jednotkami. Většinou vymyslejí různé varianty tabulek s označením bytů a datumy, objevují se i návrhy vytvořit aplikaci nebo úklid zaplatit. Pokud s tím nepřijdou sami, nabídne učitel i otočnou ručičku (viz obrázek 1).

Žáci různá řešení porovnávají a hodnotí. Zároveň si tím ujasňují, co jsou vlastně kritéria kvality takového systému. Mnohdy se ukáže, že už ten požadavek spravedlivosti je potřeba prodiskutovat pečlivěji, že pravidelné střídání nemusí odpovídat skutečné situaci.

Co se týče technické stránky, identifikují žáci součásti a fungování cedule s ručičkou jako informačního systému. Rozmýšlejí, jaká data a jakým způsobem systém „ukládá“, na jakém je postaven „hardware“, jak vypadají související procesy (co přesně znamená poloha ručičky, kdo a za jakých okolností ji mění, na čem stojí důvěra v informace na ceduli). Jaké konkrétní úkoly umožňuje splnit, resp. jaké uživatelům poskytuje služby. Zkouší zformulovat návod k použití a zjišťují, že je to potřeba, protože intuíce různých lidí vede k různým postupům. Srovnávají výhody a nevýhody oproti jiným řešením. Domýšlejí, co a proč se může pokazit, resp. k jakým neobvyklým situacím může dojít a jak je lze se kterým typem cedule řešit (výpadky v úklidu kvůli dovolené či nemoci, ztráta cedule, vandalismus, nedostatečný úklid a různé podvody, příliš opotřebovaná osička znevýhodňující paní Hudákovou, děti, které na ručičku nedosáhnou...). Navrhují vylepšení cedule, která umožní tyto situace řešit lépe.






Obrázek 1: Příklad řešení pro férový úklid na chodbě

Tento příklad ukazuje, že i zcela jednoduché „systémy“ otvírají zajímavé a hodnotné otázky. Zároveň žákům umožňují se na ně soustředit lépe než u technicky pokročilejších systémů.

6.2 Připomínač léků v Coda.io

Druhý příklad je výsledkem žakovské práce. Slouží k připomínání léků. Na obrázku 2 vidíme rozhodující část uživatelského rozhraní (konkrétní léčiva jsou zvolena nesmyslně).

Jméno léku	Den	Za jak dlouho беру lék	Balení	Vzít lék
Brufen	Dnes	0 hrs 27 mins ago		<input type="button" value="Vzal jsem lék"/>
Stacyl	Dnes	2 hrs 3 mins		<input type="button" value="Vzal jsem lék"/>
Oscillococnium	Zítřa	14 hrs 33 mins		<input type="button" value="Vzal jsem lék"/>

Obrázek 2: Příklad elektronického informačního systému

Žák při tvorbě tohoto systému řešil principiálně stejné otázky, s tím, že technické řešení je samozřejmě značně pokročilejší. Na středoškolské úrovni je ale dosažitelné. Tato konkrétní práce ani nevyžadovala relaci mezi dvěma tabulkami. Na druhé straně ale pracuje s interakcí pomocí tlačítek. Tlačítko se aktivuje, když je čas vzít lék (ne dřívě). Jeho stisk potom přenastaví čas příštího brání léku o určenou časovou periodu později. Od toho se odvíjí obsah dalších sloupců (zejm. výpočet zbývajících času) a zpětně i opětovná aktivace tlačítka.

Žák svůj projekt zpracoval v aplikaci Coda.io. Ta je určená pro tvorbu interaktivních dokumentů například v malých firmách, přesto (nebo právě proto) nabízí dobře vyvážené možnosti pro výuku informačních systémů na školách. Tabulky jsou databázové v tom smyslu, že všechny řádky tabulky mají stejné sloupce a tabulky lze mezi sebou provazovat. Navíc Coda umožňuje různé tvořit pohledy na stejná data (pro různé potřeby – např. přehled všech léků seřazených podle abecedy, léků, které máme brát dnes v pořadí podle času, léků, které už jsme dnes vzali atp.). Většina takových úkolů se dá splnit snáze než v klasickém tabulkovém kalkulátoru a snáze než v klasické databázi.

Databázové pojetí tabulek umožňuje také výrazně přehlednější a příjemnější zápis vzorců. Ty jsou jednotné pro celý sloupec, problematika absolutních a relativních odkazů v databázi nedává smysl. Na ostatní data se navíc neodkazujeme souřadnicemi, ale lidsky srozumitelným pojmenováním. Další usnadnění plyne z tečkové notace, díky níž lze řadu funkcí volat jako metodu dat, a ušetřit si tak vnořování a počítání uzavíracích závorek. Příklad viz obrázek 3 se vzorcem pro sloupec *Den* v tabulce z obrázku 2. Zelené pole značí použití obsahu sloupce *Kdy budete brát lék* z téhož řádku tabulky. Ikona na konci zeleného pole značí datový typ datum.

```
If(
  Kdy budete brát lék .DateTimeTruncate("day") = Today(),
  "Dnes",
  If(
    Kdy budete brát lék .DateTimeTruncate("day") = Today() + 1,
    "Zítřa",
    "Jindy"
  )
)
```

Obrázek 3: Ukázka vzorce v Coda.io

Další výhody Cody jsou na straně vstupů. Coda integruje formuláře, nabízí tlačítka, která po stisku automaticky mění data v tabulkách podle požadavků uživatele. Další automatizované úpravy lze spouštět periodicky podle času (např. každou nedělní půlnoc připravit řádek tabulky pro další týden) nebo na základě specifické změny v datech. Všechny změny se obratem promítají do všech souvisejících tabulek. Jako poslední zmíníme možnost integrace s dalšími nástroji a službami. Žáci například (jen zkusmo) implementovali do správce domácích úkolů automatické posílání e-mailů učitelům s předdefinovanou pokornou žádostí o posun termínu, pokud úkol neoznačili včas jako splněný.

Celkově se Coda jeví jako ideální nástroj pro výuku informačních systémů. Obsahuje dostatečně silné databázové mechanismy (lze dokonce hovořit i o efektivitě různých návrhů struktury dat) a zároveň umožňuje velmi flexibilní zobrazování dat (skrývání sloupců i filtrování řádků, řazení, podmíněné formátování a další) a bohaté možnosti interaktivity vůči uživatelům. Přitom nevyžaduje příliš hluboké technické dovednosti, všechny základní prvky lze spravovat přímo prostřednictvím grafického uživatelského rozhraní. Pro pokročilejší věci je potřeba si osvojit jazyk použitý pro vzorce. To ale není nadměrně obtížné a v důsledku to jen posílí porozumění tomu, jak celá věc funguje.

6.3 Další náměty

Pro lepší představu uvedeme ještě další náměty, které žáci zpracovávají. Některé jsou nápadem učitelů, některé naopak reálně zpracované projekty žáků, které zpravidla plynou z potřeb někoho z jejich okolí.

Náměty záměrně umožňují různou úroveň pokročilosti. Ve skutečnosti umožňují také velmi různé způsoby řešení. To je záměr, je úkolem žáků zvolit, jak se k problému postaví, co vše a jak příjemně uživateli umožní.

Například půjčovnu lze řešit jedinou tabulkou, kde k jednotlivým položkám můžeme vepisovat, kdo je má půjčené. Můžeme také přidat, odkdy výpůjčka trvá. Z toho případně počítat, jak je to dlouho. Při vrácení obsah těchto sloupců vyprázdňujeme. Když si chceme evidovat i kontakt, musíme ho do tabulky vepisovat pokaždé znovu. Když se tomu budeme chtít vyhnout, založíme druhou tabulku, kde můžeme spravovat data o lidech, a z tabulky věci k vypůjčení se do tabulky lidí jen odkazovat. To nám ale pořád neumožní pracovat s historií výpůjček, což ale uživatel možná chce. Pro tu bychom museli například založit tabulku výpůjček jako takových, a z ní se odkazovat do tabulek s věcmi a s lidmi.

Podobnou škálu různě pokročilých řešení nabízí i ostatní náměty. Poznamenejme ještě, že v předchozím odstavci komentujeme potřeby uživatele ve vztahu ke struktuře databáze. V souladu s uvedenými úpravami v tabulkách je ale třeba pokaždé revidovat i související procesy (kdy, kam a co přesně do které tabulky zapisujeme – jak probíhá výpůjčka, vrácení, urgence, inventura

atp.) a uživatelské rozhraní (jaká data zobrazovat, jakou formou, jakou nabídnout interaktivitu, aby to uživatelé nezahltlo, a naopak podpořilo hladké a správné provádění potřebných úkonů).

Nyní už k samotným příkladům možných nebo skutečných žákovských projektů:

- Evidence domácí knihovny, filmotéky, ...
- Správce (nejen školních) úkolů, pro jednotlivce nebo tým.
- Hlasování, třeba o tématu k diskusi, o otázkách pro významného hosta, o deskové hře pro daný večer, ...
- Rodinný nákupní seznam.
- Sledování rodinných výdajů.
- Sledování spotřeby automobilu.
- Plánování rodinného nebo skupinového výletu, nebo jiné akce.
- Rozdělování služeb na domácí práce (mytí nádobí, vyklízení myčky, venčení psa, ...).
- Správa členů klubu.
- Pomůcka pro hraní Dungeons&Dragons.
- Tréninkový deník.
- Sběrka hokejových kartiček, známek, DVD, rubikových kostek, krabiček od zápalek, čajů, Pogů, ...
- Půjčovna (čehokoliv).
- Přehled školního hodnocení.
- Evidence docházky, počítání absencí.
- Plánovač dárků, párovač přání a dárků, secret Santa.
- Evidence skladových zásob, třeba malé kavárny nebo e-shopu.
- Receptář s filtrováním podle dostupných surovin.
- Pomůcka pro rozhodčího třeba sportovní gymnastiky nebo krasobruslení.
- Kombinování outfitů, hráčů ve sportovním týmu, ...
- Kalkulačka pro výpočet ceny dortu podle vrstev, příchutí a dalších voleb.

7 ZÁVĚR

V článku jsme popsali pojetí okruhu informačních systémů ve výuce informatiky aktuálně zaváděné do českých základních a středních škol. Shrnuli jsme okolnosti jeho vzniku a výsledek popsaného procesu. Jmenovali jsme podstatné přínosy, které zařazení informačních systémů do výuky přináší. Na jednu stranu jsou to databáze, cílem práce v tomto okruhu je ale také vývoj použitelných řešení reálných potřeb reálných uživatelů.

Protože bez předchozí zkušenosti nemusí být zřejmé, jakou úroveň pokročilosti RVP očekává, uvedli jsme pro představu dvě konkrétní ukázky.

BIBLIOGRAFICKÉ ODKAZY

- [1] VANÍČEK, J. Towards a compulsory computing curriculum at primary and lower-secondary schools: the case of Czechia. In: Barendsen, E., Chytas, Ch. (eds): *Informatics in Schools. Rethinking Computing Education. ISSEP 2021*. Lecture Notes in Computer Science, vol. 13057. Springer, Cham, 2021. s. 109 - 120.
- [2] MŠMT: *Opatření ministra školství, mládeže a tělovýchovy, kterým se mění Rámcový vzdělávací program pro základní vzdělávání*. MŠMT, Praha, 2021. https://www.msmt.cz/file/54860_1_1.
- [3] LESSNER, D. Analýza významu pojmu „Computational Thinking“. *Journal of Technology and Information Education*. 2014, 6 (1), s. 71–88.
- [4] MŠMT. *Strategie digitálního vzdělávání do roku 2020*. Ministerstvo školství, mládeže a tělovýchovy ČR, Praha, 2014.
- [5] VANÍČEK, J, LESSNER, D. Bobřík učí informatiku (2. díl seriálu – Procházení grafů). *MATEMATIKA-FYZIKA-INFORMATIKA*. Olomouc, 2014, 23(2), s. 147–158.
- [6] BALANSKAT, A., ENGELHARDT, K. *Computing our future*. Brussels, European Schoolnet, 2015.
- [7] VÚP. *Rámcový vzdělávací program pro gymnázia*. Výzkumný ústav pedagogický v Praze, Praha, 2007. ISBN 978-80-87000-11-3.
- [8] MOLNÁR, Z. *Podnikové informační systémy*. Praha: ČVUT, 2009.
- [9] BELL, T. a MORGAN, J. *Computer Science Field Guide*. University of Canterbury, Canterbury, 2012. <http://www.cosc.canterbury.ac.nz/csfieldguide/>.
- [10] VANÍČEK, Jiří and Miroslava ČERNOCHOVÁ. Didaktika informatiky na startu. In: STUHLÍKOVÁ and JANÍK, eds. *Oborové didaktiky v České republice: stav, vývoj a perspektivy*. MuniPress, Brno (2015). ISBN 80-210-7769-0.
- [11] FILIPI, Z., MAINZ, D., FADRHONC, J. *Práce s daty pro 5. až 7. ročník základní školy*. Západočeská univerzita, Plzeň, 2020. <https://pracesdaty.zcu.cz>.

IT v Nitre: Súťaž v programovaní pre stredoškóľákov

IT in Nitra: Programming Contest for Secondary Schools

Gabriela Lovászová
FPVaI UKF v Nitre
Tr. A. Hlinku 1
949 01 Nitra
Slovensko
glovaszova@ukf.sk

Viera Michaličková
FPVaI UKF v Nitre
Tr. A. Hlinku 1
949 01 Nitra
Slovensko
vmichalickova@ukf.sk

ABSTRACT

Programming contests are an important motivating factor for learning. In Slovakia, there are several competitions for secondary school pupils having focus on different target groups and with different forms of organization. The aim of this paper is to introduce a new programming contest IT in Nitra, which extends the offer of competitions with a new format. Rules of the competition, types of tasks, organization details and evaluation of the results of previous contest's editions are presented. In discussion, we include some observations on knowledge and skills of the competing students as well as reflection on the preparation and implementation of the contest.

Keywords

Programming Contest. Python. Automatic Evaluation of Solutions.

ABSTRAKT

Súťaže v programovaní predstavujú významný motivačný faktor pre učenie sa. Na Slovensku existuje viacero súťaží určených pre žiakov stredných škôl so zameraním na rôzne cieľové skupiny žiakov a s rôznymi formami organizácie. Cieľom príspevku je predstaviť novú súťaž v programovaní IT v Nitre pre stredoškóľákov, ktorá rozširuje ponuku súťaží o nový formát. Predstavené sú pravidlá súťaže, typy úloh, organizácia a zhodnotenie výsledkov doterajších dvoch ročníkov súťaže. Diskusia k hodnoteniu výsledkov obsahuje niektoré zistenia o vedomostiach a zručnostiach súťažiacich žiakov a reflexiu týkajúcu sa prípravy a realizácie súťaže.

Kľúčové slová

Súťaž v programovaní. Python. Automatické vyhodnocovanie riešení.

1 ÚVOD

Súťaž je formou vyučovania, ktorá má významný prínos pre motiváciu žiakov učiť sa. Súťaže v programovaní sú pre žiakov príležitosťou konfrontovať svoje vedomosti a zručnosti so žiakmi z iných škôl a regiónov, riešiť iné ako štandardné školské úlohy, zažiť súťažnú atmosféru, nadviazať priateľské kontakty so žiakmi s podobnými záujmami.

Pre žiakov stredných škôl existuje na Slovensku viacero súťaží v programovaní. Najprestížnejšou súťažou je Olympiáda v informatike [1]. Ide o viackolovú súťaž jednotlivcov. Súťažiaci riešia najprv úlohy v domácom kole, najlepší postupujú do krajského a potom do celoštátneho kola. Z úspešných riešiteľov celoštátneho kola sa na sústredení vyberá tím, ktorý reprezentuje Slovensko na medzinárodnej olympiáde. V olympiáde sa riešia náročné úlohy zostaviť algoritmus s najlepšou výpočtovou zložitouťou. Súťaž má dve kategórie, okrem hlavnej kategórie A,

z ktorej je možné postúpiť na medzinárodnú olympiádu, sa súťaží v kategórii B určenej pre nižšie ročníky s menšou náročnosťou úloh. Forma súťaže je prezenčná, okrem domáceho kola, ktoré je korešpondenčné.

Ďalšími súťažami s podobným zameraním úloh ako Olympiáda v informatike (správnosť a výpočtová zložitnosť algoritmu) sú Korešpondenčný seminár z programovania (KSP) [2] a Zenit v programovaní [3]. Obe súťaže sú viackolové. KSP je korešpondenčná súťaž so záverečným viacdňovým sústredením, na ktoré sú pozvaní najlepší riešitelia. Zenit v programovaní je prezenčná súťaž s automatickým vyhodnocovaním riešení.

Na programovanie v jazyku Python sú zamerané súťaže Python Cup [4] a Palma Junior [5, 6]. Python Cup je jednokolová súťaž s úlohami na programovanie grafiky, organizuje sa on-line v prostredí domovských škôl súťažiacich. Palma Junior je súťaž jednotlivcov alebo dvojčlenných tímov v dvoch kategóriách. Organizuje sa niekoľko školských kôl formou on-line a jedno finálové kolo prezenčne.

V tomto príspevku predstavíme novú súťaž v programovaní pre žiakov stredných škôl IT v Nitre [7].

2 Súťaž IT v Nitre

IT v Nitre je súťaž v programovaní v jazykoch Python, C a Java, ktorú organizuje Katedra informatiky na Fakulte prírodných vied a informatiky Univerzity Konštantína Filozofa v Nitre. Je určená pre všetkých študentov stredných škôl so záujmom o programovanie. Koná sa prezenčne v počítačových učebniach katedry (s výnimkou 1. ročníka, kedy sa súťaž kvôli pandemickým opatreniam realizovala on-line). V roku 2023 sa uskutoční jej 3. ročník.

2.1 Pravidlá súťaže

Súťaž je zameraná na testovanie algoritmického myslenia a programátorských zručností vo zvolenom programovacom jazyku. Súťažiaci pracujú v on-line prostredí systému Priscilla [8]. Vytvárajú programy na riešenie desiatich rôzne náročných problémových úloh. Na riešenie týchto úloh majú 90 minút, úlohy môžu riešiť v ľubovoľnom poradí. Správnosť riešenia sa overuje automaticky porovnaním výstupov vytvoreného programu s výstupmi vzorového programu pre rovnaké vstupy. Súťažiaci dostáva informáciu o miere správnosti svojho riešenia priebežne počas súťaže, a to po každom pokuse o odovzdanie svojho programu. Svoj program tak môže opravovať a odovzdávať opakovane, do výsledkov sa započítava najlepšie ohodnotená verzia riešenia. Počas súťaže nie je dovolené používať iné pomôcky ako papier a pero. Pre úspech v súťaži je dôležitá nielen správnosť odovzdaných riešení, ale aj čas potrebný na ich naprogramovanie. Za každú ušetrenú minútu získava súťažiaci 3 body navyše. Časový

bonus sa priznáva len tým riešiteľom, ktorí dosiahnu aspoň 30% z celkového počtu bodov. Hodnotiacia komisia ihneď po ukončení súťaže vyhodnotí riešenia a posúdi správnosť automaticky pridelených bodov. Úspešným riešiteľom sa stáva každý súťažiaci, ktorý získa minimálne 30% bodov za vyriešené úlohy.

Registrácia pre školy je dostupná na webovej stránke www.itvnitre.sk. Systém Priscilla, v prostredí ktorého súťaž prebieha, je voľne dostupný na adrese <https://priscilla.fitped.eu/>. Študenti sa doň môžu prihlásiť už pred súťažou a vyskúšať si, ako budú počas súťaže pracovať. K dispozícii je aj archív súťaže so zadaniami úloh z prechádzajúcich ročníkov.

2.2 Typy úloh

Súťažné úlohy vychádzajú z obsahu vyučovania programovania na stredných školách. V ľahších úlohách je potrebné spracovať jednoduchší vstup, zvyčajne vo forme reťazca (overiť nejaké kritérium, získať z reťazca informácie potrebné na výpočet požadovanej hodnoty, transformovať vstupný reťazec do iného formátu a pod.). V náročnejších úlohách sú na vstupe štruktúrované dáta – postupnosti alebo tabuľky. Výstupom požadovaným v zadaní môže byť hodnota jednoduchého, ale aj štruktúrovaného typu. Zadaná sú uvádzané vo forme krátkeho textu, ktorý stručne opisuje problém s uvedením viacerých príkladov vstupných a výstupných dát.

Pri hodnotení úloh súťažiaci nevidia, na akých vstupných dátach sa jeho program testoval a aké výstupné hodnoty sa očakávali. Súťaž má dynamický charakter. Súťažiaci riešia pomerne veľký počet úloh v relatívne krátkom časovom intervale, vyberajú si ich v ľubovoľnom poradí, riešenie vytvárajú často iteratívne podľa výsledkov overenia získaných zo servera po poslednom pokuse o odovzdanie. Výhodu majú tí súťažiaci, ktorí sú schopní rýchlo čítať s porozumením a sú zruční v programovaní niektorých typických prvkov riešení súvisiacich najmä s načítaním vstupov a prípravou výstupu v požadovanom tvare, príp. s usporadúvaním dát pomocou vstavaných funkcií a písaním vlastných funkcií.

2.3 Organizácia

Technicky je súťaž zabezpečovaná systémom Priscilla [8], ktorý bol vyvinutý v rámci medzinárodného projektu Work-Based Learning in Future IT Professionals Education (FITPED) [9]. Ide o komplexnú mikrolearningovú platformu založenú na aktívnom učení sa a gamifikácii zameranú primárne na podporu vyučovania programovania vo viacerých programovacích jazykoch. Aktuálne sa rozširuje o kurzy z rôznych oblastí umelej inteligencie. Systém Priscilla sa používa prostredníctvom webového prehliadača a využívajú ho študenti vysokých (ale aj stredných) škôl vo viacerých krajinách. V špecializovaných kurzoch integruje série mikrolekcií s praktickými programátorskými zadaniami. V mikrolekciách sa strieda nový obsah s aktivizujúcimi, interaktívnymi prvkami (otázkami a krátkymi úlohami rôznych typov) zameranými na zapamätanie a porozumenie vysvetlených pojmov, zápisov a súvislostí. Praktické zadaná umožňujú aplikovanie nových vedomostí v praxi (vyžaduje sa napísanie zdrojového kódu riešenia). Správnosť riešení sa vyhodnocuje automaticky na základe pripravených testovacích prípadov. Študenti získavajú za činnosti v systéme body, svoje výsledky si môžu porovnať v rebríčku úspešnosti s ostatnými študentmi. Systém podporuje realizovanie testov a súťaží, obsahuje tiež nástroje užitočné pre analýzu edukačných dát [10].

Počas súťaže sú pre sprevádzajúcich učiteľov ako aj ďalších zúčastnených realizované workshopy a prednášky zamerané na

moderné technológie a inovatívne didaktické postupy vo vyučovaní programovania, ale aj iných oblastí informatiky.

3 Zhodnotenie predchádzajúcich ročníkov súťaže

V školskom roku 2019/2020 prebehlo pilotne nultý ročník súťaže, ktorého sa zúčastnilo 29 stredoškôľakov a 8 študentov bakalárskeho stupňa v študijnom programe Aplikovaná informatika. Cieľom nultého ročníka súťaže bolo otestovať pravidlá a formu súťaže. Súťažilo sa v jazykoch Java a Python. Na základe skúseností a spätných väzieb sme upravili pravidlá súťaže. V nasledujúcich dvoch školských rokoch prebehli dva ročníky súťaže, prvý ročník sa konal on-line kvôli opatreniam v súvislosti s pandémiou Covid-19.

3.1 Účasť a výsledky

V Tabuľke 1 uvádzame počty škôl a študentov stredných škôl, ktorí sa do súťaže zapojili v dvoch doterajších ročníkoch, a dosiahnuté výsledky.

Tabuľka 1: Kvantitatívne vyhodnotenie niektorých aspektov súťaže

ročník	1.	2.
Šk. rok	2020/2021	2021/2022
Počet súťažiacich	86	32
Počet úspešných riešiteľov	31	15
Priemerný počet získaných bodov	267 / 1000	295 / 1000
Počet škôl	15	8

V súťaži je možné dosiahnuť najviac 1000 bodov za správne riešenie (100 bodov za každú z 10 úloh) a bonusové body za čas. Úlohy sú navrhnuté s cieľom získať rozlišujúce hodnotenie [11], v ktorom je výkon žiaka porovnávaný s výkonmi iných žiakov. Nie je cieľom, aby všetci súťažiaci vyriešili všetky úlohy, ale aby sa vytvoril dostatočný rozptyl vo výsledkoch. Väčšina žiakov nestihne vyriešiť v časovom limite všetky úlohy, výhodu majú žiaci, ktorí okrem algoritmického myslenia majú aj dobré zručnosti v písaní programového kódu.

Väčšina súťažiacich si ako programovací jazyk volila Python. Záujem o jazyk C bol malý (v prvom ročníku 4 žiaci, v 2. ročníku 1 žiak).

3.2 Príklad súťažnej úlohy

Ako ukážku zadania uvedieme a podrobnejšie analyzujeme jednu úlohu z 2. ročníka súťaže.

Zadanie úlohy: Športovci z rôznych krajín získali na olympiáde medaily. Zlatá medaila má hodnotu 3 body, strieborná 2 body a bronzová 1 bod. Vytvorte program, ktorý riadky tabuľky s medailovou bilanciou usporiada podľa toho, koľko majú jednotlivé krajiny bodov. Na vstupe je počet krajín uvedených v tabuľke (celé číslo), následne na samostatných riadkoch pre každú krajinu počty získaných medailí v poradí: zlaté, strieborné, bronzové. Oddeľovačom je čiarka.

Príklad vstupu a výstupu:

vstup	výstup
4	2,2,2
0,0,10	0,0,10
2,2,2	3,0,0
0,0,1	0,0,1
3,0,0	

V úlohe mali žiaci vyriešiť niekoľko problémov:

- načítanie štruktúrovaného vstupu,
- výpočet počtu bodov,
- zoraďovanie riadkov tabuľky podľa počtu bodov,
- výpis štruktúrovanej hodnoty.

Problém a. Pri riešení načítania vstupu volili žiaci niekoľko stratégií: načítať vstup ako zoznam reťazcov:

```
['0,0,10', '2,2,2', '0,0,1', '3,0,0']
```

zoznam zoznamov reťazcov:

```
[['0', '0', '10'], ['2', '2', '2'], ['0', '0', '1'], ['3', '0', '0']]
```

zoznam zoznamov celých čísel:

```
[[0, 0, 10], [2, 2, 2], [0, 0, 1], [3, 0, 0]]
```

Využívali pritom imperatívne techniky programovania – v cykle (`for`) menili hodnotu zoznamu pridávaním ďalšieho prvku (`append`), alebo deklaratívne techniky – generovali zoznam pomocou generátorovej notácie (`list comprehension`) alebo upravovali prvky zoznamu na celé čísla mapovaním (`map`).

Problém b. Výpočet počtu bodov žiaci realizovali jednoduchým výrazom s tromi hodnotami štruktúrovanej premennej (počet zlatých, strieborných, bronzových medailí), alebo v cykle s tromi opakovaniami. Definícia vlastnej funkcie na výpočet bodov bola zriedkavá (len 2 prípady).

Výsledky výpočtu žiaci ukladali buď do samostatnej štruktúry, ktorú so vstupom spájalo len poradie prvkov, alebo ich pridávali k vstupnému zoznamu zoznamov ako tretí rozmer. Niektorí žiaci si vytvorili slovník s počtami bodov ako kľúčmi a vstupnými údajmi ako hodnotami. Tieto riešenia však boli len čiastočne správne, pri rovnakých bodových hodnotách sa údaje prepisovali (strácali). Výpočet bodov ad-hoc bez ukladania sa vyskytol zriedkavo.

Problém c. Jednou skupinou riešení na usporiadanie postupnosti podľa počtu bodov bolo použitie vstavanej metódy `sort`, alebo funkcie `sorted`. Problémom bolo, že vstupnú postupnosť bolo treba usporiadať podľa iného ako predvoleného kritéria a v opačnom ako predvolenom poradí. Častým riešením bolo vytvorenie samostatného číselného poľa s počtami bodov, jeho utriedenej verzie a prevrátenie do opačného poradia, a následné zisťovanie pôvodnej pozície v neusporiadanom poli pre spárovanie usporiadaných počtov bodov s neusporiadaným vstupom. Obrátenie poradia realizovali žiaci buď rezom `[::-1]` alebo funkciou `reversed`. Lepšie riešenia triedili zoznam podľa počtu bodov uložených spolu so vstupnými údajmi v dvojici alebo v dvojprvkovom zozname:

```
[(10, '0,0,10'), (12, '2,2,2'), (1, '0,0,1'), (9, '3,0,0')]
```

Sofistikovanejšie riešenia použili vo funkciách `sort`, `sorted` voliteľné parametre `reverse` na zadanie opačného smeru usporiadania a `key` na definovanie kritéria (zriedkavé).

Druhou skupinou riešení bolo naprogramovanie vlastného triediaceho algoritmu. Vo väčšine prípadov žiaci naprogramovali triedenie výberom maxima do nového poľa (nie na mieste), v jednom prípade bublinové triedenie na mieste, v jednom triedenie porovnávaním všetkých dvojíc prvkov.

Problém d. Pri výpise výsledkov mali jednoduchšiu úlohu tí, ktorí si vstup ponechali vo forme zoznamu reťazcov. Ostatní žiaci spätne vytvárali reťazce z jednoduchších hodnôt, alebo použili formátovaný výpis.

Uvádzame jedno autorské riešenie s ad-hoc výpočtom bodov s využitím funkcionálnych techník programovania, ktoré sa v riešeniach žiakov objavovali v rôznej miere (Obrázok 1) a druhé autorské riešenie bez funkcionálnych techník programovania s predvýpočtom a uložením počtu bodov do štruktúry k vstupným údajom (Obrázok 2).

```
def body(ret):
    z = ret.split(',')
    return int(z[0])*3+int(z[1])*2+int(z[2])

n = int(input())
medaily = [input() for i in range(n)]

for x in sorted(medaily, key = body, reverse = True):
    print(x)
```

Obrázok 1: Autorské riešenie 1

```
def body(ret):
    z = ret.split(',')
    return int(z[0])*3+int(z[1])*2+int(z[2])

n = int(input())

medaily = []
for i in range(n):
    riadok = input()
    medaily.append((body(riadok), riadok))

medaily.sort(reverse=True)

for x in medaily:
    print(x[1])
```

Obrázok 2: Autorské riešenie 2

3.3 Diskusia

Z kvalitatívnej analýzy riešení sme získali zaujímavé poznatky o algoritmických a programátorských vedomostiach žiakov a o programovacích technikách, ktoré ovládajú a používajú:

- Žiaci používajú podľa očakávania prevažne imperatívny štýl programovania, ale nie sú zriedkavé ani niektoré deklaratívne techniky. Z deklaratívnych techník najčastejšie využívajú generátorovú notáciu na vytvorenie zoznamu pri spracovaní riadkového textového vstupu na postupnosť čísel. Výnimočne sa v riešeniach objavuje funkcia ako parameter, dokonca anonymná (`lambda`). Prekvapivé je zriedkavé definovanie vlastnej funkcie.
- Žiaci majú problémy s viac ako dvojrozmernými štruktúrami. V analyzovanom prípade radšej vytvárali viac jednoduchších štruktúr.
- Žiaci nemajú vedomosti o triediacich algoritmoch, riešenia v analyzovanom prípade boli väčšinou intuitívne, alebo boli použité vstavané funkcie.

Z analýzy výsledkov nultého a prvých dvoch ročníkov súťaže vyplývajú pre autorov zadaní nasledujúce poznatky:

- Je potrebný kvalitný výber testovacích prípadov, aby automatické vyhodnocovanie rozlíšilo správne od čiastočne správnych riešení.
- Výsledky riešenia zadaní musia byť z veľkej množiny, aby náhodné generovanie výstupov nemalo veľkú pravdepodobnosť na úspech.
- Náročnosť, množstvo a typy úloh sú zvolené správne, výsledky ukazujú dostatočný rozptyl vo výkonoch žiakov.
- Za zváženie stojí zvýšenie bonusu za čas, aby sa podporil taktický aspekt súťaže.
- Úspešnosť riešenia úlohy nemusí zodpovedať jej náročnosti, úlohu zohráva aj fakt, že väčšina súťažiacich nestihne riešiť všetky úlohy. Ak riešia úlohy po poradí, je možné, že úspešnosť riešení závisí aj od poradia úloh. Za zváženie stojí miešanie poradia úloh pre súťažiacich, aby sme dostali objektívnejší obraz o náročnosti úloh.

Podľa výsledkov z predchádzajúcich ročníkov usudzujeme, že väčšina súťažiacich si bude aj naďalej voliť ako programovací jazyk súťaže Python, v menšej miere jazyk C (najmä študenti stredných odborných škôl). O jazyk Java zatiaľ nebol záujem. Nultého, pilotného ročníku súťaže sa zúčastnili aj študenti bakalárskeho študijného programu Aplikovaná informatika, ktorí v 1. ročníku navštevovali úvodný kurz programovania v jazyku Java. Z analýzy riešení vyplynulo, že programy napísané v Jave boli dlhšie, menej prehľadné, mnohí študenti neboli schopní riešenie dokončiť pre problémy s načítavaním štruktúrovaného vstupu. Zručné používanie jazyka Python ostatných súťažiacich zo stredných škôl zvýhodňovalo. Vzhľadom na objektívne odlišnosti programovacích jazykov preto zachováme oddelené vyhodnocovanie účastníkov súťaže podľa toho, ktorý z jazykov v súťaži budú používať.

Skúsenosti s dvomi organizačnými formami súťaže nám dávajú možnosť porovnať výhody a nevýhody prezenčnej a on-line formy. Výhodou prezenčnej formy je osobný kontakt súťažiacich. Počas prestávok sme svedkami rozhovorov a nadväzovania kontaktov žiakov z rôznych škôl. Súťažiaci a ich sprevádzajúci učitelia majú tiež príležitosť navštíviť univerzitné pracovisko organizátorov a zúčastniť sa sprievodného programu. Nevýhodou prezenčnej formy je, že počet účastníkov je obmedzený priestorovými možnosťami univerzity. Nutnosť fyzického presunu obmedzuje súťaž na regionálnu. Naproti tomu pri on-line forme vieme osloviť súťažiacich z celého Slovenska bez obmedzení, avšak vytráca sa cenný spoločenský aspekt súťaže.

4 Záver

Súťaž IT v Nitre dopĺňa ponuku súťaží v programovaní pre stredoškôlkov. Nezameriava sa len na mimoriadne nadaných študentov, ale na všetkých študentov so záujmom o programovanie

s cieľom prehĺbiť tento záujem a vytvoriť ďalší priestor na rozvíjanie schopností riešiť problémy a rozvoj programátorských zručností. Súťaž je atraktívna modernou formou, ale aj vecnými cenami pre víťazov. Učitelia oceňujú osobné stretnutia a sprievodný program, ktorý sa realizuje počas súťaže a v čase jej vyhodnocovania. Diskusie s učiteľmi za praxe sú obohatením aj pre organizátorov. Riešenia súťažiacich sú cenným zdrojom informácií o skutočnom obsahu a kvalite vyučovania programovania na slovenských stredných školách. Kvalitatívna analýza riešení poukázala na niektoré miskoncepce alebo málo trénované zručnosti (efektívna práca s údajovými štruktúrami, používanie vlastných funkcií, funkcionálne techniky programovania v jazyku Python), na ktoré sme reagovali odbornou publikáciou pre učiteľov informatiky [12].

POĎAKOVANIE

Súťaž IT v Nitre a tento príspevok sú finančne podporené z projektu FITPED-AI č. 2021-1-SK01-KA220-HED-000032095 programu ERASMUS+ Európskej komisie.

BIBLIOGRAFICKÉ ODKAZY

- [1] Olympiáda v informatike. <http://oi.sk/>
- [2] Korešpondenčný seminár z programovania. <https://www.ksp.sk/>
- [3] Zenit v programovaní. <https://zenit.ksp.sk/>
- [4] Python Cup. <http://edu.fmph.uniba.sk/PythonCup/>
- [5] Palma Junior. <https://di.ics.upjs.sk/palmaj/>
- [6] ŠNAJDER, L., GUNIŠ, J. *Tvorba úloh pre programátorské súťaže*. Košice : UPJŠ, 2014. 79 s. ISBN 978-80-8152-139-3
- [7] IT v Nitre. <https://www.itvnitre.sk/>
- [8] Priscilla Platform. <https://fitped.eu/priscilla-platform/>
- [9] Work-Based Learning in Future IT Professionals Education (FITPED). <https://fitped.eu/>
- [10] SMYRNOVA-TRYBULSKA, E. et al. *Microlearning : New Approaches to a More Effective Higher Education*. Cham : Springer, 2022. 230 s. ISBN 978-3-031-13359-6. DOI=<https://doi.org/10.1007/978-3-031-13359-6>
- [11] TUREK, I. *Didaktika*. Bratislava : Wolters Kluwer, 2014. 618 s. ISBN 978-80-8168-004-5
- [12] LOVÁSZOVÁ, G., MICHALIČKOVÁ, V., KVAŠŠAYOVÁ, N. *Programovanie v jazyku Python: Vybrané kapitoly pre učiteľov informatiky*. Nitra : UKF, 2022. 80 s. ISBN 978-80-558-1951-8. DOI=<https://doi.org/10.17846/FPVaI-2022-34>

Propedeutické využitie digitálnej modelovej železnice k programovaniu mikrokontroléra Arduino

A propaedeutic use of a digital model railway to program an Arduino microcontroller

Miroslav Melicherčík
Fakulta prírodných vied
Univerzity Mateja Bela
Tajovského 40
974 01 Banská Bystrica
Slovenská republika
miroslav.melichercik@umb.sk

Vladimír Siládi
Fakulta prírodných vied
Univerzity Mateja Bela
Tajovského 40
974 01 Banská Bystrica
Slovenská republika
vladimir.siladi@umb.sk

Michal Povinský
Fakulta prírodných vied
Univerzity Mateja Bela
Tajovského 40
974 01 Banská Bystrica
Slovenská republika
michal.povinsky@umb.sk

ABSTRACT

Programming itself presents a complex process in which it is necessary to link algorithmic thinking, with knowledge of the programming language as well as the technologies. Our aim is to propose an approach that offers the possibility of motivating pupils to learn programming using experimentation. For this purpose, we have linked a digital railway model with an Arduino development board and then controlled the locomotives via a mobile phone communicating via Bluetooth with the Arduino. As part of the above, it was essential to obtain and use several knowledge and skills such as wiring electrical circuits, microcontroller programming, programming mobile applications and creating and printing 3D models. The proposed procedure was tested on a sample of 16 students, where it was shown that although not all technologies were studied in depth, it provides a good steppingstone for their further studies and also increases motivation for further study.

Keywords

Programming. Motivation. Arduino. Digital railway model. MIT App Inventor. 3D modeling.

ABSTRAKT

Programovanie ako také predstavuje zložitý proces, v ktorom je potrebné prepojiť algoritmicke myslenie, so znalosťou programovacieho jazyka ako aj používaných technológií. Našou snahou je navrhnúť prístup, ktorý ponúka možnosť motivácie žiakov k štúdiu programovania s využitím experimentovania. Za týmto účelom sme prepojili model digitálnej železnice s vývojovou doskou Arduino a následne ovládali lokomotívy prostredníctvom mobilného telefónu komunikujúceho cez Bluetooth s Arduino. V rámci uvedeného bolo nevyhnutné získať a využiť viacero poznatkov a zručností ako zapájanie elektrických obvodov, programovanie mikrokontroléra, programovanie mobilných aplikácií a vytváranie a tlač 3D modelov. Navrhnutý postup bol vyskúšaný na vzorke 16 študentov, kde sa ukázalo, že hoci neboli všetky technológie preštudované do hĺbky, poskytuje to dobrý odrazový mostík pre ich ďalšie štúdium a zároveň zvyšuje motiváciu k ďalšiemu štúdiu.

Kľúčové slová

Programovanie. Motivácia. Arduino. Model digitálnej železnice. MIT App Inventor. 3D modelovanie.

1 ÚVOD

Komerčné používanie mikrokontrolérov v zariadeniach bežného života, ich dostupnosť na trhu a nízka cena je výzvou pre súčasného učiteľa. Veľké množstvo lacných snímačov, motorčekov a ďalších súčiastok vytvára priestor tvoriť zaujímavé aktivity, ktoré môžu pomôcť motivovať žiakov zaujímať sa o problematiku internetu vecí, v optimistickjších prípadoch vytvárať priestor pre rozvoj kreativity žiakov v hľadaných nových, nekonvenčných prototypov inteligentných zariadení pre každodenný život.

Pomerne nízka cena komponentov umožňuje učiteľovi vytvárať jednoduché zázemie pre potreby žiackeho experimentovania. Experimentovať sa dá vytváraním zaujímavých hardvérových riešení, programovaním mikrokontrolérov, alebo spojením oboch aktivít do jednej rozsiahlejšej.

Z detstva si pamätáme nadšenie, ktoré majú deti, často aj ich otcovia, pre modelové železnice. Ani túto oblasť neobišla digitalizácia. Na trhu sa okrem klasických analógových modelových železníc objavili digitálne modelové železnice, ktoré obohatili hru s vláčikmi o nové možnosti. Využili sme možnosť prepojiť vlastné riadenie mikrokontrolérom s ľahko dostupnou digitálnou modelovou železnicou typu H0.

Využitie spojenia ekosystému mikrokontrolérov s motivačnými výhodami modelovej digitálnej železnice sa dá použiť na prilákanie žiakov k štúdiu napr. programovania. Na školách sú na experimentovanie momentálne najpopulárnejšie a najviac používané vývojové dosky Arduino, Micro:bit a Raspberry Pi. Ich využitie v edukácii je opísané v mnohých vedeckých a popularizačných článkoch. Použitie vývojovej dosky, ako je napríklad Arduino, môže pomôcť zlepšeniu znalostí žiakov z logických obvodov [1, s. 7] (využitia hardvéru), programovania [1, 2] (tvorba softvéru), ale aj oboznámiť žiakov so základmi elektroniky. Niektorí autori uvádzajú dokonca, že je možné s použitím Arduina jednoduchšie učiť začiatočníkov (nevynímajúc deti) programovať [3]. V tom istom zdroji predstavujú autori učebné materiály postavené na modeli ADDIE (Analysis – Design – Development – Implementation – Evaluation) [4]. Podobne populárne ako využívanie Arduina, či už vo vyučovacom procese, alebo v krúžkovej činnosti, tak je aj používanie vývojovej dosky BBC Micro:bit [5, 6]. Overili sme si to napr. na tvorivých dielnach pre žiakov stredných škôl [7,8].

V článku nemáme ambíciu vytvárať podobné výučbové materiály založené na iteračnom princípe. Sústreďujeme sa len na jednu zo

základných úloh, s ktorou sa je potrebné vysporiadať hneď na začiatku realizácie zámeru a to prechodom na riadenie digitálnej modelovej železnice mikrokontrolérom. Nad túto základnú úlohu je potom možné navrhnuť a implementovať rôzne rozšírenia.

2 DIGITÁLNA MODELOVÁ ŽELEZNICA OVLÁDANÁ MIKROKONTROLÉROM

Základným rozdielom medzi analógovým a digitálnym modelom železnice je v spôsobe ovládania lokomotív. Zatiaľ čo v analógovej verzii je regulovateľný zdroj jednosmerného prúdu priamo pripojený na koľaje (každá koľaj vedie jeden pól), pričom smer pohybu lokomotívy je priamo závislý od polarít napájania a rýchlosť od veľkosti napätia. V prípade, že sa na trati bude nachádzať viac lokomotív, budú sa všetky pohybovať približne rovnako rýchlo a v rovnakom smere a nie je možné ovládať každú z nich samostatne. Pri použití digitálne ovládanej železnice je trať trvalo napájaná striedavým prúdom, pričom do jeho frekvencie sú modulované riadiace povely adresované pre jednotlivé lokomotívy (smer a rýchlosť jazdy, svetlá, atď.). Takto je možné nezávisle ovládať viacero lokomotív nachádzajúcich sa na tej istej trati.

Model digitálnej železnice preto ponúka širšiu možnosť experimentovania pri ovládaní železnice pomocou mikrokontroléra a umožňuje riešiť hneď niekoľko úloh, ktoré vytvárajú výzvy pre experimenty žiakov:

- nahradenie riadenia dodávaného k modelovej železnici vlastným riadením s mikrokontrolérom,
- nahradenie štandardného napájania digitálnej železnice vlastným napájacím zdrojom,
- vytvorenie aplikácie pre ovládanie,
- tlač 3D komponentov na prichytenie nového riadenia a súčiastok železnice.

Ako je vidieť, problematika je široká a zasahuje do oblastí hardvéru, softvéru a 3D tlače.



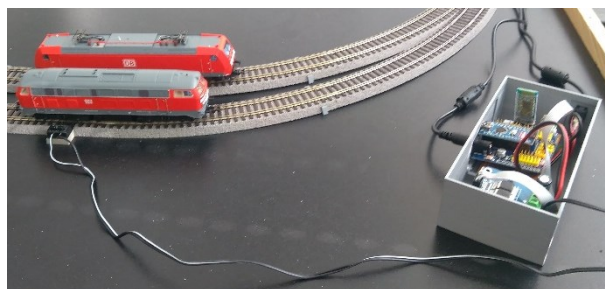
Obrázok 1: Pôvodné riadenie modelovej železnice

Použitá modelová železnica *DB SmartControl Light HO*¹ obsahuje dve vlakové súpravy (rušeň a vagóny), koľaje, dve mechanické výhybky a digitálne ovládanie *PIKO SmartControllight*

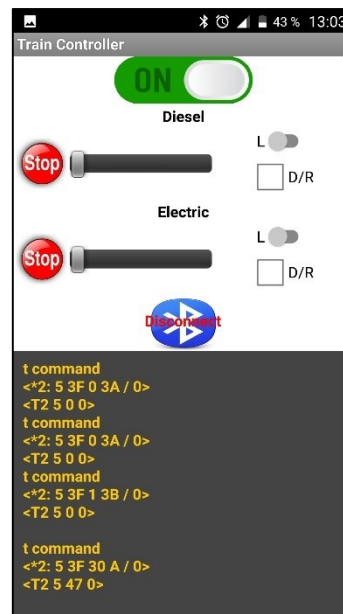
¹ <https://www.htmodel.sk/startset-2-vlakove-supravy-db-smartcontrol-light-ho/>

s napájaním. Na ovládanie a programovanie lokomotív sa štandardne používa ovládač, ktorý je dodávaný ako súčasť balenia (pozri Obrázok 1).

My sme sa rozhodli nahradiť digitálne ovládanie vlastným riešením. Pôvodné digitálne ovládanie sme nahradili mikrokontrolérom Arduino NANO. Trať sme napájali prostredníctvom H-mostíka (pozri Obrázok 2). Na ovládanie sme vyvinuli mobilnú aplikáciu pre zariadenia s OS Android (pozri Obrázok 3). Komunikácia medzi Arduino a mobilným zariadením bola zabezpečená prostredníctvom Bluetooth modulu. Okrem toho bolo potrebné doplniť zapojenie o ďalšie logické členy na generovanie signálu, vypínač napájania a konektory pre jednoduchšie a spoľahlivejšie zapájanie zostavy.



Obrázok 2: Vyvinuté riadenie a napájanie modelovej železnice osadené vo vytláčenom obale na 3D tlačiarňi



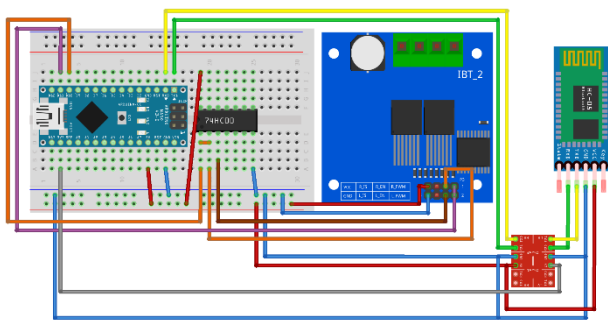
Obrázok 3: Vyvinutá aplikácia na ovládanie modelovej železnice

2.1 Zapojenie hardvéru

Základ ovládania tvorí vývojová doska Arduino NANO s mikrokontrolérom, ktorá je osadená do nadstavby IO shield,

vďaka čomu je možné jednoduché pripájanie snímačov. IO shield je umiestnený na doske plošných spojov. Na doske plošných spojov sú tiež umiestnené: konektory pre napájanie Arduina s regulátorom napätia na 5 V, ovládanie H-mostíka a päťica pre pripojenie Bluetooth modulu s prevodníkom úrovne napätia 5 – 3,3 V, integrovaný obvod HC7400. Pre názornosť uvádzame schému zapojenia pomocou kontaktného poľa (pozri Obrázok 4).

K digitálnym pinom Arduina D10 a D11 je pripojený H-mostík typu BTS7960. Na pine D10 je generovaný obdĺžnikový signál, ktorý je použitý pre generovanie napájacieho striedavého prúdu pre trať. Pin D10 na Arduine je prepojený s pinom L_EN na H-mostíku, čo umožňuje generovať kladnú časť napájacieho signálu. Keďže polaritu výstupného napätia použitého H-mostíka je potrebné ovládať pomocou dvoch pinov L_EN, R_EN, použili sme integrovaný obvod zapojený ako invertor signálu z pinu D10 na Arduine a invertovaný signál sme pripojili na pin R_EN na H-mostíku, čím sme zabezpečili generovanie zápornej časti napájacieho signálu. Pin D11 na Arduine slúži na zapnutie/vypnutie napájania trate z H-mostíka a je spojený s pinmi L_PWM a R_PWM.



Obrázok 4: Schéma zapojenia

Bluetooth modul HC-05 je pripojený k sériovému rozhraniu Arduina na pinoch D0 a D1. Keďže Arduino pracuje s logickou úrovňou signálu 5 V a Bluetooth modul s úrovňou signálu 3,3 V, je potrebné konvertovať úroveň signálu medzi oboma zariadeniami pomocou prevodníka. Pri zapojení sme vychádzali z dokumentácie knižnice DCC++ [9].

H-mostík je napájaný jednosmerným prúdom pri napätí 15 V, pričom z toho istého zdroja je napájané aj Arduino pomocou regulátora napätia na 5 V. Toto zapojenie môže viesť k nestabilite ovládania pri prechode vlaku cez výhybku, kedy môže dôjsť ku krátkodobému skratu medzi koľajami nápravou. Väčšina súčasných používaných zdrojov zareaguje na tento skrat znížením výkonu skôr, ako je ho možné detegovať pomocou H-mostíka a softvérovo túto situáciu ošetriť, čo môže viesť k poklesu napätia v Arduine a jeho následnému reštartovaniu. Možným riešením je použitie iného typu H-mostíka, ktorý umožňuje automatické odpojenie v prípade skratu alebo oddelením napájania Arduina od H-mostíka.

Táto časť riešenia vytvára priestor pre motivovanie k získaniu znalostí a základných zručností v oblasti elektroniky (napr. zapájanie obvodov, meranie, spájkovanie, atď.).

2.2 Softvér ovládača

Hlavnou úlohou ovládača je zabezpečiť napájanie trate spolu s modulovaním signálov pre vykonanie povelov pre lokomotívy. Jednotlivé povelové sú do Arduina odosielené cez sériové rozhranie a Bluetooth modul, ku ktorému sa pripája mobilná aplikácia.

Alternatívne je možné na ovládanie použiť aj desktopovú aplikáciu komunikujúcu bezdrôtovo cez Bluetooth alebo cez USB kábel.

Program pre Arduino je navrhnutý s použitím knižnice DCC++ [9]. Táto knižnica pozostáva z dvoch hlavných častí: základnej stanice a ovládača. Základná stanica predstavuje zariadenie, ktoré je priamo pripojené k trati a prostredníctvom Arduina a ovládača motorov, resp. H-mostíka umožňuje napájať, ovládať a monitorovať trať. Ovládač obsahuje grafické rozhranie pre používateľa, ktoré umožňuje vyslať povel a sledovať stav trate. Pre naše účely sme použili modul knižnice s názvom *Text Command Syntax*, ktorý umožňuje Arduinu prijímať textové reťazce so zakódovaným povelením cez sériové rozhranie. Pre aktivovanie modulu je potrebné v súbore `DCCpp.h` definovať makro `#define USE_TEXTCOMMAND`. Nasledujúci zdrojový kód zabezpečí spracovanie zadaných povelov.

```
#include "DCCpp.h"
void setup()
{
  Serial.begin(9600);
  DCCpp::begin();
  DCCpp::beginMain(UNDEFINED_PIN, DCC_SIGNAL_PIN_MAIN, 11, UNDEFINED_PIN);
}
void loop()
{
  DCCpp::loop();
}
```

Príkazom `Serial.begin(9600)` sa inicializuje sériová komunikácia medzi aplikáciou ovládača a Arduinom. Metóda `DCCpp::begin()` slúži na inicializovanie DCC++ knižnice. Metóda `DCCpp::beginMain()` zabezpečí inicializovanie ovládania pripojenej trate, pričom táto metóda má štyri argumenty určujúce čísla pripojených pinov: smer motora (v našom prípade nezapojený), signál (generuje signál na pine D10), zapnuté (ovláda zapnutie napájania trate pinom D11), monitor prúdu (v našom prípade nezapojený). Metóda `DCCpp::loop()` zabezpečí opakované sledovanie, spracovanie a vykonanie prijatých povelov.

Jednotlivé povelové sa ukladajú do vnútorných registrov knižnice DCC++ a tieto sú opakovane vysielané do trate, vďaka čomu sú doručené lokomotíve aj neskôr alebo v prípade neskoršieho nakoľajenia.

Táto časť riešenia poskytuje priestor pre motivovanie k získaniu znalostí z programovania Arduina ako aj drôtovej a bezdrôtovej komunikácie Arduina s počítačom alebo mobilnými zariadeniami.

2.3 Aplikácia na ovládanie

Vytvorený ovládač umožňuje prijímanie povelov v podobe textových reťazcov cez sériové rozhranie. To umožňuje poslať povel priamo cez USB kábel pripojený k počítaču alebo bezdrôtovo cez Bluetooth modul. Keďže sme sa z dôvodu pohodlnejšieho ovládania rozhodli vytvoriť vlastnú mobilnú aplikáciu, pre komunikáciu sme zvolili technológiu Bluetooth. Mobilná aplikácia bola vytvorená v prostredí MIT App Inventor, ktoré umožňuje jednoducho navrhovať a vytvoriť mobilnú aplikáciu pre OS Android s použitím blokového programovania.

Základné funkcionality aplikácie (pozri Obrázok 3) boli prispôbené opisanej modelovej železnici. Aplikácia umožňuje odoslať povel pre zapnutie a vypnutie napájania trate, ovládať

rýchlosť a smer pohybu jednotlivých lokomotív, ovládať zapnutie a vypnutie funkcie lokomotív rozsvietení svetlá a tiež možnosť núdzového zastavenia lokomotív. Aplikácia zároveň umožňuje výber Bluetooth zariadenia na pripojenie a zobrazenie prebiehajúcej komunikácie (textových povelov a odpovedí) s Arduino. V použitej modelovej železnici sú dve lokomotívy s adresami 4 a 5, použitými ako hodnotami pre parameter CAB v nasledujúcich poveloch.

Nasledujúci blok kódu zabezpečí výber a pripojenie mobilného zariadenia k Bluetooth modulu. Po úspešnom pripojení sa v aplikácii zobrazí možnosť ovládania trate.

```

when BTN_BT .Click
do
  if BluetoothClient1 .IsConnected
  then
    call BluetoothClient1 .Disconnect
    set BTN_BT .Text to "Connect"
    set TextBox1 .Text to "Disconnected"
    set HorizontalArrangement_OnOff .Visible to false
    set VerticalArrangement_D .Visible to false
    set VerticalArrangement_E .Visible to false
  else
    set TextBox1 .Visible to false
    set LIST_BT .Visible to true
    set LIST_BT .Elements to BluetoothClient1 .AddressesAndNames

when LIST_BT .AfterPicking
do
  if call BluetoothClient1 .Connect
  address LIST_BT .Selection
  then
    set BTN_BT .Text to "Disconnect"
    set LIST_BT .Visible to false
    set TextBox1 .Visible to true
    set TextBox1 .Text to join "Connected"
    set HorizontalArrangement_OnOff .Visible to true
    set VerticalArrangement_D .Visible to true
    set VerticalArrangement_E .Visible to true
    
```

Ďalší blok príkazov umožňuje núdzové zastavenie lokomotívy, plynule regulovateľné ovládanie rýchlosti pohybu lokomotívy a voľbu smeru pohybu lokomotív.

```

initialize global SPEED_D to 0

when BTN_STOP_D .Click
do
  set SLIDE_D .ThumbPosition to 0
  set global SPEED_D to 0
  if CheckBox_D .Checked
  then
    call BluetoothClient1 .SendText
    text "<t 25-1>"
  else
    call BluetoothClient1 .SendText
    text "<t 25-10>"

when SLIDE_D .PositionChanged
thumbPosition
do
  set global SPEED_D to round get thumbPosition
  if CheckBox_D .Checked
  then
    call BluetoothClient1 .SendText
    text join "<t 25"
    join get global SPEED_D
    join "1>"
  else
    call BluetoothClient1 .SendText
    text join "<t 25"
    join get global SPEED_D
    join "0>"

when CheckBox_D .Changed
do
  if CheckBox_D .Checked
  then
    call BluetoothClient1 .SendText
    text join "<t 25"
    join get global SPEED_D
    join "1>"
  else
    call BluetoothClient1 .SendText
    text join "<t 25"
    join get global SPEED_D
    join "0>"
    
```

Pre vykonanie uvedených povelov sú použité textové reťazce s nasledujúcou syntaxou:

```
<t REGISTER CAB SPEED DIRECTION>
```

Parameter REGISTER určuje celé číslo predstavujúce číslo vnútorného registra použitého v rámci knižnice DCC++ pre uloženie povelov. Parameter CAB určuje adresu lokomotívy, pre ktorú je povel určený. Parameter SPEED určuje rýchlosť pohybu v rozsahu 0 – 126, pričom hodnota –1 je použitá pre núdzové zastavenie. Parameter DIRECTION s hodnotou 0 alebo 1 určuje smer pohybu lokomotívy.

Nasledujúci blok príkazov slúži na zapnutie a vypnutie napájania trate pomocou textových reťazcov:

```

when BTN_Off .Click
do
  call BluetoothClient1 .SendText
  text "<1>"
  set BTN_Off .Visible to false
  set BTN_On .Visible to true

when BTN_On .Click
do
  call BluetoothClient1 .SendText
  text "<0>"
  set BTN_On .Visible to false
  set BTN_Off .Visible to true
    
```

Pomocou ďalšieho bloku kódu je možné ovládať funkciu osvetlenia lokomotívy.

```

when SW_L_D .Changed
do
  if SW_L_D .On
  then
    call BluetoothClient1 .SendText
    text "<f 5 144>"
  else
    call BluetoothClient1 .SendText
    text "<f 5 128>"
    
```

Pre vykonanie uvedených povelov sú použité textové reťazce s nasledujúcou syntaxou:

```
<f CAB BYTE>
```

Parameter CAB určuje adresu lokomotívy, pre ktorú je povel určený. Hodnota pre parameter BYTE sa určí na základe vzťahu $F0 - F4$ môžu nadobúdať hodnotu 0 alebo 1. Funkcia F0 zodpovedá osvetleniu lokomotívy, ostatné funkcie (napr. zvukové efekty) nie sú podporované pri použitých lokomotívach, preto hodnoty parametrov F1 – F4 môžu zostať rovné 0.

Monitorovanie komunikácie zabezpečuje nasledujúci blok kódu.

```

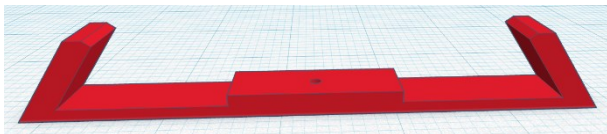
when Clock1 .Timer
do
  if BluetoothClient1 .IsConnected and call BluetoothClient1 .BytesAvailableToReceive > 0
  then
    set TextBox1 .Text to join call BluetoothClient1 .ReceiveText
    numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
    join " "
    join TextBox1 .Text
    
```

Ako vidieť, pri tvorbe softvérového ovládača a softvérovej aplikácie bolo použitých viacero nástrojov: Arduino IDE s knižnicou DCC++ a MIT App Inventor.

2.4 Tvorba 3D modelov a ich tlač

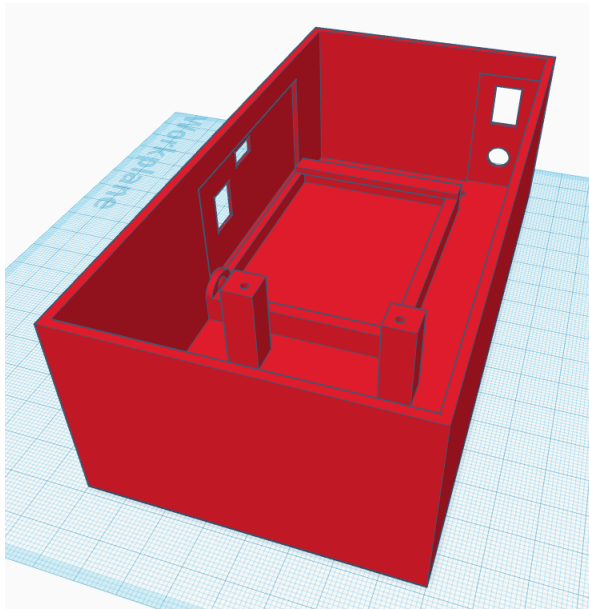
Ďalšou oblasťou experimentovania žiakov, s cieľom vzbudiť záujem o informačné technológie (konkrétne počítačovú grafiku), ktorá sa ponúka pri realizácii modelovej železnice, je 3D modelovanie. 3D modely boli vytvárané v on-line nástroji TinkerCAD a tlačené na 3D tlačiarňami ZMorph.

V prvom prípade bolo potrebné zabezpečiť vytvorenie koľajiska uchytením k panelu. K tomuto účelu bol navrhnutý model plastových úchytky s možnosťou opakovaného pripevnenia a vybratia koľajnice (pozri Obrázok 5).



Obrázok 5: Model plastovej úchytky

Okrem toho bola vymodelovaná krabička na osadenie súčiastok ovládača: Arduino NANO aj so shieldom, H-mostík, konektory a vypínač (pozri Obrázok 6).



Obrázok 6: Model krabičky na osadenie súčiastok ovládača

Reálne použitie oboch súčiastok je vidieť na obrázku (pozri Obrázok 2).

Pri vytváraní modelov bolo nevyhnutne potrebné dodržať presné rozmery, ako napr. umiestnenie otvorov pre konektory a skrutky, šírka koľajnice, atď.

3 TVORIVÉ DIELNE

Navrhnutý experiment s modelovou železnicou bol zrealizovaný v rámci dvoch tvorivých dielní: *Od návrhu k prototypu* [10] a *Programujeme železnicu* [11], ktoré pozostávali každá z dvoch stretnutí. Na dvoch stretnutiach sa venovala pozornosť 3D modelovaniu a 3D tlači, na ostatných dvoch programovaniu Arduina a jeho prepojeniu s modelovou železnicou. Do tvorivých dielní boli zapojení šesťnásť študenti bakalárskeho stupňa štúdia aplikovanej informatiky a vývoja softvéru. V úvode prvej časti oboch tvorivých dielní študenti vyplňali vstupný dotazník², v ktorom odpovedali na 18 otázok, ktoré boli rozdelené do 4 skupín: základné údaje o študentovi (5 položiek), očakávania od tvorivej dielne (4 otázky), podrobnejšie informácie o účastníkovi (6 otázok), vízie respondenta o jeho budúcnosti v oblasti IT (3 otázky). Pre potreby tohto príspevku nás zaujímala otázka zameraná na samohodnotenie študenta v oblasti programátorských zručností. Ako vidieť v tabuľke (pozri Tabuľku 1), tvorivých dielní sa zúčastnili študenti, ktorí hodnotili svoje znalosti a zručnosti z programovania väčšinou ako priemerné.

Tabuľka 1: Odpovede študentov na požiadavku „Ohodnoť svoje znalosti z kódovania na nasledujúcej škále.“

Škála	Začínajúci programátor – Programátor expert									
Stupeň	1	2	3	4	5	6	7	8	9	10
Odpovede	2	–	1	3	4	3	2	1	–	–

Zaujímali nás tiež odpovede na otázku, v ktorej sme zisťovali očakávania účastníkov, čo sa týka inšpirovania a motivácie k ďalšiemu štúdiu nových technológií a programovania.

Tabuľka 2: Odpovede študentov na tvrdenie „Účast' na tomto podujatí ma bude inšpirovať k ďalšiemu štúdiu nových technológií a programovania.“

Škála	Vôbec nesúhlasím – Úplne súhlasím				
Tvorivá dielňa: Od návrhu k prototypu					
Stupeň	1	2	3	4	5
Odpovede	–	–	2	3	11
Tvorivá dielňa: Programujeme železnicu					
Stupeň	1	2	3	4	5
Odpovede	–	–	2	4	10

Ako vidieť v tabuľke (pozri Tabuľku 2), na tvorivé dielne prišli už prevažne motivovaní študenti, ktorí očakávali, že technológie, s ktorými sa stretnú ich budú motivovať ďalej sa vzdelávať v oblasti nových technológií a programovania.

Keďže išlo o študentov vysokoškolského štúdia, tak pri téme tvorby 3D modelov pre tlač komponentov modelovej železnice sme nechali študentom voľnosť pri návrhu komponentov, podmienkou bolo, že sa do nich budú osádzať snímače slúžiace na ovládanie modelovej železnice. Študenti sa tak venovali napr. návrhu a modelovaniu súčastí návěstí, semaforov, výhybiek, perónov, tunelov, staníc, atď. Náš koncept modelov úchytkov a krabičky bol uvedený ako príklady na uvedenie do problematiky.

² Dotazník projektu Meet and Code.

Cieľom v časti z programovania Arduina bolo „pohnutie vláčika“ naprogramovanou aplikáciou. Študentom sa cieľ podaril splniť a lokomotívu modelovej železnice vytvoreným softvérom počas konania tvorivej dielne rozšlybali.

Po poslednej hodine každej tvorivej dielne sme študentov znovu požiadali o vyplnenie dotazníka. V tomto prípade išlo o výstupný dotazník, ktorý pozostával z 27 otázok, ktoré boli rozdelené do 6 skupín: základné údaje o študentovi (2 položky), o splnení očakávaní (6 otázok), o pomoci akcie (5 otázok), o odporúčaní akcií typu Meet and Code (5 otázok), podrobnejšie informácie o účastníkovi (6 otázok), otázky o budúcnosti respondenta v oblasti IT (3 otázky).

Pre porovnanie uvádzame odpoveď na tú istú otázku ako bola v úvodnom dotazníku, ktorá zisťovala, či účasť na tvorivých dielnach inšpirovala študentov k ďalšiemu učeniu sa novým technológiam a programovaniu (pozri Tabuľku 3).

Tabuľka 3: Odpovede študentov na tvrdenie „Účasť na tejto akcii ma inšpirovala k ďalšiemu učeniu sa novým technológiam a programovaniu.“

Škála	Vôbec nesúhlasím – Úplne súhlasím				
Tvorivá dielňa: Od návrhu k prototypu					
Stupeň	1	2	3	4	5
Odpovede	–	–	–	2	3
Tvorivá dielňa: Programujeme železnicu					
Stupeň	1	2	3	4	5
Odpovede	–	–	1	1	4

Nanešťastie počet odovzdaných výstupných dotazníkov bolo výrazne menej ako v prípade vstupných dotazníkov. Z tohto dôvodu, z dôvodu už motivovaných študentov na začiatku tvorivých dielní, ale aj z dôvodu celkovo štatisticky nereprezentatívnej vzorky respondentov, nie je možné formulovať relevantné závery, ktoré by potvrdili alebo vyvrátili náš predpoklad možnosti motivovať študentov k štúdiu nových technológií a programovania uvedenými technológiami. Experiment bude potrebné ešte zopakovať na väčšej vzorke.

4 ZÁVER

Napriek tomu, že nebol z našej strany štatisticky vyhodnotený očakávaný prínos spojenia modelovej železnice a mikrokontroléra Arduino pre potreby motivácie študentov k učeniu sa nových technológií a programovania, môžeme konštatovať, že je možné spojiť modelovú železnicu a jej ovládanie mikrokontrolérom Arduino pre potreby vzdelávacieho procesu. V našom návrhu sme nepostupovali spôsobom detailného pochopenia každej z použitých technológií. Na jednej strane, sme uprednostnili širokospektrálny náhľad na použité technológie s uprednostnením zážitku a jednoduchého experimentovania žiakmi (nie bádania). Na druhej strane, takéto zážitkové experimentovanie predpokladá, že učiteľ vykoná množstvo prípravných aktivít tak, aby žiaci mohli bezproblémovo testovať dostupné možnosti. Čím sú žiaci mladší, tým je väčšia experimentálna záťaž na strane učiteľa. Toto tvrdenie vychádza aj z toho, že samotná modelová železnica, bez problematiky riešenia ovládania, je ako hračka určená pre vek nad 14 rokov.

Opisovaný koncept prepojenia modelovej železnice s ovládaním prostredníctvom vývojovej dosky Arduino bude v blízkej

budúcnosti rozšírený o ovládanie prehadzovania mechanických výhybiek pomocou servomotorov, o optické snímače prechodu vlaku určenými sekciami koľajiska [12], atď., kde je vytvorený priestor aj pre ďalšie úlohy z oblasti 3D tlače, napr. prevod otáčania servomotoru na posun prehadzovania výhybky. Našou ambíciou po doplnení modelovej železnice o spomínané komponenty aplikovať na ovládanie vlakových súprav prvkami umelej inteligencie a to či už na báze vyhodnocovania získaných signálov zo snímačov alebo pomocou rozpoznávania obrazu z kamery. Tento aspekt môže byť použitý pri vyučovaní predmetu umelá inteligencia a počítačové videnie vo vysokoškolskej forme štúdia informatiky.

POĎAKOVANIE

K realizácii projektu a vytvoreniu tohto príspevku pomohla finančná podpora iniciatívy Meet and Code 2022.

BIBLIOGRAFICKÉ ODKAZY

- [1] BASHIR, A., et al. Effectiveness of using Arduino platform for the hybrid engineering education learning model. In: *2019 Advances in Science and Engineering Technology International Conferences (ASET)*. IEEE, 2019. p. 1-6.
- [2] KÜRŞAT, A., ZAFER, T. Analyzing the effects of Arduino applications on students' opinions, attitude and self-efficacy in programming class. *Education and Information Technologies*, 2021. p. 1143-1163.
- [3] NOVÁK, M., KALOVÁ, J., PECH, J. Use of the Arduino platform in teaching programming. In: *2018 IV International Conference on Information Technologies in Engineering Education (Inforino)*. IEEE, 2018. p. 1-4.
- [4] ADDIE Model: Instructional Design. *IJET [online]*. 2017, [cit. 2023-03-06]. Dostupné tiež na: <https://educationaltechnology.net/the-addie-model-instructional-design/>. ISSN 2476-0730.
- [5] SENTANCE, S. Teaching with physical computing in school: the case of the micro: bit. *Education and Information Technologies*, 2020, 25: 2577-2603.
- [6] SCHMIDT, A. Increasing Computer Literacy with the BBC micro: bit. *IEEE Pervasive Computing*, 2016, p. 5-7.
- [7] VOŠTINÁR, P., KNEŽNÍK, J. Experience with teaching with BBC micro: bit. In: *2020 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2020. p. 1306-1310.
- [8] VOŠTINÁR, P., MELICHERČIK, M., VAGAČ, M. Teaching programming through the Arduino workshops at secondary schools. In: *2020 18th International Conference on Emerging eLearning Technologies and Applications (ICETA)*. IEEE, 2020. p. 749-755.
- [9] DCC++. *DCCWiki*. Dostupné tiež na: https://dccwiki.com/Main_Page
- [10] Od návrhu k prototypu. *Meet and Code*. Dostupné tiež na: <https://meet-and-code.org/sk/sk/event-show/8980>
- [11] Programujeme železnicu. *Meet and Code*. Dostupné tiež na: <https://meet-and-code.org/sk/sk/event-show/8697>
- [12] KAVANKOVA, I. Model Train Controlled by Microcomputer. In: *Silhavy, R. (eds) Software Engineering Perspectives in Systems*. CSOC 2022. Lecture Notes in Networks and Systems, vol 501. Springer, Cham. Dostupné tiež na: https://doi.org/10.1007/978-3-031-09070-7_14

Aplikovanie vyvíjanej metodiky na rozvoj informatického myslenia v kombinácii s divadlom - prvotné výsledky

Developing methodology was applied to develop CT thinking in combination with a drama – preliminary results

Karolína Miková

Univerzita Komenského v Bratislave
Mlynská dolina
84248 Bratislava
Slovensko
mikova@fmph.uniba.sk

Zuzana Kubincová

Univerzita Komenského v Bratislave
Mlynská dolina
84248 Bratislava
Slovensko
kubincova@fmph.uniba.sk

ABSTRACT

In this paper, we describe the planning, progress and results of the application of the currently developed methodology. The methodology is developed as part of the Erasmus project Fertile and focuses on the development of Computational thinking skills among students at all school levels using educational robotics, in combination with another subject from the field of art (drawing, music, literature, or drama), while it should use blended learning methods. The results presented in this article are from the first verification of a problem-based project created based on the methodology at a private secondary school with 9th grade students. The project combines the theme of Greek myths and robotics, applied specifically with LEGO Spike Prime. At the end of the article, we present recommendations for the further development of the methodology and for the use of the selected robotic kit.

Keywords

Educational robotics. Computational thinking. Drama. LEGO SPIKE Prime. Secondary school.

ABSTRAKT

V tomto článku opisujeme plánovanie, priebeh a výsledky aplikovania aktuálne vyvíjanej metodiky. Metodika je vyvíjaná v rámci Erasmus projektu Fertile a zameriava sa na rozvoj zručnosti informatického myslenia u žiakov všetkých stupňov škôl pomocou edukačnej robotiky, v kombinácii s ďalším predmetom z oblasti umenia (výtvarná, hudobná, literatúra, či dráma), pričom by mala využívať metódy kombinovaného vyučovania (ang. Blended learning). Výsledky, ktoré v tomto článku prezentujeme sú z prvého overenia problémovo orientovaného projektu vytvoreného na základe metodiky na súkromnej základnej škole so žiakmi 8-ročného gymnázia. Projekt spája tému grécke báje a robotika, aplikovanú konkrétne pomocou LEGO Spike Prime. V závere článku uvádzame odporúčania pre ďalší vývoj metodiky aj pre používanie vybranej robotickej stavebnice.

Kľúčové slová

Edukačná robotika. Vývoj Metodiky. Informatické myslenie. Predstavenie. LEGO SPIKE Prime. Druhý stupeň ZŠ.

1 ÚVOD

Po zavedení lockdownu vzniklo pre učiteľov množstvo problémov ako realizovať výučbu [1]. Medzi najčastejšie preberané problémy patrili rôzne psychické úzkosti ale aj rôzne technické problémy [2].

Jak na strane žiakov, tak na strane učiteľov. Z pohľadu výučby negatívne zasiahlo toto online vzdelávanie obzvlášť edukačnú robotiku. Zvyčajne sa vyučuje na hodinách informatiky, ktorá buď nebola vyučovaná vôbec, alebo ak bola, zameriavala sa na základné digitálne zručnosti, v lepšom prípade na programovanie – ako sme sa dozvedeli z rozhovorov, ktoré sme realizovali s učiteľmi. Tieto rozhovory boli súčasťou väčšieho výskumu¹, ktorý sme robili s pedagógmi základných a stredných škôl, v rámci projektu Erasmus Fertile². Medzi učiteľmi išlo len o ojedinelé prípady, kedy sa pustili do vyučovania edukačnej robotiky aj počas lockdownu. Na základe tejto skúsenosti, ktorú mali aj kolegovia vo viacerých krajinách venujúcich sa vzdelávaniu pedagógov a edukačnej robotike zároveň, vznikla myšlienka vytvoriť metodiku a platformu, ktorá by spájala a podporovala vyučovanie edukačnej robotiky aj v situáciách, ako bol lockdown a to pomocou kombinovaného vyučovania (angl. *blended learning*) [3]. Keďže v zahraničí je bežné prepájanie predmetov, tak ako najatraktívnejšia kombinácia bolo umenie, resp. „arts“. V našich podmienkach pod tým rozumieme výtvarnú výchovu, hudobnú výchovu, drámu, či literatúru. Túto myšlienku rozvíja a podporuje projekt Erasmus Fertile s názvom „Artful educational robotics to promote computational thinking in a blended learning context“, ktorý má ako jeden zo svojich cieľov vytvorenie metodiky pre učiteľov, podľa ktorej by učelia boli schopní aj sami písať kvalitné medzi-predmetové, problémovo orientované projekty. Má vzniknúť aj platforma, ktorá by podporovala vzájomnú komunikáciu a kooperáciu, ako medzi učiteľmi, tak medzi žiakmi.

U nás na Slovensku sa veľmi ťažko realizujú medzi-predmetové projekty, ktoré potrebujú spoluprácu viacerých učiteľov. Počúvame to od rôznych učiteľov z praxe naprieč celým Slovenskom, a dokazujú to aj mnohé tvrdenia učiteľov dopĺňujúceho pedagogického štúdia (DPŠ), s ktorými sa stretáme každý semester, či učiteľia účastní v našom výskume. Vytvorenie metodiky v projekte má za cieľ túto spoluprácu uľahčiť a pomôcť s prípravou, alebo s ponúknutím hotových exemplárnych materiálov, ktoré by spájali edukačnú robotiku a ďalšie predmety ako napr. hudobnú výchovu, výtvarnú výchovu, literatúru, drámu a pod, pričom by sa zameriavali primárne na rozvoj zručností informatického myslenia [4].

Aktuálne sa projekt nachádza v štádiu, kedy výskumníci zo všetkých zúčastnených krajín (Grécko, Španielsko, Česko a Slovensko) overujú na svojich školách prvotný návrh metodiky.

¹ Bližšie opíšeme v kapitole 3. Metodológia práce

² fertile-project.eu/

V tomto článku opisujeme našu konkrétnu skúsenosť s aplikovaním tejto metodiky na jednej základnej škole, kde sme odučili problémovo orientovaný projekt a na základe takto zozbieraných výsledkov budeme navrhovať ďalšie zmeny vo vývoji metodiky.

2 VYVÍJANÁ METODIKA

Metodika bola vyvinutá na základe štúdie [5], ktorá sa zaoberala rozvojom zručností infromatického myslenia počas aktivít s edukačnou robotikou. Zároveň sme sa pri jej tvorbe snažili zohľadniť požiadavky projektu a to, že metodika bude určená učiteľom. Ako základnú koncepciu sme si zvolili problémovo orientované vyučovanie [6], ktoré sa ukazuje ako vhodné v kombinácii s edukačnou robotikou. Metodika aktuálne pozostáva z piatich krokov:

1. **Porozumenie zadaniu** – Žiaci identifikujú problémy a podmienky v zadanej výzve.
2. **Generovanie nápadov** – Žiaci načrtnú jeden alebo viac nápadov, ktoré by mohli spĺňať podmienky dané v zadaní problému.
3. **Formulácia riešenia** – Myšlienka sa majú pretransformovať do formulácie riešenia tak, aby zohľadňovali požiadavky problému, čím mobilizujú vedomosti súvisiace s charakteristikami robota a artefaktu.
4. **Práca na riešení** - Krok je úspešne vykonaný, keď je formulované riešenie úplne dokončené prostredníctvom konštrukcie artefaktu a naprogramovaného robota.
5. **Kontrola riešenia** – Kým robot vykonáva príkazy, študenti pozorujú realizovaný artefakt a naprogramovaného robota a hodnotia zhodu so stanovenými podmienkami a primeranosť.

Jednotlivé kroky majú svoje členenia, v ktorých si majú učitelia vybrať z ponúkaných možností. Takéto sprevádzanie učiteľov pri spisovaní projektu má za cieľ zabezpečiť kvalitný rozvoj zručností žiakov v oblasti infromatického myslenia [7]. Aktuálne sa metodika zameriava na tieto hlavné zručnosti:

- Abstrakcia
- Dekompozícia
- Identifikovanie vzorov
- Algoritmické myslenie
- Hodnotenie

3 METODOLÓGIA PRÁCE

Skôr ako opíšeme výskumný problém, ciele výskumu, zber dát a výber účastníkov, by sme chceli zdôrazniť, že projekt prebieha už rok a v rámci tohto obdobia sme zrealizovali prieskum pomocou špeciálne navrhnutého online dotazníka pre učiteľov edukačnej robotiky. Dotazník bol rozposlaný vo všetkých krajinách projektu, v tomto príspevku sa však odvolávame len na výsledky zozbierané na Slovensku. Taktiež sme realizovali niekoľko riadených skupinových rozhovorov s učiteľmi informatiky vybranými na základe výsledkov tohto dotazníka. Išlo o 15 pedagógov zo základnej, strednej a vysokej školy. Údaje z rozhovorov boli nahrávané a kvalitatívne analyzované pomocou otvoreného kódovania [8]. Na výsledky z tohto dotazníka a rozhovorov sa odvolávame na viacerých miestach v článku, konkrétne napríklad v prvej kapitole.

3.1 Cieľ

Výskumným problémom, ktorý sme si stanovili a ktorý opisujeme v tomto článku bolo overiť kvalitu a potenciál navrhnutéj

metodiky. Teda, či sa dá podľa nej zrealizovať problémovo orientovaný projekt v kombinácii s edukačnou robotikou a divadelným vystúpením tak, aby rozvíjal niektoré (minimálne jednu) zručnosti infromatického myslenia.

Ako hlavný cieľ sme si preto stanovili navrhnuť v kooperácii s učiteľmi vybranej základnej školy roboticko-umelecký projekt podľa metodiky, zrealizovať ho a analyzovať zozbierané dáta.

3.2 Participanti

Výber účastníkov do tohto výskumu bol ovplyvnený školami, ktoré sú zahrnuté do projektu ako partnerské a učiteľmi umenia a informatiky, ktorí na týchto školách pracujú. Na základe dostupnosti (času a miesta) sme si vybrali jednu školu, kde boli učitelia oboch predmetov ochotní v danom termíne spolupracovať. Táto škola má isté špecifiká, ako slovné hodnotenia od prvého stupňa ZŠ aj na vysvedčení, zaradenie projektového vyučovania do školského roka, či polročné testy z hlavných predmetov.

Pre projekt bola vybraná trieda žiakov deviatego ročníka (14-16 roční žiaci) súkromnej základnej školy v hlavnom meste. Vybraná trieda pre lockdown počas covid pandémie nepracovala s robotmi a teda išlo o ich prvé stretnutie s edukačnou robotikou. Žiaci v minulosti síce pracovali v prostredí Scratch, avšak nemali nejaké hlbšie vedomosti o programovacích konceptoch. Trieda podľa triednej učiteľky aj ich učiteľky informatiky je šikovná, snaživá a pracuje sa s ňou dobre.

3.3 Zber dát

Počas testovania sme nemali zabezpečené povolenie pre nahrávanie video a audiozáznamov, takže sme v súlade s GDPR zbierali len také typy údajov, ktoré by nezachytávali tváre osôb a iné citlivé údaje o nich. Teda sme zaznamenávali fotografie a videonahrávky zostrojených modelov a ďalších rekvizít [9]. Tiež sme zozbierali údaje z pracovných listov, ktoré žiaci vyplňali počas prvého stretnutia. A záverečné dáta z krátkeho online dotazníka, v ktorom sa žiaci na záver projektu vzájomne hodnotili. Posledný typ dát bol rozhovor s učiteľkou informatiky pred, počas a po testovaní.

4 PRIEBEH TESTOVANIA

4.1 Príprava

Na začiatok sme mali niekoľko spoločných online stretnutí, kde sme navrhli a spísali projekt podľa odporúčaní metodiky, podľa aktuálnych možností školy, dohodli sme konkrétne termíny a pripravili potrebné materiály. Metodika, ktorú sme chceli overiť, obsahuje niekoľko krokov (pozri Kapitolu 2), ktoré zabezpečujú kvalitné aplikovanie problémovo orientovaného vyučovania. Podľa týchto krokov, resp. fáz sme s učiteľkou literatúry a učiteľkou informatiky naplánovali, ako bude náš projekt na škole prebiehať. Najprv sme vybrali vhodnú tému. Tá bola na literatúre – Grécke báje. V rámci nej mali študenti v skupinách naštudovať pridelenú báju, spracovať ju tak, aby ju mohli zahráť ako krátku scénu a aby si ju ich spolužiaci zapamätali. Dohodli sme sa, že do týchto vystúpení žiaci majú zakomponovať robota. A to tak, že využijú buď pohyb, zvuk, alebo svietenie. Robot mohol dotvárať prostredie, v ktorom sa bája odohrávala, ale mohol predstavovať aj konkrétnu postavu z báje.

Keďže žiaci vybranej triedy dovtedy nepracovali s edukačnou robotikou, chceli sme sa v projekte zamerať na využitie takých vstupno-výstupných komunikátorov, ktoré sú pre robotiku typické a robia ju zaujímavejšou v porovnaní s bežným programovaním. Naším cieľom teda bolo, aby sa študenti najprv oboznámili so

stavebnicou ako takou, jej hardvérovým zložením, ale aj príkazmi, ktoré ovládajú zvuk, svietenie a základný pohyb. Vytvorili sme preto pre tento účel krátky pracovný list, ktorý žiakov počas pomerne krátkeho času efektívne viedol pri oboznamovaní sa s prostredím a ovládaním robota.

Pred začatím testovania na škole bolo potrebné aktualizovať softvér v robotoch, nainštalovať aplikácie do tabletov a spárovať tablety s robotmi. Keďže sme mali dva rôzne typy tabletov a na jednom type modelov bol problém s inštaláciou softvéru, tak sme týmto procesom strávili niekoľko, nie málo hodín. Napriek tomu sa pri overovaní ukázal istý problém, ktorý opisujeme v kapitole 4.3. Zhrnutie.

Istý čas nám zabralo v rámci prípravy aj nabíjanie samotných robotov, ktoré trvalo niekoľko hodín, čo v porovnaní napr. s Ozobotmi bolo niekoľko násobne dlhšie.

Pred testovaním bolo treba priniesť stavebnice a tablety do školy, zabezpečiť ich pripojenie na školskú wifi a odskúšať ich. Keďže išlo až o 10 sád, vyžadovalo si to návštevu školy navyše.

4.2 Priebeh

Samotný projekt bol navrhnutý na 5 stretnutí. Uvádzame stretnutia a nie vyučovacie hodiny, lebo išlo o 45 alebo 90 minútové stretnutia. Samotné testovanie, na ktorom sme sa zúčastnili ako výskumníci, prebiehalo počas dvoch dní, kedy sme boli na troch stretnutiach so žiakmi, pričom každé trvalo 2 vyučovacie hodiny.

4.2.1 Prvé stretnutie

Žiaci na hodinách literatúry s učiteľkou prebrali tému gréckej báje, kde dostali zadanie projektu. Boli rozdelení do skupín po 4-5 žiakov a každá skupina dostala pridelenú jednu báju, ktorú si mala nastudovať, spracovať, vytvoriť vlastný scenár predstavenia na 5-10 min a naučiť sa podľa scenára vystupovať.

4.2.2 Druhé stretnutie

Počas ďalšieho stretnutia žiaci v skupine dotvorili rekvizity a zrealizovali v rámci triedy samotné vystúpenia, kde prezentovali jednotlivé skupiny svojim spolužiakom nacvičené príbehy.

4.2.3 Tretie stretnutie

V rámci hodiny literatúry bola žiakom predstavená druhá časť projektu, kde boli oboznámení s problémovou úlohou, že do svojich scenárov budú mať zakomponovať roboty. Za domácu úlohu mali identifikovať možnosti v scenári, kde by to bolo najvhodnejšie. Preto na ďalších stretnutiach nemali väčšie problémy s brainstormovaním návrhov ako zakomponovať roboty, ale skôr so samotným zrealizovaním nápadu.

Doposiaľ na stretnutiach so žiakmi bola účastná len učiteľka literatúry. Na ďalších sme boli prítomní my, ako výskumníci, aj ako hlavní vyučujúci.

4.2.4 Štvrté stretnutie

Triedu sme mali celú, nerozdelenú, teda okolo 22 žiakov počas dvoch vyučovacích hodín, t.j. 2x45 min v ich triede bez počítačov. Pôvodné štvorice sme rozdělili na dve dvojice a počas tejto hodiny pracovala každá dvojica s jednou robotickou stavebnicou, jedným tabletom a jedným pracovným listom. Cieľom bolo, aby sa naučili zapnúť aplikáciu, spárovať robota s tabletom, rozhábať ho rôznymi príkazmi, nahráť, editovať a spustiť zvuky, či rozsvietiť rôzne farby pomocou LED svetiel na kočke. Taktiež mali za úlohu premyslieť si ešte raz scenár a preveriť, či im LEGO PRIME Spike umožňuje realizovať nápad, ktorý mali na zapojenie robota do predstavenia. V prípade, ak im ostal čas, mali začať pracovať na konštrukcii modelu.

Žiaci boli naozaj šikovní a nezaznamenali sme žiadne výrazné problémy s porozumením. Jedine neprejavili záujem vyplňať pracovný list, aj keď išlo o minimálnu prácu a niekoľkokrát sme ich na to upozornili. (pozri Obr.1 a Obr.2) Každá skupina stavala iný model a potrebovali sme overiť kvalitu ich porozumenia príkazom na programovanie správania robota. Tiež sme si všimli, že počas druhej vyučovacej hodiny bola ich pozornosť výrazne slabšia a horšie sa sústredili. Prejavil sa fakt, že nie sú zvyknutí pracovať 90 min bez prestávky.

MENO: *Silvia*
MENO: *Práček*

Názov báje: *O-tylos*
Názov robota: *Spike 7*
tablety: očka 5

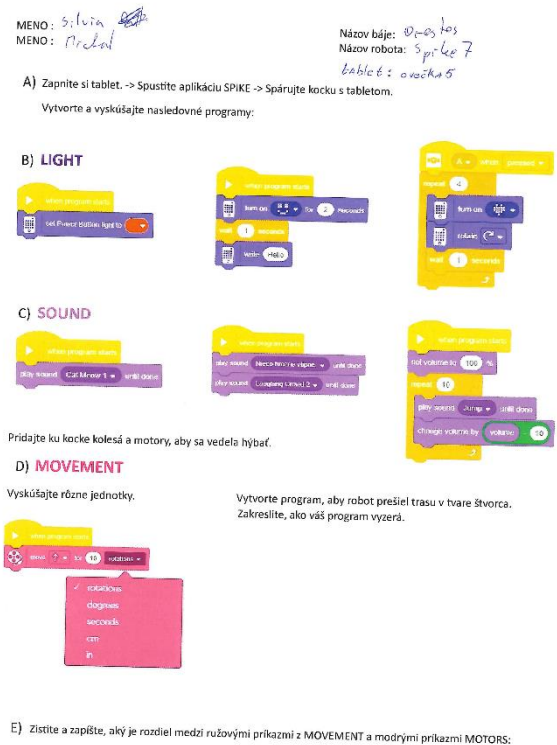
A) Zapnite si tablet. -> Spustíte aplikáciu SPIKE -> Spárujte kočku s tabletom.
Vytvorte a vyskúšajte nasledovné programy:

B) LIGHT

C) SOUND

D) MOVEMENT

E) Zistíte a zapíšte, aký je rozdiel medzi ružovými príkazmi z MOVEMENT a modrými príkazmi MOTORS.



Obr.1 Ukážka prvej strany vyplneného pracovného listu

MENO: _____
MENO: _____

Názov báje: _____
Názov robota: _____

F) Prediskutujte vo svojej skupine (4/5 člennej), ako by ste chceli využiť robotov pri prezentovaní svojej báje. Môžete využiť maximálne dvoch robotov, pričom vždy budete pracovať s jedným robotom maximálne dvaja. Sem stručne opíšte svoje výsledné návrhy:

*- hoci v prvej časti máme 2 roboty, jedna z nich bude hrať úlohu a druhá bude mať
nahrávanie a zvuky. Druhý robot má byť s pomocou svetiel robotu.*

Obr.2 Ukážka druhej strany toho istého vyplneného pracovného listu

4.2.5 Piate stretnutie

Toto stretnutie sme zrealizovali dvakrát, lebo sme žiakov rozdělili na polovicu tak, aby v každej skupine boli prezentované 3 gréckej báje. S oboma polovicami triedy sme mali stretnutie trvajúce dve vyučovacie hodiny. Na tomto stretnutí sa počas prvej vyučovacej hodiny snažili dokončiť svoje rozpracované modely, ktoré mali rozpracované a odložené ešte z predošlého stretnutia. Každá skupina mala k dispozícii dva roboty, keďže na predošlej hodine pracovali v dvojiciach. Počas druhej vyučovacej hodiny prezentovali svoje predstavenia spolu s robotickými modelmi. Na

Obr. 3 je niekoľko ukážok modelov vytvorených žiakmi. Po každom predstavení nasledovalo krátke hodnotenie od spolužiakov, ktoré bolo vytvorené v google forms a prihlasovali sa doň cez naskenovaný QR kód. Na predstavení sa zúčastnili okrem nás výskumníkov, aj obe pani učiteľky – literatúry a informatiky.

Otázky, ktoré boli v hodnotiacom formulári boli nasledovné:

1. Tvoje meno:
2. Akú báj hodnotíš:
3. Ohodnot' zakomponovanie robota do prezentovanej báje. Škála: 1-5
4. Ako by si hodnotil využitie robota LEGO SPIKE vzhľadom na možnosti, ktoré poskytuje? Škála: 1-5.
5. Za čo by si skupinu pochválil? Čo spravila ozaj výborne pri tomto vystúpení?



Obr.3 Ukážky niektorých modelov robotov a rekvizít použitých pri divadelnom vystúpení na poslednom stretnutí

4.3 Zhrnutie

Zozbierané dáta sme spracovali, analyzovali a zistené závery uvádzame v tejto kapitole. Rozdelili sme ich do nasledovných troch skupín: Hardvér a softvér, Vedomosti a schopnosti žiakov a Úprava metodiky.

4.3.1 Hardvér a softvér

Kvalitatívny výskum umožňuje zameranie sa na zaujímavé výsledky aj v prípade, že neboli výskumníkmi stanovené ako výskumný cieľ. To sa stalo aj nám. Jeden z výsledkov, ktorý sme zistili, bol mimo našich pôvodných cieľov. A to ohľadne používanej robotickej stavebnice LEGO PRIME Spike [10]. V minulosti sme aj pri iných rôznych projektoch pravidelne používali LEGO stavebnice, konkrétne LEGO WeDo (obe verzie) a nezaznamenali sme žiadne problémy. Avšak pri tomto overovaní, sme aj napriek predpripravenému párovaniu tabletov a robotov a aj napriek aktualizácií softvéru museli bojovať s množstvom problémov: náhodné odpájanie robotov od tabletu; zastavenie, až zamrznutie softvéru, resp. program po spustení nereagoval, alebo robot nereagoval na spustenie príkazu; enormne dlhé čakanie na znovu-spustenie softvéru a to v úplne náhodných situáciách nezávislých od konkrétneho tabletu či konkrétneho robota a pod.

Jedným z dôvodov, ktoré mohlo tieto problematické situácie spúšťať bola nová verzia aplikácie, ktorú v decembri spustilo LEGO Education a ktorá robila aj iným kolegom podobné problémy.

Naopak milým prekvapením bola výdrž batérie pri kockách LEGO SPIKE, ktoré po testovaní nestratili ani 10% z celkového nabitia.

Ako naozaj náročné sa nám ukázala aj samotná príprava na testovanie. V prípade, že tento proces bude zabezpečovať škola, resp. vyučujúci/a informatiky, môže to pre časovú náročnosť viesť k zníženej ochote takéto projekty realizovať.

4.3.2 Vedomosti a schopnosti žiakov

Ako sme písali v predošlých častiach, išlo o triedu, ktorá bola ochotná pracovať a bola zvyknutá na projektový spôsob výučby, teda nezaznamenali sme potrebu výrazného motivovania k práci a vynucovania pozornosti. Napriek tomu sa efektívnosť práce žiakov znižovala zvyšujúcim sa časom trvania stretnutia. Ukázalo nám to potencionálne riziko pri realizovaní projektov a nutnosť zväziť dĺžku trvania práce žiakov, v závislosti od veku.

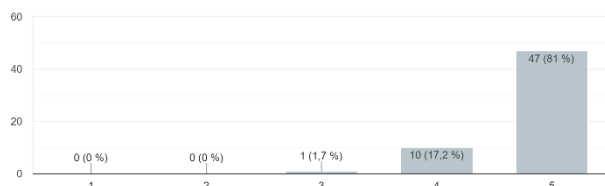
Aplikovanie kombinovaného učenia metód bolo v tomto konkrétnom prípade problémové, lebo táto konkrétna robotická hračka nemá simulátor, a žiaci nie sú zvyknutí v tomto období mávať domáce úlohy, takže sme využili aspoň online hodnotiaci nástroj. Pri tom sa nám ukázali rôzne zaujímavosti, nie vždy pozitívne.

Pri vzájomnom hodnotení sme očakávali, že žiaci nebudú úplne objektívni, ale budú k sebe nadpriemerne zhovievaví, čo sa aj potvrdilo. Na otázku 3 ohľadne zakomponovania robota do prezentovania dal len v jednom prípade najhoršie bodové hodnotenie jeden žiak – 3 body (pozri Obr.4). Pri 4-tej otázke ohľadne využitia potenciálu robota Spike boli žiaci kúsok prísnejší (pozri Obr. 5). Viac ráz ohodnotili spolužiakov 3 bodmi a raz dokonca 2.

Sebahodnotenie použilo 8 žiakov z celkového počtu 24, avšak nie všetci označili, že hodnotia svoj tím, takže bolo potrebné dohľadanie. Z týchto ôsmich len jeden žiak v otázke 4 nedal sám sebe plný počet bodov, inakšie všetky sebahodnotenia boli maximálne.

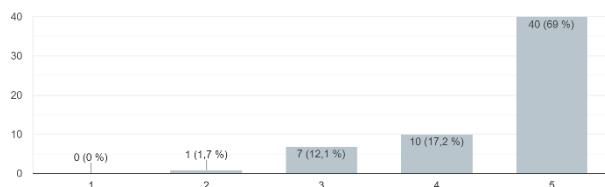
Zaujímavé bolo, že aj keď boli všetky skupiny zložené zo 4 žiakov, a na oboch posledných stretnutiach prezentovali vždy po tri skupiny, počet hlasov, ktoré dostali jednotlivé skupiny bol rôzny a pohyboval sa od 6 do 10.

Odhodnoť zakomponovanie robota do prezentovanej báje.
58 odpovedí



Obr.4 Výsledky k otázke 3. Hodnotenie spolužiakov o zakomponovaní robota do báje.

Ako by si ohodnotil využitie robota Spike, vzhľadom na možnosti, ktoré poskytuje?
58 odpovedí



Obr.5 Výsledky k otázke 4. Hodnotenie spolužiakov o využití potenciálu robota.

Taktiež bolo zaujímavé, že žiaci nehodnotili priebežne po každom predstavení, ako sme ich vyzývali, ale viacerí hodnotili naraz dve vystúpenia. Akoby si najprv všetko pozreli a až potom hodnotili. Otázka ostáva, či je to dobré, aby vedeli objektívnejšie hodnotiť, alebo to zhoršilo ich objektívnosť, lebo zabudli na predošlé predstavenia a ich kvality a chyby.

Poslednú otázku v hodnotení (čo spravila skupina pri vystúpení výborne) sme nechali otvorenú, lebo nás zaujímalo na akú oblasť sa žiaci zamerajú. Či to bude robotika, programovanie, herecký výkon, alebo kreativita pri vytváraní deja a kulís. Z celkových 57 odpovedí sa 21 týkalo robotiky, 18 hereckého výkonu a 18 všeobecných pochval. Keď sa lepšie pozrieme na hodnotenia robotiky, tak len jedna sa týkala dobrého naprogramovania a dve konštrukcie robota. Ostatné boli zamerané skôr na ideu akou bol robot využitý.

4.3.3 Úprava metodiky

Pri práci s metodikou v prípravnej časti sme zaznamenali dva menšie problémy.

Metodika nerátala s projektami s krátkym trvaním a jednotlivé kroky obsahovali ďalšie časti. Tie sú vhodné, ba dokonca potrebné, ak projekt trvá dlhší čas. Napríklad 6-10 vyučovacích hodín. Avšak pre projekty, ako ten náš, ktorý bol naplánovaný na 5 hodín, pričom samotné rozvíjanie informatického myslenia sa dialo v podstate na troch stretnutiach. Práca sa metodikou bola v tomto bode pre nás nepohodlná a treba ju upraviť, alebo navrhnúť alternatívu.

Druhý problém bol, že zručnosti, ktoré metodika ponúkala boli veľmi všeobecné a nezameriavali sa na rôzne ďalšie zručnosti, ktoré pri riešení problémov žiaci zapájajú. Pri takejto stručnej a všeobecnej ponuke zručností sa nám ťažko zaraďovali činnosti z projektu do čiastkových krokov metodiky. Neboli sme si vždy istí, či sme zvolili vhodnú zručnosť, lebo takmer všetky nám spadali do jednej kompetencie. Metodika taktiež neriešila čiastkové napĺňanie krokov.

5 Diskusia a plány do budúcnosti

Na základe našich zistení, ktoré sme získali počas aplikovania metodiky, sme si uvedomili, že je potrebné vytvoriť kvalitnú a zrozumiteľnú príručku s návodmi, ktorá bude pomáhať učiteľom vedieť správne využívať metodiku, keďže samotný proces vysvetlenia nie je až taký jednoduchý. S týmto ráta aj náš Erasmus projekt a sme radi, že sme mohli takto získať prvé skúsenosti, čo sú podstatné kroky pri oboznamovaní učiteľov s metodikou.

Ďalším zistením je fakt, že žiaci v tomto veku nepracujú efektívne, ak sa doba ich samostatnej práce predĺži na dlhšie, ako sú zvyknutí, aj keď môže ísť pre nich o zaujímavú tému. Je preto potrebné dávať si pozor pri plánovaní, ako dlhé budú stretnutia so žiakmi a len v prípade nutnosti naťahovať dĺžku stretnutia na viac ako jednu vyučovaciu hodinu.

Otázkou, na ktorú sa bude treba zamerať a prediskutovať ju aj s ďalšími kolegami je otázka hodnotenia. Nakoľko je dobre aplikovať prísnejšie a možno aj inštruktívnejšie orientované vzájomné hodnotenie, alebo ostať vo forme voľného hodnotenia, z ktorého môže učiteľ lepšie spoznať charakter a povahu žiakov. Resp. otvára sa otázka, aký koncept (vzájomného) hodnotenia do takéhoto projektu zaradiť.

Ohľadne informatické myslenia vieme vyjadriť len subjektívne hodnotenie, že študenti bez pomoci učiteľa navrhli a zrealizovali kvalitné a vtipné riešenia ako po dizajnovej, či technickej, tak aj programátorskej stránke. Avšak úplne objektívne posúdenie nevieme spraviť zo zozbieraných dát aj pre problémy, ktoré sme mali s občasne nefunkčným softvérom. Pre kvalitnejšie posúdenie rozvoja zručností je nutné spraviť viaceré merania. V prípade zložitejšieho projektu, kde žiaci už dlhšie pracovali s robotickou stavebnicou si vieme predstaviť zamerať sa na podrobné skúmanie zručností informatického myslenia.

Do budúcnosti by sme mohli zvážiť možno jednoduchšiu robotickú stavebnicu, ktorá sa bude ľahšie prenášať a bude mať stabilne fungujúci softvér. Prípadne použiť takú robotickú stavebnicu, ktorou škola disponuje. Uvedomujeme si, že ohľadne aktualizácií softvéru nemáme na situáciu žiaden dosah, napriek tomu však môže tento fakt výrazne narušiť plynulosť priebehu výučby.

V ďalších krokoch projektu budeme vytvárať s učiteľmi z praxe exemplárne príklady projektov, ktoré sa budú zameriavať na ďalšie témy ako napr. hudobná, či výtvarná. V každom prípade sa plánujeme zamerať aj na ďalšie stupne vzdelávania ako je stredná a vysoká škola.

6 Záver

Cieľom nášho výskumu bolo navrhnúť roboticko-umelecký projekt podľa metodiky, zrealizovať ho a analyzovať zozbierané dáta. Projekt sme navrhli v spolupráci s učiteľkami informatiky a literatúry 8-ročného súkromného gymnázia pre žiakov kvarty. Téma projektu bola navrhnúť a zrealizovať predstavenie konkrétnej gréckej báje s použitím robotov LEGO PRIME Spike tak, aby ich predstavenie bolo atraktívnejšie. Z programátorskej časti mohli použiť príkazy na ovládanie motorov, zvukov a svetiel použili motory. Overenie prebehlo počas dvoch stretnutí, ktoré trvali 90 min.

Na základe analýzy zozbieraných dát sme odhalili niekoľko zaujímavých zistení. A to, že softvér pre prácu s LEGO PRIME Spike po decembrovej aktualizácii má viacero bagov, že žiaci ZŠ optimálne pracujú len počas 45 minút a následne ich pozornosť výrazne klesá, že pri vzájomnom hodnotení treba zvažovať výhody a nevýhody inštrukcionistického prístupu pri hodnotení a mali by

sme zväziť využitie robotických stavebníc dostupných na škole. V rámci samotnej metodiky by bolo treba ponúknuť pestrejšiu škálu zručností pre informatické myslenie a zohľadniť v metodike projekty, ktoré trvajú jednu či dve vyučovacie hodiny.

Do budúcnosti plánujeme naše zistenia zdieľať s ďalšími kolegami a zohľadniť ich vo výslednom návrhu metodiky. V najbližšom polroku máme naplánované a už rozbehnuté ďalšie dva projekty, v ktorých bude metodika použitá v kombinácii s hudobnou a výtvarnou výchovou.

POĎAKOVANIE

Tento výskum bol podporený z projektu VEGA 1/0621/22 ako aj z projektu APVV-20-0353 a FERTILE 2021-1-EL01-KA220-HED-000023361.

BIBLIOGRAFICKÉ ODKAZY

- [1] Prieskum zameraný na online a diaľkové vzdelávanie. <https://www.schooleducationgateway.eu/sk/pub/viewpoints/surveys/survey-on-online-teaching.htm>
- [2] The Global Report on the Status of Teachers 2021. <https://eiie.io/StatusOfTeachers2021E>
- [3] ANDERSEN, K., GEIRSDOTTIR, G., THORSTEINSSON, S.E., THORBERGSSON, H., GUDMUNDSSON, K.S. Engineering education case studies: Engaging students in blended learning. In 2018 IEEE International Professional Communication Conference (ProComm). p. 203-206. IEEE, 2018.
- [4] WING, JM. Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 366, no. 1881 (2008): p. 3717-3725.
- [5] ATMATZIDOU, S., DEMETRIADIS, S. Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. In *Robotics and Autonomous Systems*, 75, p.661-670, 2016.
- [6] DENIS, B., HUBERT, S. Collaborative learning in an educational robotics environment. In *Computers in human behavior*, 17(5-6), p.465-480, 2001.
- [7] SELBY, C., WOOLLARD, J. Computational thinking: the developing definition. ACM: 2010.
- [8] ŠVAŘÍČEK, R., ŠEĎOVÁ, K. et al. *Kvalitativní výzkum v pedagogických vědách*. Praha: Portál, 2014. 384 s. ISBN 978-80-262-0644-6.
- [9] CRESWELL, J. *Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research*. 3 vyd. New Jersey: Pearson Education, 2008. ISBN-13: 978-0-13-207208-0
- [10] LEGO SPIKE Prime. <https://www.lego.com/en-sk/product/lego-education-spike-prime-set-45678>

Ako posúdiť vhodnosť robotickej stavebnice - LEGO SPIKE Prime

How to assess the suitability of a robotic kit - LEGO SPIKE Prime

Karolína Miková

Univerzita Komenského v Bratislave
Mlynská dolina
84248 Bratislava
Slovensko
karolina.mikova@fmph.uniba.sk

Barbora Stenová

Univerzita Komenského v Bratislave
Mlynská dolina
84248 Bratislava
Slovensko
stenova7@uniba.sk

ABSTRACT

In this article, we are dedicated to identifying the criteria that would be starting point when assessing the adequacy of the LEGO Spike Prime robotic kit in teaching informatics. We carried out the research of the first phase with pupils of the lower secondary school and pupils of the 1st year of high school. Our results from the first phase of testing show that the main aspect of this adequacy is time adequacy, which is a consequence of several criteria. We identified the following criteria: Cognitive difficulty, Model construction, Motivation, Creativity and Research. Based on the students' skills in these areas, it is possible to estimate the optimal time requirement so that the students take away quality knowledge from the lessons.

Keywords

Robotics. LEGO SPIKE. Appropriateness. Criteria. Computer science.

ABSTRAKT

V tomto článku sa venujeme identifikovaniu kritérií, ktoré by boli východiskové pri posudzovaní primeranosti robotickej stavebnice LEGO Spike Prime pri vyučovaní informatiky. Výskum prvej fázy sme realizovali so žiakmi 2. stupňa základnej školy a žiakmi 1. ročníka strednej odbornej školy. Naše výsledky z prvej fázy testovania ukazujú, že hlavným aspektom tejto primeranosti je časová primeranosť, ktorá je dôsledkom viacerých kritérií. Identifikovali sme tieto kritéria: Kognitívna náročnosť, Konštrukcia modelu, Motivácia, Kreativita a Bádanie. Na základe zručností žiakov v týchto oblastiach je možné odhadnúť optimálnu časovú náročnosť tak, aby si žiaci z vyučovania odniesli kvalitné poznatky.

Kľúčové slová

Robotika. LEGO SPIKE. Primeranosť. Kritéria. Informatika.

1 EDUKAČNÁ ROBOTIKA

Robotické hračky sú moderné pomôcky, pomocou ktorých vieme vyučovať informatiku. Rozvíjajú veľa zručností u žiakov ako napríklad jemnú motoriku, kreativitu, plánovanie, spoluprácu, kritické a algoritmické myslenie. Podnecujú žiakov vyjadrovať svoje vlastné myšlienky a nápady, ktoré vkladajú, či už do konštrukcie modelu, alebo tvorby programu [1, 2]. Ak sú tieto robotické hračky programovateľné, vieme pomocou nich vyučovať aj konkrétne programovacie konštrukty [3]. Otázkou ostáva, či primerane. Primerane veku, primerane kognitívnym vedomostiam.

Napríklad sa začne používať textový programovací jazyk príliš skoro, alebo sa napr. zavedie používanie LEGO NXT robotov pre 6. ročník ZŠ. Pri rôznych situáciách stretávame učiteľov/ky, ktorí sa pýtajú na vhodné robotické stavebnice do ich škôl, pre ich žiakov. Ako však odporučiť vhodnú stavebnicu? Aké sú oblasti, v ktorých ju treba posudzovať?

V našich výskumných projektoch sa preto v posledných rokoch zameriavame na analyzovanie rôznych softvérov, detských programovacích jazykov, robotických stavebníc a robotov [4,5]. Analyzujeme reakcie žiakov z pohľadu kognitívnej náročnosti pri práci s rôznymi robotmi a robotickými stavebnicami v kombinácii s rôznymi vzdelávacími materiálmi, analyzujeme možnosti vybraného robota na vyučovaní informatiky (programovania) a pod. Hľadáme rôzne kritéria, pomocou ktorých by sme vedeli jednotlivé robotické pomôcky a materiály k nim posudzovať. Takéto kritéria by mali vytvárať ucelený systém, ktorý by v budúcnosti učiteľom pomáhal pri výbere robotických pomôcok.

V tomto príspevku sa zaoberáme jednou konkrétnou stavebnicou a to analýzou robotickej stavebnice LEGO Spike Prime [6], ktorá je náhradou za známou stavebnicu LEGO NXT, resp. jej modernejšieho nástupcu LEGO EV3.

2 LEGO SPIKE PRIME

Je robotická stavebnica od firmy LEGO, ktorá nahradila predošlú verziu LEGO EV3 Mindstorms [7], ktorá je určená pre žiakov vo veku 10 a viac rokov. Rovnako ako LEGO WeDo 2 [8] poskytuje veľkú kreativitu v konštrukciách modelov.

2.1 Hardvérové zloženie

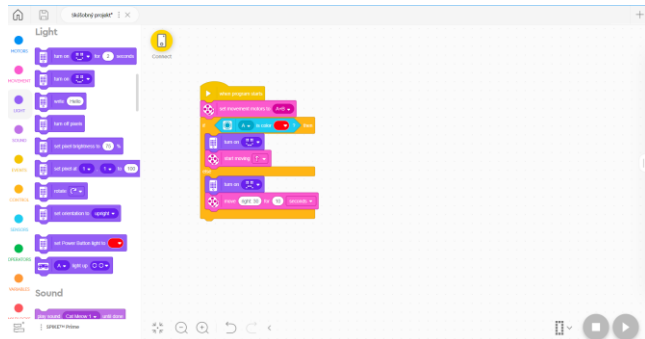
Základná sada tejto stavebnice (pozri Obr.1) obsahuje cez 500 farebných LEGO dielikov, programovateľný smart Hub so svetelnou maticou o rozmeroch 5x5 dielikov, centrálnym tlačidlom, ktoré svieti, pravým a ľavým tlačidlom, integrovaným reproduktorom a senzorom náklonu. Táto robotická stavebnica má širokú škálu interaktivity, ktorú zabezpečujú okrem Hub zariadenia aj ďalšie senzory: senzor sily, vzdialenosti, farebný senzor a 2 stredné a 1 veľký motor. Tieto zariadenia sa pripájajú do 6-tich portov na Hube. Vstavaná batéria sa nabíja cez microUSB kábel. Umožňuje viaceré spôsoby programovania pohybu, ako relatívny ale aj absolútny pohyb. Pri programovaní pohybu môžeme nastaviť dĺžku kroku, čas behu motora, alebo uhol o koľko sa má motor otočiť.



Obrázok 1. Základný set Stavebnice LEGO SPIKE Prime [6]

2.2 Aplikácia

Programovacie prostredie je založené na základe programovacieho jazyka Scratch (pozri Obr.2), k dispozícii je aj prepnutie do textového programovania v jazyku Python. Existujú dve verzie: desktopová aplikácia a online verzia. Hub k aplikácii pripájame pomocou USB kábla alebo Bluetooth. V aplikácii sú, resp. boli 2 módy: „download mode“, kedy sa program načíta a uloží do Hub na zvolenú pozíciu a po odpojení Hubu od aplikácie je tento program uložený na danej pozícii a program vieme kedykoľvek spustiť. Druhý mod „Streaming mode“ - program sa vykonáva v reálnom čase a počas behu programu môžeme meniť parametre sa po aktualizácii v decembri 2022 už nenachádza. Program sa do Hub neuloží a nevieme ho spustiť po odpojení. Aplikácia obsahuje tiež návody na konštrukciu modelov (so zameraním na inžinierstvo a informatiku), rôzne lekcie s ukázkami programov a následnými úlohami, ktoré vedú k riešeniu zadaného problému a zlepšovaniu funkcie robotického hračky. Lokalizácia do slovenčiny nie je možná.



Obrázok 2. Ukážka online aplikácie aj s programom pre LEGO SPIKE Prime

3 METÓDY VÝSKUMU

Témou nášho výskumu, ktorý prezentujeme v tomto príspevku je výučba s robotickou stavebnicou LEGO Prime Spike, primárne so žiakmi na 2. stupni ZŠ. Na Slovensku neexistujú overené a štátnym orgánom schválené výučbové materiály, ktoré by určovali primerané zaradenie konkrétnych robotických stavebníc na stupne

vzdelávania, tobôž do jednotlivých ročníkov. Zvolili sme si preto vek, ktorý odporúča výrobca stavebnice.

Nami vybraná robotická stavebnica LEGO PRIME Spike má síce aplikáciu, ktorá ponúka rôzne medzi-predmetové aktivity, avšak pre systematické vyučovanie programátorských konceptov nie sú tieto materiály úplne vhodné. Aby sme vedeli určiť primerané zaradenie úloh do ročníkov, dali sme si za cieľ iteratívnym spôsobom preskúmať, ako reagujú žiaci na 2. stupni ZŠ, resp. žiaci vo veku 10-15 rokov na túto stavebnicu na hodinách informatiky v rôznych ročníkoch. Pre potreby výskumu sme si zvolili kvalitatívny výskum vývojom [9], ktorý nám umožňoval aktívne zasahovať do deja ako učiteľ, aj ako výskumník. A to priebežne a počas ďalších etáp.

3.1 Ciele výskumu

V našom výskume máme ako primárny cieľ identifikovať kritéria, resp. oblasti, podľa ktorých vieme posúdiť vhodnosť zaradenia stavebnice LEGO Spike Prime do výučby, respektíve do ročníkov. Druhým cieľom je vybrať najvhodnejšiu postupnosť modelov, ktoré budú umožňovať systematické rozvíjanie vedomostí a zručností žiakov pri používaní programovacích konštruktov. Výsledkami druhého cieľa sa v tomto príspevku nebudeme zaoberať, ale čiastočne ho budeme zohľadňovať vo výsledkoch. V rámci našej práce sme si teda na základe zvolených cieľov a výskumného problému položili nasledujúcu výskumnú otázku:

V akých oblastiach vieme posudzovať vhodnosť stavebnice LEGO PRIME Spike pre vyučovanie informatiky?

Aby sme mohli získať dáta potrebné na zodpovedanie našich cieľov, museli sme navrhnuť také aktivity, ktorých obsah bol v súlade so Štátnym vzdelávacím programom [10]¹ a ktoré zohľadňovali kognitívny stav žiakov v daných ročníkoch. Tento krok nepokladáme za cieľ nášho výskumu, ale skôr ako tvorbu nástroja na zber dát, preto sa samotnými otázkami a úlohami nebudeme podrobnejšie v článku zaoberať. Ich obsah sme komunikovali s odborníkmi a tvorili sme ich na základe viacerých didaktických zásad [11, 12].

3.2 Výber vzorky

Našu výskumnú vzorku tvorili žiaci 5. až 8. ročníka pomerne veľkej spojenej základnej školy v hlavnom meste, pričom škola nemá špeciálne zameranie. V každom ročníku sme pracovali s polovicou triedy, v priemere 10 - 13 žiakov. Túto vzorku sme vyberali podľa vekového odporúčania od firmy LEGO pre danú robotickú stavebnicu. Na základe zistení z testovania sme sa rozhodli otestovať prácu s touto stavebnicou aj so staršími študentmi, aby sme porovnali kognitívnu a aj časovú náročnosť práce s touto robotickou stavebnicou. 14 študentov prvého ročníka strednej technickej školy (tiež z hlavného mesta) prišlo na Univerzitu, kde pracovali na 2 aktivitách s robotickou stavebnicou. Išlo o rovnaké aktivity, ako mali žiaci na základnej škole.

3.3 Metódy zberu dát

Na zber dát sme využili rôzne metódy kvalitatívneho výskumu. Išlo o pozorovanie a zapisovanie všetkých javov a dejov, ktoré sa diali na overovacích hodinách. Na týchto hodinách sme zaznamenávali aj fyzické dáta ako videozáznam, fotografie, vyplnené pracovné listy a písomné spätné väzby žiakov. Hodiny sme nahrávali na

¹ Aj keď aktuálne prebieha kurikulárna reforma, ktorá každú chvíľu dá do platnosti nový ŠVP, v čase výskumu platil pôvodný inovovaný ŠVP.

videokameru, aby sme sa vedeli k overovaniu vrátiť a zachytiť aj detaily práce žiakov s robotickou stavebnicou LEGO Prime Spike. Dáta tvorili aj pracovné listy, s ktorými žiaci v priebehu hodín pracovali a zapisovali si tam svoje myšlienky a aj odpovede na otázky. To nám pomohlo k lepšej predstave, ako žiaci premýšľajú pri procese programovania robotickej stavebnice. Využili sme aj metódu zberu terénnych zápiskov priamo v priebehu výučby.

3.4 Metódy Spracovania dát

Získané dáta sme následne podrobili analýze, aby sme dostali odpovede na naše výskumné otázky a vedeli splniť ciele nášho výskumu. Vzhľadom na povahu zozbieraných dát sme samozrejme zvolili kvalitatívne metódy analýzy. Ako jednu z výskumných stratégií, na zodpovedanie našich výskumných otázok a dosiahnutie cieľov výskumu, sme použili techniku kontrastovania [13]. Táto technika využíva porovnávanie dát dvoch najrozdielnejších prípadov a jej výsledkom je škála, do ktorej sa umiestňujú dáta zo všetkých ostatných prípadov. V našom prípade išlo o dáta z vyučovacích hodín s robotickou stavebnicou LEGO Spike Prime. Ako hraničné prípady sme si zvolili vyučovacie hodiny v piatom ročníku ZŠ a v prvom ročníku SŠ. Pre tieto prípady sme okódovali dáta a špecifikovali vlastnosti, ktoré mali odlišné. Týmto spôsobom nám vznikli krajné hranice výslednej škály. Následne sme do vytvorenej škály zasádzovali ostatné prípady - dáta z vyučovacích hodín v ostatných ročníkoch. V tejto technike sa kladie dôraz na špecifikované vlastnosti jednotlivých prípadov a ich umiestnenie do škály. Škálu, teda východiskové vlastnosti sme viac krát upravovali tak, aby jej výsledná podoba zohľadňovala všetky špecifické vlastnosti jednotlivých prípadov.

4 VÝSLEDKY

V tejto kapitole prezentujeme výsledky vyššie opísanej analýzy, ktoré sú odpoveďou na našu výskumnú otázku, aké sú oblasti, v ktorých môžeme posudzovať primeranosť robotickej stavebnice LEGO Prime Spike pri využití informatiky. Predstavíme nami vytvorené oblasti, resp. kritéria, ktoré sme spozorovali pri realizovaní testovania. Tieto kritéria, alebo oblasti opisujú zručnosti žiakov, ktoré sa rozvíjajú pomocou výučby s LEGO SPIKE PRIME. Tieto spozorované zručnosti vieme zadeliť aj do troch domén :

- kognitívna (bádanie a kognitívna náročnosť)
- psychomotorická (konštrukcia)
- afektívna (kreativita a motivácia)

Jednotlivé oblasti sme identifikovali a posudzovali aj na základe pedagogickej komunikácie [14], ktorú sme zaznamenávali a analyzovali. Išlo o komunikáciu medzi žiakom a učiteľom, medzi žiakmi a niekedy aj komunikáciu so sebou samým. Otázky, ktoré kládli žiaci v rámci tejto pedagogickej komunikácie nám odhaľovali úroveň zručnosti žiakov v rôznych oblastiach. Pozorovali sme, kedy a aké otázky kládli žiaci. Či išlo o otázky neporozumenia, ako často sa pýtali, či za akým účelom bola otázka položená. Otázky žiakov nám odkryli, či si boli žiaci svojou prácou istí, či rozumeli zadaniam, či potrebovali pomoc učiteľa alebo či potrebovali zrýchliť tempo, lebo im bolo všetko jasné. Oblasti, ktoré sme identifikovali boli: Kognitívna náročnosť, Konštrukcia modelu, Motivácia, Kreativita a Bádanie. V nasledujúcich kapitolách tieto oblasti priblížime podrobnejšie.

4.1 Kognitívna náročnosť

Pri hodnotení náročnosti jednotlivých úloh s LEGO SPIKE Prime sme brali do úvahy čas vypracovania, náročnosť programovania, samostatnosť riešenia, teda či žiaci pracovali samostatne, alebo

potrebovali našu pomoc pri riešení úlohy. Pozorovali sme ako často žiaci niečo nevedeli, akých chýb sa dopúšťali a z akých dôvodov. Sledovali sme aj, či žiaci rozumejú zadaniam a pokynom.

V piatom, dokonca aj v šiestom ročníku preberané koncepty boli pre žiakov nadsadené nad ich vedomosti a schopnosti programovania. Žiaci sa veľa pýtali, žiadali pomoc pri vypracovaní niektorých úloh, často robili chyby aj opakovane tie isté. Ak im bola ponúknutá zo strany učiteľa aktívna pomoc, tak úlohy zvládli, avšak nepochopili nadväznosti medzi úlohami a novým vedomostiam porozumeli len povrhu. Ôsmaci zas k aktivite pristupovali ľahostajne a považovali viaceré úlohy za triviálne, pre nich nedostačujúce. To však bolo ovplyvnené aj chýbajúcou motiváciou a lenivosťou žiakov ôsmeho ročníka, pretože v niektorých úlohách sa aj oni dopúšťali chýb a pýtali si pomoc učiteľa. Teda ich posúdenie náročnosti bolo neadekvátne. Stredoškólači sa už na úlohy pozerali z iného pohľadu, keďže po polroku na strednej škole už mali základné skúsenosti s programovaním a tiež aj s programovaním robotickej hračky Micro:bit. Niektorí sa dokonca s programom iniciatívne hrali, vytvárali optimálnejšie riešenia, či skúšali rôzne sekvencie príkazov.

V tomto bode by sme spomenuli aj najčastejšie chyby a miskoncepce, ktorých sa žiaci počas výučby s robotickou hračkou dopúšťali:

4.1.1 Nepochopenie zadania

Žiaci sa opakovane pýtali na postup práce, alebo opakovane im muselo byť vysvetlené zadanie. Prevažne išlo o žiakov 5. a 6. Ročníka. Príklad rozhovoru:

Žiak: "Naprogramoval som prvý program, spustili sme ho, čo teraz?"

Učiteľ: "Skúste teraz spoločne napísať do tabuľky, čo si myslíte, že jednotlivé príkazy v tabuľke znamenajú."

Žiaci: "No dobre."

4.1.2 Spustenie viacerých rôznych programov naraz

Žiaci mali za úlohu postupne vyskúšať viaceré programy a napísať ako sa ich model bude správať v jednotlivých prípadoch. Niektorí žiaci naprogramovali a spustili všetky programy naraz, čím vytvorili paralelný proces a nevedeli správne odčítať správanie modelu. Alebo častou chybou u piatakov a šiestakov bolo, že skúšali spustiť iba osamotené príkazy, ktoré neposkladali do programu.

4.1.3 Rozdiel medzi príkazmi pohybu

Mladší žiaci nevedeli dostatočne rozlíšiť príkaz pohybu do nekonečna a na konečný počet otáčok. Ich pozornosť bola zameraná len na to, že sa motor točí a detailnejšie nepozorovali vlastnosti tohto pohybu. Práve táto chyba ukazuje, že žiaci piateho a šiesteho ročníka ZŠ nevnímajú programovanie dostatočne komplexne a nedokážu ešte poriadne uchopiť tento koncept vlastným bádáním.

4.1.4 Rýchlosť otáčania motora

Rýchlosť otáčania motora je veľmi problematický koncept tejto robotickej stavebnice. Prvou chybou, s ktorou sme sa u žiakov stretli bolo nesprávne určenie najmenej a najväčšej rýchlosti. Žiaci základnej školy častokrát označili za najmenšiu rýchlosť 0%, kedy sa motor ani nespustí. Za najväčšiu zas označovali veľké čísla nad 100% a nespozorovali, že 100% je maximum a pri každom väčšom čísle sa motor točí rovnakou rýchlosťou.

4.1.5 Poradie príkazov

Ďalšou často vyskytovanou chybou bolo nesprávne poradie príkazov v programe. Napríklad aj poradie príkazov na rozbehnutie motora. Žiaci zadávali príkaz nastavenia rýchlosti až po príkaze na spustenie motora. Problémom však bolo, že v niektorých prípadoch to v takomto poradí fungovalo, čo si nevieme vysvetliť. (zvažujeme možnosť, že išlo o chybu v softvéri)

Mnoho ďalších chýb pramenilo z nepozornosti žiakov a z časového stresu. U mladších žiakov sme spozorovali labilnejšiu pozornosť, rýchlo sa nechali rozrušiť a potom sa ťažko vedeli znova naplno sústrediť. Starší žiaci (8. ročník ZŠ a 1. ročník SŠ) už dokázali udržať svoju pozornosť pri práci dlhší čas a sústrediť sa na riešenia problémov. Aj z tohto dôvodu robili menej chýb.

4.2 Konštrukcia modelu

Táto kategória definuje zložitost' konštrukcie. Pozorovali sme, či žiaci skladajú daný model intuitívne/ľahko, alebo či sa často pozerajú do návodu, alebo či sa pri konštrukcii mylia a musia konštrukciu rozobrať a následne sa vrátiť pár krokov späť. Rovnako aj čas, ktorý potrebovali na postavenie modelu. Samozrejme sme prihliadali aj na správnosť konštrukcie.

Skonštruovanie modelu trvalo žiakom pomerne dlho, hlavne v nižších ročníkoch. Konštrukcia modelu bola častokrát na úkor programovania, rozvíjania algoritmického myslenia a riešenia problémov. Aj keď sme sa snažili vybrať nenáročnejšie modely, žiaci v nižších ročníkoch ZŠ skladali model pomalšie (13-20 minút), viackrát sa pozerali na predchádzajúce kroky a stávalo sa, že poskladali časť konštrukcie zle, prípadne zrkadlovo. Z tohto dôvodu sme sa rozhodli dať do skupiny žiakov ZŠ 2 tablety, aby mohli súčasne konštruovať model robota, ktorý sa skladal z 2 častí. Stredoškólači zvládli konštrukciu modelu lepšie a za rovnaký čas, a to s použitím iba 1 tablety v skupine, čo výrazne spomalilo proces skladania modelu. Napriek očakávanej jednoduchosti zostavenia konštrukcie, sa žiaci SŠ tiež potýkali s rôznymi komplikáciami v procese skladania modelu.

4.3 Motivácia - atraktivita

Pod týmto bodom myslíme zánietenosť žiakov do vyučovacieho procesu s robotickou stavebnicou Lego Spike Prime. Pozorovali sme, či nadšenie žiakov bolo len chvíľkové, alebo trvalo počas celej hodiny. Sledovali sme zmeny nálad a aktívnu prácu žiakov počas výučby. A zároveň, či prejavovali záujem učiť sa a nadšenie z ďalších hodín.

Zánietenosť žiakov k vyučovaniu s robotickou stavebnicou LEGO Spike Prime bolo závislé od vekovej kategórie a predošlých vedomostiach a schopnostiach programovať. Žiaci piateho a šiesteho ročníka prejavili najväčší záujem, kvôli možnosti hrania sa s Legom. Avšak pri programovaní stavebnice ich zánietenosť mierne upadala, keďže preberané koncepty boli pre nich náročnejšie. Žiaci siedmeho ročníka pútavo pracovali počas celej vyučovacej hodiny. Ich záujem o nové vedomosti z oblasti robotiky bol viditeľný ako aj pri konštrukcii modelu tak aj pri programovaní stavebnice, keďže doň vkladali aj vlastnú kreativitu. Žiaci v priebehu vyučovania a aj po jeho skončení prejavovali záujem o ďalšie nové vedomosti. Kvôli veku a vyspelosti žiakov strednej školy sme spozorovali iný prístup. Nešlo už o nadšenie zo skladania Lega ako aj zo zadaní úloh, ale skôr o záujem hlbšie pochopiť súvislosti konceptov a možnostiam tejto robotickéj stavebnice, prípadne sa pretekať so spolužiakmi.

4.4 Kreativita

Neodlúčiteľnou zložkou konštruovania a programovania robotickéj stavebnice je určite aj kreativita. Pozorovali sme, či žiaci do svojich konštrukcií modelov, alebo programov vkladajú aj niečo vlastné. Kreativita rozvíja ďalšie rôzne schopnosti ako plánovanie, kritické myslenie, sebakritiku... a jednoznačne motivuje žiakov k lepším výkonom [15].

Od piateho ročníka ZŠ až po prvý ročník SŠ sa žiaci hrali pomerne rovnako s konštrukciou modelu a vylepšovali ju podľa vlastnej fantázie. Pridávali rôzne farebné dieliky, dotvárali ruky, nohy, oblečenie tanečníka. Niektorí žiaci preukázali svoju kreativitu aj pri programovaní. Využívali rôzne sekvencie príkazov na rozbehnutie rúk a nôh tanečníka. Svoju pozornosť venovali aj hudbe, ktorú si vyberali podľa vlastného vkusu. Dievčatá v 7. ročníku dokonca naprogramovali tanečníka tak, aby sa hýbal presne do rytmu hudby a skončil, keď skončí hudba. Ďalej žiaci samostatne objavovali svetelnú maticu a rozsvetľovali LED svetielka, alebo si nechali vypísať slová na svetelnej matici. Vlastnou kreativitou sa motivovali k ďalším ešte lepším výkonom.

4.5 Bádanie

V tejto oblasti sa zameriavame na mieru samostatného objavovania nových poznatkov, zručností a následného experimentovania s nimi. Pozorovali sme napríklad, či žiaci hľadaním a skúšaním samostatne prídu na to, ako zapnúť základnú jednotku, ako spustiť program, teda celkové ovládanie robotickéj stavebnice. V procese výučby s robotickou stavebnicou učíme žiakov budovať túto schopnosť objavovania a skúšania.

Žiaci 2. stupňa ZŠ nemali ešte veľmi vybudovanú schopnosť overovania a praktického skúšania programom. Niektoré úlohy z pracovného listu vypracovali iba po krátkom zamyslení bez toho, aby si to priamo overili a vyskúšali. Hlavne v piatom a šiestom ročníku sa častokrát žiaci pýtali "Ako to máme spraviť?" bez toho, aby to najskôr vyskúšali sami. Opakom boli žiaci strednej školy, ktorí s každou úlohou experimentovali, bádali a prakticky overovali svoje odpovede. S bádanim je spojený aj aha-efekt, čiže gestá, výkriky, emócie, ktoré žiaci prejavovali pri pochopení a objavení novej informácie. Tento aha-efekt bol najprítomnejší v siedmom ročníku. Starší žiaci už svoje emócie nedávali až tak hlasno najavo.

4.6 Záverečné zhrnutie

Po spracovaní všetkých dát nám vyšla ako najpodstatnejší aspekt hodnotenia primeranosti *Časová náročnosť*. Avšak po hlbšej analýze sme dospeli k záveru, že táto časová náročnosť je vlastne prirodzeným dôsledkom všetkých predošlých kritérií. Podľa toho, ako žiaci v ročníkoch dokázali v identifikovaných oblastiach úspešne pracovať, resp. mali rozvinuté vyššie menované zručnosti, by sme mali vedieť odhadnúť celkovú náročnosť LEGO SPIKE Prime. Teda celkovú časovú náročnosť celej aktivity od úvodného predstavenia aktivity, cez konštrukcie modelu, vyplňovanie pracovného listu až po programovanie jednotlivých úloh, záverečnú diskusiu a vyhodnotenia aktivity.

Obmedzenie, ktoré je dané nezávisle od robotickéj stavebnice, je vymedzenie času pre 1 vyučovaciu hodinu. To znamená, že optimálny čas celej aktivity by mal trvať 45 minút. Avšak prvé hodiny testovania ukázali, že je potrebné na vypracovanie našej prvej aktivity až dve vyučovacie hodiny, preto sme sa rozhodli v procese overovania v tejto nulte fáze výskumu aktivitu rozdeliť na dve samostatné aktivity - úvodná hodina na zoznámenie sa s robotickou stavebnicou, softvérovým prostredím a následne naväzujúca aktivita zameraná na pohyb motorov. Aj napriek

tomuto rozdeleniu piatáci nestihli vypracovať všetky úlohy z pracovného listu, keďže veľa času venovali práve konštrukcii modelu. Žiaci strednej školy boli šikovnejší a celú prvú aktivitu stihli za 60 až 70 minút, čo by nepokrylo celé 2 vyučovacie hodiny, bolo by potrebné im zadať ďalšie úlohy na zvyšný čas. Najoptimálnejšie pracovali siedmáci, ktorí stihli všetky povinné úlohy a niektoré skupinky vypracovali aj pár z rozširujúcich úloh.

Napriek tomu, že išlo o nami vytvorenú prvú verziu pracovných listov, zameraných na programovanie robotičkej hračky, ktoré sme využívali na hodinách so žiakmi, nepotrebovali do ďalšej fázy overovania ďalšie zásadné úpravy, čo do obsahu a formulácie zadání. Drobné zmeny boli síce spravené, ale jediná zásadná zmena do ďalšej fázy bude vyhradenie dlhšieho času na vypracovanie úloh. A teda rozdelenie aktivít do viacerých vyučovacích hodín.

Ako slabé miesto nášho výskumu považujeme nedostatok času. V škole, kde bolo testovanie realizované, sme mali k dispozícii dve vyučovacie hodiny, kde sme plánovali otestovať dve aktivity. Avšak kvôli vyššej časovej náročnosti aktivít, sme na základnej škole stihli otestovať iba prvú aktivitu. So žiakmi strednej školy sme potom stihli otestovať obe aktivity.

5 Záver

V tomto článku opisujeme výsledky nulte fázy výskumu vývojom, ktorý má naplánované v budúcnosti ešte ďalšie etapy overovania. V tejto etape sme preskúmali úroveň kognitívnych a psychomotorických vedomostí a schopností u priemerných žiakov druhého stupňa ZŠ a 1. ročníka SOŠ. Medzičasom sme už realizovali aj ďalšie etapy testovania a to so žiakmi nižších ročníkov 8-ročného gymnázia, ktoré zodpovedajú veku žiakov druhého stupňa ZŠ. Ide o vek, ktorý podľa našej nulte fázy vyzerá ako primeraný pre použitie tejto konkrétnej stavebnice na vyučovanie programátorských konceptov. V blízkej dobe plánujeme analyzovať údaje aj z tohto overovania, kde sme už testovali upravené materiály – pracovné listy a zameriavali sme na do akej hĺbky vieme s takto starými žiakmi preberať programátorské koncepty.

Podľa našich výsledkov z kvalitatívnej analýzy a nami vytvorenej škály, robotická stavebnica LEGO Spike Prime nie je úplne vhodná do piateho a šiesteho ročníka ZŠ. Odporučili by sme ju zaradiť až od siedmeho ročníka ZŠ, kde žiaci ešte stále prejavujú pomerne veľký záujem o prácu s LEGO stavebnicou, sú kreatívni, komunikatívni a prejavujú záujem o nové vedomosti. Stavebnica je vhodná aj pre 1. ročník SŠ, avšak bolo by potrebné aktivity a jednotlivé úlohy posunúť na vyššiu kognitívnu úroveň.

POĎAKOVANIE

Tento výskum bol podporený z projektu VEGA 1/0621/22 ako aj z projektu APVV-20-0353 a FERTILE 2021-1-EL01-KA220-HED-000023361.

BIBLIOGRAFICKÉ ODKAZY

- [1] RUBINACCI, F., PONTICORVO, M., PASSARIELLO, R., MIGLINO, O. Robotics for soft skills training. *In Research on Education and Media*, 9(2), p.20-25.S
- [2] HATZIGIANNI, M., STEVENSON, M., FALLOON, G., BOWER, M., FORBES, A. Young children's design thinking skills in makerspaces. *In International Journal of Child-Computer Interaction*, 27, p.100216. 2021.
- [3] HRUŠECKÁ, A., MIKOVÁ, K. Identifikácia gradácie infromatických konštruktov pri vyučovaní edukačnej robotiky na 1. stupni ZŠ. *In DidInfo 2021* [elektronický

dokument]. Banská Bystrica: Univerzita Mateja Bela, 2021. s. 118-122. ISBN 978-80-557-1823-1.

- [4] MIKOVÁ, K., FEJKOVÁ, V. A comparison of two (not well known) educational robotic kits for teaching computational thinking. *In ICERI2022 Proceedings*. Seville: IATED, 2022, p. 7830-7837. ISBN: 978-84-09-45476-1
- [5] MIKOVÁ, K., BUDINSKÁ, L., STENOVÁ, B.: Analýza edukačných robotických hračiek dostupných na Slovensku. *In DidInfo 2021* [elektronický dokument]. Banská Bystrica: Univerzita Mateja Bela, 2021. s. 118-122. ISBN 978-80-557-1823-1.
- [6] LEGO SPIKE Prime. <https://www.lego.com/en-sk/product/lego-education-spike-prime-set-45678>
- [7] LEGO Mindstorms EV3. <https://stavebnice-lego.heureka.sk/lego-mindstorms-31313-ev3/#prehlad/>
- [8] LEGO Education WeDo 2.0 Core set. <https://education.lego.com/en-us/products/lego-education-wedo-2-0-core-set/45300#wedo-20>
- [9] VAN DEN AKKER, J., GRAVEMEIJER, K., MCKENNEY, S., NIEVEEN, N. (Eds.). *Educational design research*. Routledge, 2006. p. 176. ISBN 9780415396356.
- [10] ŠTÁTNY PEDAGOGICKÝ ÚSTAV. *Inovovaný Štátny vzdelávací program*. <https://www.statpedu.sk/sk/svp/inovovany-statny-vzdelavaci-program/>
- [11] JANČARIKOVÁ, K. et al. *Didaktické zásady v prirodovdne vzdelávaní*. CHARLES UNIVERSITY, FACUL, Pedagogická fakulta UK 2022., 156 s. ISBN 9788076032897.
- [12] ZORMANOVÁ, L. *Obecná didaktika. Pro studium a praxi*, 1. Praha: Grada, 2014. 240 s. ISBN 978-80-247-4590-9.
- [13] ŠVAŘÍČEK, R., ŠEĎOVÁ, K. et al. *Kvalitatívny výskum v pedagogických viedach*. Praha: Portál, 2014. 384 s. ISBN 978-80-262-0644-6.
- [14] ŠEĎOVÁ, K. et al. *Výuková komunikace*. Brno: Masarykova univerzita, 2019. 176 s. ISBN 9788021095298
- [15] MIKULÁŠTÍK, M. *Tvořivost a inovace v práci manažera*, Praha : Grada, 2010, 207 s. ISBN 978-80-247-2016-6

Analyza kurzu programovania pre matematikov

Analysis of a programming course for mathematicians

Lubomír Salanci
Univerzita Komenského
Mlynská dolina
842 48 Bratislava
Slovenská republika
salanci@fmph.uniba.sk

ABSTRACT

In this paper, we analyse a programming course for students of mathematics in the 1st year at university. We focus on topics, programming languages, development environments, graphics libraries, and student assessment from a longitudinal perspective. Since there were several important changes during the course's existence, we also analyse the impact of these interventions on student success and their grades.

Keywords

Programming. Education. Analysis. Examination. Evaluation.

ABSTRAKT

V tomto článku analyzujeme kurz programovania pre prvákov vysokoškolského odboru matematika. Zameriavame sa na preberané témy, programovacie jazyky, vývojové prostredia, grafické knižnice a hodnotenie študentov z dlhodobého hľadiska. Keďže počas existencie kurzu nastalo niekoľko dôležitých zmien, analyzujeme aj vplyv týchto intervencií na úspešnosť aj výsledné hodnotenie študentov.

Kľúčové slová

Programovanie. Vzdelávanie. Analýza. Skúšanie. Hodnotenie.

1 ÚVOD

Analýzou univerzitných kurzov programovania sa zaoberajú viaceré články [1], [2]. V našom príspevku sa zameriavame kurz programovania pre prvákov matematického študijného odboru. Študenti Matematiky, Ekonomickej a finančnej matematiky, Manažárskej matematiky a Poistnej matematiky majú náš kurz odporúčaný v zimnom semestri prvého ročníka. Počas 25 rokov sme získali množstvo skúseností a údajov. Najzaujímavejšie z nich analyzujeme a vyhodnocujeme.

2 CHARAKTERISTIKA

V priebehu rokov sa výrazne menili technológie, možnosti fakulty aj požiadavky garantov matematických študijných programov. Preto sme počas existencie kurzu používali viaceré programovacie jazyky a vývojové prostredia:

- v roku 1998 jazyk C++ a prostredie Borland C,
- od roku 1999 prostredie C++Builder,
- od roku 2014 jazyk C# a prostredie SharpDevelop,
- od roku 2022 jazyk Python a IDLE.

Keďže je kurz určený pre neinformatické študijné odbory, nemali by nás prekvapiť informatické schopnosti študentov. V minulosti, na prelome milénia, k nám zväčša prichádzali študenti, ktorí nemali žiadne alebo iba veľmi malé skúsenosti s prácou s počítačom, nevedeli písať na klávesnici alebo používať myš. Takýto stav už nepozorujeme, momentálne je informatická gramotnosť študentov je na primeranej úrovni, a preto nás viac zaujímajú ich programátorské skúsenosti zo strednej školy.

V neformálnej ankete pomerne stabilne zisťujeme, že približne 50 % študentov o sebe tvrdí, že nikdy neprogramovali. Preto máme kurz nastavený tak, aby ho úspešne zvládli aj začiatočníci.

Pri uvažovaní o programovacích jazykoch a prostrediach sme museli zohľadňovať kritériá, ktoré boli stanovené garantmi. Napríklad, v roku 1998 bola požiadavka, aby bol kurz realizovaný v programovacom jazyku C alebo C++. Jazyk C# sme garantom študijných odborov sami navrhli, pretože s rastúcim počtom študentov a počítačov na fakulte nastal problém s licenciami pre C++Builder. Zo strany garantov vznikla v školskom roku 2022 požiadavka, aby sa kurz realizoval v jazyku Python. Skúsenosti z používania jednotlivých prostredí analyzujeme v kapitole 3.

Zmeny v programovacích jazykoch a prostrediach prinášali aj výzvy, keďže sme pri každej takejto zmene vytvárali kurz prakticky od začiatku. Samozrejme, pokým to bolo možné, využívali sme skúsenosti z minulých rokov. Po každej veľkej zmene nasledovalo obdobie jedného až dvoch rokov drobných úprav – napríklad, podľa spätnej väzby od kolegyň alebo kolegov, ktorí realizovali cvičenia, sme upravovali úlohy. Ak boli niektoré náročne, rozdelili sme ich na menšie časti. Ak sa ukázalo, že iné úlohy by pomohli k lepšiemu zvládnutiu skúšky, doplnili sme ich. A naopak, ak sa niektoré úlohy na cvičeniach ukázali ako zbytočný dril, zrušili sme ich. Ďalej nasledovalo niekoľkoročné obdobie, počas ktorého zostal kurz stabilný.

S tým, ako sa menil kurz a podmienky na jeho realizáciu, menili sa aj kritéria hodnotenia študentov. V kapitole 4 podrobnejšie analyzujeme vývoj tém a v kapitole 5 hodnotenie študentov.

3 ANALÝZA JAZYKOV A PROSTREDÍ

Pri každej zmene programovacieho jazyka sme vždy museli zvažovať a prehodnocovať nasledovné 3 zložky:

- programovací jazyk – ten býval približne daný, ale existujú rôzne dialekty jazykov,
- vývojové prostredie – pre jeden programovací jazyk zväčša existuje veľa prostredí,
- knižnice – môžu, ale nemusia byť zviazané s jazykom alebo vývojovým prostredím.

Pri prechode z jazyka C++ na C# sme zvažovali viaceré jazyky, ktoré sa podobajú na C: C#, Java alebo JavaScript. Rôzne jazyky majú rôzne výhody a nevýhody, pričom tie inak vníma profesionál a inak študent – začiatočník.

Pri voľbe programovacieho prostredia sme okrem SharpDevelop zvažovali aj rôzne iné varianty, ako napríklad: textový editor a kompiláciu z príkazového riadka, QtCreator, NetBeans, Visual Studio, MonoDevelop, PyCharm, Processing a ďalšie.

Asi najzaujímavejšia je problematika knižníc. Tie sa zároveň ukazujú ako najdôležitejšie pre dizajn kurzu. K programovacím jazykom existujú štandardné knižnice, napríklad pre prácu so súborami, textovým vstupom a výstupom alebo matematickými

funkciami. Pokým však potrebujeme pracovať s grafikou alebo vytvárať grafické používateľské prostredie, je situácia veľmi komplikovaná. Tvorcovia niektorých kompilátorov alebo vývojových prostredí dodávajú vlastné grafické knižnice. V iných prípadoch je na programátorovi, aké knižnice použije: QT, GTK, WinAPI GDI, Mono, a ďalšie.

Vzniká tak veľký kontrast medzi tým, ako jednoducho sa vytvárajú aplikácie pre textový / konzolový režim, a ako zložito sa pracuje s grafikou. Pritom vizualizácia výpočtov a výsledkov je pre začiatočníkov veľmi dôležitá. Na nasledujúcich ukážkach s kreslením červeného štvorca môžeme porovnať zápisy v jednotlivých jazykoch a filozofiu práce s grafikou.

Kreslenie pomocou štandardných knižníc WinAPI a GDI má s inicializáciou až 80 riadkov. Výsledok je teda neúmerne dlhý, takže ho celý ani neuvádzame. Avšak aj samotné kreslenie vyžaduje dodržanie viacerých náročných pravidiel:

```
HBRUSH red = CreateSolidBrush( RGB(255,0,0) );
...
case WM_PAINT:
    RECT rect;
    HDC dc = GetDC(hwnd);
    SetRect(&rect, 0, 0, 200, 100);
    FillRect(dc, &rect, red);
    ReleaseDC(hwnd, dc);
```

V prostredí C++Builder, ktoré používa vlastné knižnice VCL, stačí napísať iba dva príkazy:

```
Canvas->Brush->Color = clRed;
Canvas->Rectangle(0, 0, 200, 100);
```

Aj v prostredí SharpDevelop v jazyku C# stačia dva riadky, ktoré sa dajú zlúčiť do jedného (bez použitia premennej g):

```
Graphics g = CreateGraphics();
g.FillRectangle(Brushes.Red, 0, 0, 200, 100);
```

V jazyku Python a prostredí IDLE potrebujeme vytvoriť grafické okno, takže inicializácia zaberie 3 riadky. Samotné kreslenie červeného štvorca má iba jeden riadok:

```
from tkinter import *
c = Canvas(width = 500, height = 300)
c.pack()
c.create_rectangle(0, 0, 200, 100, fill = 'red')
```

Znamená to, že pokým je výpis pozdravu typu "Hello, World!" do konzoly jednoduchý, zobrazenie takéhoto pozdravu v grafickom režime môže byť v jednotlivých knižniciach rôzne komplikované – týka sa to nielen dĺžky programu, ale aj náročnosti konceptov. Preto považujeme rozmýšľanie o knižniciach, ktoré sú nie raz zviazané so samotným jazykom, prípadne aj vývojovým prostredím, za veľmi dôležité.

4 ANALÝZA TÉM

Pri voľbe tém zohráva značnú úlohu premyslené dávkovanie nových konceptov. Prvá postupnosť tém vychádzala z vlastných skúseností a zo skúseností kolegov, ktorí už dlhé roky učia programovanie. Neskôr, keď už sme lepšie rozumeli problémom našich študentov, upravovali sme aj témy, ktorým sa venujeme.

Pre porovnanie uvádzame zoznam tém, keď sme učiť začínali v C++:

1. vstup, výstup, premenné, príkaz priradenia (int, cin >>, cout <<, + - * /),
2. podmienený príkaz (if ... else, !, &&, ||, switch),
3. príkazy cyklu (while, do ... while, for),
4. funkcie bez návratovej hodnoty (parametre, viditeľnosť premenných, globálne premenné),

5. funkcie s návratovou hodnotou (volanie referenciou, ostatné typy, typ char),
6. polia, reťazce (pole, viacrozmerné polia, nulou ukončený reťazec, pole ako parameter funkcie),
7. textová obrazovka (farby, animácie, klávesnica)
8. štruktúry (typedef, struct),
9. video pamäť (bitové pole, mimoobrazovkové animácie),
10. textové súbory (zápis, čítanie - ifstream, ofstream),
11. funkcie s poľami (pretypovanie, porovnávanie, hľadanie a nahrádzanie reťazcov, o skúške).

S prechodom z prostredia MS-DOS do MS Windows sme viaceré témy zrušili, pretože sa stali neaktuálnymi. Príkladom môže byť práca s textovou obrazovkou alebo priamym prístupom do video pamäti. Hoci boli témy pre začiatočníkov užitočné, napríklad, kvôli vizualizácii, v grafickom používateľskom prostredí MS Windows existujú iné prostriedky na vizualizáciu, a navyše, priamy prístup do video-pamäte už nebol jednoduchý a prakticky možný, čím uvedené témy stratili zmysel.

Preto aj zoznam tém, keď sme končili s jazykom C#, vyzeral inak:

1. prvý program (postupnosť príkazov, kreslenie a súradnice: MessageBox, FillRectangle),
2. výrazy a premenné (číselné typy, konverzie, vstup: int, double, Convert.To..., komponent Edit),
3. cyklus (cyklus for, Random, premenná cyklu),
4. vetvenie programu (if ... else, !, &&, ||),
5. matematické úlohy (akumulácia výsledku, cyklus while, nepresnosť výpočtov a typu double),
6. podprogramy (metódy bez návratovej hodnoty, parametre, volanie metódy),
7. pole (jednorozmerné, vytvorené vymenovaním prvkov alebo pomocou new, indexy, prvky poľa),
8. myš (udalosti myši, súradnice),
9. dvojrozmerné pole (vymenované, vytvorené pomocou new, indexy, prvky poľa),
10. funkcie (metódy s návratovou hodnotou, return),
11. o skúške (priebeh, riešenie skúškového úlohy, ako postupovať, na čo si dať pozor).

Porovnaním predchádzajúcich dvoch zoznamov zistíme, že niektoré témy sme zrušili, zmenili, rozriedili alebo presunuli:

1. C++ → C# ako 1. a 2. téma,
2. C++ → C# až ako 4. téma,
3. C++ → C# 3., 5.,
4. C++ → C# 6.,
5. C++ → C# 10., znaky a referencie až v letnom semestri,
6. C++ → C# 7., 10., reťazce až v letnom semestri,
7. C++ → C# už nie,
8. C++ → C# až v letnom semestri ako triedy,
9. C++ → C# už nie,
10. C++ → C# až v letnom semestri,
11. C++ → C# reťazce až v lete, pretypovanie už nie.

Vidíme, že sme vykonali niekoľko dôležitých úprav:

- tému polia sme rozdelili na dve časti, pričom dvojrozmerné polia nasledujú až s určitým časovým odstupom,
- vlastné typy, štruktúrované premenné a prácu so súbormi sme presunuli až do letného semestra.

Textové súbory sú samé o sebe náročná téma, ktorá sa navyše preberala až ku koncu semestra, takže študenti nemali dostatok času na jej precvičenie. Presunom do letného semestra sme zabezpečili dostatočný priestor pre jej bezpečné a komfortné zvládnutie. Pri výmene jazyka C# za Python sa témy zachovali s tým, že v 7. téme už nehovoríme o poliach, ale o zoznamoch.

5 ANALÝZA HODNOTENIA

Aj pravidlá hodnotenia predmetu sa počas rokov menili. Niektoré zmeny boli nevyhnutné (napríklad, pri prechode na kreditné štúdium alebo súviseli s akreditáciou), iné nastávali z praktických dôvodov (napríklad, súviseli s počtom študentov, kapacitnými možnosťami). Zmeny v hodnotení sme realizovali postupne, a tak, ako sme s učením a hodnotením získavali skúsenosti.

Spočiatku bolo hodnotenie založené na tom, že študenti získavali body za rôznu aktivitu. Získané body sa nakoniec sčítali a podľa dosiahnutého súčtu bola pridelená známka.

Body bolo možné získať za:

- rozcvičky – riešenie krátkej úlohy na papieri na začiatku každého cvičenia,
- test – riešenie úloh na papieri na konci semestra,
- projekt – vytvorenie vlastného programu,
- skúšku – riešenie praktickej úlohy pri počítači.

Postupne sme však hodnotenie, ak nepočítame povinnú aktívnu účasť na cvičeniach, zredukovali iba na skúšku pri počítači. Viedlo nás k tomu niekoľko dôvodov.

V roku 2007 sme zrušili záverečný test. Cieľom testu bolo, aby sa študenti pred skúškou začali zamýšľať nad prípravou na skúšku. Kvôli veľkému nárastu počtu študentov okolo roku 2006 sme však už nedokázali zabezpečiť priestory, dozor, a teda ani jednotné testovanie všetkých študentov.

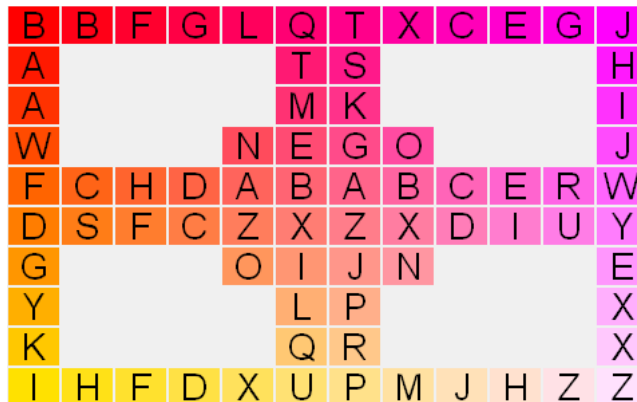
V roku 2016 sme zrušili rozcvičky. Cieľom rozcvičiek bolo, aby sa študenti počas semestra dostatočne venovali predmetu. Rozcvička sa písala na začiatku cvičenia. V rámci nej mali študenti počas 10 minút vymyslieť a napísať krátky program na tému, ktorá sa trénovala na predchádzajúcom cvičení. Postupne sa však začali ukazovať viaceré negatíva tohto prístupu, ktoré by sa dali jednoducho opísať tak, že študenti sa viac venovali rozcvičkám ako samotnému cvičeniu. Napríklad, na cvičenie neprišli pripravení z témy, ktorá sa mala aktuálne cvičiť, ale z predchádzajúcej témy, z ktorej sa písala rozcvička. Alebo sa zvyšok cvičenia venovali tomu, ako má vyzerať správne riešenie, či vzájomnému porovnávaní výsledkov z predchádzajúcej rozcvičky.

V roku 2022 sme zrušili projekt. Cieľom projektu bolo, aby sa študenti trénovali na skúšku. Medzičasom však začali prichádzať generácie, ktoré, čím ďalej, tým viac, podvádzali. Odovzdávanie projektu sa menilo na konfrontáciu toho, či študent vypracoval projekt sám alebo si ho dal naprogramovať niekým iným. Aby sa študentom neoplatilo podvádzať, znížili sme počet bodov za vypracovaný projekt. Ku koncu tak projekt riešilo iba zopár najšikovnejších študentov, čo boli práve takí, ktorí to nepotrebovali. Keďže projekt si svoj účel neplnil, zrušili sme ho.

Pri uvažovaní o praktickej skúške na počítači sme sa rozhodovali medzi tým, či budú študenti riešiť jednu komplexnú úlohu alebo viacero menších rôznych úloh. Nakoniec sme zvolili prvú možnosť.

Domnievame sa, že pri vhodne zvolenej úlohe dokážeme overiť viacero rôznych tém. Zároveň sa nazdávame, že riešenie jednej komplexnej úlohy je bližšie k praxi, teda k tomu, ako funguje reálne programovanie. Zistili sme, že pre našich študentov sú zaujímavé úlohy, ktoré vychádzajú z hlavolamov v štvorcovej sieti. Príkladom môže byť zjednodušená verzia hry Mahjongg, ako vidíme na Obrázku 1. V nej treba podľa určitých pravidiel odstraňovať dvojice kameňov s rovnakými písmenkami. Výsledný program umožní používateľovi zahrať si takúto hru, bude ho

strážiť, aby dodržiaval pravidlá hry (cieľom teda nie je vytvoriť program, ktorý takýto hlavolam vyrieši).



Obrázok 1: Ukážka úlohy zo skúšky.

Kvôli skúšaniam sme potrebovali vymyslieť veľa rôznych úloh tak, aby sa medzi jednotlivými termínmi skúšok zadania neopakovali, ale zároveň tak, aby boli úlohy porovnateľne náročné.

Preto majú všetky skúškové úlohy podobnú šablónu. Treba:

- prečítať textový súbor a podľa jeho obsahu inicializovať dvojrozmerné pole – po presune súborov do letného semestra sme túto etapu zo skúšky vypustili,
- vizualizovať obsah poľa – použiť cyklus v cykle, vypočítať súradnice a nakresliť hraciu plochu,
- spracovať vstup od používateľa – reagovanie na dve kliknutia myšou, vypočítať indexy prvkov poľa,
- aplikovať pravidlá hry – zmeniť obsah poľa, napríklad testovať okolie prvku, riadok, stĺpec,
- zobrazíť správu o skončení hry.

Zadania formulujeme tak, aby sme dokázali jednotlivé etapy riešenia vcelku presne bodovať, napríklad:

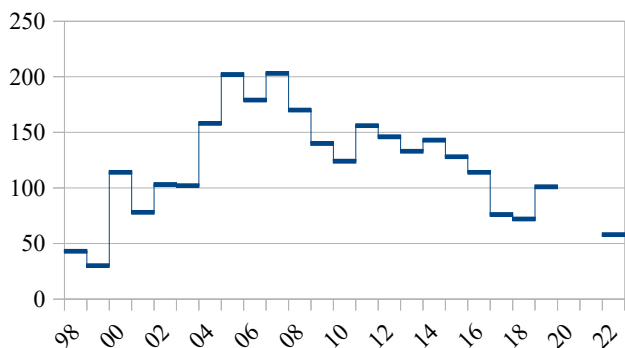
- 25 bodov získa študent za načítanie plochy,
- 25 bodov za kreslenie,
- 15 bodov za klikanie,
- 25 bodov za aplikovanie pravidiel hry,
- 10 bodov za zobrazenie konečnej správy.

Bodovanie za jednotlivé etapy sa pri rôznych úlohách mierne líši podľa toho, ako veľmi náročné na implementáciu sú pravidlá hry alebo test na koniec hry. Bodovanie etáp zvykneme rozdeliť ešte na jemnejšie časti, čím hodnotenie viac zjednotíme a spresníme.

Veľmi skoro sa však ukázalo, že bodovanie pre objektívne hodnotenie nestačí. Na začiatku sme totiž body pridelovali za kód, ktorý študent napísal. Teda, zohľadňovali sme aj to, čo svojim kódom chcel povedať. S nespokojnými študentami sme však nezriedka končili v konfrontácii typu: „prečo som dostal iba x bodov a spolužiačik / spolužiačka y .“ Ako objektívny prístup sa ukázala idea, ktorú sme prevzali z programovania riadeného testovaním (test-driven programming): body pridelujeme iba za tie časti programu, ktoré reálne fungujú a vieme ich testovať bez toho, aby sme v programe čokoľvek menili. Študentov zároveň učíme, že testovanie je pre nás akoby skúška správnosti. Pri riešení úlohy na skúške študentom nepomáhame, ani im neradíme s tým, že pre nás je veľmi dôležitou súčasťou skúšky a hodnotenia to, ako si poradia s vlastnými chybami. Ak sú študenti vopred na takýto prístup pripravení, dobre ho akceptujú.

6 VÝSLEDKY A ICH INTERPRETÁCIA

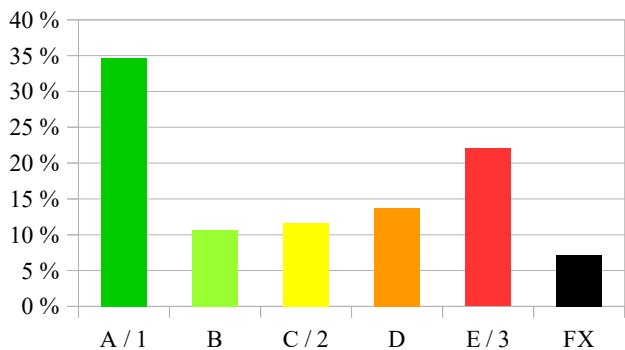
Počty študentov, ktorí absolvovali skúšku z programovania v jednotlivých rokoch, ukazuje graf na Obrázku 2. Ak študent skúšku opakoval, je započítaný iba raz. Študent, ktorý si náš predmet zapísal, ale nezúčastnil sa na skúške, nie je započítaný.



Obrázok 2: Počty študentov, ktorí absolvovali skúšku. Na osi x sú jednotlivé školské roky, na osi y počty študentov. V rokoch 2020 / 22 a 2021 / 22 skúška nebola.

Počet študentov kulminoval v rokoch 2005 / 06 až 2007 / 08, keď si predmet zapísalo okolo 250 študentov. V týchto rokoch sa skúšky zúčastnilo okolo 80% študentov, preto na grafe vidíme hodnoty mierne nad 200. Pomer, medzi tým, koľkí študenti si predmet zapísali, a koľkí z nich nakoniec skúšku absolvovali, sa z roka na rok mení. Napríklad, v školskom roku 2003 / 04 si predmet zapísalo 178 študentov, z nich bolo na skúške iba 102, teda iba 57%. Dôvodov, prečo sa študenti skúšky nezúčastňujú, býva viacero. Najčastejším je, že študenti v priebehu prvého semestra zanechávajú štúdium. Ďalším dôvodom býva, že sa chcú sústrediť na iné, pre nich dôležitejšie predmety. Skúšku z programovania potom absolvujú ako druháci.

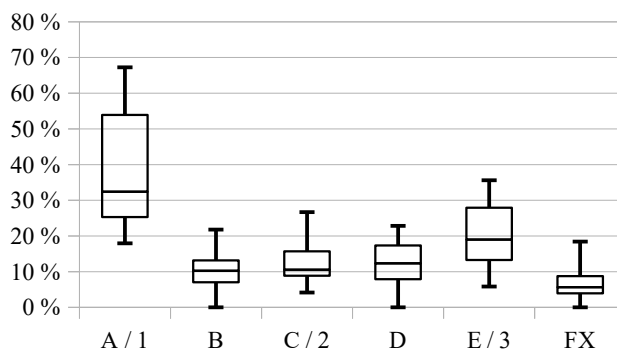
Pri tejto príležitosti je potrebné upozorniť na to, že v číslach, ktoré prezentuje Obrázok 2, ale aj ďalšie obrázky, sú zahrnutí aj iní študenti. Napríklad, v školskom roku 2003 / 04 bolo zo 178 študentov 162 prvákov. Ďalší 15 zapísaní študenti boli druháci, ktorí predmet opakovali, alebo si ho zapísali až ako druháci. Jedna študentka bola z iného študijného odboru a predmet si zapísala preto, lebo ju zaujal. Vo výsledkoch potom nerozlišujeme, koľkí študenti predmet opakujú, alebo si ho zapísali ako druháci. Študentov z iných odborov býva minimum (navyše, študentom informatiky, ktorí majú niekoľko svojich náročných kurzov programovania, nie je umožnené zapísať si náš kurz pre začiatočníkov). Preto sa domnievame, že nasledujúce výsledky nie sú týmito študentami výrazne skreslené.



Obrázok 3: Zastúpenie jednotlivých hodnotení. Os x zodpovedá známke, na osi y sú počty študentov v percentách.

Z Obrázka 3 získame predstavu o výslednom hodnotení študentov. Hodnotenie A získalo približne 35% študentov, hodnotenia B, C, D sú približne rovnomerne zastúpené – dosahuje ich okolo 10% študentov. Naopak, hodnotenie E je druhé najčastejšie, keďže ho získalo okolo 20% študentov. Hodnotenie FX získava najmenej študentov. Nazdávame sa, že takéto rozdelenie hodnotenia je dané kritériovým charakterom hodnotenia (naším cieľom nie je rozlíšiť rôzne úrovne schopností študentov).

Kvartilový graf na obrázku 4 ukazuje rozloženie a rozptyl hodnotení. Hodnotenie A dosahuje najväčší rozptyl, teda boli školské roky, keď takéto hodnotenie dosiahlo 30% študentov a boli školské roky, keď ho dosiahlo vyše 60% študentov. Je zaujímavé, že ostatné hodnotenia sú viac koncentrované okolo svojich mediánov, vrátane hodnotenie FX s najnižším mediánom necelých 6%.

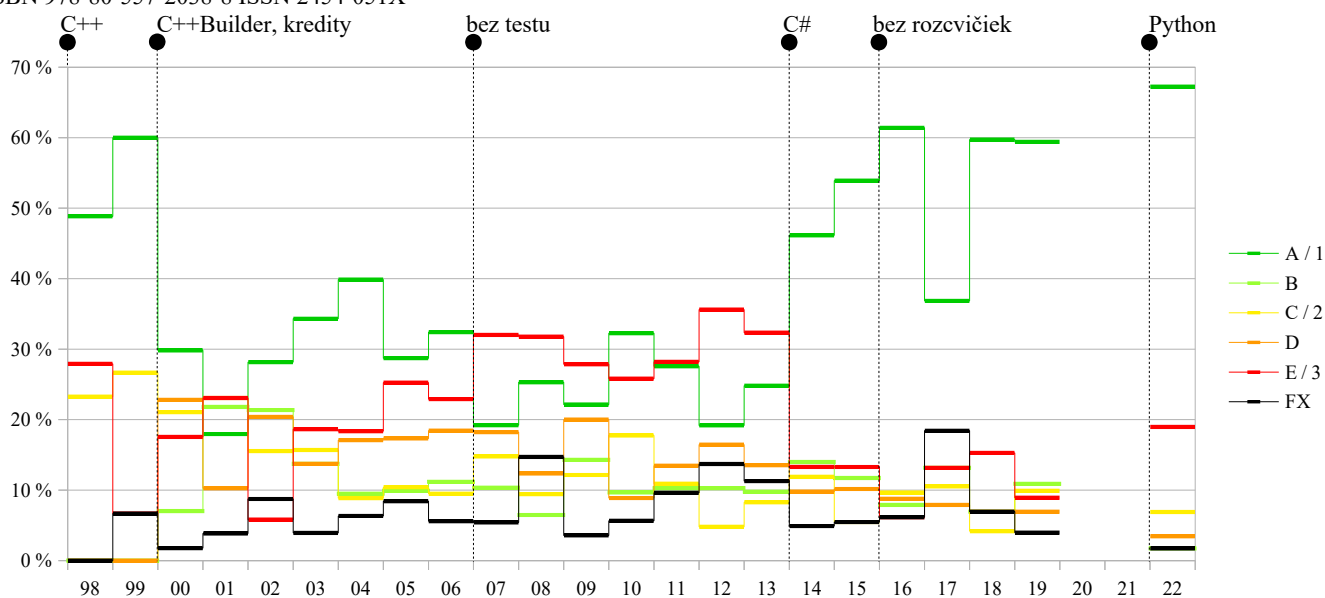


Obrázok 4: Hodnotenie študentov – detaily. Význam osí je rovnaký ako na Obrázku 3.

Vývoj hodnotenia od začiatku existencie predmetu až doposiaľ ukazuje graf na Obrázku 5. Jednotlivými farebnými líniami sú znázornené stupne hodnotenia. Zároveň sú v grafe zanesené aj vertikálne značky s dôležitými a nám známymi intervenciami.

Počas prvých dvoch školských rokov 1998 / 99 a 1999 / 20 sme používali C++ pre MS-DOS. V tomto období dosahovalo veľa študentov hodnotenie A, zároveň sme však v tomto období hľadali objektívny prístup k hodnoteniu. Od školského roku 2000 / 21 sa zaviedol kreditový systém. Zároveň sme začali používať prostredie C++Builder, čím sa výrazne zmenili typy problémov, ktoré študenti riešili. Navyše sme zaviedli exaktnejší prístup k hodnoteniu. S touto zmenou nastal pokles hodnotení A. Na tejto časti grafu až do školského roku 2013 / 14 si môžeme všimnúť, že počty hodnotení A a E sa rôzne prelínajú, ale hodnotenie E postupne, čím ďalej, tým viac, prevláda nad hodnotením A. Od školského roku 2014 / 15 sme vymenili programovací jazyk C++ za C#. Od tohto obdobia začalo výrazne prevládať hodnotenie A. Nazdávame sa však, že k tomu neprispela zmena programovacieho jazyka, ale presun niektorých problematických tém (najmä textových súborov) do letného semestra. Zdá sa, že zrušenie rozevičiek v roku 2016 / 17 nemalo na výsledné hodnotenie vplyv. V školskom roku 2022 / 23 nastala zmena programovacieho jazyka zo C# na Python. Z tohto obdobia máme zatiaľ málo údajov, ale hodnotenia sú podobné ako roky predtým.

Vráťme sa ešte do obdobia rokov 2000 / 21 až 2013 / 14 a skúsme posúdiť vplyv (ne)prítomnosti testu na hodnotenie. Hoci sa v priebehu rokov menili pravidlá hodnotenia, študenti mohli predmet úspešne absolvovať iba vtedy, ak na skúške získali približne 50 percent bodov. Znamená to, že môžeme porovnávať, ako sa v priebehu rokov vyvíjala úspešnosť študentov pri riešení skúšok.

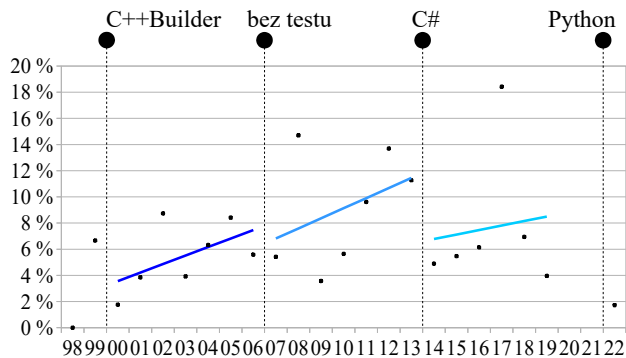


Obrázok 5: Výsledky skúšok v priebehu školských rokov. Na osi x sú jednotlivé školské roky, na osi y sú počty študentov v percentách, farebné línie zodpovedajú známke podľa legendy vpravo.

Zamerajme sa na neúspešnosť študentov, teda to, koľko percent študentov získalo v tomto období hodnotenie FX. Čierna línia na Obrázku 5 má v období rokov 2000 / 21 až 2013 / 14 komplikovaný priebeh. Pred zrušením testu získalo FX priemerne 5,88 % študentov, po zrušení testu priemerne až 9,14 % študentov. Jednoduché konštatovanie, že zrušenie testu spôsobilo nárast počtu neúspešných študentov nebude správne. Ak si dobre všimneme čiernu líniu, môžeme už v čase existencie testu pozorovať nárast hodnotení FX. Za tento trend sú zodpovedné vonkajšie okolnosti. Ak však tento trend pokračoval aj v ďalších rokoch, bude prirodzené, že priemerný počet študentov s hodnotením FX po zrušení testu vzrastie.

nadväzujú a majú v oboch obdobiach podobný sklon. Keďže kritériá pre úspešné absolvovanie skúšky zostali nezmenené, dá sa povedať, že prítomnosť testu na úspešnosť alebo neúspešnosť absolvovania predmetu nemala vplyv. Podobnou úvahou môžeme zistiť, že test mal malý vplyv na hodnotenie A a mierny vplyv na hodnotenie E.

Pre porovnanie, v období používania jazyka C# vychádza trend hodnotenia FX na 0,34 percentuálneho bodu za rok (bledomodrá úsečka). V tomto období sa však zmenili témy, a tým pádom aj skúškové úlohy, čo malo za následok aj znížený podiel hodnotení FX. Každopádne, aj v tomto období môžeme pozorovať nepredvídateľné výkyvy, napríklad v školskom roku 2017 / 18.



Obrázok 6: Lineárny fit hodnotenia FX v troch intervaloch: 2000 – 2006, 2007 – 2013 a 2014 – 2019. Význam osí je rovnaký ako na Obrázku 5.

Na grafe v Obrázku 6 sú čiernymi bodkami znázornené iba hodnotenia FX (y-ové osi majú v Obrázkoch 5 a 6 rôznu mierku). Úsečky modrej farby lineárne fitujú tieto hodnoty v jednotlivých obdobiach. Môžeme vidieť, že v období písania testu (tmavomodrá úsečka) aj po zrušení testu (prostredná modrá úsečka) malo hodnotenie FX rastúci trend so sklonom 0,65 a 0,77 percentuálneho bodu za rok a lineárnym vzťahom [3] 30% a 14% v jednotlivých obdobiach. Nízky lineárny vzťah je očakávaný vzhľadom na rozptýlenosť hodnotenia, ale pre naše účely nie je zaujímavý. Dôležitejšie je, že obidve úsečky na seba v podstate

ZÁVER

V našom kurze dlhodobo sledujeme úspešnosť študentov na skúškach a trend, akým sa hodnotenie vyvíja. V príspevku sme analyzovali výsledné hodnotenie a vplyv našich intervencií. Videli sme napríklad, že zrušenie testu sa na výslednom hodnotení neprejavilo. O tretinu však narástol počet študentov, ktorí boli na opravnej skúške, čo sme však už v príspevku neanalyzovali.

POĎAKOVANIE

Tento článok prezentuje časť výstupov naplánovaných a spolufinancovaných z grantu KEGA 010UK-4/2022.

BIBLIOGRAFICKÉ ODKAZY

- [1] MURPHY, E., CRICK, T., DAVENPORT, J. H., An Analysis of Introductory Programming Courses at UK Universities. In *The Art, Science, and Engineering of Programming*, 2017, vol. 1, no. 2, article 18. ISSN: 2473-7321.
- [2] FIGUEIREDO, J., GARCÍA-PENALVO, F. J., Teaching and Learning Tools for Introductory Programming in University Courses. In *Proceedings of the 2021 International Symposium on Computers in Education*, Spain. 2021. DOI: 10.1109/SIIIE53363.2021.9583623.
- [3] TITKO, M., NOVÁK, L., JÁNOŠÍKOVÁ, M., Praktická štatistika. EDIS UNIZA. 2021, ISBN 978-80-554-1814-8

Porozumenie zadaní úloh z programovania s grafickým výstupom nevidiacim študentom

Comprehension of programming assignments with graphical output by blind student

Mária Stankovičová
Centrum podpory študentov, UK
Šafárikovo námestie 6, P.O.BOX 440
814 99 Bratislava
Slovenská republika
maria.stankovicova@rec.uniba.sk

ABSTRACT

In our extended abstract, we analyse the comprehension of assignments in an introductory Python programming course. We focus on tasks with graphical output whose assignment is illustrated with an image. During the course, we have noticed that the interpretation of some assignments by blind students differs from the instructor's original idea. This was evident precisely in assignments where the required graphical output was represented by a picture in the assignment. This form of assignment proved to be inadequate for the blind student. We assume that the results of the analysis will help us to suggest a modification of the assignments.

Keywords

Programming. Blind student. Graphical output.

ABSTRAKT

V našom rozšírenom abstrakte analyzujeme porozumenie zadaní úloh nevidiacim študentom úvodného kurzu programovania v jazyku Python. Zameriame sa na úlohy s grafickým výstupom, ktorých zadanie je doplnené obrázkom. V priebehu kurzu sme si všimli, že interpretácia niektorých zadaní nevidiacim študentom sa líši od pôvodnej predstavy vyučujúceho. Prejavilo sa to práve v úlohách, kde požadovaný grafický výstup bol v zadaní znázornený obrázkom. Ukázalo sa, že pre nevidiaceho študenta nebola táto forma zadania postačujúca. Predpokladáme, že výsledky analýzy nám pomôžu navrhnúť úpravu zadaní.

Kľúčové slová

Programovanie. Nevidiaci študent. Grafický výstup.

1 ÚVOD

Zrakové postihnutie má vplyv na spôsob získavania informácií a vytváranie predstáv o okolitom svete. Špecifické potreby nevidiaceho študenta vyplývajú z neúplnosti až absencie zrakových vnemov. V edukačnom procese je dôležité využiť ostatné zmysly, najmä sluch a hmat [1].

Dôsledky zrakového postihnutia sú silne viazané aj na vek, v ktorom k postihnutiu došlo. Strata zraku v mladom veku umožňuje uchovať si vizuálnu spomienku a pomáha pri utváraní predstáv na základe minulých vizuálnych skúseností. Avšak pamäťové obrazy bez posilňovania slabnú. Preto je v edukačnom procese mimoriadne dôležité upozorňovať na podstatné vlastnosti a znaky jednotlivých predmetov a javov [2].

Zrakové postihnutie študentov by nemalo byť dôvodom na znižovanie nárokov a uplatňovanie neopodstatnených úľav. Nie je

dôvod na redukovanie objemu požadovaného učiva a znižovanie úrovne výučby. Potrebné je prispôsobiť metódy, formy, prostriedky a organizáciu práce.

Pri tvorbe študijných materiálov vo forme elektronického dokumentu, ktoré obsahujú obrázky, sa odporúča k obrázkom priradiť alternatívny text [3].

2 METODIKA

Pre analýzu porozumenia zadaní úloh nevidiacim študentom sme zvolili stratégiu kvalitatívneho výskumu – prípadovú štúdiu [4]. Výskumnú vzorku tvoril jeden nevidiaci študent, ktorý prišiel o zrak v detskom veku. Ako metódy zberu dát sme použili rozhovor (so študentom a vyučujúcim) a analýzu výstupov vo forme naprogramovaných úloh.

Kurz Programovanie (1) je určený pre študentov odboru matematika, na ktorom sa študenti oboznamujú so základnými konštrukciami a údajovými typmi programovacieho jazyka Python [5]. Vyučovacie formy kurzu sú prednášky a cvičenia. Počas cvičení študenti samostatne riešia sadu úloh s gradovanou náročnosťou. Svoje riešenia a prípadné problémy konzultujú s vyučujúcimi. Zadania úloh sú dostupné na webovej stránke predmetu. Niekoľko dní po cvičeniach sú zverejnené aj ukážky správnych riešení [5].

Úlohy na cvičeniach boli rôznych typov – tvorba programu s textovým výstupom, s grafickým výstupom, analýza výsledku zadaného kódu. Zadania pre tvorbu programov s grafickým výstupom boli doplnené obrázkami ilustrujúcimi výstup programu.

3 INTERPRETÁCIA VYBRANÝCH ÚLOH

Naše číslovanie úloh nezodpovedá originálnemu číslovaniu v študijných materiáloch a z priestorových dôvodov neuvádzame ani všetky skúmané zadania úloh. Pri niektorých úlohách uvádzame obrázky z výstupov programov od študenta a tiež obrázky znázorňujúce pôvodnú predstavu vyučujúceho.

3.1 Úloha 01

Vytvorte program, ktorý nakreslí 100 náhodne rozmiestnených hviezdčiek.

Študent uvažoval nad úlohou takto:

„Čo sa myslí hviezdčikou? V zadaní sa nespomína, že má študent kresliť textové hviezdčiky (cez create_text), teda ich skôr vypisovať než kresliť. Nevieť, či je toto na tom obrázku vidieť, že je to font a nie kresba, alebo či to má študentovi napadnúť, že by sa to takto dalo spraviť. Mne to nenapadlo, pôvodne som predpokladal, že je

reč o kreslených hviezdach napr. z niekoľkých zrotovaných štvorcov na sebe. Našťastie, nasledujúca úloha to vyjasnila.“

3.2 Úloha 02

Upravte predchádzajúci program tak, aby sa pred kreslením hviezdčiek nakreslil do celej grafickej plochy modrý obdĺžnik. Hviezdčičky budú biele a každá bude mať náhodne zvolenú veľkosť písma.

Študent uvažoval nad úlohou takto:

„Toto zadanie znie pomerne jednoznačne, obsahuje ale obrázok. Predpokladám, že iba na ilustráciu, ak ale vyjadruje nejaké ďalšie parametre, napríklad minimálnu a maximálnu veľkosť hviezdčiek, v mojom programe nebude reflektovaná.“

3.3 Úloha 03

Vytvorte program, ktorý nakreslí vlajku.

Na obrázku bola vlajka USA. Študent uvažoval nad úlohou takto:

„V zadaní nie je textovo uvedené, akú vlajku máme nakresliť, tá je vyobrazená na obrázku.“

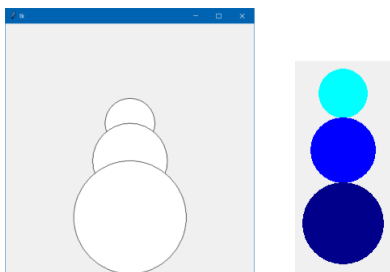
Študent sa rozhodol nakresliť slovenskú vlajku.

3.4 Úloha 04

Budeme kresliť veľa kruhov. Preto zadefinujte podprogram $kruh(x, y, r, farba)$, ktorý do grafickej plochy nakreslí kruh so stredom v bode x, y , polomerom r a zafarbený podľa parametra $farba$. Potom napríklad príkaz: $kruh(200, 100, 50, 'red')$, nakreslí červený kruh so stredom v bode 200, 100 a polomerom 50. Použite tento podprogram a nakreslite pomocou neho snehuliaka.

Študent uvažoval nad úlohou takto (Obrázok 1):

„Čo sa myslí pod ‘snehuliak’? Z koľkých gúl sa má skladat? Má mať ruky? Tvár? Metlu? Hrniec na hlave? A akým spôsobom má byť vyobrazený? Majú sa gule jednoducho dotýkať v jednom bode, alebo majú byť čiastočne vnorené do seba, ako keby boli na sebe fyzicky posadené?“

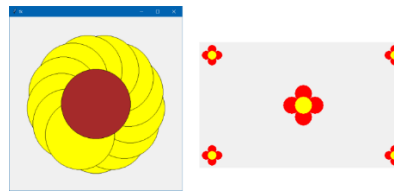


Obrázok 1: Snehuliak – vľavo je interpretácia študenta, vpravo vzor od vyučujúceho.

3.5 Úloha 05

Vytvorte podprogram $kvet(x, y, v)$, ktorý, pomocou volaní podprogramu $kruh$ z predchádzajúcej úlohy, nakreslí kvet: parametre x, y určujú pozíciu kvetu (napríklad, sú to súradnice stredu kvetu). Parameter v určuje veľkosť kvetu (napríklad, je to polomer kruhov, ktoré tvoria lupene a stred kvetu). Použite podprogram $kvet$ a vytvorte pohľadnicu.

Študentovi tiež nebolo jasné, ako presne má kvet vyzerat', ani mu nebolo zrejmé, ako má vyzerat' pohľadnica (Obrázok 2).



Obrázok 2: Kvet – vľavo je interpretácia študenta, vpravo vzor pohľadnice od vyučujúceho.

4 ANALÝZA

Pri analýze porozumenia zadani úloh nevidiacim študentom sme zistili, že zadania môžeme rozdeliť do troch základných typov.

- Zadania úloh, ktoré boli interpretované správne a tiež aj grafický výstup študenta sa zhodoval s pôvodnou predlohou.
- Zadania úloh, ktoré študent vyhodnotil ako jednoznačné, no grafický výstup sa nezhodoval s pôvodnou predlohou.
- Zadania, ktoré študent vyhodnotil ako nejednoznačné a grafický výstup sa ani nezhodoval s pôvodnou predlohou.

Pri podrobnejšom skúmaní úvah študenta sme hľadali spôsob, ako doplniť informácie k zadaniam označeným ako nejednoznačné. Zistili sme, že obrázky, ktoré plnia pomocnú úlohu pre intaktných študentov, nemajú priradený alternatívny text. V alternatívnom texte by mohol byť uvedený opis základných parametrov, ktoré má obrázok spĺňať (dotýkanie strán útvarov, umiestnenie na ploche, prekryvanie, farebná výplň...).

5 ZÁVER

Vďaka analýze porozumenia zadani úloh nevidiacim študentom sme získali hodnotnú spätnú väzbu. Úvahy a pripomienky nevidiaceho študenta nám pomôžu modifikovať zadania úloh priradením alternatívneho textu k obrázkom. Alternatívne texty budú obsahovať jednoznačné informácie o parametroch obrázka. V niektorých prípadoch zväzime doplnenie informácií priamo do zadani úloh.

POĎAKOVANIE

Naša vďaka patrí nevidiacemu študentovi, ktorý upozornil na nejednoznačnosť v zadaniach grafických úloh. Tiež za jeho ochotu a trpezlivé zaznamenanie svojich úvah. Ďakujeme aj vyučujúceho, ktorý nám poskytol texty a obrázky z učebného materiálu. Tento príspevok prezentuje časť výstupov naplánovaných a financovaných z grantu KEGA 010UK-4/2022.

BIBLIOGRAFICKÉ ODKAZY

- [1] LOPÚCHOVÁ, J. *Základy špeciálnej pedagogiky pre prácu so študentmi stredných a vysokých škôl*. Bratislava: PdF UK, 2007, 73-91. ISBN 978-80-89113-30-9
- [2] POŽÁR, L. *Psychológia detí a mládeže s poruchami zraku*. Trnava : Trnavská univerzita, 2000, ISBN 80-88774-74-8
- [3] MAMOJKA, B., REGEC, V., TEPLICKÝ, P. *Metodická príručka na vytváranie a používanie prístupných elektronických dokumentov*. UNSS 2006
- [4] ŠVAŘÍČEK, R., ŠEĎOVÁ, K. et al. 2007. *Kvalitatívny výzkum v pedagogických vedách*. Praha : Portál, strany ISBN 978-80-7367-313-0
- [5] Informačný list a stránka predmetu: https://sluzby.fmph.uniba.sk/infolist/sk/1-MAT-130_22.html
<https://edu.fmph.uniba.sk/~salanci/Python/index.html>

Dištančné vyučovanie prakticky orientovaných predmetov

Distance learning of practically oriented subjects

Veronika Stoffová

Trnavská univerzita v Trnave
Hornopotočná 23
918 43 TRNAVA
Slovensko
veronika.stoffova@truni.sk

ABSTRACT

This article informs about how teachers dealt with distance learning or combined teaching of practically oriented subjects. The author points to possible solutions, which she considers to be an adequate substitute for face-to-face teaching which preserve or raise the level and improve the quality of education. As an example, three thematic units or subjects were chosen, namely programming, construction and programming of robots and preparation of future teachers for pedagogical practice. The author mainly describes his experience in teaching using the MS Teams environment utilising virtual tools and simulation models. While the last topic concerns the higher education of future teachers, the first two topics are addressed universally with the possibility of their application at all levels of the education system.

Keywords

Distance Education. Remote programming. Virtual environments and tools. Simulation models.

ABSTRAKT

Tento článok informuje o tom, ako sa učitelia vysporiadali s dištančným vzdelávaním, prípadne kombinovaným vyučovaním prakticky zameraných predmetov. Autorka poukazuje na možné riešenia, ktoré považuje za adekvátnu náhradu za prezenčné vyučovanie, a ktoré zachovávajú, príp. zvyšujú úroveň a zlepšujú kvalitu vzdelávania. Ako príklad boli zvolené 3 tematické celky, príp. predmety a to programovanie, konštrukcia a programovanie robotov a príprava budúcich učiteľov na pedagogickú prax. Autorka opisuje hlavne svoje skúsenosti s výučbou v prostredí MS-Teams pomocou virtuálnych nástrojov a simulačných modelov. Kým posledná téma sa týka vysokoškolského vzdelávania, prvé dve témy sú riešené univerzálne s možnosťou ich aplikácie na všetkých stupňoch vzdelávacieho systému.

Kľúčové slová

Dištančné vzdelávanie. Programovanie dištančne. Virtuálne prostredia a nástroje. Simulačné modely.

1 ÚVOD

Predmet *programovanie* má špecifické postavenie v informatike na základných a stredných školách ako aj v študijných programoch vysokých škôl a univerzít. Dôležitú úlohu hrá hlavne v študijných programoch orientovaných na informatiku, a informačné a digitálne technológie. Programovanie patrí medzi kľúčové a zároveň náročné profilujúce predmety. Mnohí študenti nároky vyučujúcich a podmienky absolvovania predmetov programovania nezvládajú. Často je nesplnenie postačujúcich podmienok na najnižšiu klasifikáciu z programovania dôvodom predčasného ukončenia štúdia nejedného študenta informatiky. Dôležitosť predmetu *programovanie* a jeho vyučovania v príprave učiteľov

informatiky potvrdzuje aj skutočnosť, že didaktike programovania je venovaná zvýšená pozornosť aj v rámci prípravy učiteľov na vyučovaciu prax.

Didaktika programovania je často v rámci študijného programu samostatný vyučovací predmet, ktorý je orientovaný na teóriu a prax vyučovania programovania. V rámci pedagogickej praxe musia budúci učitelia informatiky vyučovať aj stanovený počet hodín z programovania. Takže nie je prístupné, aby budúci učiteľ informatiky ukončil vysokoškolské štúdium nevedel programovať a nebol schopný programovanie fundovane a adekvátne vyučovať. Zároveň musí byť schopný posúdiť vhodnosť a primeranosť použitého programovacieho prostredia (programovacieho jazyka) na vyučovanie na danom stupni vzdelávania ohľadom na mentálnu úroveň edukantov, rozvoja ich algoritmického a programátorského myslenia a splnenie predpokladov úspešného zvládnutia disciplíny a tematického celku [1, 2, 3, 4, 5]. Tradičná, prirodzená a najúčinnjšie vyučovacia forma praktických predmetov je prezenčná, príp. kombinovaná. Prezenčná forma vyučovania sa praktizuje na základných, stredných školách, aj v dennej forme štúdia na univerzitách. V rámci povinnej školskej dochádzky sa môže vzniknúť situácia, keď je prezenčnú výučbu nevyhnutne nahradiť dištančnou [6, 7].

Dôvodom na dištančné vzdelávanie môže byť viacero. Naposledy nás k tomu donútila pandémia COVID-19. Dôvodom dištančného vzdelávania môžu byť okrem pandémie, aj iné výnimočné okolnosti a situácie, ktoré si to vyžadujú alebo vynútili. Na základnej a strednej škole to môže byť napríklad zlý zdravotný stav žiaka, alebo určitý hendikep, ktorý neumožňuje účasť žiaka na dennej výučbe. Dištančné vzdelávanie a práca podľa individuálneho študijného plánu sa môžu dotýkať aj mimoriadne nadaných detí, ktoré sa zúčastňujú športových sústredení na zlepšenie výkonu, prípravných sústredení na účasť v medzinárodných súťažiach a podobne.

2 VYUČOVANIE POČAS PANDÉMIE

Počas pandemickej situácie s COVID19 na slovenských školách sa vyučovalo dištančne alebo kombinovanou formou. Aj keď na začiatku bolo mnoho problémov súvisiacich s adekvátnym technickým a softvérovým vybavením škôl a žiakov v domácom prostredí (hlavne v sociálne slabších rodinách). Nakoniec sa našli prijateľné riešenia [6]. Učitelia aj žiaci zvládli prácu vo zvolenom softvérovom prostredí, komunikovali náhradnými spôsobmi – prostredníctvom školského informačného systému alebo s pomocou komunikačných zariadení priamo alebo sprostredkované. Počas pandémie prebiehalo vyučovanie v obmedzenom režime, preto učitelia často redukovali pôvodný obsah výučby a niektoré témy v porovnaní s pôvodným plánom vynechali [8, 9]. Medzi vynechané tematické celky patrili

predovšetkým prakticky orientované predmety, hlavne z dôvodov nedostatku technického vybavenia [10, 11].

Žiaci kombinovanú a dištančnú formu vzdelávania vnímali rôzne. Krátkodobú zmenu zvládli celkom dobre a dá sa povedať, že bez väčších problémov. Používanie nových učebných pomôcok a digitálnych vzdelávacích technológií ich celkom bavilo. Po dlhodobom odlúčení detí od školy, od spolužiakov a kamarátov sa začali prejavovať príznaky nepohody a traumy. Okrem sociálnych väzieb so spolužiakmi a učiteľmi im chýbala aj možnosť pohybovať sa v známych priestoroch školy. Mnohí učitelia hľadali náhradné riešenia a začali ako učebné pomôcky intenzívnejšie využívať elektronické učebnice, klasické cvičebnice, didaktické aplikácie, virtuálne laboratória, virtuálne prostredky, edukačné webové stránky, vzdelávacie rozhlasové relácie a televízne programy. Z alternatívnych učebných pomôcok sa stali hlavné učebné pomôcky [12, 13, 14]. Učitelia tiež sami vytvárali rôzne vlastné aplikácie, virtuálne nástroje a triedy na spestrenie vyučovania s cieľom zvýšenia záujmu detí o školu a o vyučovanie.

Mnohí erudovaní učitelia sa snažili žiakom pomocou technológií umožniť aspoň virtuálne navštíviť svoju školu, prechádzať sa po známych priestoroch, realizovať svoje školské úlohy v triede, v laboratóriách a v iných špeciálnych zariadeniach školy. M. Zboran napr. vytvoril príbeh, ktorý žiaka vtiahol do deja a motivoval jeho snahu o úspešné vyriešenie zadaných úloh [15, 16]. Žiak sa pohyboval v priestoroch školy, riešil zadané úlohy a získaval za správne odpovede čiastkové odmeny, vďaka ktorým postupne splnil celkovú úlohu. Prvé úlohy boli zamerané na opakovanie učiva z chémie (vzhľadom na to, že druhým aprobačným predmetom tvorcu je chémia), takže okrem plnenia úloh hry si žiaci upevňovali už získané vedomosti. Takto chcel autor využiť prvky gamifikácie vo vyučovaní a podporiť získavanie a upevňovanie nových poznatkov hrovou formou.



Obrázok1: Sférická fotografia (virtuálnej) učebne výpočtovej techniky

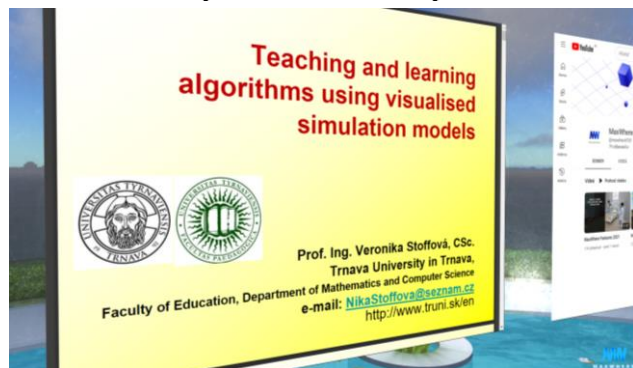
Podobné aplikácie boli vytvorené aj pre vybrané tematické celky vo viacerých ďalších odborných učebniciach, kde sa žiak mohol pripojiť s pomocou interaktívnych odkazov na virtuálne simulačné experimenty. Žiak mohol pozorovať a interaktívne meniť podmienky experimentu, čím si tiež hrovou formou získaval a upevňoval svoje vedomosti. (Pozri obr. 1.) Tak to mohol učiteľ sprístupniť vhodné (podľa vlastného uváženia a vlastných kritérií a zvážená selektované) video-tutoriály, vzdelávacie webové portály, elektronické učebnice, interaktívne didaktické materiály a pod. „z jedného miesta“.

Nie každý učiteľ je schopný vytvárať vlastné aplikácie „šité na mieru“, čo sa týka formy a obsahu prezentovaných poznatkov, ale

tí, ktorí majú chuť moderne vyučovať, majú k dispozícii širokú paletu voľne dostupných didaktických, interaktívnych aplikácií na spestrenie a zvýšenie efektivity vyučovania, či už v prezenčnej alebo v dištančnej forme [17, 18]. Na interaktívnu tvorbu vzdelávacieho prostredia aj pre neprofesionálov slúžia rôzne platformy. Mnohé z nich sú voľne dostupné, a pre záujemcov sú k dispozícii rôzne tutoriály, príručky, klasické a videonávody, ako v danom prostredí tvoriť. Nám sa osvedčil nástroj MaxWhere, ktorý umožňuje interaktívnu tvorbu didaktickej aplikácie v 3D priestore pre ľubovoľný predmet a využiť pri tom všetky svoje vopred pripravené elektronické učebnice učebné materiály, didaktické aplikácie, či už vlastné alebo voľne dostupné na Internete. Ilustračné snímky z tvorby v prostredí MaxWhere sú na obrázku 2 a 3.



Obrázok 2: Voľba prostredia na tvorbu aplikácie v MaxWhere



Obrázok 3: Tvorba aplikácie v prostredí MaxWhere

3 DIŠTANČNÉ VYUČOVANIE PRAKTICKY ORIENTO VANÝCH PREDMETOV

Realizovať praktické činnosti v rámci dištančného vzdelávania, ako je stavba robotov, vykonanie laboratórnych experimentov z chémie, z fyziky, z biológie a podobne, kde je potrebné používať špeciálne zariadenia a materiály, meracie prístroje, či chemikálie, nie je jednoduché. Je ťažké nahradiť reálny zážitok, aký ponúkajú praktické činnosti a reálne experimenty. V reálnych experimentoch možno do prijímania informácií zaradiť nielen zrak a sluch, ale aj hmat, čuch a chuť. Hoci ich podiel na prijímaní informácií v porovnaní so zrakom a sluchom je nižší, zanechávajú v pozorovateľovi silný nezabudnuteľný dojem. Často práve vôňa, chuť a hmat slúžia ako jednoznačné identifikačné atribúty na rozpoznanie objektov. Na prenos informácií, ktoré sú vnímané hmatom, čuchom a chuťovými bunkami nevymysleli zatiaľ žiadnu spoľahlivú metódu.

Do určitej miery, praktické činnosti a laboratórnu prácu dokážeme realizovať a preniesť do virtuálnej, prípadne rozšírenej reality. Školské experimenty možno vykonať v zdieľaných a vzdialených laboratóriách. Učiteľom realizovaný experiment v reálnych podmienkach školského laboratória možno sprostredkovať žiakom v reálnom čase, cez Internet alebo vyhotoviť videonahrávku, ktorú žiak môže doma prehrať aj viackrát, podľa potreby [19].

Hľadali sme možnosti, ako realizovať prakticky orientované edukačné činnosti dištančnou formou v oblasti informatiky. Ako príklad sme vybrali tri predmety, prípadne tri tematické celky: programovanie, edukačnú robotiku a pedagogickú prax. V ďalšom texte opíšeme ako sme tieto predmety vyučovali dištančne a aké výsledky sme dosahovali.

3.1 Programovanie

Vo vyučovaní programovania sa veľa nezmenilo. Už aj vzhľadom na to, že sme vždy mali k dispozícii celý rad elektronických učebných materiálov, hlavne vlastných. Prvé predmety programovania v učiteľskom štúdiu považujeme za úvod do štúdia programovania, kde sa snažíme vybudovať pevné základy tejto disciplíny. Počas dlhodobej praxe z vyučovania tohto predmetu na mnohých univerzitách sme získali bohaté skúsenosti a vybudovali sme celkom slušnú bázu elektronických učebných pomôcok na jeho efektívne vyučovanie [13, 14, 15, 19, 20]. Okrem teoretickej časti predmetu bola mimoriadna pozornosť venovaná praktickej časti predmetu. Vždy sme pracovali podľa zásad „programovať sa naučíš len programovaním“ a k tomu sme viedli aj našich študentov.

To ako sme pracovali v tomto predmete dištančnou formou počas pandémie sme podrobne referovali na mnohých konferenciách a odborných podujatiach. Počas dištančného vzdelávania sme pracovali v prostredí MS Teams a využívali sme možnosti, ktoré toto prostredie ponúka a to tak počas on-line prednášok ako aj cvičení, konzultácií, komunikácií so študentmi a skúšok [20].

Na cvičeniach aj počas konzultácií sme často využívali možnosť zdieľania obrazovky a pridelenie práva zásahu do programového kódu. Takže komunikácia a diskusia bola veľmi živá a úžitok z nej mali všetci účastníci – študenti aj učiteľ. Tak sa mohli veľmi rýchlo odstrániť „mylné koncepcie“ a študenti si vymieňali svoje skúsenosti a podelili sa o svoje nápady. Rady a usmernenia učiteľa si mohli vypočuť všetci. Intenzívnejšie sme používali interaktívne elektronické učebné pomôcky, ktoré študenti mali k dispozícii aj na domácu prácu a na prípravu na vyučovanie. Sem možno zaradiť elektronické učebnice, cvičebnice, zbierky (vzorovo) riešených úloh a námety na riešenie, edukačné webové stránky, tutoriály, videozáznamy prednášok a mnoho ďalších. Hlavne sa nám osvedčili interaktívne vizualizované simulačné modely (na stránke ide.sk).

3.2 Edukačná robotika – uplatnenie simulačných prostriedkov a virtuálnych nástrojov

Stavba a programovanie robotov sú pre žiakov základných škôl na získavanie praktických technických zručností z konštruovania funkčných zariadení veľmi dôležité. Vzhľadom na to, že rozvoj praktických zručností žiakov dištančnou formou je ťažko realizovateľný, hľadali sme možnosti a náhradné riešenia. Jedno z nich je používanie simulačných nástrojov a virtuálnych prostredí. Programovacie prostredie LEGO Digital Designer umožňuje vlastnú stavbu robota, rovnako aj Virtual Robotics Toolkit. Na

riadenie vytvorených robotov, či už z reálnych, alebo virtuálnych stavebníc sa môžu používať rovnaké programovacie prostredia, napr. LEGO MINDSTORMS Education EV3 Software for students.

Nielen LEGO MINDSTORMS EV3 robotická stavebnica má svoje ekvivalentné virtuálne prostredie, ale mnoho ďalších programovateľných hračiek a robotických stavebníc. Ako príklady uvedieme niekoľko tých, s ktorými máme skúsenosti a ktoré sme používali. Už v predškolských zariadeniach sa používajú programovateľné hračky ako sú Bee-Bot, Blue-Bot, rôzne programovateľné dopravné prostriedky (zmenšený model osobného auta, nákladného auta a pod.), ku ktorým sú vyhotovené virtuálne prostredia veľmi podobné tým reálnym. V týchto prostrediach je možné pohyb robotickej hračky programovať.

O problematike robotických stavebníc a ich využívaní v dištančnom vzdelávaní na základnej škole sme referovali na multikonferencii „Trendy vĕ vzdelávánĭ & DIDMATTECH 2022“ [22].

Mnohí učitelia, ktorým sa darilo vo vyučovaní robotiky do takej miery, že ich študenti dokázali uspieť na robotických súťažiach, chceli v príprave študentov na tieto súťaže pokračovať. Napr. učiteľ, ZŠ M. Zboran chcel zopakovať projekt robotického vysávača, ktorý úspešne realizoval v rámci krúžkovej činnosti v prezenčnej forme. Z robotickej stavebnice LEGO MINDSTORMS so žiakmi ZŠ skonštruoval robot s infračerveným a ultrazvukovými senzormi, ktoré slúžili na zaregistrovanie prekážky a zmenu smeru pohybu. Robotickú stavebnicu rozšíril dokúpeným malým statickým vysávačom. Model robota bol skonštruovaný tak, že vysávač sa dal nasadiť do stredu zo spodnej časti. Žiaci mali menšie problémy s konštrukciou, keďže sa museli prispôsobiť dokúpenému vysávaču a nastaviť vzdialenosť sacej plochy od podlahy tak, aby bola jeho účinnosť čo najlepšia. Najlepším konštrukčným riešením žiakov bol robot s dvomi motormi, vysávačom umiestneným v strede a so senzormi na bokoch robota. Súčasťou projektu bola aj skonštruovaná základňa pre vysávač, na ktorú sa vedel vrátiť po stlačení diaľkového infračerveného ovládača. Základňa obsahovala okrem ovládača aj infračervený senzor a jeden stredný motor (pozri obr. 4).

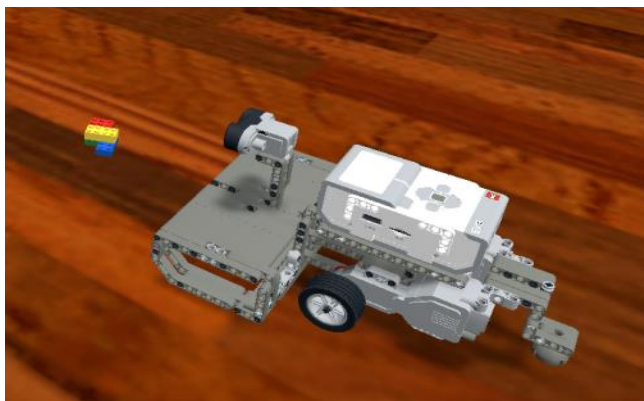


Obrázok 4: Programovateľný vysávač z reálnej robotickej stavebnice LEGO MINDSTORMS

Pandémia spôsobila, že vyučovanie prebiehalo dištančnou formou a aj po návrate žiakov do školy ešte stále nemohli pracovať s reálnymi stavebnicami z dôvodu hygienických opatrení. Hľadali sa náhradné riešenia ako by sa dalo vo vyučovaní robotiky

pokračovať. Ako vhodné riešenie sa ukázalo použitie virtuálneho prostredia Virtual Robotic Toolkit, ktoré simuluje robotické stavebnice EV3 Mindstorms.

Žiaci už mali predchádzajúcu skúsenosť s programovaním robota vo vizuálnom programovacom jazyku. Na komunikáciu so žiakmi sme používali MS Teams, ktorý mali žiaci k dispozícii bezplatne v rámci školskej licencie. Po predchádzajúcom využívaní prostredia Virtual Robotic Toolkit cez zdieľanú obrazovku, si niektorí žiaci (traja) sami zakúpili toto prostredie. Ostatní (šiesti) pracovali cez zdieľanú obrazovku učiteľa. Výsledok práce žiakov v tomto prostredí je na obr. 5.



Obrázok 5: Robotický vysávač vytvorený a naprogramovaný v prostredí Virtual Robotic Toolkit

Vzhľadom na skutočnosť, že edukačná robotika sa čoraz intenzívnejšie využíva vo vzdelávaní a často práve programovanie robotických hračiek a robotov znamenajú pre začínajúcich programátorov prvé dotyky s programovaním. Preto sa táto tematika dostala aj do študijných programov učiteľstva informatiky [22, 23, 24].

Autorka článok [6] H. Hyksová, uvádza konkrétne príklady a možnosti výučby robotiky a programovania robotov s pomocou virtuálneho robota VEX VR v dennej a dištančnej forme vzdelávania. Autorka píše aj o tom ako v dištančnej forme vyučovania vyriešili vyučovanie prakticky orientovaných predmetov na základných školách v Českej republike.

Používanie virtuálnych prostriedkov v stavbe a programovaní robotov zároveň vyriešil aj jeden veľký problém – nedostatočnú, príp. chýbajúcu technickú vybavenosť škôl. Pri používaní virtuálnych prostriedkov, každý žiak pracuje v rovnakých podmienkach.

Používanie simulačných nástrojov a virtuálnych prostredí sa nám osvedčilo aj v dištančnom vzdelávaní počas pandémie COVID–19. Podľa našich zistení bol spôsob riešenia problému s reálnym a virtuálnym robotom identické, iba v druhom prípade žiaci nemuseli konštruovať celý robot a riešiť funkčnosť dizajnu, čo ušetrilo čas a mohli sme sa viac zamerať na programovanie.

3.3 Príprava budúcich učiteľov na pedagogickú prax

Praktická príprava budúcich učiteľov informatiky je dôležitou a významnou súčasťou učiteľského štúdia. Ide o úzke prepájanie teórie s praxou. Budúci učiteľ získa ucelený obraz o svojom budúcom povolani. Často rozhoduje o tom, či nastúpi po skončení svojho štúdia do vzdelávacej praxe a či bude pokračovať vo svojej

učiteľskej profesii na ktorú sa pripravoval aj v priebehu pedagogickej praxe. Prípravu a realizáciu pedagogickej praxe budúcich učiteľov informatiky počas pandémie veľmi podrobne a precízne opisuje vo svojej dizertačnej práci N. Kvašayová [25]. Zaoberala sa metodickou prípravou študentov učiteľstva na pedagogickú prax s dôrazom na kritické a tvorivé myslenie, ale aj reflektovanou pedagogickou praxou. V práci sú kvalitatívne analyzované rôzne javy, ktoré vznikli počas pedagogickej praxe. Výskum ovplyvnila aj pandemická situácia. Je to jedinečné dielo v ktorom sa nachádza kvalitatívna analýza rôznych javov, ktoré vznikajú počas pedagogickej praxe vrátane tých, ktoré sa vytvorili v pandemickej situácii.

Riešenie pedagogickej praxe v počas pandémie si vyžadovalo nemalé úsilie a veľké pedagogické majstrovstvo zo strany univerzitného pedagóga zodpovedného za jej prípravu, organizovanie a realizáciu a tiež od cvičných učiteľov na základných a stredných školách. Často bolo nutné improvizovať a hľadať vhodné náhradné riešenia. Pedagogickú prax boli nútení realizovať dištančne v neznámych podmienkach, v málo známom prostredí, často bez potrebného technického a programového vybavenia zo strany školy a hlavne žiakov, s použitím často improvizovaných prostriedkov a pod.

Nikto totiž nepredpokladal takéto zásadné zmeny: Pripraviť budúceho učiteľa na vyučovaciu prax, ktorá môže prebiehať tak prezenčne, ako aj dištančne online formou.

Práve tieto problémy a vzniknutá nepríjemná situácia ukázala, že je nutné zmeniť a prepracovať učiteľské študijné programy a že je nevyhnutné zaradiť do učiteľskej prípravy nové predmety, príp. pozmeniť obsah tých existujúcich [23, 24].

Je dôležité aby budúci učelia v rámci vysokoškolskej prípravy získali dostatočné poznatky o formách, prostriedkoch a systémoch dištančného vzdelávania, videokonferenčných systémoch a tiež aby poznali základné školské informačné systémy (ŠIS), ktoré sa používajú v súčasnosti. Poznatky učiteľov z oblasti informačných systémov musia byť širšie a hlbšie než na úrovni zručného inteligentného používateľa. Hlavne učelia informatiky musia poznať aké služby ŠIS poskytujú, ako tieto služby fungujú a ako ich efektívne využívať. V rámci pedagogickej praxe musia získať tiež praktické skúsenosti z používania ŠIS. Pri používaní IKT, systémov a nástrojov dištančného vzdelávania a moderných ŠIS nemožno ignorovať ani počítačovú a internetovú bezpečnosť, ochranu osobných údajov a platnú legislatívu. Bezpečnosť a spoľahlivosť takýchto systémov patrí medzi základné požiadavky ich nasadenia. Na to, aby informačné systémy fungovali správne a boli dôveryhodné treba predchádzať prípadným rizikám. Preto musia byť poznatky o ŠIS zaradené do každého študijného programu učiteľského štúdia. Najlepšie by bolo zaradiť ich vyučovanie do spoločného základu.

4 ZÁVER

Vynútené dištančné vzdelávanie a realizované prieskumy počas pandémie COVID-19 odhalili rôzne problémy a potvrdili určité skutočnosti, ktoré sa v denných formách realizácie vyučovania neprejavili, príp. neboli také zjavné. Medzi nimi je nutné spomenúť tieto:

- Školy, učelia, a ani žiaci neboli a stále nie sú dobre pripravení na dištančnú formu vzdelávania;
- Učelia nemajú dostatok skúseností s vytváraním online materiálov a s poskytovaním online vzdelávania;

- Chýba koordinácia, riadenie spolupráce a plánovanie procesu tvorby vzdelávacích materiálov, tak po obsahovej ako aj formálnej stránke;
- Školské informačné systémy sa nevyužívajú v plnom rozsahu a mnohé ich funkcie nie sú učiteľom ani vedeniu školy známe;
- Chýbajúce kompetencie, profesionalita, poznatky a informácie o ŠIS pre a na vedenie základných a stredných škôl a školských zariadení ako aj ich výber pre konkrétnu školu môže byť čistá náhoda. bez presvedčenia o správnosti voľby;
- Učitelia sa nestretli a až do pandémie, nepoužívali žiadne kompletné softvérové systémy na dištančné (online) vzdelávanie v plnom rozsahu vzdelávacích programov;
- Školy využívali a využívajú rôzne systémy dištančného vzdelávania, a na niektorých školách veľmi často viac ako jeden súčasne;
- Školy a učitelia nie sú dostatočne informovaní o možnostiach dištančného vzdelávania a tiež o digitálnych vzdelávacích materiáloch, ktoré majú k dispozícii a ktoré sú voľne dostupné na vzdelávacích portáloch, webových sídlach a hniezdach.

Odporúčania:

- Je potrebné aby čerství absolventi učiteľských študijných programov prichádzali do praxe dostatočne pripravení na využívanie modernej didaktickej techniky a vzdelávacích technológií, softvérových aplikácií, ŠIS, a videokonferenčných systémov vhodných aj na dištančné vzdelávanie.
- Je nutné zaradiť do študijných programov univerzitetnej prípravy budúcich učiteľov tematické celky, ktoré by odstránili vyššie uvedené nedostatky.
- V aktuálnych učiteľských študijných programoch musia budúci učitelia získať vyššie vymenované chýbajúce zručnosti a určité skúsenosti už počas vysokoškolského štúdia a povinnej učiteľskej praxe.
- Túto akútnu spoločenskú objednávku je nutné splniť v príprave nových študijných programov v rámci pripravovanej novej akreditácie všetkých učiteľských študijných programov na slovenských univerzitách.
- Je tiež potrebné, aby školy vyriešili spôsob, ako zabezpečiť potrebné hardvérové nástroje s primeraným softvérovým vybavením pre žiakov, ktorí ich nevlástnia.
- Školy musia koordinovať a metodicky riadiť diaľkové štúdium a správne sa rozhodnúť, ktorý softvér by sa mal použiť na diaľkové štúdium, aby ich žiaci nemuseli využívať viac systémov súčasne.
- Je nutné znížiť počet použitých druhov ŠIS na základných a stredných školách a vo vzdelávacích zariadeniach. Používať len tie, ktoré sú kompatibilné s rezortným informačným systémom a sú s ním prepojené.
- Znížiť počet používaných druhov softvérových systémov na podporu a organizovanie dištančného vzdelávania. Na jednej škole používať najviac dva, ak je to opodstatnené (napr. iný na 1. stupni ZŠ) aj to tak, aby každý žiak používal z nich len jeden.
- Digitálna gramotnosť učiteľa a žiaka má byť z používania rôznych softvérových systémov a aplikácií na takej úrovni, aby neznižoval ich výkon.

- Centrálne riadiť tvorbu elektrických učebníc a učebných materiálov a ich distribúciu, aby boli pokryté aspoň kľúčové tematické celky a aby sa netvorili učebné materiály s veľkou redundanciou.
- Tiež je potrebné školám a žiakom sprístupniť celoštátne učebnice v elektronickej podobe.

POĎAKOVANIE

Článok vznikol s podporou KEGA MŠVVaŠ SR v rámci projektu: 013TTU-4/2021 Interaktívne animačno-simulačné modely pre deep learning

BIBLIOGRAFICKÉ ODKAZY

- [1] STOFFOVÁ, V. – HORVÁTH, R.: Gamification in Programming Learning. In *EDULEARN21*. Barcelona : IATED Academy, 2021. S. 9253–9258. ISBN 978-84-09-31267-2. ISSN 2340-1117.
- [2] VOŠTINÁR, P. - Motivational tools for learning programming in primary schools. In *Central European journal of new technologies in research education and practice*, vol. 2, no. 1, 2020, p. 97-106.
- [3] HORVÁTH, R. – STOFFOVÁ, V.: Project-based and Problem-based Teaching Programming. In *eLSE 2021*. Bucharest : CAROL I National Defence University Publishing House, 2021. S. 537–544. ISSN 2066-026X.
- [4] STOFFOVÁ, V. – CZAKÓOVÁ, K.: Úvod do programovania v prostredí mikrosvetov : (Vysokoškolská učebnica) / Veronika Stoffová, Krisztina Czaková ; [recenzenti: Imrich Okenka, Veronika Gabaľová]. - 1. vyd. - Komárno : Univerzita J. Selyeho, Ekonomická fakulta, 2016. - 114 s. - ISBN 978-80-8122-170-5. – Spôsob prístupu: <http://ukftp.truni.sk/epc/13632.pdf>.
- [5] HYKSOVÁ, H.: Robotic Toys and Kits in Primary School Education. In: 12th International Conference on Education and New Learning Technologies. Palma, Spain, 2020. ISBN: 978-84-09-17979-4, ISSN: 2340-1117.
- [6] STOFFOVÁ, V. – KARPIELOVÁ, M. – HORVÁTH, R. – NÉMETHOVÁ, S.: Some Possibilities of Distance Learning of Programming. In *INTED 2021*. Valencia : IATED. Int. Assoc. technology education & development, 2021. S. 4838–4844. ISBN 978-84-09-27666-0. ISSN 2340-1079.
- [7] STOFFOVÁ, V.: Využitie IKT v učiteľskej profesii – Tvorba didaktických aplikácií (Use of ICT in the teacher profession– Creation of didactic applications). In DRÁBKOVÁ Jindra a Jan BERKI, eds. *Sborník konference Didinfo 2018* [online]. Liberec, 2018, pp. 153–162, ISBN: 978-80-7494-424-6, ISSN: 2454-051X. Available at: didinfo.net/images/DidInfo/files/didinfo_2018.pdf
- [8] FESZTEROVÁ, M.: Waste as an innovative educational topic during the pandemic covid 19 in chemistry for pupils on the 7th grade of primary school. In *Edukacja w dobie pandemii*. - Radom : Uniwersytet Technologiczno-Humanistyczny im. Kazimierza Pułaskiego, 2022. - ISBN 978-83-7351-948-0. - ISSN 1642-5278, P. 129-156.
- [9] FESZTEROVÁ, M.: The Education of Pre-service Chemistry Teachers, the Content of Innovation, Methods and Forms during COVID-19. In *R&E-Source*. - ISSN 2313-1640, suppl. *Factory of Engineering Pedagogy - FEP*, č. 24 (2022), s. 1-8 [online]. DOI 10.53349/resource.2022.is24.a1104.

- [10] HYKSOVÁ, H.: Rozvoj konstrukčních dovedností – nezbytná součást robotiky. In: *Trendy ve vzdělávání*. Olomouc, 2020. ISBN: 978-80-568-0398-1. DOI: 10.5507/tvv.2020.006
- [11] STOFFOVÁ, V. – ZBORAN, M.: Shared and virtual chemistry laboratories for distance education. In: *ICERI2021 Proceedings 14th annual International Conference of Education, Research and Innovation Online Conference*. 8-9 November, Edited by: L. Gómez Chova, A. López Martínez, I. Candel Torres, IATED Academy, 2021, pp. 6526-6531, ISBN 978-84-09-34549-6, ISSN: 2340-1095, doi: 10.21125/iceri.2021.1479
- [12] PODHOROVÁ, M. – HORVÁTH, R.: Non-substance Addictions as a Modern Problem of Primary and Secondary School Learners. In *INTED2022*. Valencia : IATED, 2022, pp. 6690–6700, ISBN 978-84-09-37758-9. ISSN 2340-1079.
- [13] PŠENÁKOVÁ, I. – MINÁRIK, M., 2020 Využitie mikrokontrolera ako učebnej pomôcky = Use of the microcontroller as a teaching device. In: *Proceedings of 33. DidMatTEech 2020 Conference = Proceedings of XXXIII. DidMatTEech 2020 Conference*: New methods and technologies in education, research and practice. - [1. vyd.]. - Budapest: Eötvös Loránd University, 2020, p. 312-318, ISBN 978-963-489-244-1. - online,
- [14] CSÓKA, M. – CZAKÓOVÁ, K., 2021: Innovations in education through the application of raspberry pi devices and modern teaching strategies. In. *INTED 2021 Proceedings of the 15th International Technology, Education and Development Conference*. Valencia : IATED Academy, 2021.
- [15] STOFFOVÁ, V. – ZBORAN, M.: My virtual School – I attend school virtually. In: *INTED2022 Proceedings 16th International Technology, Education and Development Conference March 7th-8th 2022*. (Edited by: A. López Martínez, L. Gómez Chova, I. Candel Torres), [1. ed.] – Valencia : IATED Academy, 2022, pp. 5563-5569, ISBN: 978-84-09-37758-9 / ISSN: 2340-1079 doi: 10.21125/inted.2022
- [16] KOREŇOVÁ, L. - Symmetry in elementary education with the use of digital technologies and manipulations. In Szarková, D. - Letavaj, P. - Richtáriková, D. - Prašilová, M. (eds.) *Aplimat 2017* [elektronický zdroj] : proceedings. - First ed. - Bratislava: Slovak University of Technology in Bratislava, 2017, s. 836-845. - ISBN 978-80-22746-50-2
- [17] POKORNÝ, M.: E-learning in Teaching Basics of Graph Theory. In: *Central European journal of new technologies in research education and practice*. - ISSN 2676-9425. - Vol. 3, no. 3 (2021), p. 15-19.
- [18] POKORNÝ, M.: Video Lessons and E-learning Can Overcome Ban of Face-to-face Lessons in Teaching Mathematics. In: *2021 International Symposium on Educational Technology*. - Piscataway : Institute of Electrical and Electronics Engineers, 2021. - ISBN 978-1-6654-2859-0. - ISSN 2766-2144. - S. 44-47.
- [19] VÉGH, L. - From Bubblesort to Quicksort with Playing a Game. In NEUBAUER, J.- HAJKOVÁ, E. (eds.) *XXIX international colloquium on the management of educational processes*, PT 2. - 1. vyd. - Brno: Univ. defence., 2011. - ISBN 978-80-7231-812-4, s. 539-549.
- [20] STOFFOVÁ, V.- GABALOVÁ, V.- HORVÁTH, R.: How to teach programming in MS Teams, *14th annual International Conference on Education and New Learning Technologies*, 2022, Palma de Mallorca, Spain, ISBN 978-84-09-42484-9 , p. 4683-4689 doi: 10.21125/edulearn.2022.1114
- [21] ZBORAN, M. – STOFFOVÁ, V. – MIKOVÁ, K.: Robotické stavebnice v dištančnom vzdelávaní na základnej škole [electronic] = Robotic kits in distance education at primary school . In: *Trendy ve vzdělávání & DidMatTech 2022*. - Olomouc : Univerzita Palackého v Olomouci, 2022. - ISBN 978-80-244-6124-3. - S. 41-41 [online].
- [22] BUČKOVÁ, H. - DOSTÁL, J.: Innovative changes in the curriculum of information and communication technologies in the Czech Republic from the perspective of primary school teachers. In Jakab, F. (eds.) *ICETA 2020 [elektronický zdroj] : 18th IEEE International Conference on Emerging eLearning Technologies and Applications : proceedings : 3, 2020 Virtual Conference - Technical University of Košice, Slovakia*. - [1. vyd.]. - Danvers: IEEE, 2020, s. 46-51, ISBN 978-0-7381-2366-0, online
- [23] STOFFOVÁ, V.: Tanárjelöltek tanítási gyakorlatának jelenlegi kérdései, a tanárképzés digitális átalakítása. *Informatika Szakmódszertani Konferencia 2019.11.21-23., Zamárdi*. - [1. vyd.]. - Budapest: Webdidaktika Alapítvány, 2020. s. 215-230. ISBN 978-615-80608-3-7 (online)
- [24] STOFFOVÁ, V.: Školské informačné systémy v učiteľskej príprave. In: *UNINFOS 2020 - Zborník z online konferencie*, Peter Haľko, Milena Rybarčáková (eds.) 1. vyd. Prešovská univerzita v Prešove s. 62-84. ISBN 978-80-555-2685-0
- [25] KVAŠSAYOVÁ, N.: Vplyv metodickej prípravy a reflektovanej pedagogickej praxe na profesijné kompetencie budúcich učiteľov informatiky. [Dizertačná práca]. Univerzita Konštantína Filozofa v Nitre. Fakulta prírodných vied a informatiky. Školiteľ: doc. RNDr. Gabriela Lovászová, PhD. Stupeň odbornej kvalifikácie: Philosophiae Doctor. Nitra: FPVI, 2022. 194 s.

Úspěšnost žáků při řešení informatických úloh v průběhu školního dne

The Success of Pupils in Solving Informatics Tasks on a School Day

Václav Šimandl
KIN PF JU
Jeronýmova 10
371 15 České Budějovice
Česká republika
simandl@pf.jcu.cz

Jan Pršala
KIN PF JU
Jeronýmova 10
371 15 České Budějovice
Česká republika
jprsala@pf.jcu.cz

ABSTRACT

At the present time the subject of Informatics is introduced in the Czech Republic. As it is a new and particularly different subject, a question emerges at what time of day it would be the most convenient to teach this subject. The aim of our article is to explain in which lesson pupils perform the best in Informatics. We analysed performances of pupils of different age in the last year of the Bebras contest according to the time of start of the contest. We discovered that the time of learning Informatics influences especially younger pupils at elementary schools, who have better results in the morning and get worse in the course of time. The older pupils are the weaker and wider this phenomenon is – pupils at grammar schools achieve stable results during the whole monitored period, i.e. from 1st to 6th lesson.

Keywords

Timetable. Learning Informatics. The success of pupils. The Bebras contest.

ABSTRAKT

V současné době je v Česku zaváděn předmět informatika. Jelikož jde o předmět nový a značně se odlišující od výuky digitální gramotnosti, nabízí se otázka, v jaké části dne by bylo nejvhodnější tento předmět vyučovat. Cílem našeho článku je ozřejmit, v jaké době během výuky podávají žáci v informatice nejlepší výkony. Analyzovali jsme výkony žáků různého věku v uplynulém ročníku soutěže Bobřík informatiky na základě času soutěžení. Zjistili jsme, že čas výuky informatiky hraje roli především u mladších žáků na základních školách, kteří mají ráno lepší výsledky a postupem času se zhoršují. Čím starší jsou žáci, tím je tento jev slabší a širší – žáci střední školy dosahují v informatice stabilních výkonů po celé sledované období, tj. od 1. až do 6. vyučovací hodiny.

Klíčová slova

Rozvrh výuky. Výuka informatiky. Úspěšnost žáků. Bobřík informatiky.

1 ÚVOD

V současné době se na základních školách i na gymnáziích začíná vyučovat předmět „nová“ informatika. Tento předmět má u žáků rozvíjet informatické myšlení a vést je k porozumění principům digitálních technologií [1, 2]. V rámci předmětu by se měli žáci zabývat řešitelností problémů, hledáním a nalézáním jejich optimálních řešení nebo zpracováním dat a jejich interpretací [1]. Jak uvádí tzv. Modelové školní vzdělávací programy pro základní vzdělávání i pro gymnázia, výuka informatiky by měla být orientována činnostně, s aktivním žákem, který objevuje,

experimentuje, ověřuje své hypotézy, diskutuje, tvoří, řeší problémy, spolupracuje, pracuje projektově, konstruuje své poznání. Na pamětní učení a reprodukci by naopak neměl být kladen žádný důraz [3].

Rámcový vzdělávací program pro základní vzdělávání, podle něhož musí školy výuku zahájit nejpozději v září 2023, požaduje minimálně 2 hodiny informatiky týdně během 1. stupně a nejméně 4 hodiny týdně během druhého stupně [1]. Rámcový vzdělávací program pro gymnázia, podle něhož musí školy výuku zahájit nejpozději v září 2025, předpokládá alespoň 4 hodiny informatiky týdně [2].

Jelikož jde o předmět nový, nabízí se otázka, ve které části školního dne by bylo nejlepší tento předmět vyučovat s ohledem na co nejlepší výkony žáků. Obecné odpovědi mohou poskytnout zahraniční studie, avšak jejich závěry nejsou jednoznačné. Podle Mulenga a Mukuka dosahují žáci 2. stupně při dopolední výuce angličtiny a přírodních věd o něco lepších výkonů v porovnání se žáky při odpolední výuce [4]. Oproti tomu Sjosten-Bell rozdíl ve výkonech žáků 1. stupně při řešení matematických úloh na základě různé denní doby nezjistila [5]. Biggers uvádí, že výkon žáka je ovlivněn jeho biorytmem, konkrétně že klasický školní den podporuje vyšší výkony u žáků, kteří jsou aktivní ráno, než u žáků aktivních později [6]. Z této studie však není zřejmé, který typ žáka je převažující.

Protože se žádná z námi nalezených studií nezabývá výkony žáků v informatice, rozhodli jsme se realizovat výzkum zkoumající výkony žáků různého věku v rámci soutěže Bobřík informatiky, která je zaměřena na rozvoj informatického myšlení. Cílem tohoto výzkumu je zjistit, jak se mění výkony žáků při řešení informatických úloh v závislosti na části školního dne. Na základě tohoto cíle výzkumu jsme si stanovili následující výzkumné otázky:

- RQ1: Jsou výkony žáků během školního dne srovnatelné?
- RQ2: Pokud nikoliv, kdy soutěžící dosahují nejlepších výkonů?

Tyto výzkumné otázky jsme zodpovídali pro různé věkové kategorie žáků, a to vždy po dvou ročnících od 4. ročníku základní školy až po maturanty.

2 ORGANIZACE VÝUKY

Maximální počet vyučovacích hodin pro jednotlivé ročníky základní školy a středních škol jsou v Česku stanoveny legislativou [7] a uvádíme je v Tabulce 1. V téže tabulce uvádíme také průměrný počet vyučovacích hodin denně jakožto jednu pětinu

maximálního počtu hodin týdně. V Rámcovém vzdělávacím programu pro základní vzdělávání a Rámcovém vzdělávacím programu pro gymnázia jsou uvedeny povinné časové dotace pro jednotlivé stupně vzdělávání, které činí 118 hodin pro 1. stupeň ZŠ, 122 hodin pro 2. stupeň a 132 hodin pro gymnázia [1, 2]. Tyto hodnoty jsou o něco nižší než součty maximálních počtů hodin v jednotlivých ročnících. Vzhledem k relativně malým rozdílům v těchto dvou typech údajů budeme v dalším textu vycházet z maximálních počtů hodin pro jednotlivé ročníky.

Tabulka 1: Počet vyučovacích hodin podle ročníku

Typ školy	Ročník školy	Maximálně týdně	Průměrně denně
Základní škola	1. ročník	22	4,4
	2. ročník	22	4,4
	3. ročník	26	5,2
	4. ročník	26	5,2
	5. ročník	26	5,2
	6. ročník	30	6
	7. ročník	30	6
	8. ročník	32	6,4
	9. ročník	32	6,4
Střední škola	1. ročník	35	7
	2. ročník	35	7
	3. ročník	35	7
	4. ročník	35	7

Výuka na základních a středních školách začíná zpravidla v 8 hodin, ačkoliv se objevují názory navrhuující posun začátku výuky na 9. hodinu [8, 9]. Mezi vyučovacími hodinami jsou přestávky v obvyklé délce 10 minut, vyjma tzv. „velké přestávky“ o délce 15 nebo 20 minut. Odpolední vyučování je od dopoledního zpravidla odděleno pauzou o délce jedné vyučovací hodiny (a přestávkami před ní a po ní). Uvedené údaje jsou obvyklé, avšak jak připouští i legislativa, jednotlivé školy si mohou organizaci školního dne do určité míry uzpůsobit [10, 11]. Budeme-li vycházet z výše uvedených údajů, rozvrh hodin by měl přibližně odpovídat harmonogramu uvedenému v Tabulce 2.

Tabulka 2: Modelový rozvrh hodin

Vyučovací hodina	Začátek	Konec
1. hodina	8:00	8:45
2. hodina	9:05	9:50
3. hodina	10:00	10:45
4. hodina	10:55	11:40
5. hodina	11:50	12:35
6. hodina	12:45	13:30
7. hodina	13:40	14:25
8. hodina	14:35	15:20

3 SOUTĚŽ BOBŘÍK INFORMATIKY

Bobřík informatiky je online soutěž, která probíhá jako jednorázový test přímo ve školách [12]. Soutěž probíhá v 5 věkových kategoriích, které zahrnují žáky od 4. třídy základní školy až po maturitní ročníky [13]; bližší rozdělení je uvedeno v Tabulce 3. Soutěže se mohou zúčastnit celé třídy žáků nebo jen vybraní zájemci, rozhodnutí závisí na školním koordinátorovi

soutěže [14]. V nejmladší věkové kategorii Mini je dovoleno soutěžit dvojicím žáků.

Tabulka 3. Věkové kategorie soutěže Bobřík informatiky (podle [15])

Typ školy	Ročník školy	Kategorie soutěže
Základní škola	4. ročník	Mini
	5. ročník	
	6. ročník	Benjamin
	7. ročník	
	8. ročník	Kadet
9. ročník		
Střední škola	1. ročník	Junior
	2. ročník	
	3. ročník	Senior
	4. ročník	

Soutěžní test je tvořen 12 úlohami, na jehož splnění mají žáci 40 minut [16]. Za správnou odpověď se soutěžícímu přičte určitý počet bodů, za nesprávnou odpověď jsou naopak body odečteny. Pokud soutěžící v dané úloze neodpoví, nejsou mu body přičteny ani odečteny [16].

Soutěžít je možné v předem určených dnech (tyto soutěžní dny se liší dle kategorií), a to vždy od 7:30 do 15:40. V tomto ohledu soutěž respektuje organizaci školního dne na základních a středních školách tak, jak je uvedeno v předchozí kapitole. K přihlášení žáka do soutěže je potřeba tzv. školní kód, který má k dispozici školní koordinátor soutěže. Tento kód učitel sdělí před začátkem soutěže žákům, například jej napíše na tabuli [16]. Žáci by se soutěže měli účastnit pod dohledem učitele, který dbá na regulérnost soutěže. Žákům není dovoleno soutěžit opakovaně. Pokud školní koordinátor soutěže zjistí neférové jednání žáka (žák při soutěži podváděl, soutěžil opakovaně, soutěžil mimo dohled učitele apod.), má možnost žáka vyškrtnout [16], díky čemuž nebude žák veden mezi řádnými soutěžícími.

4 SBĚR DAT A ANALÝZA DAT

4.1 Sběr a selekce dat

Pro naši realizovaný výzkum jsme využili výzkumný vzorek, který tvořili všichni účastníci soutěže Bobřík informatiky z listopadového kola soutěže. Abychom zajistili integritu dat, z výzkumu jsme vyloučili soutěžící, kteří byli vyškrtnuti školními koordinátory soutěže. Celkem jsme do výzkumu zařadili 173891 soutěžících¹; počet respondentů (soutěžících) podle jednotlivých kategoriích uvádí Tabulka 4.

Tabulka 4: Počty respondentů podle věkových kategorií

Věková kategorie	Počet respondentů
Mini	46853
Benjamin	59937
Kadet	42721
Junior	17515
Senior	6865
Celkem	173891

¹ Soutěžní dvojici v tomto výzkumu chápeme jako jednoho soutěžícího (respondenta)

Respondenty v každé kategorii jsme následně rozdělili do skupin podle toho, kdy začali soutěžit. Rozhodující přitom byla hodina počátku soutěžení, vznikly tedy skupiny respondentů se začátkem soutěžení v časových oknech 7:30 (včetně) až 7:59 (včetně); 8:00 až 8:59; ..., 14:00 až 14:59, 15:00 až 15:39. Přehled počtu respondentů v jednotlivých skupinách uvádí Tabulka 5. Skupiny respondentů, které jsme začlenili do výzkumu, mají v Tabulce 5 bílý podklad, ostatní mají světle šedý podklad. Rozhodnutí o (ne)zařazení jednotlivých skupin výzkumu vysvětluje následující text.

Tabulka 5: Počty respondentů podle času začátku soutěžení v jednotlivých věkových kategoriích

Časové okno	Mini	Benj.	Kadet	Junior	Senior
7:xx	460	1250	1237	731	285
8:xx	6382	8871	6411	3698	1492
9:xx	6367	7203	4890	2382	901
10:xx	10098	12517	7206	3329	1271
11:xx	14020	13942	8153	2737	960
12:xx	7824	11836	7207	2064	814
13:xx	1295	2962	4103	1383	715
14:xx	323	1272	3325	1046	357
15:xx	84	84	189	145	70

Před dalším zpracováním dat jsme se museli rozhodnout, které skupiny v jednotlivých věkových kategoriích podrobíme dalšímu zkoumání. Ačkoliv se nabízela možnost zkoumat všechny, obávali jsme se zkresení výsledků. Je totiž pravděpodobné, že zatímco v některých částech dne soutěžily celé třídy, v jiných soutěžili téměř výlučně předem vybraní žáci, kteří mají zájem o informatiku. Lze předpokládat, že tito žáci dosahují v informatice lepších výsledků než jejich spolužáci. Z hlediska způsobu soutěžení tak lze nalézt tři typy škol:

- Školy, ve kterých soutěžily celé třídy v době výuky informatiky nebo v době výuky jiného příbuzného předmětu.
- Školy, ve kterých soutěžili vybraní žáci v době výuky informatiky nebo v době výuky jiného příbuzného předmětu; zbytek třídy pracoval na jiných úkolech.
- Školy, ve kterých soutěžili vybraní žáci ve svém volném čase (nebo v rámci volnočasových kroužků) před výukou nebo po ní.

První dva typy škol naše výsledky nezkrusují, neboť jsou vyučovací hodiny, během nichž soutěžily, rozloženy během školního dne rovnoměrně. Třetí typ škol naše výsledky může zkrusovat, neboť koncentroval žáky se zájmem o informatiku do doby po skončení povinné výuky, resp. před jejím začátkem. Z tohoto důvodu jsme výzkumu zúžili pouze na časová okna, v nichž téměř jistě probíhala povinná výuka žáků dané kategorie. Při rozhodování, zda respondenty, kteří začali soutěžit v určitém časovém okně, zařadit do výzkumu nebo nikoliv, jsme vycházeli z údajů v Tabulkách 1, 2 a 3. Jestliže například je ve 4. a 5. ročníků základní školy (kategorie Mini) průměrný denní počet vyučovacích hodin roven 5,2, je pravděpodobné, že výuka na řadě škol končí po 5. vyučovací hodině. Lze tak předpokládat, že v rámci povinné výuky začali žáci soutěžit se nejspíše v časovém okně 11:xx.

U starších věkových kategorií jsme museli zvažovat nejednotnost rozvrhů mezi školami a třídami. Některé školy (resp. část jejich tříd) mohly mít v den soutěžení odpolední vyučování, a tudíž mohli

během výuky soutěžit ještě v časovém okně 14:xx. Jiné třídy mohly výuku končit po 5. vyučovací hodině a v časovém okně 12:xx soutěžili pouze vybraní žáci v rámci zájmové činnosti. Lze očekávat, že v pozdějších časových oknech se podíl žáků soutěžících v rámci zájmové činnosti zvyšoval. Po přihlídnutí k počtu soutěžících v jednotlivých časových oknech jsme jako poslední časové okno zahrnuté do výzkumu zvolili u kategorie Mini časové okno 11:xx, u ostatních kategorií 12:xx. Jsme si vědomi, že jsme do výzkumu mohli zařadit více časových oken s tím, že v nich podíl žáků soutěžících mimo povinnou výuku nebude tak vysoký. Rozhodli jsme se však raději některé skupiny respondentů do výzkumu nezařadit a ochudit se o potenciálně zajímavé výsledky než získat výsledky, které jsou zkruslené a neodpovídají realitě. Podobně jsme přistupovali i k respondentům, kteří začali soutěžit v časovém okně 7:30 až 7:59. Ačkoliv jsme si vědomi, že některé školy začínají výuku ještě před 8. hodinou ranní, obáváme se, že řada škol nechala vybrané zájemce soutěžit ještě před začátkem výuky, což by opět zkrusleno výsledky výzkumu.

4.2 Analýza dat

Analýza dat probíhala v jednotlivých kategoriích odděleně, kdy jsme vzájemně porovnávali výkony respondentů se začátkem soutěžení ve zkoumaných časových oknech. K tomuto účelu jsme využili statistický software R. Všechny níže uvedené statistické testy jsme provedli na hladině významnosti $\alpha = 0.01$, kterou jsme zvolili s přihlídnutím k velikosti výzkumného vzorku.

K zodpovězení RQ1 se vzhledem k typu dat nabízelo využití statistický test analýzy rozptylu. U každé kategorie jsme testovali, jestli splňuje předpoklady pro použití tohoto testu. Pomocí Andersonova-Darlingova testu jsme testovali normalitu dat a pomocí Levenova testu jsme testovali homoskedasticitu. Během testování předpokladů jsme zjistili, že ani jedna kategorie dat nesplňuje předpoklady pro analýzu rozptylu a nebude možné tento test použít. Proto jsme rozdílly testovali pomocí Kruskalova-Wallisova testu.

Pokud jsme zjistili, že v některé kategorii je mezi časovými okny statisticky významný rozdíl, provedli jsme post hoc analýzu v dané kategorii s cílem zodpovědět RQ2. Post hoc analýzu jsme provedli pomocí Dunnova testu.

5 VÝSLEDKY VÝZKUMU

5.1 Jsou výkony žáků během školního dne srovnatelné (RQ1)?

V každé kategorii dat jsme otestovali normalitu pomocí Andersonova-Darlingova testu. Nulová hypotéza byla zamítnuta ve všech případech.

Poté jsme provedli Kruskalův-Wallisův test. Testovali jsme následující hypotézu H_0 proti H_1 :

H_0 : Čas výuky informatiky nemá vliv na výkon žáků.

H_1 : Čas výuky informatiky má vliv na výkon žáků.

Výsledky tohoto testu jsou zobrazeny v Tabulce 6 (dvěma hvězdičkami označujeme statisticky významné hodnoty na hladině $\alpha = 0.01$).

Tabulka 6: Výsledky Kruskalova-Wallisova testu

Věková kategorie	p-hodnota
Mini	0.0082**
Benjamin	0.0000**
Kadet	0.0000**
Junior	0.0201
Senior	0.0859

Jak je zřejmé z Tabulky 6, nulovou hypotézu jsme zamítli v případě kategorií Mini, Benjamin a Kadet, a nezamítli v případě kategorií Junior a Senior.

V případě kategorií **Junior** a **Senior**, které jsou určeny pro žáky středních škol, tedy můžeme tvrdit, že výkony žáků začínajících soutěž v době od 8:00 do minimálně 12:59 jsou srovnatelné. Jelikož je soutěž Bobřík informatiky zaměřena stejně jako školní předmět „nová“ informatika na rozvoj informatického myšlení, je možné toto tvrzení zobecnit na předmět informatika. Lze tak tvrdit, že výkony žáků středních škol v informatice jsou od 1. až do minimálně 6. vyučovací hodiny (včetně) srovnatelné.

5.2 Kdy soutěžící dosahují nejlepších výkonů (RQ2)?

U kategorií Mini, Benjamin a Kadet jsme zamítli nulovou hypotézu, tedy můžeme tvrdit, že výkony žáků v těchto kategoriích nejsou během školního dne srovnatelné. Zajímalo nás proto, jaké časové intervaly se od sebe liší.

K zodpovězení této otázky jsme využili Dunnův test, který jsme použili na každou z výše zmíněných kategorií a jehož výsledky v podobě p-hodnot uvádíme v Tabulkách 7, 9 a 11 (dvěma hvězdičkami označujeme statisticky významné hodnoty na hladině $\alpha = 0.01$).

Tabulka 7: Výsledky Dunnova testu pro kategorii Mini

	8:xx	9:xx	10:xx	11:xx
8:xx	-	0.0041**	0.0025**	0.0021**
9:xx	0.0041**	-	0.1364	0.7516
10:xx	0.0025**	0.1364	-	0.0278
11:xx	0.0021**	0.7516	0.0278	-

V kategorii **Mini** jsme odhalili, že se významně liší výsledky v časovém intervalu 8:00-8:59 oproti ostatním zkoumaným intervalům, tj. 9:00-11:59. Žáci v čase mezi 8. hodinou a 9. hodinou dosahovali v průměru lepších výsledků než v ostatních časech (viz Tabulka 8). Lze tedy říci, že žáci 4. a 5. ročníku ZŠ během 1. vyučovací hodiny dosahují lepších výkonů v informatice než během 2. až 5. vyučovací hodiny.

Tabulka 8: Průměrné bodové zisky žáků v kategorii Mini během dne

Časový úsek	Průměr
8:xx	95.2
9:xx	92.8
10:xx	91.6
11:xx	93.0

Mezi ostatními časovými intervaly jsme nezjistili statisticky významné rozdíly ve výkonech žáků. Lze tak tvrdit, že v rozmezí od 2. do 5. vyučovací hodiny jsou jejich výkony stabilní.

Tabulka 9: Výsledky Dunnova testu pro kategorii Benjamin

	8:xx	9:xx	10:xx	11:xx	12:xx
8:xx	-	0.2590	0.0001**	0.0005**	0.0000**
9:xx	0.2590	-	0.0736	0.1461	0.0002**
10:xx	0.0001**	0.0736	-	0.6109	0.0126
11:xx	0.0005**	0.1461	0.6109	-	0.0017**
12:xx	0.0000**	0.0002**	0.0126	0.0017**	-

V kategorii **Benjamin** jsme zjistili, že se významně liší výsledky v časovém intervalu 8:00-8:59 oproti intervalům 10:00-10:59, 11:00-11:59 a 12:00-12:59. Na základě porovnání průměrného počtu bodů (viz Tabulka 10) můžeme říci, že žáci v časovém intervalu 8:00-8:59 dosahovali lepších výsledků než v časových intervalech 10:00-12:59. Lze tedy říci, že žáci 6. a 7. ročníku ZŠ během 1. vyučovací hodiny dosahují významně lepších výkonů v informatice než během 3. až 6. vyučovací hodiny.

Tabulka 10: Průměrné bodové zisky žáků v kategorii Benjamin během dne

Časový úsek	Průměr
8:xx	92.8
9:xx	92.1
10:xx	90.3
11:xx	90.6
12:xx	87.3

V této kategorii jsme dále zjistili, že se významně liší výsledky v časovém intervalu 12:00-12:59 oproti intervalům 8:00-8:59, 9:00-9:59 a 11:00-11:59. Na základě porovnání průměrného počtu bodů (viz Tabulka 10) můžeme říci, že žáci v časovém intervalu 12:00-12:59 dosahovali horších výsledků než v časových intervalech 8:00-9:59 a 11:00-11:59. Lze tedy tvrdit, že žáci 6. a 7. ročníku ZŠ během 6. vyučovací hodiny dosahují horších výkonů v informatice než během 1., 2. a 5. vyučovací hodiny.

Mezi časovými intervaly 9:00-9:59, 10:00-10:59, 11:00-11:59 jsme nezjistili statisticky významné rozdíly ve výkonech žáků. Lze tak tvrdit, že v rozmezí od 2. do 5. vyučovací hodiny jsou jejich výkony stabilní.

Tabulka 11: Výsledky Dunnova testu pro kategorii Kadet

	8:xx	9:xx	10:xx	11:xx	12:xx
8:xx	-	0.4400	0.4400	0.0000**	0.0011**
9:xx	0.4400	-	0.3200	0.0000**	0.0001**
10:xx	0.4400	0.3200	-	0.0001**	0.0088**
11:xx	0.0000**	0.0000**	0.0001**	-	0.4400
12:xx	0.0011**	0.0001**	0.0088**	0.4400	-

V kategorii **Kadet** jsme zjistili, že se významně liší výsledky v časových intervalech 8:00-8:59, 9:00-9:59 a 10:00-10:59 oproti ostatním zkoumaným intervalům. Na základě porovnání průměrného počtu získaných bodů (viz Tabulka 12) lze říci, že žáci 8. a 9. ročníku ZŠ během 1. až 4. vyučovací hodiny dosahují lepších výkonů v informatice než během 5. až 6. vyučovací hodiny.

Mezi časovými intervaly 8:00-8:59, 9:00-9:59 a 10:00-10:59 jsme nezjistili statisticky významné rozdíly ve výkonech žáků. Lze tak tvrdit, že v rozmezí od 1. do 4. vyučovací hodiny jsou jejich výkony stabilní. Obdobně lze říci, že v průběhu 5. a 6. vyučovací hodiny jsou výkony žáků stabilní.

Tabulka 12: Průměrné bodové zisky žáků v kategorii Kadet během dne

Časový úsek	průměr
8:xx	70.3
9:xx	71.6
10:xx	69.9
11:xx	66.2
12:xx	66.9

Obrázek 1 obsahuje krabicové grafy každé kategorie podle času zahájení testu. Z grafů můžeme vidět výsledky popsané výše a sestupnou tendenci výkonů v testu v prvních třech kategoriích v závislosti na čase. Výsledky v kategoriích Junior a Senior jsou i zde viditelně vyrovnané.

6 ZÁVĚR

Zkoumali jsme, jestli jsou výkony žáků v informatice srovnatelné během školního dne, a pokud nikoliv, tak nás zajímalo, kdy žáci dosahují lepších výkonů. Za tímto účelem jsme analyzovali výsledky soutěže Bobřík informatiky v každé věkové kategorii.

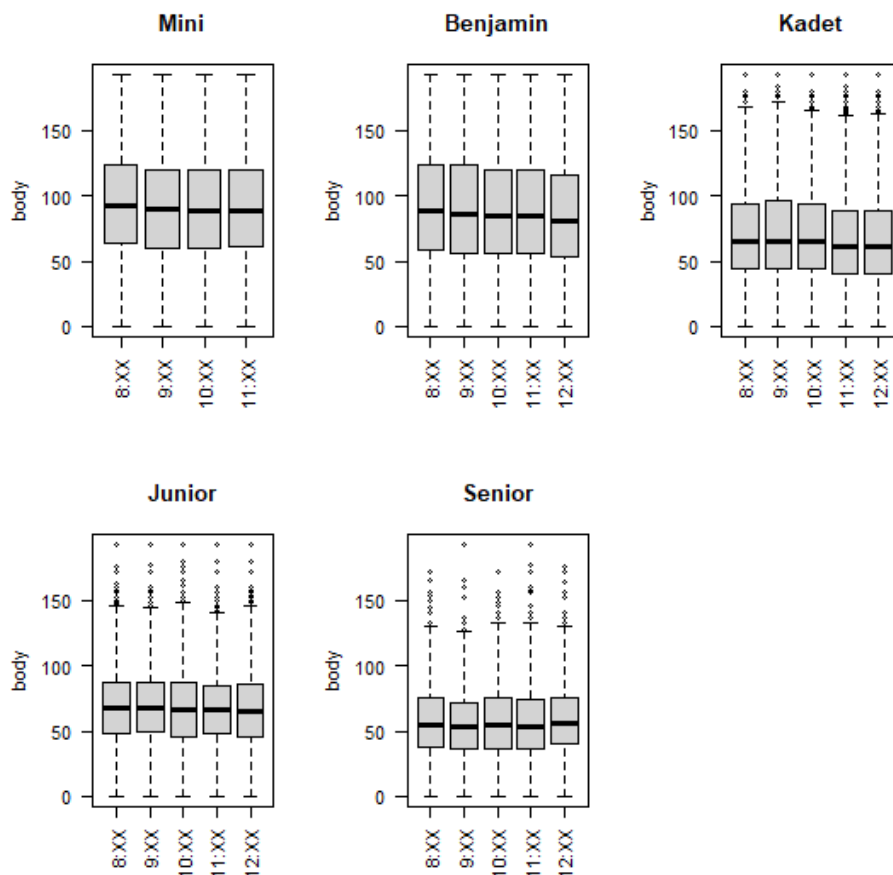
Zjistili jsme, že na středních školách jsou výsledky žáků z informatiky srovnatelné přinejmenším od 1. do 6. vyučovací hodiny. Oproti tomu se na základních školách výsledky během dne

mění a tyto změny jsou statisticky významné. U všech kategorií na základních školách (Mini, Benjamin, Kadet) jsou výsledky dosažené ráno znatelně lepší než výsledky dosažené později v průběhu dopoledního vyučování.

Nejlepších výsledků ve všech kategoriích dosahují žáci během 1. vyučovací hodiny. V kategorii Mini (4. a 5. ročník ZŠ) jsou výkony žáků významně lepší během 1. vyučovací hodiny oproti 2. až 5. vyučovací hodině, během nichž se už dále nezhoršují. V kategorii Benjamin (6. a 7. ročník ZŠ) jsou výkony žáků nejlepší během 1. (a případně 2.) vyučovací hodiny, v ostatních hodinách jsou horší – zatímco v rozmezí od 2. do 5. vyučovací hodiny jsou stabilní, během 6. hodiny jsou opět významně horší oproti předchozím hodinám. V kategorii Kadet (8. a 9. ročník ZŠ) jsou výkony stabilně nejlepší v rozmezí 1. až 4. vyučovací hodiny a významně horší jsou během 5. a 6. hodiny (v rámci nichž jsou stabilní).

Na základě vzájemného porovnání zjištění ze všech kategorií je patrné, že čas výuky informatiky hraje roli především u mladších žáků na základních školách, kteří mají ráno lepší výsledky a postupem času se horší. Čím starší jsou žáci, tím je tento jev slabší a širší (žáci v posledních ročnících základní školy dosahovali nejlepších výsledků první čtyři vyučovací hodiny). Žáci střední školy dosahují v informatice stabilních výkonů po celé sledované období, tj. od 1. do 6. vyučovací hodiny.

Obrázek 1: Krabicové grafy každé kategorie



Za limit našeho výzkumu lze považovat především nemožnost odlišit žáky, kteří soutěžili v rámci povinné výuky, od žáků, kteří se soutěže zúčastnili po výuce, resp. před ní. Jak uvádíme v kapitole 4.1, lze očekávat, že po výuce soutěžili především žáci se zájmem o informatiku, což zkruskuje výsledky výzkumu. Riziko zkruslení výzkumu jsme eliminovali tím, že jsme do analýzy dat nezařadili skupiny respondentů, kteří začali soutěžit v časových oknech, kdy s velkou pravděpodobností už neprobíhala povinná výuka. V tomto ohledu jsme se snažili o validní výzkum za cenu ochuzení o potencionálně zajímavé výsledky. Vzhledem k různorodosti harmonogramu školního dne na jednotlivých školách je však otázku, zda jsme do výzkumu neměli zařadit respondenty z více časových oken, v jejichž rámci respondenti začali soutěžit, a současně zachovali validitu výzkumu, museli bychom být schopni u každého respondenta určit, zda soutěžil v rámci povinné výuky, či nikoliv.

Dalším limitem výzkumu pro zobecnění výsledků na celý školní rok může být jednorázový sběr dat. Ten proběhl v první polovině listopadu. Není tedy jisté, zda je vývoj výkonů žáků během dne obdobný i na konci školního roku, kdy je zcela jiná délka dne, venkovní teplota apod.

Přínos výzkumu spočívá v objasnění, v jaké době během školního dne dosahují žáci základních a středních škol nejlepších výsledků. Nami zjištěné výsledky lze využít při rozhodování, zda informatiku zařazovat na začátku, nebo spíše ke konci školního dne. Náš výzkum taktéž do značné míry odpovídá na otázku, zda jsou žáci schopni plnohodnotných výkonů už od 1. vyučovací hodiny. Další výzkumy v této oblasti by se měly věnovat výkonům v dalších vyučovacích předmětech tak, aby bylo možno získat ucelený obraz této problematiky.

BIBLIOGRAFICKÉ ODKAZY

- [1] *Rámcový vzdělávací program pro základní vzdělávání* [online]. Praha : Ministerstvo školství, mládeže a tělovýchovy, 2021 [cit. 2023-01-16]. <https://www.edu.cz/wp-content/uploads/2021/07/RVP-ZV-2021.pdf>
- [2] *Rámcový vzdělávací program pro gymnázia* [online]. Praha : Ministerstvo školství, mládeže a tělovýchovy, 2021 [cit. 2023-01-16]. https://www.edu.cz/wp-content/uploads/2021/09/001_RVP_GYM_uplne_zneni.pdf
- [3] Modelové školní vzdělávací programy. In *Informatické myšlení* [online]. 2022 [cit. 2023-01-16]. <https://imysleni.cz/svp>
- [4] MULENGA, H.M., MUKUKA, A. Learning Time of Day and Students' Academic Achievement at School Certificate Level: A Case Study of Chibote Girls' Secondary School. In *Journal of Education and Practice*. 2016, roč. 7, č. 20, s. 88-93. <https://files.eric.ed.gov/fulltext/EJ1109220.pdf>
- [5] SJOSTEN-BELL, W. *Influence of Time-of-Day on Student Performance on Mathematical Algorithms* : Diplomová práce [online]. San Rafael, CA : Dominican University of California, 2005. <https://files.eric.ed.gov/fulltext/ED490712.pdf>
- [6] BIGGERS, J.L. Body Rhythms, the School Day, and Academic Achievement, In *The Journal of Experimental Education*. 1980, roč. 49, č. 1, s. 45-47, <http://doi.org/10.1080/00220973.1980.11011761>
- [7] Zákon č. 561/2004 Sb., o předškolním, základním, středním, vyšším odborném a jiném vzdělávání (školský zákon). Verze 52. In *Zákony pro lidi* [online]. 2021 [cit. 2023-01-16]. <https://www.zakonyprolidi.cz/cs/2004-561>
- [8] ZAJAC, L. Začátek výuky od devíti, chtějí Zelení. Podle ředitelů by to přineslo problémy. In *Idnes.cz* [online]. 2021 [cit. 2023-01-16]. https://www.idnes.cz/zpravy/domaci/zacatek-skolni-vyuky-9-00-hodin-reditele-skol-strany-volby-zeleni-ods-kscmano.A210906_095718_domaci_lre
- [9] MENŠÍK, J. Revoluce ve školství? Co kdyby výuka začínala až od 9:00, přemýšlí politici. In *Novinky.cz* [online]. 2022 [cit. 2023-01-16]. <https://www.novinky.cz/clanek/domaci-revoluce-ve-skolstvi-co-kdyby-skola-zacinala-az-od-9-00-premysli-politici-40403815>
- [10] Vyhláška č. 48/2005 Sb., o základním vzdělávání a některých náležitostech plnění povinné školní docházky. Verze 11. In *Zákony pro lidi* [online]. 2022 [cit. 2023-01-16]. <https://www.zakonyprolidi.cz/cs/2005-48>
- [11] Vyhláška č. 13/2005 Sb., o středním vzdělávání a vzdělávání v konzervatoři. Verze 7. In *Zákony pro lidi* [online]. 2022 [cit. 2023-01-16]. <https://www.zakonyprolidi.cz/cs/2005-13>
- [12] VANÍČEK, J. Rozvoj informatických kompetencí žáků pomocí soutěže: kauza Bobřík informatiky [online]. In *Sborník konference Počítač ve škole*. Nové Město na Moravě : Gymnázium Vincence Makovského, 2009 [cit. 2023-01-19]. <https://www.bebas.org/sites/default/files/documents/publications/Vanicek%20-2009.pdf>
- [13] Obecné informace. In *Bobřík informatiky* [online]. 2023 [cit. 2023-01-19]. <https://www.ibobr.cz/o-soutezi/obecne-informace>
- [14] Jak zorganizovat soutěž na škole. In *Bobřík informatiky* [online]. 2023 [cit. 2023-01-19]. <https://www.ibobr.cz/o-soutezi/jak-zorganizovat-soutez-na-skole>
- [15] Věkové kategorie. In *Bobřík informatiky* [online]. 2023 [cit. 2023-01-19]. <https://www.ibobr.cz/o-soutezi/vekove-kategorie>
- [16] Pravidla soutěže. In *Bobřík informatiky* [online]. 2023 [cit. 2023-01-19]. <https://www.ibobr.cz/o-soutezi/pravidla-souteze>

Aktivita EU Code Week Spievajme so Scratchom

EU Code Week activity Let's sing with Scratch

Lubomír Šnajder
Prírodovedecká fakulta UPJŠ
Jesenná 5
041 54 Košice
Slovenská republika
lubomir.snajder@upjs.sk

Katarína Brinziková
Prírodovedecká fakulta UPJŠ
Jesenná 5
041 54 Košice
Slovenská republika
katarina.brinzikova@student.upjs.sk

Ján Guniš
Prírodovedecká fakulta UPJŠ
Jesenná 5
041 54 Košice
Slovenská republika
jan.gunis@upjs.sk

ABSTRACT

As a constructionist development environment, Scratch allows you to program different types of artifacts that can be shared, commented, liked, and remixed on the cloud. In this article, we focus on the creation of music projects in Scratch realized in the framework of our own activity Let's sing with Scratch, which was part of the international initiative EU Code Week 2022. In the first part of the article, we describe the process of this activity and in the second part we evaluate the results achieved – selected music projects of students and teachers from participating schools. At the end of the article, we summarize the achieved results with benefits for students and teachers.

Keywords

Informal education. Programming. Scratch. Music. EU Code Week.

ABSTRAKT

Scratch ako konštrukcionistické vývojové prostredie umožňuje vytvárať rôzne typy artefaktov, ktoré sa dajú zdieľať, komentovať, lajkovať a remixovať na cloude. V článku sme sa zamerali na tvorbu hudobných projektov v Scratchi realizovaných v rámci vlastnej aktivity Spievajme so Scratchom, ktorá bola súčasťou medzinárodnej iniciatívy EU Code Week 2022. V prvej časti článku popisujeme priebeh tejto našej aktivity a v druhej časti vyhodnocujeme dosiahnuté výsledky – vybrané hudobné projekty žiakov a učiteľov zo zúčastnených škôl. V závere článku sumarizujeme dosiahnuté výsledky s benefitmi pre žiakov a učiteľov.

Kľúčové slová

Neformálne vzdelávanie. Programovanie. Scratch. Hudba. EU Code Week.

1 ÚVOD

Súčasťou informatického vzdelávania je aj neformálne vzdelávanie, ktoré sa realizuje prostredníctvom záujmových krúžkov, táborov, súťaží, festivalov vedy, online aktivít atď. Aj keď neformálne vzdelávanie je dobrovoľné, organizované pre vybranú skupinu žiakov, bez nutnosti existencie kurikula a systematického prístupu, má veľký potenciál pre skvalitnenie informatického vzdelávania. Okrem prípravy žiakov na súťaže, či doučovania žiakov, poskytuje žiakom príjemnejšie prostredie na riešenie neštandardných problémov a tvorbu rôznych artefaktov bez strachu zo zlej známky. Pre učiteľov je neformálne vzdelávanie šancou vyskúšať výučbu nových tém ešte pred ich zavedením do formálneho vzdelávania, či rozvíjať témy, na ktoré sa nenašiel čas vo formálnom vzdelávaní pre všetkých žiakov. Súčasná školská informatika má veľký potenciál rozvíjať informatické myslenie, kolaboratívne a komunikačné spôsobilosti, čítanie s porozumením, a to zaradovaním nových (aj interdisciplinárnych) tém a prístupov

v informatickom vzdelávaní [1]. Jedným z príkladov zaujímavej interdisciplinárnej témy je programovanie hudby a zvukov.

Viaceri výskumníci a pedagógovia sa zaoberajú edukačnými aktivitami prepájajúcimi programovanie s hudbou a zvukmi vo vývojových prostrediach využívaných v našich školách, napr. Scratch [2, 3, 4], Imagine Logo [5], MIT App Inventor [6, 7], MakeCode [8, 9], EV3-G [10, 7], Python [11]. Pre rozvíjanie algoritmického myslenia slabozrakých a nevidiacich žiakov boli vyvinuté špecializované programovacie prostredia využívajúce zvuky a tóny ako napr. Torino či Coshi [12]. Na umeleckých školách sa na tvorbu skladieb a na spracovanie zvukov využívajú špecializované programovacie prostredia ako napr. Sonic Pi [13] či PureData [14].

Na našom pracovisku je problematika programovania hudby a zvukov súčasťou syllabov predmetov QPR1 Programovanie v Scratchi a TSM1b Tvorba a spracovanie multimédií. V rámci Klubu učiteľov informatiky sme sa na stretnutí 13. 12. 2022 venovali problematike programovaniu hudby a zvukov v jazykoch Scratch, Python a Sonic Pi. V minulom akademickom roku sme v priebehu obdobia 11.–24. 10. 2021 zorganizovali pre žiakov ZŠ, SŠ, ich učiteľov a tiež pre študentov učiteľstva informatiky vlastnú aktivitu Programujeme hudbu a zvuky, ktorá bola súčasťou medzinárodnej iniciatívy EU Code Week 2021 [15].

V nadväznosti na uvedenú aktivitu sme v tomto akademickom roku 2022/2023 v rámci EU Code Week [16] zrealizovali vlastnú aktivitu Spievajme so Scratchom, ktorej cieľom, priebehom a výsledkom sa venujeme v ďalšom texte tohto článku.

2 POPIS AKTIVITY

Aktivitu Spievajme so Scratchom sme realizovali v priebehu obdobia 11. – 23. 10. 2022 ako vlastné verejné podujatie (s kódom cw22-louWz), do ktorého sa mohli zapojiť školy so svojimi vlastnými lokálnymi EU Code Week podujatiami. Na webovej stránke aktivity [17] sme uviedli ciele, očakávaný výsledok a priebeh aktivity spolu s odporúčaniami a dosiahnutými výsledkami.

2.1 Ciele a výsledok aktivity

Hlavnými cieľmi aktivity boli:

- prehĺbenie pozitívneho postoja žiakov a učiteľov k programovaniu ako k prostriedku na vytváranie a prezentovanie vlastných hudobných projektov,
- rozvíjanie multimediálnej gramotnosti, programátorských zručností a tvorivosti žiakov a učiteľov,
- uvedomenie si základov kódovania hudby prostriedkami programovacieho jazyka.

Očakávaným výsledkom aktivity bolo vytvorenie zbierky hudobných projektov naprogramovaných v Scratchi žiakmi a

učiteľmi zo zúčastnených škôl a ich publikovanie na Scratch cloude.

2.2 Priebeh aktivity

Na začiatku sme oslovili školy ohľadom zapojenia sa do tejto aktivity. Následne táto aktivita prebehla v dvoch etapách:

- Prvá etapa: 11. – 16. 10. 2022
Kontaktné osoby (učitelia) na svojich školách informovali žiakov a prípadne učiteľov o tejto aktivite a zistili záujem o zapojenie sa. Škola takto mohla pridať svoje vlastné lokálne EU Code Week podujatie ako súčasť nášho verejného podujatia.
Žiaci samostatne alebo s pomocou učiteľa (vo výučbe, či na krúžku) preštudovali ukážky hudobných projektov na Scratch cloude v štúdiu Spievajme so Scratchom a začali programovať vlastné hudobné projekty.
Dňa 11. 10. 2022 v čase 16:00 – 17:00 sme prostredníctvom videokonferenčnej platformy Big Blue Button zorganizovali online tutoriál k tvorbe hudobných projektov v jazyku Scratch.
- Druhá etapa: 17. – 23. 10. 2022
Žiaci a učitelia naprogramovali svoje hudobné projekty a publikovali ich na Scratch cloude v štúdiu Spievajme so Scratchom [18].
Kontaktné osoby (učitelia) zaregistrovali odkazy na publikované Scratch projekty ako výstupy tejto aktivity za svoju školu vyplnením online formulára, kde uviedli aj pre stručnú štatistiku pre potreby EU Code Week 2022.

2.3 Odporúčania k vývoju Scratch projektov

V rámci tejto aktivity sme sa zamerali na nasledujúce formy naprogramovaných hudobných projektov:

- **pesnička** s vlastným spevom so statickou či animovanou grafikou súvisiacou s obsahom piesne, prípadne textom piesne,
- **hudobná skladba** bez spevu prehraná viacerými hudobnými nástrojmi (skutočné nástroje alebo syntetické pomocou príkazov Scratchu),
- **hudobný nástroj** realizovaný ako interaktívny projekt využívajúci stláčanie klávesov, klikanie na rôzne postavy reagujúce rečou, hudbou či zvukovým efektom,
- **iná** vyššie neuvedená forma hudobného projektu.

K návrhu, programovaniu a publikovaniu Scratch projektov sme autorom poskytli nasledujúce odporúčania:

- Do tejto aktivity sa môžu zapojiť žiaci, či už samostatne alebo v tímoch, a zároveň aj samotní učitelia danej školy (nielen informatiči). Takto sa učitelia budú môcť zdokonaľiť v programovaní, a tiež si užiť kreatívne chvíle pri tvorbe hudobných projektov.
- Prioritou pri tvorbe projektov by nemala byť technická náročnosť a komplexnosť programového kódu, ale skôr dobrý nápad a využiteľnosť vytvoreného projektu v praxi.
- Pri tvorbe projektov v Scratchi môžeme použiť nasledujúce bloky:
 - zo skupiny Zvuk: **zahraj zvuk, zastav všetky zvuky, zmeň efekt, nastav efekt, zruš**

zvukové efekty, zmeň hlasitosť, nastav hlasitosť, hodnota hlasitosť,

- z pripojeného rozšírenia Hudba: **zahraj tón, zabubnuj, nehraj, nastav nástroj, nastav tempo, zmeň tempo,**
- z pripojeného rozšírenia Text to Speech: **povedz, nastav hlas, nastav jazyk,**
- zo skupiny Zisťovanie: **hodnota hlasitosť,**
- zo skupiny Udalosti: **pri hlasitosť.**
- Autori projektov by mali podľa svojich možností používať vlastné obrázky a zvuky, v prípade cudzích by mali byť v niektorej z Creative Commons licencií s uvedením odkazu pod obrázkom.
- Autori projektov sa môžu inšpirovať študovaním cudzích projektov, ktoré môžu komentovať, lajkovať či remixovať.
- Autori projektov by mali pred publikovaním projektu skontrolovať a vyčistiť programový kód, či v ňom nie sú nepotrebné nezapojené bloky príkazov.
- Autori projektov by mali na stránke svojich projektov v časti Instructions uviesť stručnú anotáciu projektu s jeho ovládaním a v časti Notes and Credits uviesť názov svojej školy a triedy, prípadne poďakovanie inému autorovi, od ktorého sa inšpirovali.

2.4 Ukážkové hudobné projekty

Pre inšpiráciu participujúcich učiteľov a žiakov k tvorbe ich hudobných projektov sme vytvorili 12 hudobných projektov, ktoré sme publikovali na Scratch cloude v štúdiu Spievajme so Scratchom [18] a prezentovali na webinári 11. 10. 2022. Prehľad ukážkových hudobných projektov s ich stručným popisom uvádzame v nasledujúcom zozname¹.

- **Experimentujeme so zvukmi – 586081785**
Projekt, v ktorom žiaci klikaním na bloky a stláčaním klávesov preskúmajú ako fungujú vybrané príkazy a parametre prehrávania zvukov, rôzne efekty prehrávania zvukov, príkazy a nastavenia syntézy reči, príkazy prehrávanie tónov pre rôzne nástroje a rôzne bicie.
- **Vianočný a novoročný pozdrav – 465773242**
Ukážka hudobného pozdravu s trojhlasne zaspievanou vianočnou koledou so súbežne spustenou animáciou tancujúceho kocúrka, otáčajúceho sa pozdravného textu a padajúcich klonov snehovej vločky.
- **Viachlasná pesnička – 14862688**
Ukážka pesničky s vlastnou päťhlasnou skladbou s výberom jednotlivých prehrávaných hlasov so súbežnou animáciou vybraných postáv.
- **Rokenrol verzia 0 – 744061140**
Ukážka najjednoduchšej verzie skladby rokenrol zadanej postupnosťou 56 príkazov na prehrávanie tónov.
- **Rokenrol verzia 1 – 586101575**
Ukážka vylepšenej verzie skladby rokenrol zadanej pomocou troch vlastných procedúr bez parametrov s ôsmymi príkazmi na prehrávanie tónov.
- **Rokenrol verzia 2 – 586103330**
Ukážka ďalšej vylepšenej verzie skladby rokenrol zadanej len pomocou jednej vlastnej procedúry s parametrom s ôsmymi príkazmi na prehrávanie tónov.

¹ V nasledujúcej časti uvádzame len čísla projektov bez prefixu s webovou adresou <https://scratch.mit.edu/projects/>

- **Rokenrol verzia 3 – 586107107**
Ukážka ešte viac vylepšenej verzie skladby rokenrol zadanej pomocou dvoch zoznamov (s osemprvkovým základným motívom tónov a so sedemprvkovým zoznamom transpozícií tónov) prehrávaných pomocou cyklov a jednej vlastnej procedúry s parametrom (transpozície).
- **Skladba typu toccata – 13845210**
Ukážka vlastnej skladby (inšpirovanej skladbou J. S. Bacha Tokáta a fuga d-moll, BWV 565), zadanej pomocou dvoch zoznamov (základného motívu a transpozícií) a prehrávanej pomocou vnorených cyklov so súbežnou vizualizáciou výšky prehrávaných tónov.
- **Karaoke piesne – 2669976**
Ukážka detskej pesničky zadanej pomocou cyklu a procedúr bez parametrov pre každý verš piesne so súbežným prehrávaním melódie a animovaným kurzorom pod aktuálnym textom.
- **Virtuálne bicie – 743228952**
Virtuálne bicie ovládané pomocou klávesnice s vybraným bicím nástrojom so súbežne prehrávaným cyklickým podkladom.
- **Hudobný nástroj ovládaný micro:bitom – 744703375**
Hudobný nástroj s bicími ovládanými tlačidlami a s melódiou s výberom tónov podľa náklonu BBC micro:bitu.
- **Virtuálny klavír ovládaný pohybom tela – 12814519**
Virtuálny jednooktávový klavír ovládaný tromi spôsobmi – pohybom tela (prsta) pred webkamerou, stláčaním klávesov klávesnice, klikaním myši na zobrazené klávesy.

3 VYHODNOTENIE AKTIVITY

Úlohou žiakov bolo samostatne alebo s pomocou učiteľa (vo výučbe, či na krúžku) preštudovať ukážky hudobných projektov na Scratch cloude v štúdiu Spievajme so Scratchom a začať programovať vlastné hudobné projekty. Žiaci a učitelia naprogramovali svoje hudobné projekty a publikovali ich na Scratch cloude v štúdiu Spievajme so Scratchom. Spolu bolo odovzdaných 11 projektov zo ZŠ, 8 projektov, pri ktorých nebola uvedená škola, 7 projektov z UKF v Nitre a 12 našich ukážkových a motivačných projektov z UPJŠ v Košiciach.

Zoznam všetkých projektov²:

- PROJEKT 1: **Matúš pieseň** (Autor: Nixiy6)
- PROJEKT 2: **Alex Prešov pieseň** (Autor: Nixiy16)
- PROJEKT 3: **PO pieseň** (Autor: Nixiy16)
- PROJEKT 4: **Simonka&Laura** (Autor: 4ZS_Michalovce)
- PROJEKT 5: **Musicamy** (Autor: 4ZS_Michalovce)
- PROJEKT 6: **Rusínska koleda** (Autor: 4ZS_Michalovce)
- PROJEKT 7: **Mining_GKM** (Autor: danil_GGG)
- PROJEKT 8: **Vo vesmíre** (Autor: jurajVav)
- PROJEKT 9: **Tanečnica zo žaby** (Autor: InMaVI)
- PROJEKT 10: **Hudobný koncert** (Autor: Beffsie)
- PROJEKT 11: **Andante koncert** (Autor: MandarinkaDarinka)
- PROJEKT 12: **Twinkle Twinkle Little Star** (Autor: 21lulu21)
- PROJEKT 13: **Billie Jean** (Autor: danokas)
- PROJEKT 14: **Multiinstrumentalista** (Autor: ceceteam)
- PROJEKT 15: **Matrix** (Autor: gabilov)
- PROJEKT 16: **Vystúpenie** (Autor: hrabalka)
- PROJEKT 17: **nočné mesto** (Autor: VEVA148)
- PROJEKT 18: **KLAVÍR** (Autor: Lukasenucko588)

² Zoznam všetkých projektov s odkazom na webovú adresu je uvedený v PRÍLOHE

- PROJEKTY 19 – 30: sú uvedené v podkapitole 2.4.
- PROJEKT 31: **Happy b-day!** (Autor: cibereova)
- PROJEKT 32: **Klavír scratch** (Autor: cibereova)
- PROJEKT 33: **Klavír-JK copy remix** (Autor: cibereova)
- PROJEKT 34: **nabrala je jagode petrovke** (Autor: cibereova)
- PROJEKT 35: **Untitled** (Autor: wanda_demjan)
- PROJEKT 36: **Teletabies_by Štefka** (Autor: cibereova)
- PROJEKT 37: **Spievame a hráme so Scratchom** (Autor: RadoZajic)
- PROJEKT 38: **LoFi paradička** (Autor: RadoZajic)

Každý projekt bol výnimočný, tvorivý, nápaditý a niektoré projekty programátorsky gradovali. Počuli sme napr. pieseň „A od Prešova“, ktorú naspieval žiacky kolektív. Pieseň graficky spracovali a naprogramovali v navzájom podobnom štýle (PROJEKTY 1 – 3). Moderné skladby, rap, hudobné nástroje a dokonca aj Rusínske koledy sa rozozvučali v projektoch od žiakov zo Základnej školy z Michaloviec. Zaujal nás komixový rozhovor postáv, jeden z nich aj v anglickom jazyku (PROJEKTY 4 – 6). Zahrali sme sa na baníka, objavovali sme aj vesmírne prevrpenia. V týchto projektoch sa hudba objavila iba vo forme zvukových efektov (PROJEKTY 7, 8).

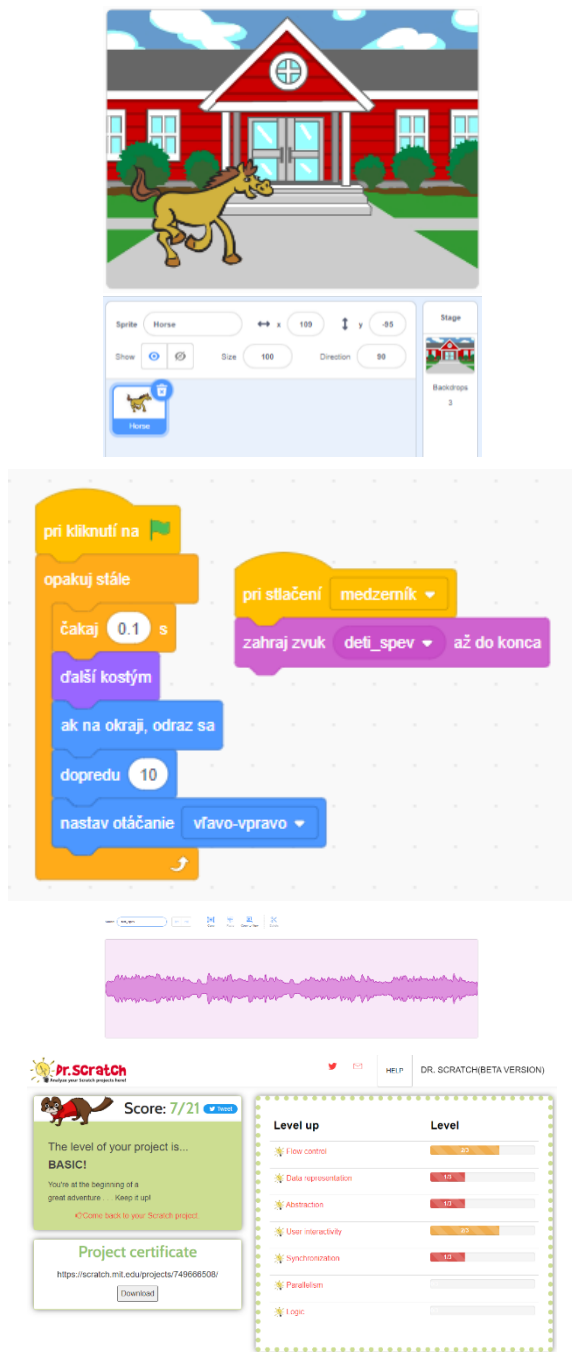
Nasledovali naozaj skvelé a prepracované projekty v podaní študentov z UKF v Nitre. Kúzelné efekty premenili žabu na tanečničku, ktorá tancuje na našu hru na klavíri. Zvuk sa prehráva stlačením kláves na počítačovej klávesnici. Boli sme na hudobných koncertoch, kde sme si mohli zvoliť hudobný nástroj, nastaviť rýchlosť pohybu tanečníka. Výučbový projekt, ktorý podľa návodu učí používateľa ako zahrať konkrétnu skladbu. Pri tanečných kreáciách postavičiek pri skladbe Billy Jean sa nedalo len tak ostať sedieť na stoličke. Autori vytvorili tanečný klíp s presným nástupom postáv. Na záver sa vypíše text „dovidenia“, ktorý je tiež naprogramovaný (PROJEKTY 9 – 15).

Jemné baletné vystúpenie nás opäť usadilo. Pozreli sme si nočné mesto so sprievodnými tónmi. Hrali sme na klavíri pomocou kláves, nie len na virtuálnej klávesnici, ale aj na tej ozajstnej, počítačovej (PROJEKTY 16 – 18).

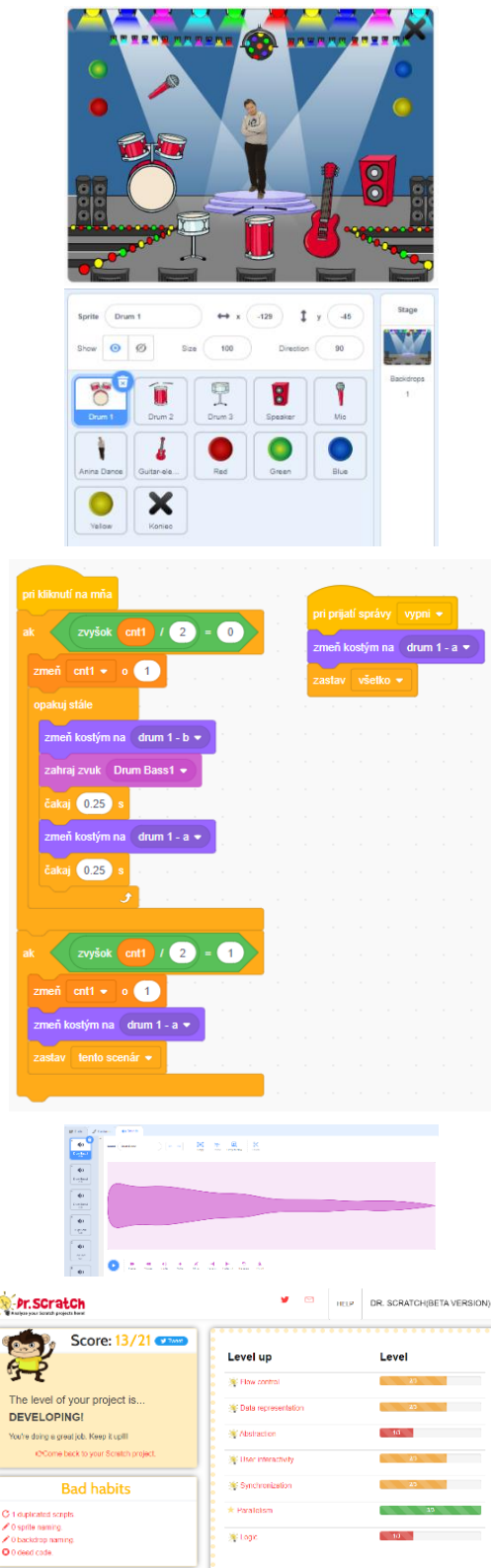
Ukážkové motivačné projekty (PROJEKTY 19 – 30) spomenuté tiež v kapitole 2.4 nás mali vtiahnuť do problematiky hudby a programovania, nájdete v nich množstvo inšpirácie a skvelých hudobných ukážok.

Niektoré projekty (PROJEKTY 31 – 38) boli publikované do Scratch cloudu po termíne, ale v súbore CodeWeek2022-vystup.docx [19] ich nájdete všetky.

V tomto súbore [19] sú uvedené všetky projekty s popisom a s náhľadom postáv, kódu resp. kódov, výsledku hodnotenia podľa Dr. Scratch [20] (pozri Obrázok 1 a Obrázok 2).



Obrázok 1 Ukážka a výsledok analýzy jednoduchého programu (PROJEKT 2)



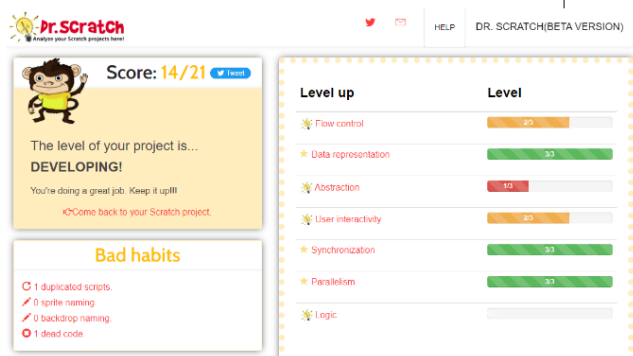
Obrázok 2 Ukážka a výsledok analýzy zložitejšieho programu (PROJEKT 10)

V ďalšej časti analyzujeme vnútornú stránku projektov – ako sú naprogramované, kde hodnotíme projekty kvalitatívne na základe nášho skúmania riešenia projektov aj kvantitatívne pomocou analytického nástroja Dr. Scratch.

Dr. Scratch sa zameriava na nasledovných 7 zložiek:

- riadenie toku (Flow control),
- reprezentáciu údajov (Data representation),
- abstrakcie (Abstraction),
- používateľskú interaktivitu (User interactivity),
- synchronizáciu (Synchronization),
- paralelizmus (Parallelism),
- logiku (Logic).

Maximálne dosiahnuteľné skóre podľa Dr. Scratch je 21 a teda každá zložka môže nadobudnúť najviac tri body. Jednotlivé zložky sú prehľadne znázornené farebnými škálami (pozri Obrázok 3).



Obrázok 3: Ukážka analýzy projektu v Dr. Scratch

Projekty boli veľmi rôznorodé, nápadité a mnohé aj inšpiratívne. PROJEKT 7 *Mining_GKM* dosiahol podľa Dr. Scratch hodnotenie 17 z 21. Tento projekt bol vytvorený na spôsob hry, kde žiak evidentne remixoval hotovú hru (extrémne vysoká zložitosť kódu) a len niektoré prvky v nej upravil a doplnil. V hre boli použité zvukové efekty. Nevieme však posúdiť, akým spôsobom boli nahrávané. Učítelia a žiaci často zabúdajú na autorské práva.

Tri projekty dosiahli ďalšie najvyššie hodnotenie a to 14 z 21. Na tieto hudobné projekty sa zameriame. PROJEKT 14 *Multiinštrumentalista* (pozri Obrázok 4) je výbornou ukážkou spojenia znalostí hudby a programovania s využitím zoznamov, do ktorých sa nahodia číselné hodnoty príslušných nôt. Navyše, keď sa prispôbi trvanie tónu pomocou hodnôt v inom zozname a využije sa cyklus, tak známa melódia je na svete. Veľkosť kódu sa využitím zoznamov navyše značne skráti. PROJEKT 15 *Matrix* (pozri Obrázok 5) zaujme spočiatku známou filmovou zvúčkou, ktorá je naozaj tón po tóne naprogramovaná. Sprievodným obrazovým javom je padanie písmen a čísel, ktoré sa navyše vyskytujú presne na pozícii klaviatúry.

Veľmi nápadité a originálne riešenie. Viachlasná pesnička s vysokým skóre *5voice song2* (pozri Obrázok 6) je ukážkový program, v ktorom používateľ vstupuje do programu interaktívne. Vyberá si z piatich postavičiek, kde každá má nahratý iný hlas. Spolu znejú ako výborná natrénovaná acapella, pretože nábeh melódie každej postavičky je v presnom takte.



Obrázok 4: Multiinštrumentalista

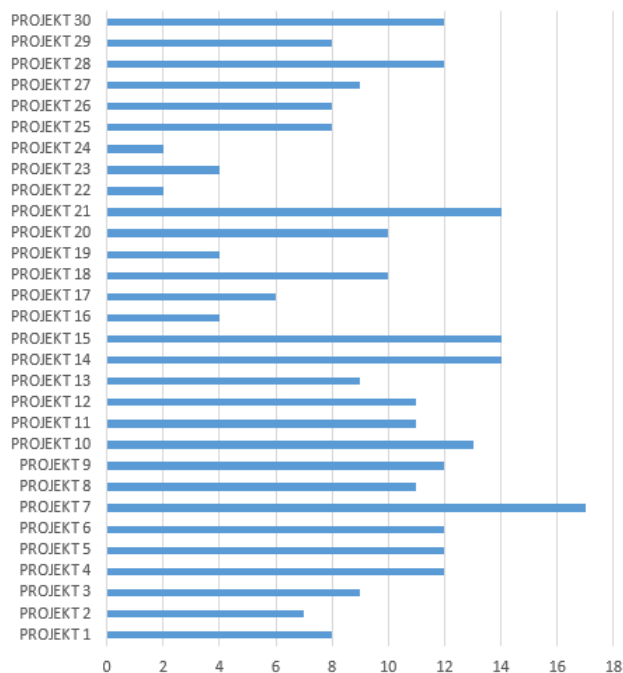


Obrázok 5: Matrix



Obrázok 6: 5voices song2

Bodové hodnotenie všetkých načas odovzdaných projektov podľa Dr. Scratch je uvedené na Obrázok 7.



Obrázok 7: Počty dosiahnutých bodov 30 projektov podľa Dr. Scratch

Početnosti dosiahnutých bodov 30 projektov podľa Dr. Scratch uvádzame v Tabuľka 1.

Tabuľka 1: Početnosti bodov 30 projektov podľa Dr. Scratch

Získané body / maximum	Počet projektov
17 / 21	1
14 / 21	3
13 / 21	1
12 / 21	6
11 / 21	3
10 / 21	2
9 / 21	3
8 / 21	4
7 / 21	1
6 / 21	1
4 / 21	3
2 / 21	2
SPOLU	30

Z nášho pohľadu projekty pôsobia zábavne, poučné a rôznorodo – našli sme tu spievané piesne, hudobné nástroje, hry, interaktivitu, tanec, skvelých programátorov, ale aj nádejných programátorov. Náročnosť programov bola od 2 do 17 bodov podľa 21 bodovej škály Dr. Scratch. Zo skúmaných 30 projektov bolo priemerne 9,5 bodov. Čísla sú jeden parameter, ale obľúbenosť druhý. Projekty boli osobne overené v praxi na vyučovaní informatiky na základnej škole ako prehliadka projektov. Obrovský ohlas mali skladby spievané žiakmi s jednoduchými kódmi (tým žiaci aj rozumeli), tiež projekty, do ktorých žiaci mohli interaktívne vstupovať a tvoriť vlastné melódie pomocou ponúknutých nástrojov. Veľmi zábavne napr. pôsobil projekt Loľi paradíčka alebo Spievame a hráme so Scratchom, na čo sa hneď žiaci pýtali: „A to môžeme vytvoriť aj my?“ Vždy je vhodné neostať pri jednoduchých kódoch, ale smerovať žiakov k efektívnosti, klásť otázky typu, či sa niečo v kóde neopakuje, či nie je jednoduchšie použiť zoznam ako rozsiahly kód, smerovať ich aj k národnej hrdosti ľudovej tvorby aj vo zvukovom podaní, ale aj v obrazovom napr. kreslenie ľudových vzorov. Veríme, že online aktivita Spievajme so Scratchom tvorcom, ale aj používateľom otvorila a otvorí brány do ďalšieho hudobno-programátorského poznávania.

4 ZÁVER

V príspevku sme opísali priebeh a výsledky online aktivity Spievajme so Scratchom ktorá bola súčasťou medzinárodnej iniciatívy EU Code Week 2022. Aj keď sa do aktivity nezapojilo veľa škôl, potešilo nás zapojenie rôznych stupňov škôl (od základnej až po vysokú), kde autormi rôznorodých projektov boli nielen žiaci, ale aj ich učelia. Väčšinou túto aktivitu žiaci realizovali doma alebo na krúžku ako nepovinné zadanie. Na jednej základnej škole ju realizovali priamo vo výučbe informatiky a hudobnej výchovy. Na UKF v Nitre aktivitu realizovali v niekoľkých predmetoch ako povinné v iných ako nepovinné zadanie. Táto aktivita ukázala, že má potenciál umožniť žiakom precvičiť a zdokonaľiť svoje programátorské zručnosti a skúsenosti, ukázať im na význam programovania ako spôsobu uvažovania a vytvárania užitočných artefaktov s prepojením na rôzne oblasti – v našom prípade hudbu. Aj keď sme túto aktivitu chápali primárne ako neformálne vzdelávanie žiakov, vďaka webinára a našim ukázkam ju považujeme aj za neformálne vzdelávanie učiteľov informatiky, či budúcich učiteľov informatiky. Veríme, že táto aktivita bude pre učiteľov informatiky

inšpirujúca a užitočná pri ich výučbe programovania v rámci formálneho aj neformálneho vzdelávania.

POĎAKOVANIE

Radi by sme sa poďakovali za spoluprácu učiteľom zapojených základných a stredných škôl – Márii Vavrovej, Márii Mrázovej, Nikole Pasternákovéj, Radovanovi Zajícovi, Zlatici Cibereovej, a tiež kolegyniam z UKF v Nitre – Gabriele Lovászovej a Viere Michaličkovej za zapojenie študentov učiteľstva do tejto aktivity a aj za ich vlastné tvorivé výstupy.

Tento článok bol vytvorený v rámci projektu KEGA 012UPJŠ-4/2021 Vývoj digitálnej knižnice interdisciplinárnych STEAM projektov a jej implementácia do informatického, matematického a prírodovedného vzdelávania na stredných školách.

BIBLIOGRAFICKÉ ODKAZY

- [1] KALAŠ, I. Informatika na križovatke. Môj názor. In *Medzinárodná konferencia o vyučovaní informatiky, DidInfo 2021* [online]. Banská Bystrica: Univerzita Mateja Bela, Fakulta prírodných vied, Katedra informatiky a Technická univerzita v Liberci, Fakulta prírodovedne-humanitní a pedagogická, 2021. s. 16–24. [cit. 2023-01-10]. Dostupné na internete: http://www.didinfo.net/images/DidInfo/files/DIDINFO_2021_zbornik.pdf.
- [2] VANÍČEK, J. et al. *Programování ve Scratch pro 2. stupeň základní školy*. [online]. 2020. Dostupné na internete: <https://imysleni.cz/ucebnice/programovani-ve-scratchi-pro-2-stupen-zakladni-skoly>.
- [3] ČERNOCHOVÁ, M. et al. *Programování ve Scratch pro pokročilé – projekty pro 2. stupeň základní školy*. [online]. Dostupné na internete: <https://imysleni.cz/ucebnice/programovani-ve-scratchi-ii-projekty-pro-2-stupen-zakladni-skoly>.
- [4] ŠNAJDER, L. Scratch as a glue for funny programming, curiosity and music creation. In *Information and Communication Technology in Education : Proceedings [elektronický zdroj]*. Ostrava: University of Ostrava, 2014. s. 228 – 236. ISBN 978-80-7464-561-7.
- [5] ŠNAJDER, L. Metodika výučby digitálneho spracovania a programovania zvukov. In *Zborník konferencie DidInfo 2011*. Banská Bystrica: Univerzita Mateja Bela, 2011. s. 35 – 48. ISBN 978-80-557-0142-4.
- [6] LOVÁSZOVÁ, G. et al. *Mobilné technológie vo vyučovaní informatiky*. Nitra: Univerzita Konštantína Filozofa, 2016. ISBN 978-80-558-1104-8.
- [7] TKÁČOVÁ, Z. et al. *Zbierka inovatívnych metódik z Informatiky pre 2. stupeň základných škôl a stredné školy*. 1. vyd. Bratislava: Centrum vedecko-technických informácií SR, 2020. 507 s. ISBN 978-80-89965-60-1.
- [8] VOŠTINÁR, P. et al. Ako spestriť výučbu informatiky na školách? In *Didinfo 2020*. Liberec: Technická univerzita v Liberci, 2020. s. 180 – 187.
- [9] PECH, J. et al. *Robotika pro základní školy: programujeme micro:bit pomoci Makecode*. [online]. 2022. Dostupné na internete: <https://imysleni.cz/ucebnice/18-robotika-pro-zakladni-skoly-programujeme-micro-bit-pomoci-makecode>.
- [10] JAKEŠ, T. et al. *Robotika s LEGO Mindstorms pro 2. stupeň základní školy*. [online]. Dostupné na internete:

<https://imysleni.cz/ucebnice/robotika-na-2-stupni-zakladni-skoly-s-lego-mindstorms>.

- [11] GUNIŠ, J. et al. *Riešenie problémov a programovanie* [online]. 1. vydanie. vyd. Bratislava: Centrum vedecko-technických informácií SR, 2020. ISBN 978-80-89965-62-5.
- [12] BUDINSKÁ, L. et al. Microworld for learning programming friendly to blind students. In *TANGNEY, B. et al. Ed. Proceedings of the 2020 Constructionism Conference*. Dublin, Ireland: The University of Dublin, Trinity College Dublin, 2020. s. 403 – 412.
- [13] *Sonic Pi – The Live Coding Music Synth for Everyone*. [online]. 2023. Dostupné na internete: <https://sonic-pi.net/>.
- [14] *Pure Data – Pd Community Site*. [online]. 2023. Dostupné na internete: <http://puredata.info/>.
- [15] *EU Code Week 2021 – Programujeme hudbu a zvuky (11. – 24. 10. 2021)*. [online]. 2021. Dostupné na internete: <https://sound.ics.upjs.sk/2021/>.
- [16] *EU Code Week*. [online]. 2022. Dostupné na internete: <https://codeweek.eu/>.
- [17] *EU Code Week 2022 – Spievajme so Scratchom (11. – 23. 10. 2022)*. [online]. 2022. Dostupné na internete: <https://sound.ics.upjs.sk/2022/>.
- [18] ŠNAJDER, E. *Štúdio Spievajme so Scratchom*. [online]. 2022. Dostupné na internete: <https://scratch.mit.edu/studios/32202418>.
- [19] BRINZIKOVÁ, K. *Analýza projektov EU Code Week 2022 aktivity Spievajme so Scratchom*. [online]. 2023. Dostupné na internete: <https://lnk.sk/ctwq>.
- [20] *Dr. Scratch*. [online]. 2022. Dostupné na internete: <http://www.drscratch.org/>.

PRÍLOHA

Zoznam všetkých projektov:

- PROJEKT 1 – **Matúš_pieseň** (Autor: Nixiy6)
 - <https://scratch.mit.edu/projects/749665326>
- PROJEKT 2 – **Alex_Prešov_pieseň** (Autor: Nixiy16)
 - <https://scratch.mit.edu/projects/749666508>
- PROJEKT 3 – **PO_pieseň** (Autor: Nixiy16)
 - <https://scratch.mit.edu/projects/749752614>
- PROJEKT 4 – **Simonka&Laura** (Autor: 4ZS_Michalovce)
 - <https://scratch.mit.edu/projects/749863643>
- PROJEKT 5 – **Musicamy** (Autor: 4ZS_Michalovce)
 - <https://scratch.mit.edu/projects/749859359>
- PROJEKT 6 – **Rusínska koleda** (Autor: 4ZS_Michalovce)
 - <https://scratch.mit.edu/projects/746149971>
- PROJEKT 7 – **Mining_GKM** (Autor: danil_GGG)
 - <https://scratch.mit.edu/projects/602231727>
- PROJEKT 8 – **Vo vesmíre** (Autor: jurajVav)
 - <https://scratch.mit.edu/projects/749323641>
- PROJEKT 9 – **Tanečnica zo žaby** (Autor: InMaVI)
 - <https://scratch.mit.edu/projects/749028830>
- PROJEKT 10 – **Hudobný koncert** (Autor: Beffsie)
 - <https://scratch.mit.edu/projects/747513448>
- PROJEKT 11 – **Andante koncert** (Autor: MandarinkaDarinka)
 - <https://scratch.mit.edu/projects/747527279>
- PROJEKT 12 – **Twinkle Twinkle Little Star** (Autor: 21lulu21)
 - <https://scratch.mit.edu/projects/749319549>
- PROJEKT 13 – **Billie Jean** (Autor: danokas)
 - <https://scratch.mit.edu/projects/749374077>
- PROJEKT 14 – **Multiinštrumentalista** (Autor: ceceteam)
 - <https://scratch.mit.edu/projects/749029936>
- PROJEKT 15 – **Matrix** (Autor: gabilov)
 - <https://scratch.mit.edu/projects/745448863>
- PROJEKT 16 – **Vystúpenie** (Autor: hrabalka)
 - <https://scratch.mit.edu/projects/749968132>
- PROJEKT 17 – **nočné mesto** (Autor: VEVA148)
 - <https://scratch.mit.edu/projects/748995781>
- PROJEKT 18 – **KLAVÍR** (Autor: Lukasenucko588)
 - <https://scratch.mit.edu/projects/748552238>
- PROJEKTY 19 – 30 sú uvedené v podkapitole 2.4.
- PROJEKT 31 – **Happy b-day!** (Autor: cibereova)
 - <https://scratch.mit.edu/projects/751386054>
- PROJEKT 32 – **Klavír scratch** (Autor: cibereova)
 - <https://scratch.mit.edu/projects/751387505>
- PROJEKT 33 – **Klavír-JK copy remix** (Autor: cibereova)
 - <https://scratch.mit.edu/projects/751389432>
- PROJEKT 34 – **nabrala je jagode petrovke** (Autor: cibereova)
 - <https://scratch.mit.edu/projects/751065225>
- PROJEKT 35 – **Untitled** (Autor: wanda_demjan)
 - <https://scratch.mit.edu/projects/751391996>
- PROJEKT 36 – **Teletabies_by Šteška** (Autor: cibereova)
 - <https://scratch.mit.edu/projects/752051576>
- PROJEKT 37 – **Spievame a hráme so Scratchom** (Autor: RadoZajic)
 - <https://scratch.mit.edu/projects/746542302>
- PROJEKT 38 – **LoPi paradička** (Autor: RadoZajic)
 - <https://scratch.mit.edu/projects/12814519/>

Informatické koncepty pre žiakov nižšieho sekundárneho vzdelávania so záujmom o programovanie

Informatics concepts for lower secondary students with an interest in programming

Monika Tomcsányiová

Katedra didaktiky MFI

Fakulta matematiky, fyziky a informatiky

Univerzity Komenského v Bratislave

Slovenská republika

monika.tomcsanyiova@fmph.uniba.sk

ABSTRACT

In our article, we focus on what other concepts, beyond those directly listed in the iŠVP, should be mastered by students interested in programming. We specifically focus on those students who have already participated, or who want to participate, in the Scratch Cup competition. We examined various foreign curricula and materials that are devoted to programming and searched to see if they include IT concepts that are not included in the Slovak curriculum. Subsequently, we chose two specific concepts, creating new blocks with inputs and cloning, and by analyzing student solutions from the competition, we investigated how difficult these concepts are for students and what types of errors occur in student solutions to competition tasks. Based on the findings, we designed and created two self-study materials for students interested in programming, which will help them understand and use these concepts when solving tasks.

Keywords

Lower Secondary Education. Scratch. Procedures with Parameters. Cloning.

ABSTRAKT

V našom článku sa venujeme tomu, aké ďalšie koncepty, nad rámec tých, ktoré sú priamo uvedené iŠVP, by mali zvládnuť žiaci so záujmom o programovanie. Špecificky sa venujeme tým žiakom, ktorí sa už zúčastnili, resp. ktorí sa chcú zúčastniť, súťaže Scratch Cup. Preskúmali sme rôzne zahraničné kurikula a materiály, ktoré sa venujú programovaniu a vyhľadávali sme, či sú v nich zahrnuté informatické koncepty, ktoré slovenské kurikulum neobsahuje. Následne sme si vybrali dva konkrétne koncepty, vytváranie nových blokov so vstupmi a klonovanie, a analýzou žiackych riešení zo súťaže sme skúmali, nakoľko sú tieto koncepty pre žiakov náročnejšie a aké typy chýb sa vyskytujú v žiackych riešeniach súťažných úloh. Pre žiakov so záujmom o programovanie sme na základe zistení navrhli a vytvorili dva materiály na samoštúdium, pomocou ktorých tieto koncepty pochopia a využijú pri riešení úloh.

Kľúčové slová

Nižšie sekundárne vzdelávanie. Scratch. Podprogramy s parametrami. Klonovanie.

1 ÚVOD

Informatika pre žiakov nižšieho sekundárneho vzdelávania, teda pre žiakov 2. stupňa základnej školy a nižších ročníkov osemročných gymnázií, obsahuje podľa Inovovaného Štátneho vzdelávacieho programu [1] ako jednu z oblastí Algoritmické

riešenie problémov. Súčasťou tejto oblasti je aj vyučovanie programovania pre danú skupinu žiakov a to vo vhodnom programovacom jazyku, ktorý v dokumente [1] nie je priamo uvedený. Dokument obsahuje výkonový a obsahový štandard separátne pre žiakov po ukončení 6. ročníka a následne po ukončení 8. ročníka. Keďže schválený Rámcový učebný plán pre 2. stupeň [2] stanovuje pre nižšie sekundárne vzdelávanie štyri hodiny informatiky, tak v 9. ročníku ZŠ, resp. v kvarte, žiaci informatiku nemajú.

Keď sa pozrieme do obsahového štandardu [1] pre oblasť, do ktorej spadá programovanie, je v ňom uvedené napr. to, že žiak po ukončení 8. ročníka dokáže použiť (programovací) jazyk na popis riešenia problému – aplikovať pravidlá, konštrukcie jazyka, že žiaci majú dokázať aplikovať pravidlá konštrukcie jazyka pre zostavenie postupnosti príkazov, interpretovať postupnosť príkazov a hľadať chybu a opraviť ju. Žiaci majú tiež vedieť riešiť problémy, ktoré vyžadujú známy počet opakovaní (cyklus) a zapísať riešenie takéhoto problému pomocou programovacieho jazyka. Z konštrukcie vetvenia majú žiaci rozpoznať situácie a podmienky, keď treba použiť vetvenie, rozpoznať, aká časť algoritmu sa má vykonať pred, v rámci a po skončení vetvenia. Majú tiež dokázať zostaviť a zapísať podmienku, vyriešiť problémy, ktoré vyžadujú vetvenie s jednoduchou podmienkou (bez logických spojok), zapísať riešenie problému s vetvením pomocou jazyka a interpretovať algoritmy s vetvením. Na konci 8. ročníka majú žiaci vedieť spracovať vstup od používateľa: majú rozpoznať situácie, keď treba čakať na vstup, zapísať algoritmus, ktorý reaguje na vstup, interpretovať zapísané riešenie, vytvoriť hypotézu, ako neznámy algoritmus spracuje zadaný vstup. Podobne vymedzený vzdelávací obsah nájdeme v článku [3]. V ňom sa pre žiakov vo veku viac ako desať rokov okrem iného uvádza, že žiaci môžu využiť modularitu na zavedenie nových slov (príkazov) do používaného programovacieho jazyka, teda že žiaci majú dokázať vytvoriť podprogramy. V slovenskom iŠVP [1] nie je explicitne uvedené, že žiaci majú dokázať navrhovať a používať pomenované postupnosti príkazov (podprogramy). Jedným z ďalších dôležitých konceptov informatiky je práca s parametrami. V článku [3] autori uvádzajú, že parametre sú silným konceptom programovania a aj to, že pre žiakov je to jeden z najväčších skokov vo výučbe programovania. Tento koncept navrhujú implementovať pre žiakov od dvanásť rokov. V slovenskom iŠVP [1] sa tento koncept vôbec neuvádza. V [4] uvádzajú prehľad národných vzdelávacích programov vo viacerých krajinách. Informácie v materiáli sú zaujímavé spracované a uvádzajú, že v tom čase, v roku 2012, väčšina štátov, okrem Slovenska, Poľska a Slovinska deklaruje

znižovanie počtu hodín, ktoré sú venované vyučovaniu digitálnej gramotnosti a navyšovanie počtu hodín venovaných informatike, resp. computer science (CS). Skúmaním rôznych kurikul autorí článku okrem iného zistili, že pri interpretácii vzdelávacích programov môže byť problematická aj terminológia, ktorá sa používa v súvislosti s informatikou a ktorá môže mať v jednotlivých krajinách mierne odlišný význam. Je potrebné uvedomiť si, že stratégie implementované rôznymi krajinami tiež úzko súvisia s ich politickými a vzdelávacími systémami. Zaujímavá je implementácia kurikula univerzity v Cambridge, pozri [5], ktorá ponúka svoje vlastné kurikulum a bohatú podporu učiteľov vo forme školení a materiálov. Jednou z oblastí „computing“ kurikula je aj programovanie, ktoré „pomáha študentom pochopiť konštrukcie programovacích jazykov a porozumieť prínosu informatikov do bežného života“. Okrem samotného kurikula je dôležité, aby mal učiteľ k dispozícii vhodné materiály. Množstvo zaujímavých zadaní pre prostredie Scratch pre žiakov nižšieho sekundárneho vzdelávania nájdeme v [6]. Úlohy pre prvý stupeň sú na stránkach [7], pričom Scratch sa v tomto prípade používa aj na rozvoj matematických schopností žiakov.

2 ŽIACI SO ZÁUJOMOM O PROGRAMOVANIE

Obsahový a výkonový štandard uvedený v iŠVP učiteľ na hodinách informatiky realizuje pomocou vhodných úloh a aktivít s dôrazom na to, aby všetci žiaci v triede dosiahli úroveň určenú v dokumente [1]. Učiteľ má však takmer v každej triede aj žiakov, ktorí počas hodín prejavujú zvýšený záujem o daný predmet, v našom prípade o záujem informatiku a programovanie. Učiteľ by mal pre týchto žiakov pripraviť nielen úlohy, ktoré rieši so všetkými žiakmi, ale aj ďalšie činnosti, napr. náročnejšie zadania. Žiakov so záujmom o programovanie učiteľ môže zapájať do súťaží alebo im navrhuje, aby vytvárali rôzne komplexnejšie projekty, pomocou ktorých sa zdokonalia v programovaní. Pre týchto žiakov je určená aj súťaž Scratch Cup.

Aké informatické koncepty a programátorské konštrukcie jazyka Scratch, nad rámec iŠVP, by mali zvládnuť žiaci so záujmom o programovanie, aby dokázali vyriešiť úlohy súťaže? Vhodnosť nami navrhovaných nových konceptov odôvodníme analýzou riešení úloh súťaže Scratch Cup. V rámci prieskumu podľa úspešnosti úloh určíme aj náročnosť dvoch konkrétnych konceptov, ktoré sú nad rámec súčasného iŠVP [1].

3 SÚŤAŽ SCRATCH CUP

Súťaž Scratch Cup je určená pre žiakov nižšieho sekundárneho vzdelávania. Súťaž je organizovaná už od roku 1997, avšak žiaci počas jej existencie používali rôzne programovacie prostredia. Všetky tieto prostredia boli založené na programovacom jazyku Logo: v rokoch 1997 až 2007 sa programovalo v prostredí Comenius Logo, od roku 2004 do roku 2020 mohli žiaci riešiť zadania aj v prostredí Imagine Logo a od roku 2014 môžu riešiť zadania aj v prostredí Scratch. Súťaž má jedno kolo, v ktorom žiaci počas deväťdesiatich minút riešia tri súťažné úlohy, pozri [8]. Prvá úloha je venovaná kresleniu korytnačou grafikou. Súčasťou tejto úlohy je aj návrh a implementácia podprogramov s parametrom. V druhej úlohe žiaci navrhujú animácie, do ktorých doplnia program. Pri riešení tejto úlohy žiaci využívajú zmeny kostýmov a tiež posielanie správ. Tretia úloha je komplexný program, v ktorom musia žiaci použiť viaceré informatické koncepty a konštrukcie programovacieho jazyka Scratch. Jej riešenie často vyžaduje klonovanie postáv.

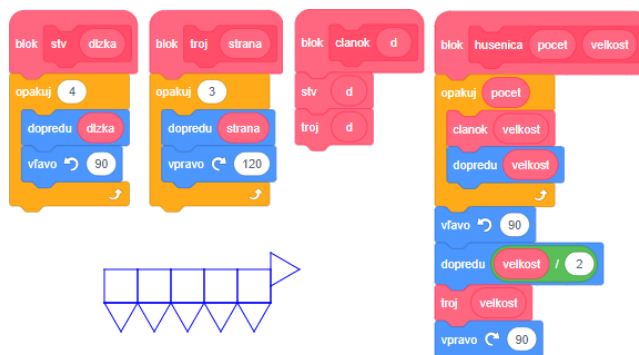
4 NOVÉ KONCEPTY V SÚŤAŽI

Informatické koncepty, ktoré majú zvládnuť žiaci v nižšom sekundárnom vzdelávaní podľa iŠVP obsahujú postupnosť príkazov, cyklus aj podmienky. Ďalším dôležitým konceptom programovania je navrhovanie a vytváranie podprogramov (v prostredí Scratch sa používa pojem vlastný blok, ale ďalej v článku budeme používať všeobecný pojem podprogram). Tento koncept môže byť pre žiakov so záujmom o programovanie prezentovaný ako podprogramy bez parametrov alebo aj ako podprogramy s parametrom, prípadne s viacerými parametrami (v terminológii Scratch sa pre parameter používa názov vstup, ale kvôli konzistencii s bežnou informatickou terminológiou budeme používať slovo parameter). V rámci skúmania žiackych riešení súťaže budeme zisťovať, či žiaci dokážu správne používať podprogramy s parametrom, resp. akým spôsobom žiak rieši úlohu, ak nerozumie parametrom v podprogramoch.

Druhým, podľa nás dôležitým, programátorským konceptom pre žiakov so záujmom o programovanie, je pochopenie a využívanie princípov objektovo-orientovaného programovania. Tento princíp je v prostredí Scratch implementovaný tak, že programátor navrhne tzv. prototyp postavy, následne tomuto prototypu nastaví vlastností (napr. kostým, pozíciu, smer, atď.) a pripraví preň scenár. Počas behu programu potom prototyp tzv. klonuje (v informatickej terminológii by sme povedali, že z navrhutej triedy vytvára objekty). Treba si však uvedomiť, že prototyp nie je v zmysle OOP plnohodnotnou triedou. Pri klonovaní postavy vznikne nový objekt, ktorý existuje výhradne len počas behu programu a ktorému sa dajú programom meniť (niektoré) jeho vlastnosti. V prostredí Scratch však už nie je možné používať ďalšie koncepty OOP, ako napr. zapúzdrenie, či polymorfizmus. Dedičnosť pre klony funguje automaticky, teda klony dedia kostým, pozíciu, natočenie a ďalšie vlastnosti prototypu. V ďalšej časti článku sa budeme venovať vyššie uvedeným dvom konceptom a budeme zisťovať, ako žiaci riešili konkrétne úlohy, v ktorých sú tieto koncepty obsiahnuté.

4.1 Podprogramy s parametrami

Jednou z úloh súťaže Scratch Cup 2020 bolo zadanie Húsenica. Text zadania žiakom presne určoval, že majú vytvoriť podprogramy s parametrom na kreslenie štvorca, trojuholníka, článku húsenice a celej húsenice. Prvé tri podprogramy bolo potrebné navrhnuť s jedným parametrom, ktorý určoval dĺžku strany pre kreslený útvar. Pri kreslení článku húsenice mali žiaci využiť podprogramy pre štvorec a trojuholník.



Obrázok 1: Vzorové riešenie úlohy Húsenica

Podprogram húsenica bolo potrebné navrhnuť s dvoma parametrami. Prvý z nich určoval počet článkov a druhý dĺžku strán pre štvorec a trojuholník v každom článku húsenice. Na obrázku 1

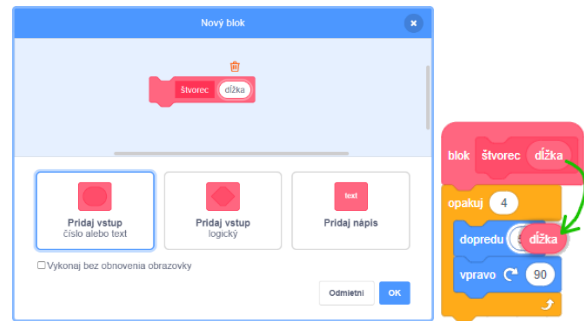
vidíme vzorové riešenie úlohy aj tiež čiarovú kresbu húsenice nakreslenú príkazmi korytnačej grafiky, ktorá mala byť výsledkom programu.

Súťaže Scratch Cup 2020 sa zúčastnilo celkom 217 žiakov, úlohu Húsenica začalo riešiť 154 žiakov. Úplne správne riešenie malo 23 súťažiacich. Ďalších 18 žiakov malo správne časť riešenia, ktorá sa týkala práce s parametrami v podprogramoch, ale títo žiaci nedokázali dokončiť riešenie tak, aby nakreslili požadovaný obrázok húsenice. Pri skúmaní žiackych riešení sme tiež zistili, že ďalších 31 riešení síce nakreslí požadovaný obrázok húsenice, ale parametre v podprogramoch žiaci vo svojich riešeniach buď vôbec nepoužili alebo podprogramy s parametrom síce vytvorili (zrejme pretože to bolo priamo vyžadované zadaním úlohy), ale parametre nedokázali použiť pri použití príslušných podprogramov v postupnosti príkazov. Parametre vo svojich riešeniach títo žiaci najčastejšie nahrádzali globálnou premennou, ktorú použili rôznymi spôsobmi tak, aby nahradila parameter. Z riešení, ktoré vidíme na obrázku 2, môžeme usudzovať, že žiaci zrejme nerozumejú na čo je parameter určený a akým spôsobom by ho mali vo vnútri podprogramu použiť. Keďže cieľom bolo nakresliť výsledný obrázok, žiaci pri testovaní svojho riešenia hľadali iný spôsob, ktorý by zabezpečil prenos veľkosti strán medzi jednotlivými procedurami. Tým bolo použitie globálnej premennej. Konkrétne riešenie na obrázku 2 ukazuje aj „chybu“ prostredia Scratch, ktoré nevyžaduje, aby mal podprogram pri použití určenej parametre a teda dovoľuje žiakovi nechať políčko pre vstup prázdne tak, ako to vidíme v hlavičke bloku článok, resp. pri použití blokov štvorec a trojuholník.



Obrázok 2: Chyby pri používaní parametrov

Z preskúmaných žiackych riešení úlohy Húsenica teda usudzujeme, že hoci žiaci dokážu nakresliť požadovaný obrázok pomocou podprogramov, je pre nich oveľa náročnejšie navrhnuť podprogram s vhodným parametrom a správne tento podprogram použiť v inom podprograme. Myslíme si však, že žiaci so záujmom o programovanie by sa mali (napr. v rámci mimoškolskej činnosti alebo samoštúdia) naučiť vytvárať a správne a efektívne používať parametre v podprogramoch. Keďže ide o koncept nad rámec súčasného iŠVP, učiteľ by mohol dať týmto žiakom napr. vhodné úlohy aj so vzorovými riešeniami a následne ďalšie úlohy na precvičovanie konceptu práce s parametrami. Pre žiakov, ktorí majú záujem zúčastniť sa súťaže Scratch Cup, sme pripravili niekoľko ďalších úloh, ktoré sú vhodné ako úvodné zadania pre prácu s podprogramami, pozri [9]. Žiaci si v tomto materiáli najprv pripomenú vytváranie a použitie nových blokov bez parametrov. Potom sa pomocou návodu krok po kroku oboznámujú s pojmom vstupu a jeho vytvorením pre nový blok. Materiál tiež obsahuje ukážku, ako novému bloku v prostredí Scratch určiť, že bude mať vstup a ako tento vstup pomenovať a použiť ho vo vnútri podprogramu, pozri obrázok 3.

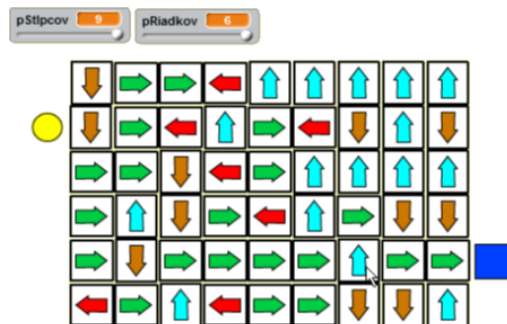


Obrázok 3: Vytvorenie nového bloku so vstupom a ukážka použitia vstupu pre základný blok dopredu

Následne materiál obsahuje niekoľko jednoduchších aj náročnejších gradovaných úloh. Keďže predpokladáme, že žiaci so záujmom o programovanie pochopia tento koncept pomerne rýchlo, úlohy gradujú rýchlejšie, ako v iných materiáloch. Tento materiál obsahuje aj viaceré zadania zo súťaže Imagine Logo Cup, v ktorej sa pri riešení úloh (do roku 2020) používalo prostredie Imagine Logo, pozri [10]. Zadania úloh sme však preformulovali a upravili tak, aby využívali terminológiu prostredia Scratch. Je celkom možné, že žiaci so záujmom o programovanie by boli schopní porozumieť aj zadaniam, ktorých text obsahuje namiesto slov „blok“, či „vstup“, všeobecnejšie slová z programovania ako „procedúra“, či „parameter“. Domnievame sa však, že pochopiť takto formulované zadanie by mohlo byť pre žiakov kognitívne náročnejšie a niektorým z nich by to mohlo prekážať pri pochopení úlohy a jej následnom riešení.

4.2 OOP v prostredí Scratch

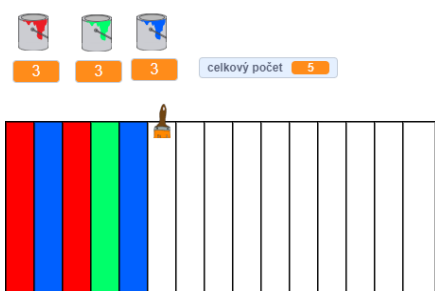
Sú žiaci sekundárneho vzdelávania schopní pochopiť a používať OOP pomocou klonovania postáv? Je koncept vytvárania prototypov a ich klonovania, tak ako je implementovaný v prostredí Scratch, dostatočne zrozumiteľný? Pre analýzu toho, či žiak rozumie a vie využívať klonovanie je potrebné si uvedomiť špecifiká zadania, v ktorého riešení je potrebné použiť klonovanie a ako sa toto zadanie líši od zadaní, v ktorom stačí vytvárať postavy duplikovaním. V súťaži Scratch Cup je koncept klonovania súčasťou zadaní (väčšinou tretej úlohy) hneď od jej pilotného ročníka v roku 2014, pozri obrázok 4.



Obrázok 4: Vytvorenie štvorcovej hracej plochy podľa hodnôt posúvačov v zadaní Cesta (Scratch Cup 2015)

Text zadania úlohy Cesta priamo určoval, že hráč môže pred štartom programu zmeniť počet riadkov a počet stĺpcov pomocou premenných, ktoré sú zobrazené ako tzv. posúvače. Je zrejme, že takéto zadanie vlastne nie je možné riešiť bez využitia klonovania.

Na zistenie toho, ako náročné je klonovanie v prostredí Scratch sme analyzovali žiacke riešenia úlohy Mosty (Scratch Cup 2020). Zistili sme však, že túto úlohu malo úplne správne vyriešenú iba 6 žiakov a ďalší 3 žiaci ju mali vyriešenú takmer správne, pričom súťaže sa zúčastnilo 217 žiakov. Úlohu ani nezačalo riešiť 177 žiakov a ďalších 15 žiakov z nej získalo iba 0 alebo 1 bod, čo vlastne znamená, že aj týchto žiakov môžeme pripočítať k tým, ktorí úlohu neriešili. Malý počet riešení niektorej (väčšinou najnáročnejšej) úlohy súťaže môže mať v konkrétnom ročníku súťaže rôzny dôvod. Jedným z nich môže byť to, že žiaci popri riešení zvyšných dvoch úloh nemali dostatok času, aby riešili a vyriešili aj túto úlohu. Ďalším dôvodom by mohlo byť, že žiaci nedokážu použiť klonovanie, ktoré je pre správne riešenie úlohy nevyhnutné. Na základe nízkeho počtu riešení v ročníku 2020 sme nechceli rozhodnúť o tom, či je vhodné zaradenie klonovania do rozšíreného kurikula informatiky pre žiakov so záujmom o programovanie. Preto sme preskúmali riešenie úlohy Natieranie plota z ročníka Scratch Cup 2021, pozri obrázok 5.



Obrázok 5: Úloha Natieranie plota (Scrach Cup 2021)

Pri prvom preskúmaní počtu riešení sme zistili, že túto úlohu riešilo oveľa viac žiakov a aj správnych riešení bolo podstatne viac ako pri úlohe Mosty. V roku 2021 sa súťaže zúčastnilo 185 žiakov, zadanie tretej úlohy Natieranie plota malo správne vyriešené 12 žiakov a takmer správne riešenie malo ďalších 14 súťažiacich. Keďže úloha mala vopred pevne stanovený, a to dokonca pomerne malý, počet konkrétnych objektov, jej riešenie mohlo viesť žiakov k vytváraniu postáv duplikovaním aj klonovaním. Pre určenie toho, či žiaci rozumejú technike klonovania postáv a vedia ju využiť pri riešení úloh nás ďalej zaujímalo, koľkokrát žiaci, ktorí úlohu iba začali riešiť, vo svojich programoch použili duplikovanie a koľkokrát klonovanie. Prezretím žiackych riešení sme zistili, 32 žiakov použilo klonovanie. Duplikovanie použilo vo svojom riešení 27 žiakov. Zaujímavé na odovzdaných riešeniach tejto úlohy je to, že úlohu začalo riešiť až 103 súťažiacich, čo je v porovnaní s predchádzajúcim ročníkom oveľa vyšší počet. Uvedomujeme si, že koncept klonovania, teda vytváranie objektov v budúcnosti (programom), je pre žiakov náročný. Aj napriek tomu však veríme, že pre tých žiakov, ktorí majú o programovanie záujem, môže byť dostatočne pochopiteľný. Pri príprave žiakov na súťaž môže učiteľ okrem zadaní z predchádzajúcich ročníkov súťaže použiť aj materiál [11]. Materiál obsahuje aj pracovný list, ktorý podrobne popisuje vytváranie štvorcovej siete. Riešením tejto úlohy, ktorá je rozdelená na podúlohy sa žiaci môžu naučiť pomerne rýchlo a rutinne vytvárať klony postavy, ktoré sú pravidelne rozmiestnené v mriežke. Túto techniku môžu potom následne použiť pri riešení rôznych úloh a to nielen v samotnej súťaži Scratch Cup.

5 ĎALŠIE INFORMATICKÉ KONCEPTY

Existujú ešte aj ďalšie informatické koncepty, o ktorých nevieme na základe preskúmaných riešení súťažných úloh usúdiť, či ich žiaci nižšieho sekundárneho vzdelávania so záujmom o programovanie zvládnu pochopiť a používať? Doteraz sme v súťaži nemali zadanie, ktoré by od žiakov explicitne vyžadovalo prácu s údajovou štruktúrou zoznam a ani také, v ktorom by bolo potrebné použiť tzv. vlastné premenné klonovaných postáv. Sú tieto koncepty aj pre žiakov so záujmom o programovanie náročné natoľko, že (zatiaľ) nemá význam pripravovať úlohy a ďalšie materiály, pomocou ktorých by sa s nimi títo žiaci oboznámili? Bude to potrebné niekedy v budúcnosti? V súčasnosti sa tvorí nový obsahový a výkonový štandard pre informatiku nižšieho sekundárneho vzdelávania. Problémom podobných reforiem však býva, že odhadovanie vhodnosti, resp. náročnosti informatických konceptov, ktoré štandard určí pre žiakov, nie je také priamočiare a jednoduché, ako by sa na prvý pohľad mohlo zdať. Z výsledkov výskumu predchádzajúcich štandardov by však malo byť možné určiť, či a nakoľko prevziať obsahový štandard, resp. čo do neho doplniť, alebo naopak, ktoré koncepty z neho vynechať. Bude práca s údajovou štruktúrou zoznam konceptom v štandarde pre všetkých žiakov. Alebo autori pripravovanej reformy usúdia, že tento koncept je pre žiakov nižšieho sekundárneho vzdelávania príliš náročný?

BIBLIOGRAFICKÉ ODKAZY

- [1] Inovovaný štátny vzdelávací program pre nižšie stredné vzdelávanie – 2. stupeň ZŠ. Dostupné tiež na: <https://www.minedu.sk/data/att/22648.pdf>
- [2] Rámcový učebný plán. Dostupné tiež na: <https://www.minedu.sk/ramcove-ucebne-plan/>
- [3] DAGIENĚ, V., HRONKOVIČ, J., LACHER, R. Designing Informatics Curriculum for K-12 Education: From Concepts to Implementations. In *Informatics in Education*. 2021, vol. 20, no. 3, p. 333-360. Also available at: <https://files.eric.ed.gov/fulltext/EJ1320406.pdf>
- [4] GUERRA, V., KUHN, B., BLÖCHLIGER, I. Informatics at school - Worldwide: An international exploratory study about informatics as a subject at different school levels, 2012, 67 p. Also available at: https://redmine.lusy.fri.uni-lj.si/attachments/download/605/Informatics%20at%20school%20-%20Worldwide_def.pdf
- [5] Cambridge Lower Secondary Computing. Also available at: <https://www.cambridgeinternational.org/programmes-and-qualifications/cambridge-lower-secondary/curriculum/computing/>
- [6] BRENNAN, K. Creative Computing a design-based introduction to computational thinking. 2021. Also available at: <https://scratched.gse.harvard.edu/resources/scratch-curriculum-guide-draft.html>
- [7] Education Endowment Foundation. (2018). ScratchMaths: Projects. Retrieved from Also available at: <https://www.ucl.ac.uk/ioe/research/projects/ucl-scratchmaths>
- [8] Súťaž Scratch Cup. Dostupné tiež na: <http://edu.fmph.uniba.sk/ScratchCup/>
- [9] Príprava na Scratch Cup: Nové bloky. Dostupné tiež na: https://bit.ly/SCup_Nove_bloky

[10] Prostredie Imagine Logo. Dostupné tiež na:
<https://imagine.input.sk/>

[11] Príprava na Scratch Cup: Nové bloky. Dostupné tiež na :
https://bit.ly/SCup_Klonovanie

Vzdělávání budoucích strojních a průmyslových inženýrů v oblastech důležitých pro Industry 4.0

Educating future mechanical and industrial engineers in areas relevant to Industry 4.0

Zdeněk Ulrych

Katedra výpočetní a didaktické techniky, ZČU
Klatovská tř. 51
306 14 Plzeň
Česká republika
ulrychz@kv.d.zcu.cz

Miroslav Malaga

Katedra průmyslového inženýrství a managementu, ZČU
Univerzitní 8
306 14 Plzeň
Česká republika
malaga@fst.zcu.cz

ABSTRACT

The paper describes an approach to teaching algorithmization and programming basics to future mechanical engineers. These students have almost zero knowledge of both algorithmization and programming when they arrive at engineering colleges. STEM programmable building blocks were used to build our models to teach algorithmization and to introduce various aspects of Industry 4.0. Most future mechanical engineers will never be computer scientists, but they have to have at least a basic understanding of the field. The paper describes the students' responses to a questionnaire on this approach to teaching algorithmization and programming fundamentals which we designed specifically for non-computer scientists.

Keywords

Education. Programming. Robotics. Industry 4.0. Fischertechnik, Hardware programming. Industrial engineering. Mechanical engineering.

ABSTRAKT

Článek popisuje zvolený přístup při výuce algoritmicke a základů programování budoucích strojních inženýrů. Tito studenti mají při příchodu na vysokou školu se strojírenským zaměřením téměř nulové znalosti algoritmicke i programování. Z tohoto důvodu byla právě pro výuku algoritmicke a přiblížení vybraných aspektů Industry 4.0 zvoleno využívání STEM programovatelných stavebnic na námi navržených modelech. Je samozřejmě nutné brát v potaz, že budoucí strojní inženýři nikdy nebudou informatici, ale že musí získat alespoň základní principiální přehled i v tomto oboru. Na základě realizovaných dotazníků jsou v článku popsána hodnocení studentů na zvolený přístup k výuce algoritmicke a základů programování řešená právě pro ne informatiky.

Klíčová slova

Vzdělávání. Programování. Robotika. Industry 4.0. Fischertechnik, Programování hardware. Průmyslové inženýrství. Strojní inženýrství.

1 ÚVOD

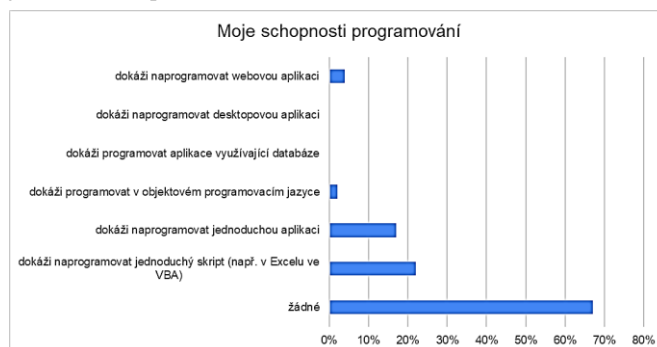
Článek se zabývá výukou budoucích strojních inženýrů na vysoké škole a zaměřuje se na výuku algoritmickeho myšlení s následným navázáním výuky na pochopení fungování vybraných informatických aspektů Industry 4.0. Prvním krokem vedoucím k modernizaci výuky na strojní fakultě ZČU v Plzni bylo

dotazníkové šetření s cílem získat informace o aktuálním stavu informatických a programátorských znalostí studentů při zahájení studia na strojní fakultě. Na základě zjištěného stavu byl zvolen pro výuku STEM přístup s využitím programovatelných STEM stavebnic Fischertechnik.

Koncept STEM vznikl v USA v 90. letech minulého století pro označení vzdělávání v oborech přírodní vědy (Science), techniky (Technology), inženýrství (Engineering) a matematiky (Mathematics) [1]. I v dnešní době je koncept dále rozvíjen a rozšiřován např. na STEAM (A – arts, schopnost tvořit, formulovat, prezentovat), STREAM (R – reading, schopnost čtení ve smyslu schopnosti si nastudovat, vytvořit literature review, aplikovat kritické myšlení, přemýšlet kreativně) či STEAMIE (IE – include everyone, tedy každý může být vzděláván) [2]. Řešení postavených na konceptu STEM je na trhu velké množství, velmi rozšířený je např. vzdělávací jednodeskový počítač BBC Micro:Bit [3], RaspberryPi [4], Arduino [5], Edison robot [6], nebo komplexní stavebnice např. společností Lego [7], nebo Fischertechnik [8].

2 VÝCHODISKA NAVRŽENÉHO ŘEŠENÍ

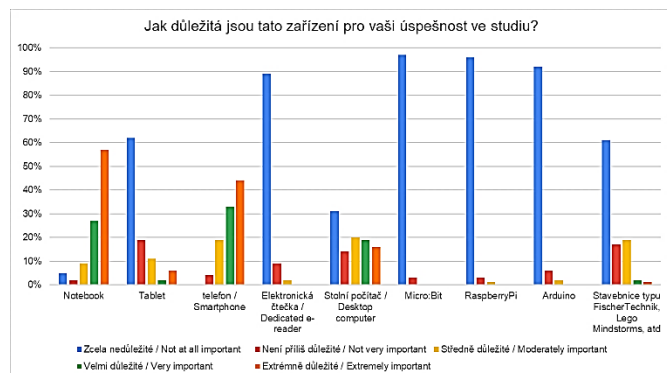
Pro účely zjištění vstupních znalostí studentů strojního inženýrství nastupujících do 1. ročníku jsme realizovali vstupní dotazníkové šetření. V rámci tohoto vstupního dotazníkového šetření jsme získali celkem 123 odpovědí. Z dotazníkového šetření vyplynulo, že 67 % studentů nemá žádné znalosti programování, 22 % studentů dokáže naprogramovat jednoduchý skript např. v MS Excel pomocí VBA (viz. obr. 1) a jen 17 % studentů dokáže naprogramovat jednoduchou aplikaci.



Obrázek 1: Dotazník – schopnosti programování

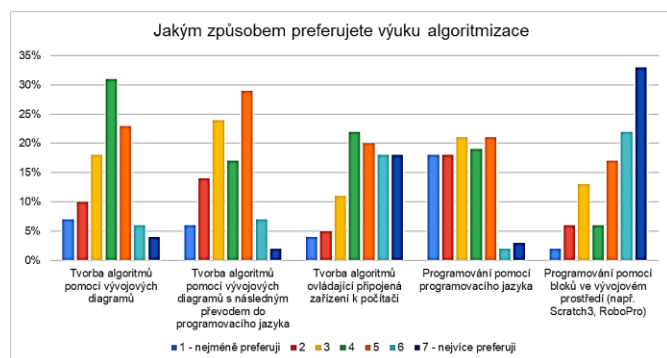
Pokud se podíváme na dotaz (obr. 2), jak vnímají studenti důležitost různých zařízení pro úspěšné absolvování studia na strojní fakultě,

tak považují za nejdůležitější notebook a telefon. Je zajímavé, že sami si částečně uvědomují, že jako budoucí strojní inženýři budou muset také mít nějaké algoritmické myšlení. Z tohoto pohledu jako vhodný nástroj k výuce algoritmizace/programování považují programovatelné stavebnice typu FischerTechnik, Lego Mindstorms atd.



Obrázek 2: Dotazník – důležitost zařízení pro studium

Po absolvování předmětů zabývajících se algoritmizací a programováním v jazyce C#/WPF studenti v dotazníkovém šetření z pohledu výuky algoritmizace sami nejvíce preferují programování pomocí bloků ve vývojovém prostředí (např. Scratch3, RoboPro), viz obr. 3.



Obrázek 3: Dotazník – preference způsobu výuky algoritmizace

3 IMPLEMENTACE STEM DO VÝUKY PRŮMYSLOVÝCH INŽENÝRŮ

Z výše uvedených důvodů proto bylo na fakultě strojní na ZČU navržena od akademického roku 2019/2020 výuka využívající koncept STEM.

3.1 KONCEPT STEM

V rámci vzdělávání strojních inženýrů je potřeba provázat znalosti hlavně matematiky, mechaniky, konstruování, elektrotechniky a elektroniky, informatiky a programování, schopnosti prezentovat a pochopitelně vysvětlit svou práci včetně jejich výsledků, a to vše kreativní cestou. To není v silách jednoho předmětu, jedné katedry ani jedné fakulty, ale je potřeba někde začít. Hlavním cílem je poukázat na souvislosti a návaznosti jednotlivých částí studia a rozšířit schopnosti algoritmizace, programování, analytického myšlení i představivosti v kontextu komplexní problematiky konceptu Industry 4.0.

3.1.1 Fischertechnik education – STEM Engineering

Pro začátek jsme z nabídky stavebnic Fischertechnik zvolili Fischertechnik education – STEM Engineering. Jedná se o polytechnickou stavebnici umožňující stavbu kostry modelů ze stavebních bloků a mechanických součástí (včetně převodovek, ozubených kol apod.), výkonových součástí (elektromotory, krokové motory, ventily, kompresory, USB kamera, optický barevný senzor, světelné moduly...) a součástmi pro sběr/vyhodnocení údajů (např. NTC rezistor pro měření teploty). Stavebnici lze v případě potřeby rozšířit o další mechanické i výkonové součástky. K oživení a řízení modelů pak slouží programovatelná řídicí jednotka a tvorba řídicích programů primárně probíhá ve vývojovém prostředí ROBO Pro, kde se programuje vytvářením vývojových diagramů formou skládání a propojování programovacích bloků. V případě potřeby lze programovat řídicí kód i v jazycích C/C++ [9], nebo Python [10].

Součástí stavebnice je i návod na stavbu modelů automatizovaného skladu, automatizovaného třídění výrobků, robotické ruky, AGV (automated guided vehicle – automaticky řízené vozidlo) včetně řídicích programů. Na obrázku 4 je ukázka automatizovaného skladu.



Obrázek 4: Fischertechnik – automatizovaný sklad [11]

Výhodou do budoucna je schopnost komunikace mezi jednotlivými řídicími jednotkami přes drátové propojení využívající na pozadí I2C sběrnici, popř. v případě programovacích jazyků C/C++ a Python lze u modelů realizovat komunikaci přes standardy WIFI, nebo Bluetooth. Tím se otevírá možnost vytvářet např. modely automatizovaných výrobních hal a s možností stavby modelů a jejich komunikace mezi sebou s ohledem na trendy dnešní doby, tedy požadavky Industry 4.0.

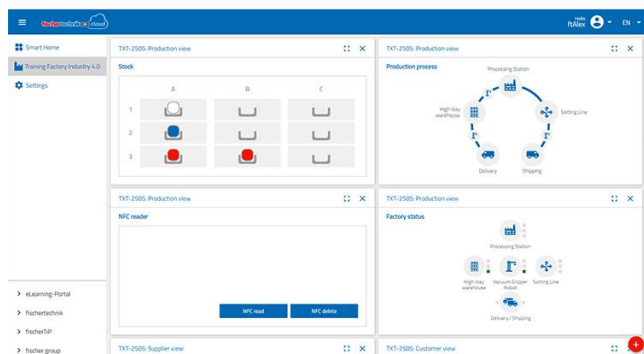
3.1.2 STEM stavebnice a Industry 4.0

V rámci předchozího textu bylo řečeno, že jednotlivé řídicí jednotky/modely stavebnice od Fischertechnik spolu mohou komunikovat. Katedra KPV na ZČU navíc pořídila od výrobce sestavený cvičný model výrobní haly se zaměřením na Industry 4.0, nazvaným Training Factory Industry 4.0, viz obr. 5.



Obrázek 5: Fischertechnik Training Factory Industry 4.0

Model se skládá z robotického otočného ramene s vakuovým systémem pro uchopení/upuštění materiálu a NFC čtečkou (Near Field communication, technologie umožňující bezdrátovou výměnu dat), automatizovaného skladu, obráběcího zařízení, dopravníku, sortovacího zařízení pracujícího na principu detekce barvy a stanice pro hlídání stavu prostředí (hlídání teploty, vlhkosti, tlaku a kvality vzduchu) s otočnou kamerou. Data ze simulace jsou ukládány do cloudu, kde je možné si je prohlížet a pracovat s nimi z pohledu dodavatele, výrobce, nebo zákazníka. Na následujícím obrázku 6 je ukázka dashboardu pro pohled ze strany výrobce, kde je vidět obsazení skladu, výrobní procesy, informace z NFC čtečky a stav továrny.



Obrázek 6: Dashboard Training Factory Industry 4.0 [12]

Model je naprogramovaný v jazyce C/C++ a zdrojové kódy jsou dostupné na githubu [13]. Po důkladném seznámení s modelem a následně zdrojovými kódy je možné upravit si program podle vlastních potřeb. Pokud však vezmeme v potaz, že budoucí strojní inženýři nejsou informatici, tak zvolený přístup výrobce je pro ně z pohledu vlastních úprav a rozšíření modelu nepoužitelný. Z tohoto důvodu je na fakultě stojní vyvíjen vlastní přístup k tvorbě továrny s prvky Industry 4.0 tak, aby zohledňoval informatické znalosti, které získají během studia na strojní fakultě (zaměřeno na ne informatiky) a jejich schopnosti.

3.1.3 Využití stavebnic STEM v praxi strojního inženýra

Využití modelů (jak education stavebnic, tak modelu Training Factory Industry 4.0) je možné jak pro výuku studentů, kdy si vyzkouší proces výroby hned ze tří úhlů pohledu, tedy z pohledu dodavatele (dodání materiálu), výroby (naskladnění materiálu, obrobění, vyskladnění materiálu), i zákazníka (nákup výrobku), tak pro pracovníky výrobních podniků, kterým je možné nasimulovat

potřebné stavy/události a bez nepříjemných faktických následků je připravit na jejich řešení ve skutečné praxi.

Stavebnice Fischertechnik se po prvních odučených semestrech s jejím využitím jeví jako vhodná pro vytváření fyzických modelů a simulací (přinejmenším jejich zjednodušenou variantou), které mohou být zajímavým doplňkem např. pro diskrétní simulaci a softwarovou vizualizaci. Modely a jejich fyzická tvorba včetně naprogramování mohou ukázat slabiny/zranitelnosti skutečného systému, který při vytvoření pouze diskrétní simulace, nebo vizualizace mohou zůstat nepovšimnuty.

Další neméně zajímavou možností je vytváření fyzických modelů (neodkazujících na konkrétní reálné existující systém) a k nim vytváření např. diskrétních simulací pro pochopení souvislosti mezi modelem a simulací.

3.2 Využití stavebnice Fischertechnik

V rámci seminářů, kde se využívají stavebnice Fischertechnik se studenti seznámí s tematickými moderními technologiemi a přístupy. Jsou to hlavně základy kybernetiky, řídicích systémů, sensorové techniky, motorizace, automatizace, robotiky, digitální komunikace a programování. Okrajově pak s mobilní robotikou, konstruováním, mechanickými systémy a návrhem systémů.

Stavebnice obsahuje programovatelnou řídicí jednotku. Program se vytvoří pomocí vývojového prostředí ROBO PRO a do řídicí jednotky (obr. 7) se následně nahraje buď přes USB kabel, nebo bezdrátově (WIFI, Bluetooth).



Obrázek 7: ROBOTICS TXT Controller (řídicí jednotka)

3.2.1 Navržené modely

Při navrhování modelů pro výuku se vycházelo z premisy, že modely musí být rychle realizovatelné, aby je studenti mohli rychle sestavit a věnovat se jejich řízení. Na druhou stranu modely musí být dostatečně složité, aby šla vymyslet rozumná složitost pro logiku řízení těchto modelů.

Pro výuku algoritmického myšlení byly po analýze postupně navrženy následující modely:

- **Větráček** – jedná se o jednoduché zařízení s jedním spínačem a motorem, který otáčí vrtulí. Cílem cvičení je ukázat, jak se programuje řídicí jednotka a propojení řídicí jednotky s počítačem.
- **Model větrné elektrárny** – model obsahuje navíc oproti předchozímu příkladu optické moduly, tedy žárovkou/led a světelný senzor. Cílem cvičení je ukázat větvení v programu.
- **Vysoušeč rukou** – cílem cvičení je vytvořit a naprogramovat světelnou bránu.
- **Manuálně ovládaný pojezd** – slouží pro zopakování vyhodnocování podmínek (větvení programu). Model se

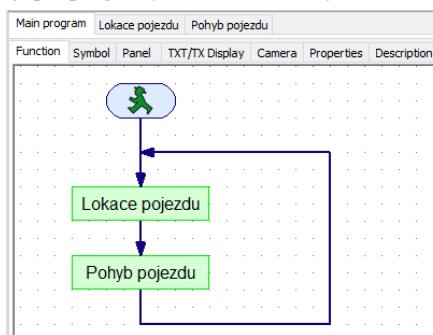
skládá ze dvou tlačítek a motoru. Motor je spojen s převodovkou, která přenáší otáčky na výstupu elektromotoru na ozubené kolečko, které je v kontaktu s ozubeným pásem pevně spojeným s nosníkem.

- **Pojezd mezi pevně danými body** – zde se studenti naučí vytvářet pro program a model grafický ovládací panel, vytvářet a využívat podprogramy, pracovat s elementy typu Command, Panel display, rozhodováním na základě podmínky a zopakují si vyhodnocování podmínek (větvení programu), využijí spínače jako skutečné koncové spínače. Cílem je mít pojezd, který po stisknutí tlačítka na ovládacím panelu (softwarové tlačítko) přesune pojezd na požadovanou pozici. Model se skládá ze dvou tlačítek a motoru. Motor je spojen s převodovkou, která přenáší otáčky na výstupu elektromotoru na ozubené kolečko, které je v kontaktu s ozubeným pásem pevně spojeným s nosníkem.
- **Systém pro průběžné měření teploty se záznamem dat do souboru** – v tomto cvičení se studenti naučí odečítat hodnoty z NTC odporu, který je využitelný pro měření teplot, omezit počet cyklů na požadovanou hodnotu, použít proměnou a pracovat se soubory, konkrétně zapisovat hodnoty do CSV souboru.
- **Počítání otáček obyčejného motoru** – cvičení je zaměřeno na počítání otáček u nekrokových motorů. V nástroji pro programování tedy studenti musí navrhnout takový algoritmus, nebo použít takový element, který dokáže „počítat“ pulzy (spojení/rozpojení) koncového spínače.
- **Jednoduchý obráběcí stůl s pevným natáčením** – v tomto cvičení se studenti naučí řídit krokový motor, který lze využít všude, kde je potřeba buď přesně krokovat pohyb, nebo synchronizovat pohyb více motorů.
- **Synchronizování ukazatelé** – v tomto cvičení se naučí synchronizovat 2 krokové motory.
- **Logický motor** – v tomto cvičení se studenti naučí využívat logické operátory a matematické operace, zopakují si podprogramy a naučí se předávat potřebné hodnoty do podprogramu, a naopak z podprogramu je vracet do programu.
- **Matematika** – v tomto cvičení si ukážeme, že lze provádět i matematické operace a vytvořit program, který např. pouze provádí potřebný výpočet i bez potřeby hw modelu. Studenti si zopakují práci s proměnnými, podprogramy, předávání parametrů do podprogramu a vrácení parametru z podprogramu.
- **Větráček II – návrat k základům a jejich rozšíření** – v tomto cvičení se studenti naučí pracovat s ukazatelem, Slidery (posuvníky), vytvářet GUI pro ovládání modelu na dotykovém displeji řídicí jednotky a nahrání do řídicí jednotky tak, aby program fungoval samostatně bez propojení řídicí jednotky s počítačem. Navíc je ukázáno, že lze zapojovat komponenty paralelně.
- **Kontrola barvy** – v tomto cvičení se studenti naučí vyhodnocovat barvy za pomoci analogového barevného senzoru a zopakují si zavedení proměnné a její zobrazení pomocí Text display, aby si mohli kalibrovat barevný senzor. Toto cvičení je možné vypracovat i při nahrazení optického barevného senzoru webkamerou, jen se v takovém případě musí model i program adekvátně upravit.

- **Pneumatika** – v tomto cvičení je ukázáno, že v jednom programu lze vytvořit libovolný počet funkcí, které se vykonávají současně, ovládání solenoidového ventilu a jednocestného pneumatického válce.
- **Alarm** – v tomto cvičení je ukázáno vyhodnocení pohybu před kamerou, využívání zvuků a jsou zopakovány podprogramy.
- **Vyhodnocení barvy** – v tomto cvičení se studenti naučí zjišťovat převládající barvu v zorném poli kamery a zopakují si práci s podprogramy a matematické a logické operace.
- **Vyhledávání objektu** – v tomto cvičení se studenti naučí vyhledávat objekt pomocí USB kamery, vyloučit prostor, který nechceme pomocí kamery vyhodnocovat a zopakují si práci s text display a dalšími grafickými ovládacími elementy.

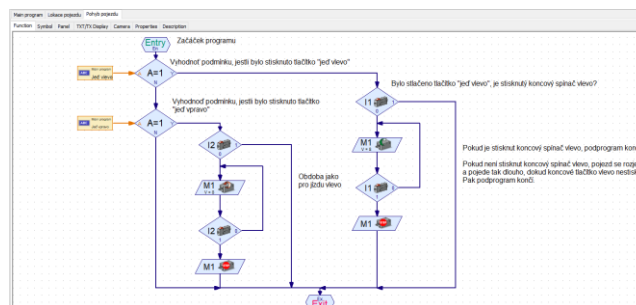
3.2.2 Ukázka vytvořeného programu

Zvolená programovatelná stavebnice Fischertechnik education – STEM Engineering využívá pro naprogramování řídicí jednotky vývojové prostředí ROBO Pro. Zde se samotný algoritmus navrhuje pomocí objektů připomínající vývojové diagramy. Na následujícím obrázku, viz obr. 8, je ukázka těla hlavního programu, kde se volají podprogramy v nekonečném cyklu.



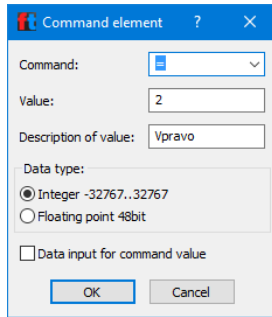
Obrázek 8: Hlavní program s využitím podprogramů

Ukázka jednoho z podprogramů je na následujícím obrázku 9.



Obrázek 9: Podprogram pro pohyb pojezdu

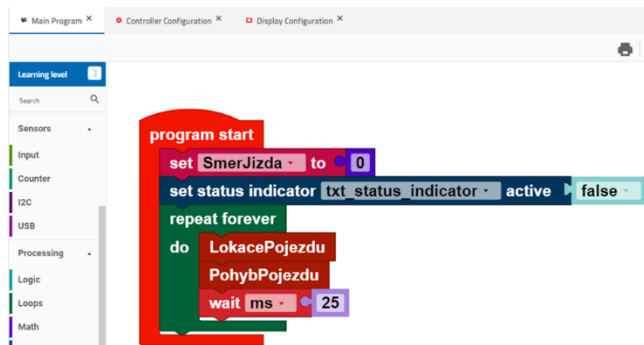
Každý objekt je možné nastavit v samostatném okně (obr. 10).



Obrázek 10: Ukázka nastavení stavu - stisknutí spínače

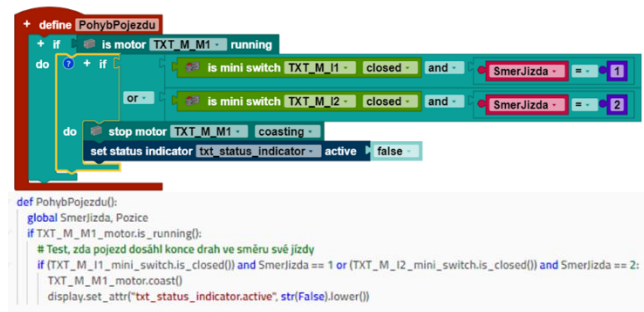
3.2.3 Budoucí úpravy

V roce 2022 společnost Fischertechnik vydala novou řídicí jednotku, kterou je možné programovat pomocí nového vývojového prostředí Robo Pro Coding. Toto vývojové prostředí podporuje tvorbu programu pomocí objektů připomínající Scratch, nebo pomocí programovacího jazyka Python. Z tohoto důvodu se připravují nové výukové podklady využívající toto nové vývojové prostředí. Na následujícím obrázku 11 je znázorněno, jak by vypadalo tělo hlavního programu pro stejný příklad uvedený v předchozí podkapitole.



Obrázek 11: Hlavní program s využitím podprogramů

Na obrázku 12 je ukázka podprogramu pro prohyb pojezdu.



Obrázek 12: Podprogram pro pohyb pojezdu

Druhou možností, jak je možné programovat řídicí jednotku je využít jazyk Python. Jen pro představu je na obrázku 12 také znázorněn podprogram PohybPojezdu v Pythonu.

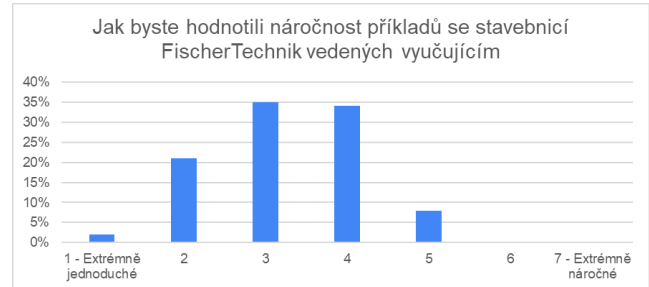
4 DISKUSE VÝSLEDKŮ

Výuka na uvedených modelech probíhala dvojitým způsobem. Některé modely/programy, nebo jen část tvorby modelu/programu

byla vedena vyučujícím, kde vyučující vysvětloval důležité aspekty potřebné pro sestavení fyzického modelu, nebo potřebných objektů pro sestavení programu. Většina hodin však byla řešena badatelsky orientovanou výukou. Studenti pracovali ve skupinách. Badatelská výuka se opírala především o následující badatelské kroky:

- 1) zadání úkolu,
- 2) návržení algoritmu nebo program,
- 3) otestování a upravení algoritmu nebo program,
- 4) prezentování a zhodnocení algoritmu nebo program.

Na konci semestru studenti dostali dotazník, kde se vyjadřovali k proběhlé výuce. Na dotazník odpovědělo 98 studentů, viz obr. 13 a obr. 14.



Obrázek 13: Dotazník - Jak byste hodnotili náročnost příkladů se stavebnicí FischerTechnik vedených vyučujícím



Obrázek 14: Dotazník - Jak byste hodnotili náročnost samostatných příkladů se stavebnicí FischerTechnik

Z dotazníků jasně vyplývá, že podle studentů zvolená náročnost příkladů odpovídala jejich schopnostem a dosavadním znalostem na kterých stavěli a ze kterých vycházeli.

V dotazníku byly také volné odpovědi, kde mohli studenti vyjádřit svůj názor na zvolený způsob výuky.

Na otázku „Ohodnoťte přínosnost výuky algoritmizace pomocí stavebnice FischerTechnik“ odpovídali např. následovně:

- Velmi přínosné.
- Velmi přínosná a záživná
- Myslím si, že přínosnost této výuky je vysoká, neboť si studenti mohou vyzkoušet praktické využití těchto technologií.
- Lze si lépe představit funkce programu
- Určitě zajímavá výuka, neboť student si může vyzkoušet nejen programovat, ale i spouštět, a hlavně fyzicky pozorovat, jak se programovatelný prvek chová.
- Výborná ukázka programování využitě v praxi.
- Hlavní přínos vidím v tom, že něco naprogramuji a hned mohu program spustit a vidím reálné výsledky mé práce

na stavebnici. Není to jen psaní programu ve vývoj. prostředí a výsledkem je pouze výpis do "kolonky".

Na otázku „Co se Vám líbilo na výuce pomocí stavebnice FischerTechnik“ odpovídali např. následovně:

- Práce ve skupinách.
- Poměrně jednoduché a příjemné ovládní, možnost vyzkoušení si praktických příkladů, práce ve skupině.
- Výuka formou hry, vyzkoušení funkčnosti našeho programu.
- Mechanická funkčnost zařízení, které jsme si naprogramovali.
- Relativně přehledné programování.
- Lépe představitelné programování.
- Možnost vyzkoušet si vlastní naprogramování různých operací. Zpestření výuky "hrou".
- Každý ocení, když se mu povede něco "vyrobit" či sestrojít. A s výukou našich učitelů to šlo skoro samo. Líbí se mi styl, jakým to funguje, že to vlastně celé zprovozním sama.
- Líbilo se mi, že člověk vidí svoji práci na vlastní oči, navíc to není jen čistá teorie.

Z odpovědí je patrné, že studenti velmi oceňují provázání programování s hardwarovým modelem. Díky tomuto propojení okamžitě vidí, jak jimi navržené programy ovlivňují chování hardwarového modelu.

Z dotazníků také vyplývá, že náročnost příkladů by mohla být i zvýšena. Z tohoto důvodu předpokládáme, že navrheme komplexnější model, který studenti již nebudou sestavovat. Model by mohl obsahovat řadu různých typů senzorů. Studenti by na tomto předem postaveném komplexnějším modelu programovali logiku, kde by museli řešit vzájemnou interakci různých senzorů a definované požadované logiky.

5 ZÁVĚR

Princip STEM a programovatelné stavebnice, které na jeho základu vznikly jsou nejen cestou, jak studentům umožnit vidět hmatatelné výsledky své práce například na fyzickém modelu automatizovaného skladu, který se chová podle studentem naprogramovaných pravidel, ale hlavně studenty naučí přemýšlet v potřebných souvislostech, systémově a s ohledem na okolí navrženého a oživeného systému. Např. průmyslový inženýr v praxi pak může používat tyto stavebnice/systémy nejen pro tvorbu fyzických modelů a simulací, ale i pro vytváření prvotních návrhů systémů, nebo návrhu řešení problémů reálných systémů. V některých případech je možné projekty postavené na STEM stavebnicích, nebo systémech využít v praxi jako součást/doplňek velkých systémů, např. pro sběr dat.

BIBLIOGRAFICKÉ ODKAZY

- [1] Koncept STEM, Národní ústav pro vzdělávání. *Národní ústav pro vzdělávání* [online]. [cit. 20.02.2023]. Dostupné z: <http://www.nuv.cz/p-kap/koncept-stem>
- [2] What Is STEM? (And STEAM, and STREAM?) – Niche Blog. *Niche: Explore Schools, Companies, and Neighborhoods* [online]. Copyright © 2019

- [3] Niche.com Inc. [cit. 20.02.2023]. Dostupné z: <https://www.niche.com/blog/stem-vs-steam-vs-stream/>
- [4] Micro:bit Educational Foundation | micro:bit. *Micro:bit Educational Foundation | micro:bit* [online]. Copyright © Copyright Micro [cit. 15.02.2023]. Dostupné z: <https://microbit.org/>
- [5] Teach, Learn, and Make with Raspberry Pi – Raspberry Pi. *Teach, Learn, and Make with Raspberry Pi – Raspberry Pi* [online]. Dostupné z: <https://www.raspberrypi.org/>
- [6] Arduino - Home. *Arduino - Home* [online]. Dostupné z: <https://www.arduino.cc/>
- [7] Edison Programmable Robot - Ideal for school classroom education. *Edison Programmable Robot - Ideal for school classroom education* [online]. Dostupné z: <https://meetedison.com/>
- [8] The Lego Learnign System. *Object moved* [online]. Copyright ©2023 The LEGO Group. [cit. 20.02.2023]. Dostupné z: <https://education.lego.com/en-us/>
- [9] STEM Sets - fischertechnik. [online]. Copyright © [cit. 15.02.2023]. Dostupné z: <https://www.fischertechnik.de/en/products/learning/ste-m-kits>
- [10] fischertechnik GmbH GitHub. *The world's leading software development platform · GitHub* [online]. Copyright © 2023 [cit. 10.02.2023]. Dostupné z: <https://github.com/fischertechnik>
- [11] fischertechnik TXT community firmware. [online]. Dostupné z: <https://cfw.ftcommunity.de/ftcommunity-TXT/en/programming/python/tutorial-1.html>
- [12] Studica.com - fischertechnik Education STEM Engineering. [online]. Copyright © [cit. 20.02.2023]. Dostupné z: https://www.studica.com/fischertechnik-education-stem-engineering-519341_5
- [13] Dashboard - fischertechnik. [online]. Copyright © [cit. 10.02.2023]. Dostupné z: <https://www.fischertechnik.de/en/service/elearning/teaching/dashboard>
- [14] TxtSmartFactoryLib: Main Page. [online]. Dostupné z: https://fischertechnik.github.io/txt_training_factory_doc/html/index.html

Sady bobřích úloh se sestavováním programu z bloků

Sets of bebras tasks with assembling programming code from blocks

Jiří Vaníček

Jihočeská univerzita v Českých Budějovicích,
Pedagogická fakulta
Jeronýmova 10
37115 České Budějovice
Česko
vanicek@pf.jcu.cz

Karolína Miková

Univerzita Komenského,
Fakulta matematiky, fyziky a informatiky
Mlynská dolina F1
84248 Bratislava
Slovensko
karolina.mikova@fmph.uniba.sk

ABSTRACT

The article deals with a new type of tasks from the Bebras informatics challenge, in which the student solves a problem by assembling a program from blocks. A module enabling the implementation of such tasks was developed and verified, including four microworlds forming frameworks for programming tasks. In addition to the competition tasks themselves, sets of follow-up tasks compiled into a curriculum introducing the student to programming in a specific microworld have been created. For many of the contestants, it was the first encounter with programming in their lives. The article aims to describe the creation of these sets of tasks with the requirements that they contain elements of familiarization, elements of learning programming, do not lack gradation and avoid errors such as moving forward too quickly. The created and verified sets of tasks and can serve as a motivational tool for pupils and the inexperienced teachers as a guide to the basics of programming

Keywords

Programming. Algorithmization. Block oriented environment. Curriculum. Bebras challenge. iBabor. Programming microworld.

ABSTRAKT

Článek se zabývá novým typem úloh ze soutěže Bobřík informatiky, v nichž žák řeší problém sestavením programu z bloků. Byl vyvinut a ověřen modul umožňující implementaci takových úloh včetně čtyř mikrosvětů tvořících rámce pro programovací úlohy. Kromě samotných soutěžních úloh byly vytvořeny sady navazujících úloh, sestavených do kurikula, seznamujícího žáka s programováním v konkrétním mikrosvětě. Pro řadu soutěžících šlo o první setkání s programováním v životě. Článek cílí na popis vzniku těchto sad úloh s požadavky na přítomnost prvků seznámení, prvků učení se programování, gradaci a absenci chyb např. příliš rychlého postupu vpřed. Vytvořené sady úloh byly ověřeny a mohou učitelé posloužit jako motivační nástroj pro žáky a nezkušenému učitelé jako vodítko pro základy programování.

Klíčová slova

Programování. Algoritmizace. Blokově orientované prostředí. Kurikulum. Bobřík informatiky. iBabor. Programovací mikrosvět.

1 PROGRAMOVACÍ BOBŘÍ ÚLOHY

Jednou z cest rozvíjení inženýrského myšlení je již po řadu let školní soutěž Bebras Challenge [1]. Situační úlohy používané v české verzi soutěže Bobřík informatiky¹ jsou zaměřeny na konkrétní inženýrský koncept. Soutěž probíhá formou online testu, v němž soutěžící odpovídá výběrem jedné z nabízených možností, častěji však interaktivně, označováním objektů nebo manipulací s objekty po ploše.

Situační „bobří“ úlohy rozvíjejí různé aspekty inženýrského myšlení a mezi nimi i algoritmizaci. Typickými algoritmickými úlohami, které se v soutěži objevují, jsou hledání počátečního či koncového stavu po vykonání daného algoritmu, porovnání více algoritmů oproti zadání úlohy, posouzení pravidel k provedení výpočtu, hledání chyby v algoritmu nebo jeho optimalizace. Mezi algoritmickými úlohami doposud chyběly úlohy, na něž se odpovídá sestavením programu, což v soutěži nabídka inženýrských úloh ochuzovalo. Bránily tomu technické možnosti testovací aplikace a s tím související malá nabídka takových úloh v mezinárodní databázi. U jiných druhů úloh, např. grafových nebo kódovacích, je potřebná interaktivita pro vytváření kvalitních úloh na dostatečné úrovni.

Hledali jsme řešení, které by umožnilo do stávajícího systému soutěže implementovat úlohy, v nichž by soutěžící stejně jako v programovacích prostředích používaných na školách mohli sestavovat svůj program z bloků a testovat jej. Jde o běžně používanou koncepci blokového programování, která je přítomná v učebnicích informatiky nebo robotiky a tudíž žákům blízká.

Analyzovali jsme známá bloková programovací prostředí (např. Scratch, Blockly, Makecode, Go Vex) z pohledu jejich využitelnosti pro vytvoření softwarového modulu z pohledu pedagogického a implementačního a vyvinuli jsme modul, který jsme implementovali do soutěžního testu Bobříka informatiky. Rozhodli jsme se vyjít z prostředí Blockly [2], které při naplnění představ o tom, jak budou soutěžící bloky sestavovat, poskytovalo nejsnadnější cestu implementace do software soutěžního testu, v němž musí koexistovat vedle sebe celá řada různých modulů, které musí komunikovat s jádrem testové aplikace bez vzájemného rušení. Blockly bylo ovšem potřeba dovyvinout, aby komunikovalo s testovou aplikací a databází a poskytovalo zpětnou vazbu i pro potřebu případného výzkumu.

¹ www.ibobr.cz

Vyvinutý modul Blockly vizuálně vypadá jako aplet, v němž na levé straně je umístěna nabídka bloků s příkazy, které uživatel vytáhne na programovací plochu uprostřed apletu a připojí je k bloku *Po spuštění*. Napravo v apletu je vizualizace, ekvivalent scény ve Scratch, na němž se odehrává situace, která je programována, a ovládací prvky spuštění programu (Obrázek 1).

Při běhu sestaveného programu modul kontroluje pravidla (např. opuštění hrací plochy postavou, náraz do překážky apod.) a po jeho skončení splnění podmínek zadání úlohy (např. maximální povolený počet bloků a podle povahy úlohy počet ušlých kroků, dosažení cílové pozice, sesbírání předmětů, vykreslení daného obrázku, pohyb po požadované trajektorii, výpis správných hodnot atd.) V zadáních různých úloh lze jednotlivé bloky skrývat.

2 VYTVOŘENÉ MIKROSVĚTY

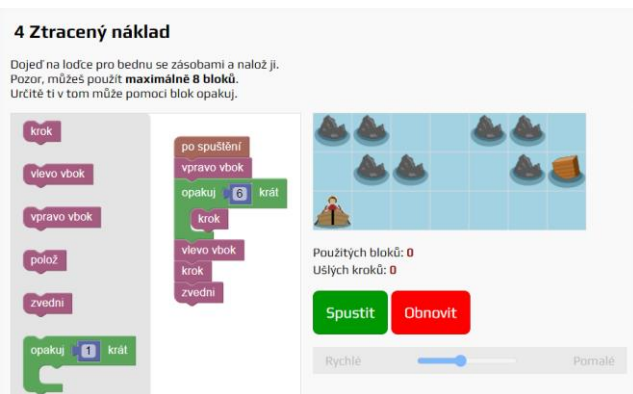
K tomuto softwarovému prostředí jsme vyvinuli čtyři šablony programovacích úloh. Každá z nich měla naplňovat roli Papertova mikrosvětů, který si žák vytváří a který mu usnadňuje vstup a orientaci v programování. Dbali jsme na naplnění vlastností mikrosvětů, jako jsou omezené sady základních příkazů, jasná okamžitá zpětná vazba, nízký práh a vysoký strop [3], tedy rychlý vzhled do problematiky a rychlý počáteční pokrok. Ke každé šabloně daného mikrosvětů je možno pouhou úpravou vstupních dat vytvářet různé konkrétní programovací úlohy, navzájem odlišné situací, náročností nebo grafikou.

Zde stručně představíme vytvořené mikrosvěty:

2.1 Robot Karel

Mikrosvět robota Karla [4] má ústřední postavu, která se podle programu pohybuje po čtvercové síti a přitom sbírá a pokládá předměty (které nelze programovat). Základní sada pěti bloků umí přesouvat postavu na sousední pole hrací plochy, otáčet jí oběma směry o pravý úhel, odebrat předmět z pole, na němž se postava nachází, nebo na prázdné pole předmět položit. Postava dále umí detekovat předmět na poli, na němž stojí, a překážku pohybu na poli před sebou. K vykonávacím blokům byly přidány bloky *Opakuj*, představující cyklus s pevným počtem opakování, *Opakuj dokud* jako cyklus s podmínkou ukončení, *Když* jako rozhodovací blok a blok podmínky umožňující detekci objektu nebo překážky.

Typickými úlohami v tomto prostředí byly dojet na dané místo, vyhnout se překážce, sesbírat (pravidelně rozmístěné) objekty nebo najít cestu, skládající se ze stejných opakujících se částí. Postavy mají tvar člověka, bobříka, robota, auta, lodí apod.



Obrázek 1: Vzhled modulu Blockly v mikrosvětě Robot Karel. Uprostřed je sestavený program připravený ke spuštění jako žakovské řešení dané úlohy.

2.2 Film

Pro druhý mikrosvět jsme se inspirovali v prostředí Blockly games - Movie [5], v němž postava animuje svůj pohyb v čase, mění své souřadnice a velikost (Obrázek 2). Ovládá ji jediný základní příkaz *Postava* se čtyřmi parametry, který postavu umístí na *pozice x, y* (v rozsahu 0–100, který má hrací plocha v obou směrech). Dalšími dvěma parametry jsou *velikost* postavy (hodnota 100 odpovídá rozměrům hrací plochy) a *natočení* postavy proti základnímu svislému směru.

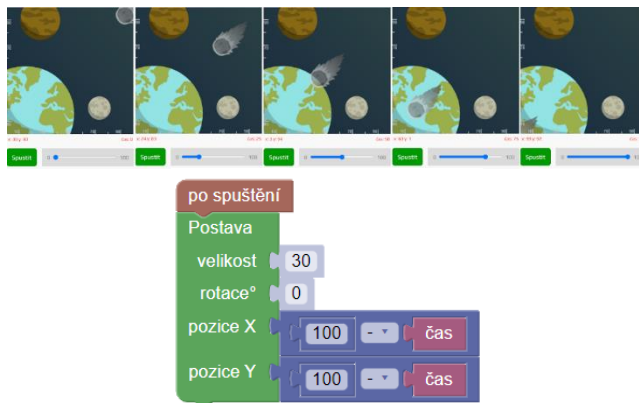
Jako hodnota parametru se kromě číselných může použít proměnná *čas*, která se po spuštění programu plynule mění od 1 do 100. Strukturu každého programu v tomto prostředí lze popsat takto: Pro *čas = 1* do *100* (vykonej blok *Postava*), v němž žák v podstatě pouze nastavuje parametry pro blok *Postava*. Použití proměnné *čas* jako hodnoty parametru umožňuje dynamiku a tím animaci postavy během vykonávání programu.

Pokud např. použijeme proměnnou *čas* jako parametr *pozice x*, bude se souřadnice *x* plynule měnit od 0 do 100 a objekt se bude rovnoměrně pohybovat zleva doprava. Hrací plocha má v obou směrech rozměr 100, tak aby postava během své animace přešla přes celou plochu.

Blok *Postava* byl v nabídce bloků doplněn blokem pro vytváření matematických výrazů s početními operacemi $+$, $-$, $*$, $/$, $^$ pro náročnější programování pohybů (např. při chůzi zprava doleva se *pozice x* zmenšuje, potřebujeme parametr $100 - \text{čas}$). Dále byl doplněn blokem pro větvení *Když (- jinak)*, kontrolující podmínku času (např. zda čas překročil určitou hodnotu) a umožňující nastavit pro různé časové intervaly různé sady parametrů bloku *Postava* a tím skládat více jednoduchých pohybů za sebou do složitějšího (např. postava se chvíli pohybuje vodorovně a poté svisle).

Typickou úlohou je nastavit parametry bloku *Postava* tak, aby postava vykonala stejnou animaci jako vzorový objekt, tzv. stín postavy. Animaci tohoto stínu lze spustit opakovaně nebo ji lze pomocí posuvníku nastavit do libovolného času.

První dva mikrosvěty Robot Karel a Film byly popsány v [6].



Obrázek 2: Rozfázovaná animace na scéně mikrosvětů Film. Na obrázcích letí šedý stín asteroidu, jehož pohyb je třeba naprogramovat (dolní část obrázku ukazuje správné řešení, *pozice x* i *y* asteroidu se s rostoucím časem zmenšují).

2.3 Želva

Třetí mikrosvět je tradiční želví grafika [7], obdobná rozšíření *Pero* ve Scratch [8]. Postava se pohybuje ve středové souřadné soustavě se středem soustavy v místě, kde se postava nachází, a s hlavním směrem ve směru svého aktuálního pohybu. Její obrázek ukazuje směr, do něhož je postava natočena (Obrázek 3).

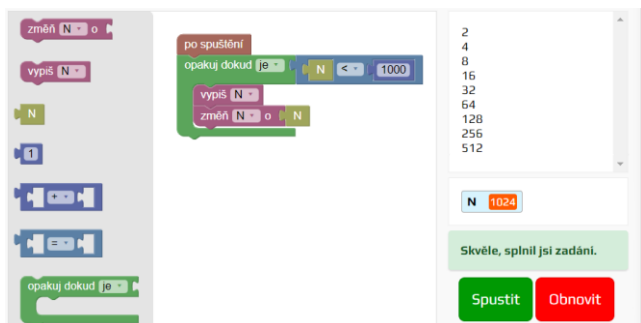
Základní sada bloků obsahuje pohyb vpřed (o počet kroků – pixelů) a otočení (oběma směry ve stupních), dále zapnutí a vypnutí kreslicího pera. Je doplněna blokem *Opakuj* s pevným počtem opakování. Parametry pro bloky pohybu nelze editovat libovolně, hráč vybírá ze sady nabízených čísel v rozbalovacím menu, které jsou v různých úlohách různé. Typickou úlohou je sestavit program, který nakreslí stejný obrázek, jako je v zadání nakreslen na scéně šedou barvou.



Obrázek 3: Scéna mikrosvětů Želva s ornamentem k nakreslení a sestaveným programem před spuštěním

2.4 Proměnné

Poslední mikrosvět je zaměřen na práci s proměnnými. Postrádá postavu a grafický výstup, programuje se tu výpis číselných hodnot, obdoby tisku na tiskárně s použitím příkazu `print`. Namísto scény je zde list papíru, na který se hodnoty vypisují. K práci s číselnými hodnotami se používají proměnné v tradiční představě pojmenovaných krabiček s hodnotami na papírku uvnitř. Základními bloky tohoto mikrosvětů jsou přiřazení hodnoty do proměnné (příkaz `let`), změna hodnoty v proměnné (blok `změň ve Scratch`), blok pro samotnou hodnotu proměnné, použitelný ve výrazech (Obrázek 4), a vypsání jedné nebo dvou hodnot do jednoho řádku (příkaz `print`). Toto vypsání přitom jako parametr neumožňuje použít výraz nebo číslo, ale pouze hodnotu proměnné. K základním blokům byly přidány bloky *Opakuj*, představující cyklus s pevným počtem opakování, bloky pracující s podmínkami porovnávacími dvě hodnoty `>`, `=`, `<` *Opakuj dokud* a *Když (- jinak)*, dále blok pro vytváření matematických výrazů se základními početními operacemi a blok s hodnotou čísla pro použití v podmínkách a výrazech.



Obrázek 4: Úloha mikrosvětů Proměnné k vypsání mocnin dvou menších než 1000 s odezvou na sestavený program. Obtížnost úlohy je zvýšena absencí bloku k nastavení hodnoty proměnné N.

Tento mikrosvět umožňuje použití úloh z tradičního pojetí výuky programování o práci s číselnými hodnotami. Typickou úlohou v tomto světě je vypsání předepsané řady čísel, zadanou jejím výpisem nebo popisem. Těžší úlohy lze vytvářet např. použitím bloků pro výrazy, omezením počtu použitelných proměnných v úloze nebo vypnutím příkazů ke změně hodnot proměnných.

3 SADY NAVAZUJÍCÍCH ÚLOH

Ke každému mikrosvětů jsme vytvořili sadu na sebe navazujících úloh. Gradování úloh v nich odráželo zvyšující se náročnost zadání. Takto vytvořené sady měly vícenásobné využití. Charakter úloh pro mikrosvětů byl kromě vlastní situace a grafického vzhledu ovlivněn dalšími faktory. Nejdůležitějším byl rozsah příkazů, které jednotlivé mikrosvětů nabízejí, a způsob, jakým se žákovské řešení kontrolovalo. Z těchto možností vyplynula do jisté míry výzva pro nás autory využít potenciál prostředí s malým počtem příkazů pro kognitivně zajímavé problémy, nabízející vtipná řešení. Kromě zkušeností, které máme z tvorby různých učebnic [9, 10], jsme vycházeli ze studií, zabývajících se analýzou infromatických konceptů a kritérií pro posuzování gradovatelnosti [11, 12].

3.1 Sada úloh pro mikrosvět Želva

Pro podrobnější seznámení jsme si k popisu vybrali prostředí želví grafiky. Jde o známý koncept kreslení čar pohybující a otáčející se postavy, na jehož základě již vznikly různé publikace [13, 14], učebnice [15, 16, 17] a další materiály. Tyto materiály byly inspirativní, ale nekorespondovaly s námi zvolenými cíli. Proto jsme většinu úloh vytvářeli od začátku sami.

Víceř generové studie ukazují [18, 19], že motivace v zadání úlohy spolu s obrázky děvcátům pomáhají k lepším výsledkům. Snažili jsme se proto vybrat příběh, který dává dostatek prostoru pro kreativní zadání a zároveň v něm reálný objekt zanechává stopu. Zvolili jsme jeden motiv, vinoucí se všemi úlohami, s tématem starověkého Egypta a tesání do kamene. Snahu zajímavými příběhy vtáhnout řešitele do příběhu jsme podpořili obrázky k naprogramování.

Kromě motivace nám šlo primárně o seznámení s infromatickými koncepty, jejichž náročnost a rozsah jsme postupně zvyšovali. Popis těchto konceptů pro mikrosvětů Robot Karel a Film lze najít v [20].

V článku [11] rozebírají autorky gradující úlohy pro robotické stavebnice ve čtyřech oblastech (ovládání, náročnost podložky robota, stav robota a programovací konstrukty). Touto optikou jsme pohlédli na naši gradující sadu úloh a objevili jsme v ní čtyři oblasti, podle kterých dokážeme přesněji popsat faktory gradace. Vytvořili jsme přehledovou tabulku, v jejích sloupcích jsou tyto objevené oblasti: Programovací koncepty, Nové příkazy, Geometrické aspekty mikrosvětů a Kognitivní operace. Jednotlivé řádky tabulky reprezentují všech 12 úloh sady. Vyplněná pole ve sloupcích tabulky reprezentují úlohu, v níž se objevil nový příkaz či bylo potřebné použít nový programátorský koncept. Jestli zůstalo pole prázdné, v dané úloze se z dané oblasti (v daném sloupci) nic nové neobjevilo.

Specifickým pro toto prostředí byl grafický aspekt (otočení, natočení). Náročnost úlohy se promítla do jeho nastavení v těle programu. Např. umístění otočení uvnitř cyklu je spíše náročnější než před ním. Ve třetím sloupci se jedná o koncepty, související s procesem kreslení. Ve čtvrtém sloupci popisujeme myšlenku, resp. kognitivní činnost, kterou musel žák při řešení vykonat.

Přidáváním nových prvků do úloh v uvedených čtyřech oblastech a jejich kombinováním v různých polích řádku, tedy ze čtyř různých oblastí, jsme mohli jemně modifikovat obtížnost úloh.

Tabulka 1: Gradace obtížnosti úloh mikrosvětů Želva v různých aspektech

Úloha	Programovací koncepty	Nové příkazy	Geometrické aspekty mikrosvětů	Kognitivní operace
1	Sekvence příkazů, parametr u příkazu	Krok s parametrem Otočení o pravý úhel	Úvodní nastavení směru kreslicí postavy Levo-pravá orientace	Pochopení příkazů, jejich řazení Spuštění programu
2	Jednoduchý cyklus	Opakuj n krát		Propedeutika k optimalizaci délky kódu
3	Dva cykly za sebou	Otočení s parametrem	Otočení o jiný než pravý úhel Návrat na původní místo	
4	Více cyklů za sebou			Identifikace posloupnosti zkracujících se délek
5	Vnořený cyklus			Optimalizace délky kódu
6			Změna směru mezi dvěma cykly používajícími otáčení	Identifikace stejného, pouze pootočeného vzoru
7	Více vnořených cyklů za sebou			Identifikace opakujícího se vzoru Řešení problému chybějícího parametru v nabídce
8		Přestaň kreslit Začni kreslit	Vícenásobná změna směru mezi dvěma cykly Kreslení přerušovaných čar	Pochopení příkazů k přepínání stavu.
9	Vnořený cyklus s delší sekvencí ve vnějším cyklu		Nastavení směru na konci vnořeného cyklu.	Identifikace opakujícího se vzoru
10		Použití záporného parametru	Otáčení v cyklu o jiný než pravý úhel Pohyb tam a zpět	Identifikace dvou prolínajících se opakujících se vzorů
11			Řešení šikmého pohybu pomocí jeho vodorovné a svislé složky	Dekompozice obrázku
12	Vícenásobné vnoření cyklů		Nahrazení posunutí o nedostupnou vzdálenost kombinací pohybů	Vícenásobná dekompozice obrázku. Identifikace opakujících se vzorů při jejich otočení Optimalizace délky kódu

4 OVĚŘENÍ A ZPĚTNÁ VAZBA

Vytvořené sady úloh byly ověřovány takto: pro každý ze čtyř vytvořených mikrosvětů jsme sestavenou sadu úloh nejprve nabídli dvěma školám, které na projektu spolupracovaly. Úlohy zde řešilo vždy několik tříd, celkově ověřovalo kolem 160 žáků. Toto pilotní ověření sloužilo především k odstraňování chyb jak v úlohách, tak v samotných mikrosvětech. Následně jsme tyto sady úloh otevřeli ostatním školám buď jako přípravu na loňské či letošní národní kolo (sady robot Karel nebo Želva), nebo jako přípravu postupujících do ústředního kola (sady Film a Proměnné). Tyto dvě sady jsme po ústředním kole zpřístupnili či zpřístupníme i ostatním školám jako mimořádný nesoutěžní „bobí“ test.

Dalšími úlohami, vytvořenými ze šablon těchto mikrosvětů, byly soutěžní úlohy. V národním kole 2021 každý ze 109 442 soutěžících absolvoval tři úlohy z mikrosvětů robota Karla. Při ústředním kole pak 358 soutěžících absolvovalo tři úlohy z mikrosvětů Film a tři úlohy z mikrosvětů robota Karla; 40 % všech soutěžních úloh tohoto kola bylo programovacích s bloky.

Tabulka 2 ukazuje počet žáků, kteří si tyto testy vyzkoušeli mimo vlastní soutěž (viz Tabulka 2).

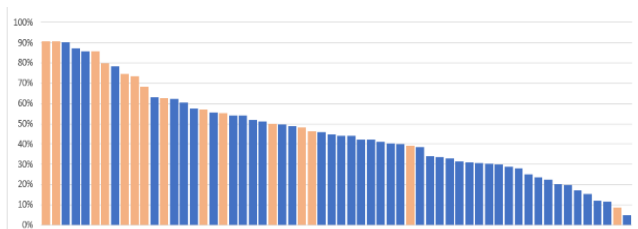
Tabulka 2: Počet žáků zkoušejících si testy mimo soutěž

Mikrosvět	počet žáků	termín
Robot Karel	45 000	září – říjen 2021
Film	540 postupujících	leden – únor 2022
Film	10 500	květen – červen 2022
Želva	71 600	říjen 2022
Proměnné	450 postupujících	leden 2023
Proměnné		květen – červen 2023

V národním kole 2022 každý z 183 564 soutěžících absolvoval dvě úlohy z mikrosvětů Želva a ve starších kategoriích úlohy z ostatních mikrosvětů. V ústředním kole pak 365 soutěžících absolvovalo po šesti blokových úlohách z celkových patnácti; po dvou z prostředí Proměnné, Film a Robot Karel.

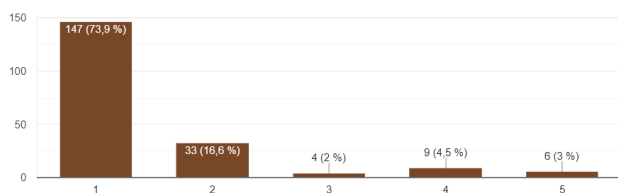
Zajímali jsme se, nakolik jsou nové programovací úlohy náročné. Analýzou úspěšnosti soutěžících ve všech kategoriích národního

kola jsme zjistili, že tyto programovací úlohy nejsou obtížné. Dokumentuje to graf na obrázku 5, na němž se programovací úlohy vyskytují více vlevo, mezi úlohami s vyšší úspěšností.



Obrázek 5: Úspěšnost v řešení soutěžních úloh národního kola 2021 soutěžícími. Výšky sloupců ukazují, kolik % soutěžících úlohu vyřešilo správně. Úlohy se sestavováním programového kódu jsou označeny oranžovou barvou.

Zajímá nás též názor učitelů na tyto úlohy – jak je vnímají, jak reagují žáci, o jak vhodné úlohy jde. V anketě odpovídalo 199 učitelů informatiky a výsledek shrnutý v grafu na obrázku 6 lze hodnotit jako úspěšný.



Obrázek 6: Výsledky anketní otázky o přínosu úloh s blokovým programováním mezi školními koordinátory soutěže, kteří tyto úlohy hodnotili školní známkou.

5 VÝHLED, ZÁVĚR

Předpokládáme, že vlastnictví unikátního typu úloh v rámci společnosti zemí pořádajících soutěž Bebras Challenge nám umožní zkoumat, nakolik jsou soutěžní otázky tohoto typu vhodné pro takový typ soutěže, zdali např. nepřenesou na sebe příliš mnoho pozornosti a tudíž spotřebovaného času soutěžícími. Předpokládáme, že při dlouhodobém využití tohoto modulu budeme vyvíjet nové úlohy a možná nové typy soutěžních úloh. Věříme, že seznámení s těmito soutěžními otázkami, které jsou typově jiné než klasické programovací úlohy, upevní na školách představu o programování jako vzdělávacím tématu, které poskytuje dostatek podnětů k rozvoji mentálních schopností žáka a které nabízí problémové situace blízké reálnému životu. Jako součást tvorby prostředí pro úlohy se sestavováním programového kódu jsme vyvinuli modul, umožňující export dat, popisujících, jak soutěžící sestavují programový kód. Při každém spuštění uživatelského programu se jeho program uloží na server a my máme možnost sledovat, jak se v průběhu řešení mění podoba žákova programu, jak se mění jeho strategie řešení úlohy, jak se žák seznamuje s novými bloky nebo programovacími koncepty, jak reaguje na neúspěch a v čem je pro něj daná úloha obtížná. Předpokládáme, že v důsledku budeme schopni sledovat, jak se žáci učí programovat, jak rozumí konceptům a jaké si vytváří miskoncepty. Jeden takový výzkum, týkající se porozumění konceptu cyklu, jsme již realizovali [21]. Sady úloh s mikrosvětvy Robot Karel [22] a Film [23] lze najít v archivu soutěže Bobřík informatiky; odkazy jsou v literatuře.

PODĚKOVÁNÍ

Vývoj prostředí a úloh byl podpořen z projektu TAČR TL03000222 „Rozvoj inženýrského myšlení pomocí situačních algoritmických problémů“.

BIBLIOGRAFICKÉ ODKAZY

- [1] DAGIENĚ, V. The Bebras Contest on Informatics and Computer Literacy – Students Drive to Science Education. In *Joint Open and Working IFIP Conference, ICT and Learning for the Net Generation*, 2008. p. 214–223.
- [2] Modul Blockly. <https://developers.google.com/blockly>
- [3] RESNICK, M. *Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play*. MIT Press, Cambridge MA. 2017.
- [4] PATTIS, R.E. *Karel the Robot: Gentle Introduction to the Art of Programming with Pascal*. John Wiley & Sons. 1981.
- [5] Blockly games. <https://blockly.games/>
- [6] VANÍČEK, J., ŠIMANDL, V., DOBIÁŠ, V. Bebras Tasks Based on Assembling Programming Code. In: Bollin, A., Futschek, G. (eds) *Informatics in Schools. A Step Beyond Digital Education. ISSEP 2022*. Lecture Notes in Computer Science, vol 13488. Cham: Springer, 2022, s. 113-124. https://doi.org/10.1007/978-3-031-15851-3_10
- [7] PAPERT, S. *Mindstorms: children, computers, and powerful ideas*. Basic Books, New York, NY, USA. 1980.
- [8] RESNICK M., a další. Scratch: Programming for all. In *Communications of the ACM*, 2009, vol. 52, no. 11, p. 60-67.
- [9] KALAŠ, I., MIKOVÁ, K.: *Základy programování ve Scratch pro 5. ročník základní školy*. Učebnice. České Budějovice: Jihočeská univerzita v Českých Budějovicích, 2020.
- [10] VANÍČEK, J., NAGYOVÁ, I., TOMCSÁNYIOVÁ, M.: *Programování ve Scratch pro 2. stupeň základní školy*. Učebnice. České Budějovice: Jihočeská univerzita v Českých Budějovicích, 2020.
- [11] HRUŠECKÁ, A., MIKOVÁ, K. Identifikácia gradácie inforatických konštruktov pri vyučovaní edukačnej robotiky na 1. stupni ZŠ. In *DidInfo 2021*. Banská Bystrica: Univerzita Mateja Bela, 2021. s. 118-122. ISBN 978-80-557-1823-1.
- [12] GUJBEROVÁ, M., KALAŠ, I. Designing productive gradations of tasks in primary programming education. In *Proceedings of the 8th workshop in primary and secondary computing education (WiPSCE '13)*. Aarhus :ACM, 2013, p. 108-117. DOI= <https://doi.org/10.1145/2532748.2532750>
- [13] PELÁNEK, R. *Želví grafika, Exkurze do programování, geometrie a umění*. Computer Press: Brno, 2018. 80 s. ISBN 9788025149058
- [14] TRŽILOVÁ: *Logo a matematika* (1993). České Budějovice: Jihočeská univerzita. <https://docplayer.cz/17910249-Jihoceska-univerzita-v-ceskych-budejovicich-pedagogicka-fakulta-logo-a-matematika-ucebni-text-pro-studenty-vyberoveho-seminare.html>
- [15] BLAHO, A., KALAŠ, I. *Tvorivá informatika (1. zošit z programovania)*. Bratislava: Slovenské pedagogické nakladateľstvo - Mladé letá, 2005.

- [16] BLAHO A., KALAŠ I.: *Imagine Logo – programování pro děti*. Computer Press, 2006. 48 s.
- [17] BLAHO, A., SALANCI, E., ŠIMANDL, V. *Programování v jazyce Python pro střední školy*. České Budějovice: Jihočeská univerzita v Českých Budějovicích, 2020. <https://imysleni.cz/ucebnice/zaklady-programovani-v-jazyce-python-pro-stredni-skoly>
- [18] HUBWIESER, P., HUBWIESER, E. GRASWALD, D. How to attract the girls: Gender-specific performance and motivation in the Bebras challenge. In *Informatics in Schools: Improvement of Informatics Knowledge and Perception: 9th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, ISSEP 2016*. Münster : Springer International Publishing, 2016, s. 40-52.
- [19] BUDINSKÁ, L., MAYEROVÁ, K. Graph Tasks in Bebras Contest: What does it have to do with gender? In *Proceedings of the 6th Computer Science Education Research Conference. CSERC '17*. HELSINKI : ACM, 2017, p. 83-90. DOI = <https://doi.org/10.1145/3162087.3162102>
- [20] VANÍČEK, J., ŠIMANDL, V., DOBIÁŠ, V. Situational algorithmic tasks with the assembling of program code as a tool for developing computational thinking. In *Journal of Technology and Information Education 2022*, roč. 14, č. 2, s. 101-119. DOI= 10.5507/jtie.2022.014. https://jtie.upol.cz/artkey/jti-202202-0001_situacni_algoritmicke_ulohy_se_sestavovanim_programeho_kodu_jako_nastroj_rozvijeni_informatickeho_mysleni.php
- [21] VANÍČEK, J., DOBIÁŠ, V., ŠIMANDL, V. Understanding loops: What are the misconceptions of lower-secondary pupils?. In: *Informatics in Education, 2023*. (in print) DOI= 10.15388/infedu.2023.20. <https://infedu.vu.lt/journal/INFEDU/article/744/info>
- [22] Bobřík informatiky (2022a). Karel: Sada edukačních programovacích úloh. Jihočeská univerzita v Českých Budějovicích. <https://www.ibobr.cz/test/archiv-predspustenim/2021/495>
- [23] Bobřík informatiky (2022b). Film: Sada edukačních programovacích úloh. Jihočeská univerzita v Českých Budějovicích. <https://www.ibobr.cz/test/archiv-predspustenim/2021/494>

IT activities for marginalized children

Patrik Voštinár
Univerzita Mateja Bela
Tajovského 40
97401 Banská Bystrica
Slovenská Republika
patrik.vostinar@umb.sk

Róbert Címer
American Center Banská Bystrica
Štátna vedecká knižnica v Banskej Bystrici
Lazovná 9
975 58 Banská Bystrica
Slovenská Republika
banskabystrica@americanspaces.sk

ABSTRACT

The aim of this article is to present activities for marginalized children aimed at developing algorithmic thinking and the basics of programming. Children from these groups have more problems in schools when learning informatics, mainly due to the absence of IT technology - computers and tablets in their homes. Some children do not even have electricity in their homes. In the article, we describe the IT activities that we did with these children during the summer camp and workshops at the premises of the American Center in in Banská Bystrica State Scientific Library.

Keywords

Programming for children. Teaching programming. Marginalized groups. Roma youth. Ozobot. App Inventor. Micro:Bit. American center.

1 INTRODUCTION

Nowadays, it is essential that elementary school students already know how to work with information and communication technologies (ICT). The need to master a computer is slowly becoming a necessity for today's school graduates. The closure of schools during the Covid-19 pandemic forced most primary school students to use computers and mobile phones for teaching. The problem was that marginalized groups do not have enough equipment for their children to be able to fully control a computer or tablet.

In 2018 (before the pandemic), the Office of the Government Plenipotentiary for Roma Communities published a publication [1] on its website, which summarizes the main findings of the survey of the same name conducted in 2018, which provided the most comprehensive overview of the living and income conditions of marginalized Roma households from Slovak communities. The research sample consisted of 5,000 respondents from more than 1,000 randomly selected households from marginalized communities in Slovakia. The results showed that the monthly sums available to marginalized groups were not even sufficient to cover basic life needs (food, housing, clothing, health care), and some households did not even have electricity installed. For this reason, children who grow up in such households do not have sufficient experience working with computers. They may lack this skill in their future profession.

The impact of computers on teaching has been investigated by several educators in their research. Based on researches [2-4], we can conclude that the use of ICT can improve the students' attitude towards the topic, each student chooses his own pace that suits him, students have immediate feedback and it is suitable for children with specific learning disabilities (it helps pupils with dyslexia, dysgraphia). Marginalized children are often not motivated to go to school, to pay attention, many of them suffer from various learning disabilities. Also for this reason, the use of

ICT in the education of marginalized children can positively affect their future.

According to the International Standard Classification of Education (ISCED), students should start teaching programming in regional education as early as 1st level. However, we often encounter the fact that computer science lessons not only at the first but also at the second level of elementary school are mostly about working with office tools (MS Office) or playing games. If children from marginalized groups come to elementary schools where computer science classes are less specialized, then these children have nowhere to learn to work with ICT technologies.

Recently, several educational aids, educational applications, and websites have been created that make it possible to teach programming in a more engaging way - closer to children. These sites and aids are starting to be sought after not only by teachers of computer science, but also of other subjects. We also owe this development to the STEM (science, technology, engineering and math) curriculum. Thanks to this education, it is possible to use various robotic kits, microcontrollers and create interesting inter-subject projects, which mostly need to be programmed, and that is also why it is important that students in the first grade of elementary schools are taught programming.

Several researches have been devoted to the use of the Scratch environment in the teaching process, e.g. [5] dealt with research on the influence of computational thinking during the use of the Scratch language, research [6] was devoted to the effectiveness of the use of the Scratch language.

We have described the suitability of using the MakeCode Arcade environment for the development of 2D games, even with the proposed methodology and research, in the publication [7]. The authors [8] used MakeCode Arcade as the main programming language in teaching programming focused on game development.

Around the world, several researches have looked at the use of Minecraft: Education Edition in the education process. In her research, Čujdiková [9] described a case study on the use of the Minecraft game in improving computational thinking and computer science teaching at an Italian high school. Kane et al. [10] presented an educational game for children based on Minecraft that taught the basics of Python programming.

In their publication, the authors [11] wrote about the experience with teaching the creation of mobile applications in the App Inventor tool. Ts. St. Georgiev [12] from Ruse University did research on the use of the MIT App Inventor programming environment at different levels of education.

About the implementation of Ozobot robots in education in Slovak and Czech schools wrote the following authors [13-14].

The aim of this article is to present our experience with teaching marginalized groups through various aids for the development of the use of ICT devices and algorithmic thinking.

2 American Center Banská Bystrica

The American Center in Banská Bystrica State Scientific Library provides information about the USA and helps promote mutual understanding and connection between Slovaks and Americans. The center provides information on study opportunities in the USA and presents American culture, history and values. Currently, it also organizes programs focused on STEM education and media literacy. Programs are often supported by the US Embassy in Bratislava. In the center, students and teachers have access to English learning materials such as dictionaries, linguistic publications and workbooks. The center is equipped with laptops, micro:bits, 3D printers and 3D pens for the public. Among the events, there are also workshops for teachers and students. [15] Figure 1 shows the American Center in Banská Bystrica State Scientific Library.



Figure 1: American Center in Banská Bystrica

3 Community center Sásová

The community center in Sásová has been operating informally in Banská Bystrica since 1994. The civic association Nádej Deťom (Hope for Children) was registered in April 1998. It is aimed at families living in marginalized communities in Banská Bystrica and its surroundings. The vision of the community center is the development of recipients of social services with the aim of using their own potential in order to become full-fledged members of society.

The activities of the community center include various club activities with children and youth so that children are better prepared for the teaching process at schools, help with homework, work with IT technologies.

4 2021 Summer IT camp

We organized the first summer IT camp for marginalized children at the premises of the American Center in Banská Bystrica State Scientific Library on August 16-20, 2021. During this one-week camp, six children tried various activities aimed at developing algorithmic thinking and programming. During each day, the children tried different activities that lasted 3.5 hours.

- Day 1 – Micro:bit programming
- Day 2 – Robotics using Ozobot Evo and Edison
- Day 3 – App Inventor and app programming
- Day 4 – 3D modelling and 3D printing
- Day 5 – 2D game programming in MakeCode Arcade and virtual reality

4.1 Micro:bit

During the first day, the children were introduced to the micro:bit educational board. They tried to program their own micro:bit name tag, the use of various sensors and extensions (NeoPixel led strip), they assembled the melody to the song "Kohútik jarabý..." ("Rooster Rooster..."), they sent each other messages using radio communication, and finally they programmed a rock-paper-scissors game, which can be seen in Figure 2. The children really liked the micro:bit educational board, but the biggest problems for them were working with the mouse (as they were not very used to it) and copying the generated HEX file to the micro:bit educational board folder.

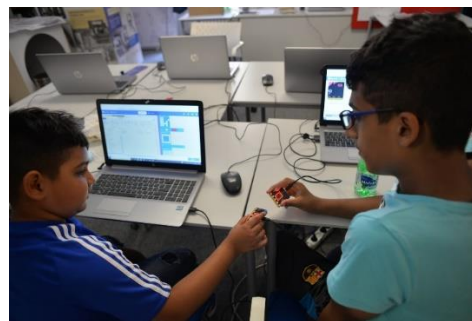
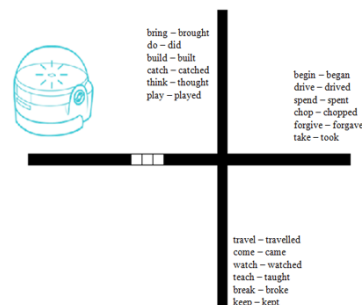


Figure 2: Rock-paper-scissors game on the micro:bit educational board

4.2 Ozobot EVO and Edison

During the second day, children tried programming two types of educational robots - Ozobot Evo and Edison. We started with the Ozobot Evo robot, which we programmed using markers. Their first task was to draw a black fixed path along which the Ozobot walked. After this activity, we gave the students pre-prepared tasks on paper, in which they drew color codes to determine the direction of the robot. These tasks were focused on cross-subject linking - tasks on English, mathematics, technology. Figure 3 shows a task focused on English, which we are a student of - to draw the color codes so that the robot goes to the correct forms of the verbs.



Nauč Ozobota po
Anglicky

Figure 3: Ozobot lesson focused on English

After these activities, we switched to the Edison robot, which we programmed again without a computer. The advantage of the Edison robot is that it allows reading barcodes using sensors located on the bottom of the robot. While working with this robot, we also focused on the reasons why robots are becoming more and more widespread in the world and what the individual sensors are used for - obstacle avoidance, line tracking, infrared sensor, etc. Figure 4 shows an activity with the Edison robot.



Figure 4: Edison Robot

These activities were very interesting for the children, as they had not encountered such robots in regular computer science lessons. Discussion revealed that they had already seen such aids, but did not use them, as they do not attend the robotics clubs organized at their schools.

4.3 App Inventor

During the third day, the children tried programming mobile applications for the Android operating system. The biggest problem with this activity was that the children who had a smartphone (not all children had a cell phone or a smartphone) had a full memory or a very old version of the Android operating system, on which our application¹ did not work. We developed applications in the App Inventor¹ online tool, which is already more demanding. We showed them how to create a simple app that included buttons to change a background color at random, open a screen with a custom drawing editor, and a screen that generated random colors, sizes, and positions of bubbles on the screen. As a second application, they tried to program the Mole Mash game. These activities were more challenging for them, as similar problems appeared as when working with the micro:bit educational board - controlling the computer using a computer mouse. Figure 5 shows a preview of the first App Inventor application created by the children.

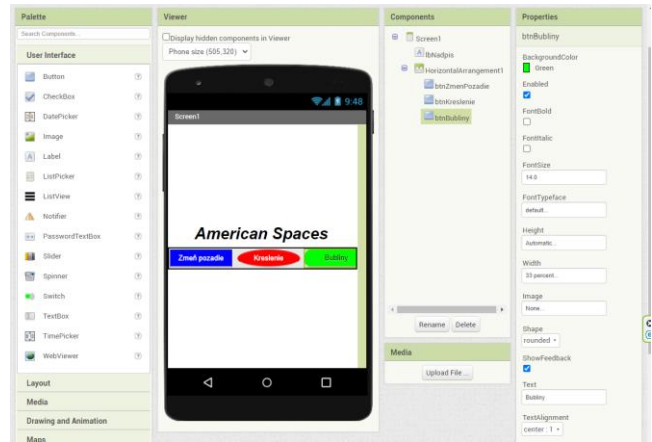


Figure 5: App Inventor – first application

4.4 3D modelling and 3D printing

On the fourth day, we showed the children how they can very easily model a 3D object. On the online site www.tinkercad.com, the children tried to model a basket token with their name and a key tag. After modeling the objects, we introduced them to the principles of 3D printing and had their pendants printed. While their pendants were being printed, we showed the children 3D Polaroid pens, with which they drew 2D objects and later a 3D cube. Figure 6 shows the work with 3D pens (the pendant is modeled on the laptop in the background). This activity interested them more than programming mobile applications, robots and micro:bit.



Figure 6: 3D modelling and working with 3D pens

4.5 Programing 2D games via MakeCode Arcade and virtual reality showcase

On the last day, the children created 2D games in the MakeCode Arcade² environment, which they then tried to play on special Kitronic Arcade game consoles. Figure 7 shows children working with the MakeCode Arcade environment.

¹ <https://appinventor.mit.edu/>

² <https://arcade.makecode.com/>



Figure 7: Game development via MakeCode Arcade

At the very end of the camp, the children tried to play virtual reality games using the Oculus Quest 2 glasses, which we created at the Faculty of Natural Sciences, Informatics Department of Matej Bel University in Banská Bystrica.

5 IT workshops

After the end of the summer camp, we received positive feedback from the leadership of the civic association Nádej Deťom (Hope for Children). This is also why we decided to continue these activities for marginalized groups. We decided to implement a regular 90-minute workshop per week on various STEM tools. During these workshops we used the following software and hardware:

- Scottie Go! – board game
- Minecraft:Education Edition
- Micro:bit and Lego expansion
- Micro:bit expansions - connecting LED lights, buttons, etc.

From the workshops, they liked the work in Minecraft: Education Edition the most, as this game enjoys worldwide popularity among children. The hardest part for them was working with micro:bit extensions – connecting LED lights, buttons. Figure 8 shows an example of working with Minecraft:Education Edition.



Figure 8: Minecraft: Education Edition – block task

During spring break 2023, we agreed with the community center on daily activities - workshops in the American Center in Banská Bystrica State Scientific Library. Each workshop lasted 90

minutes and was attended by 6-9 children from marginalized groups. Every day we did different activities:

- Working with 360° photos in CoSpaces
- Augmented reality and Merge Cube in CoSpaces
- Ozobot EVO
- Micro:bit
- Minecraft:Educational Edition

Figure 9 shows a sample of the activity done by the students in the CoSpaces environment.

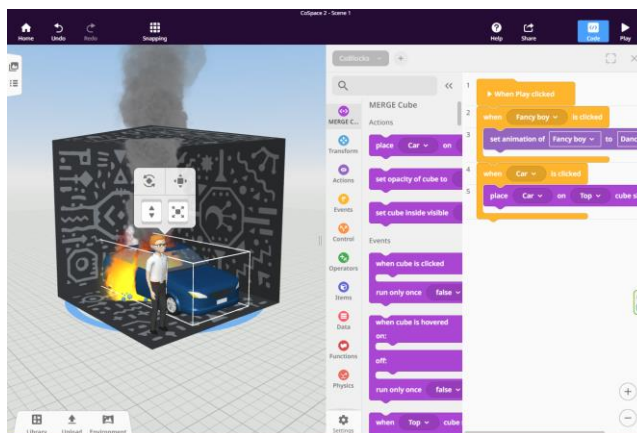


Figure 9: CoSpaces environment with Merge Cube

6 CONCLUSION

In the contribution, we presented our experience with teaching marginalized groups during the IT summer camp in 2021, during the IT workshops in 2022 and 2023. We received feedback continuously during the activities (from their assistant, who was present during all activities) and after the first IT camp. The feedback was very positive, according to the leadership of the civic association Nádej Deťom (Hope for Children), our activities helped them in teaching at the elementary school. The observation showed that in the initial activities the children were more timid, it was more difficult for them to work with the computer. With each subsequent workshop, the children became smarter and progress was visible. For this reason, we would like to continue similar activities in other summer camps and IT workshops.

ACKNOWLEDGMENT

The contribution was created thanks to the support of the US Embassy in Bratislava and American Center in Banská Bystrica State Scientific Library.

BIBLIOGRAFICKÉ ODKAZY

- [1] GRAUZELOVÁ, T., MARKOVIČ, F. *Príjmy a životné podmienky v marginalizovaných rómskych komunitách*, 41 s., 2018, ISBN 978-80-969971-9-0.
- [2] NOCAR, D., BÁRTKOVÁ, E. *Motivace nadaných žáků a studentů k řešení úloh pomocí ICT.*
- [3] UHLÍČOVÁ, L. *Zlepšenie postoja žiakov k preberanej téme.*

- [4] VANÍČEK, J. Výsledky průzkumu mezi školními koordinátory. Web soutěže Bobřík informatiky [online]. České Budějovice: Jihočeská univerzita,
- [5] FAGERLUND, J., HÄKKINEN, P., VESISENAHO, M., VIIRI, J. Computational thinking in programming with Scratch in primary schools: A systematic review. In *Computer Applications in Engineering Education*. 2021, vol. 29, no. 1, p. 12– 28.
- [6] KALELIOGLU, F., GÜLBAHAR, Y. The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. In *Informatics in Education*. 2014, vol. 13, n. 1, s. 33-50. ISSN 1648-5831.
- [7] VOŠTINÁR, P. MakeCode Arcade: Interesting environment for programming 2D games. In *IEEE world conference on engineering education (edunine): proceedings: the future of engineering education: current challenges and opportunities*. Piscataway: Institute of Electrical and Electronics Engineers, 2021a, p. 1-6. ISBN 978-0-7381-1321-0.
- [8] MOSTER, M. et al. "Can You Help Me?" An Experience Report of Teamwork in a Game Coding Camp for Autistic High School Students. In *2022 IEEE/ACM 44th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*. 2022, p. 50-61. ISBN 978-1-6654-9592-9.
- [9] ČUJDÍKOVÁ, M. Create Minecraft Fame, Save the World. In *ECGBL 2019 13th European Conference on Game-Based Learning*. 2019.
- [10] KANE, I. et al. Escape from the Python's Den: An Educational Game for Teaching Programming to Younger Students. In *Proceedings of the 2019 ACM Southeast Conference*, 2019, p. 279-280.
- [11] GRAY, J., ABELSON, H., WOLBER, D., FRIEND, M. Teaching CS principles with app inventor. In *Proceedings of the 50th Annual Southeast Regional Conference (ACM-SE '12)*. New York: Association for Computing Machinery, p. 405–406.
- [12] GEORGIEV, T. Students' Viewpoint about Using MIT App Inventor in Education. In *42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija: IEEE, 2019, p. 611-616. ISBN 978-953-233-098-4.
- [13] PICKA, K., DOSEDLA, M., STUCHLIKOVA, L. Robotic didactic aid Ozobot in Czech schools. In *2020 18th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 2020, p. 525-533.
- [14] ŽÁČEK, M., SMOLKA, P. Development of Computational thinking: Student motivation using Ozobot. In *Proceedings of the 2019 3rd International Conference on Education and E-Learning*. 2019. p. 36-40.
- [15] American Center Banská Bystrica.
<https://www.americanspaces.sk/>

Deskové hry jako prostředek pro rozvoj informatického myšlení studentů učitelství informatiky

Board games as the means of development of computational thinking of computer science teacher students

Viola Vrbová
Základní škola Horšovský Týn
Zámecký park 3
346 01 Horšovský Týn
Česko
violavrb@students.zcu.cz

Zbyněk Filipi
KVD FPE ZČU
Klatovská tř. 51
301 00 Plzeň
Česko
filipiz@kvd.zcu.cz

ABSTRACT

The paper is based on a Bachelor's thesis that focuses on board games that can be a means of developing activities in which computational thinking is manifested. In the paper we introduce three board games and described the elements of computational thinking applied in them. The research conducted on computer science teacher students and their relationship to board games shows that university students play board games and can recognize elements of computational thinking in the selected board games. Board games can provide an opportunity for the development of computational thinking not only for adults and computer science teacher education students who have not undergone the development of computational thinking in formal education at lower levels. However, it is important to see them as developmental.

Keywords

Computational thinking. Board games. Informal learning.

ABSTRAKT

Príspevek je založený na bakalárskej práci, jež je zaměřena na deskové hry, které mohou být prostředkem pro rozvoj činností, v nichž se projevuje informatické myšlení. V práci představujeme informatické myšlení a jeho charakteristiky. Hraní deskových her navozuje aktivní stav mysli, který je výhodný pro rozvoj nového druhu myšlení, proto jsme analyzovali tři deskové hry a popsali prvky informatického myšlení v nich uplatňované. Z provedeného výzkumného šetření týkajícího se studentů učitelství informatiky a jejich vztahu k deskovým hrám vyplývá, že studenti vysokých škol hrají deskové hry a dokážou rozpoznat prvky informatického myšlení ve vybraných deskových hrách. Deskové hry mohou poskytnout možnost rozvoje informatického myšlení nejen dospělým a studentům učitelství informatiky, kteří neprošli rozvojem informatického myšlení ve formálním vzdělávání na nižších stupních. Je však důležité je vnímat jako rozvíjející.

Klíčová slova

Informatické myšlení. Deskové hry. Neformální vzdělávání.

1 ÚVOD

V únoru 2021 proběhla v českém vzdělávání výrazná změna Rámcového vzdělávacího programu pro základní vzdělávání

(RVP ZV), v jejímž důsledku musí všechny školy začít nejpozději od září 2023 ve čtvrtém a pátém ročníku na prvním stupni a nejpozději o rok později ve všech ročnících na druhém stupni s výukou informatiky namísto dosavadní výuky informačních a komunikačních technologií. V obsahově výrazně proměněném vzdělávacím oboru je kladen důraz na rozvoj informatického myšlení (IM) žáků [1]. K podobné změně došlo na úrovni gymnázií, kde je povinnost přizpůsobit Školní vzdělávací program (ŠVP) novému kurikulárnímu dokumentu RVP G nejpozději od září 2025 [2]. Kvůli značné setrvačnosti je patrné, že až do roku 2034 se budou na vysoké školy hlásit uchazeči, kteří neprošli uceleným vzděláváním v informatice zaměřeným na rozvoj IM od primárního po vyšší sekundární stupeň soustavy ISCED. Zároveň nemusí ani vědět, co si mají pod pojmem IM představit. Tito studenti, obzvláště studenti učitelství informatiky, by měli IM rozvíjet mimo formální systém vzdělávání, ať již návštěvou neformálních kurzů či kroužků, nebo sebezvzděláváním.

Sebezvzdělávání může mít mnoho různých podob a může být velmi efektivní, pokud dotyčného baví a probíhá hrou. Hra sama o sobě může být velmi dobrým prostředkem k naučení řešení různých situací, pochopení souvislostí a rozvoji tvořivosti, jiného pohledu na věc či jiného druhu myšlení. Vše probíhá na základě postavení před problémové situace, které hráč ve hře řeší [3]. Hra jako specifická činnost člověka je součástí vývoje osobnosti, pomáhá k poznání okolního světa i vlastních možností. Důležitou roli má v každém věku, přestože v předškolním věku je nejvíce pozorovatelná a má svůj největší význam [4, s. 27]. Jediný rozdíl v hrách provázející dospělé může být ten, že již na rozdíl od dětí předškolního věku upřednostňují více strukturované hry s danými pravidly a může zde mít svůj další rozměr i spolupráce. Ideálním reprezentantem těchto her jsou deskové hry, nejlépe v unplugged podobě (ačkoli v poslední době je moderní deskové hry předělávat do digitální podoby, hrát je online a využívat možnosti počítače, který některá pravidla hlídá za hráče).

V dětství děti neznají rozdíl mezi hrou a učením, u studentů vysokých škol a dospělých je tomu už jinak. Dětem je často vštěpováno, že něco, co dělají z vlastního zájmu a nepovinně, je hra, zatímco učení je pro ně povinnost, což může ovlivnit jejich budoucí náhled na svět a vnitřní motivaci. Ve hrách dětem i dospělým méně vadí chyby a dokážou s nimi lépe pracovat a zároveň se z nich poučit [4]. K nastalé problémové situaci se již na základě zkušenosti postaví jinak a nebude již probíhat metodou pokus-omyl, ale vyšší formou řešení problému – rozdělí problém

na menší části a analyzují daný problém i prostředky, které k jeho řešení mají k dispozici. Zde již může hrát roli aplikace algoritmických postupů, který však musí být podle podstaty problému obměňován, aby byl co nejefektivnější [5, s. 101–102]. V ideálním případě si pak jedinec řešením různých problémů osvojí jakýsi způsob, jak všechny další problémy vhodně vyhodnotit a úspěšně dojít k řešení problému.

S vystavováním různým problémovým situacím se u člověka může rozvinout jiný způsob myšlení, který pro něj mohl být dosud neznámý. Tento složitý kognitivní proces slouží k již zmíněnému rozhodování, jak co nejlépe a efektivně dosáhnout kýženého cíle. Hry v jakékoli podobě poskytují velké množství podnětů, jak rozvíjet myšlení a schopnost řešit problémy.

1.1 Koncepty a postupy informatického myšlení

Význam IM se v posledních letech velmi vyvíjel. Prvním, kdo použil pojem IM, byl již v 60. letech Seymour Papert, spoluvůdce programovacího jazyka Logo, který byl navržen k výuce a rozvíjení myšlení dětí [6]. Největší kontury však pojmu vyrýsovala Jeannette M. Wing svým článkem z roku 2006 nesoucí přímo název *Computational thinking*. K IM přistupovala jako k něčemu, co představuje univerzální přístupy k řešení problémů vycházející z konceptů počítačové vědy [7]. Denning a Tedre [8] IM popsali jako mentální dovednosti a postupy vedoucí k navrhování výpočtů, které umožní počítači vykonat činnost za nás, a vysvětlení a interpretaci světa jako celku. V České republice je podstatný popis IM z webu imysleni.cz [9] jako „...*způsobu myšlení, který se zaměřuje na popis problému, jeho analýzu a hledání efektivních řešení. Nabízí nám sadu nástrojů a postupů.*“

IM je souhrnem myšlenek, konceptů a postupů, které používá informatik postavený před problémovou situací. Každý z propagátorů IM může brát různé koncepty jako ty hlavní, my uvádíme ty, na kterých shoduje více autorů [10], [11]:

- **Logické myšlení** – zahrnuje analýzu problému, která umožní učinit vhodné rozhodnutí vedoucí k dalšímu procesu a jistému závěru o situaci.
- **Algoritmy a algoritmické myšlení** – myšlení, ve kterém se pracuje s přesnými postupy řešení problémů.
- **Vzory a jejich rozpoznávání** – rozpoznání vzorů může vést k definici zobecnitelného řešení, které dále může ovlivnit automatizaci ve výpočtech, k iteraci i rekurzi.
- **Abstrakce** – jeden z nejdůležitějších vyšších myšlenkových procesů v rámci IM; poskytuje zjednodušení a zobecnění na základě podobnosti a rozdílů.
- **Evaluace** – pomáhá k určení vhodnosti řešení problému.
- **Automatizace** – klíčová část IM pro počítačovou vědu, v běžném světě se spíše jedná o posouzení, které problémy lépe vyřeší člověk nebo stroj a případná úprava pro počítač.

Nesmíme opomenout i postupy aplikované informatiky, které rovněž tvoří IM [10]:

- **Dekompozice** – náročný problém je vhodné rozdělit na menší části.
- **Vytváření počítačových artefaktů** – počítačový výtvar dokáže pomoci se simulací nějakého procesu, zároveň to může být i hlavním cílem problému, který řešíme IM.
- **Testování a debugging** – velice souvisí s vyhodnocením vhodných variant řešení.

- **Postupný vývoj** – spojuje dohromady dekompozici a testování s debuggingem v rámci komplexního řešení problému.
- **Spolupráce a kreativita** – vyjadřuje potřebu používání zmíněných konceptů a postupů v součinnosti s ostatními kolegy, protože v málokterém oboru lidské činnosti dnes někdo vytváří něco originálního a nového zcela sám a u programování to platí ještě více.

S každým dalším problémem se může IM rozvíjet, přičemž nejefektivnější rozvoj nastává ve chvíli, kdy je zapojena i vnitřní motivace k vyřešení problému. Dnes již tradičními možnostmi rozvoje nejen ve školství je využití robotických stavebnic, blokových programovacích prostředí či zajímavých sad úloh ze soutěže Bobřík informatiky. Prvky IM však můžeme identifikovat i v deskových hrách, které mají rovněž potenciál k rozvoji IM, např. právě v sebevzdělávání dospělých, kteří cíleným rozvojem IM ve formálním vzdělávání neprošli. Je otázkou, nakolik si sami uvědomí, jaké způsoby myšlení používají.

2 DESKOVÉ HRY JAKO PROSTŘEDEK K ROZVOJI INFORMATICKÉHO MYŠLENÍ

V rámci hraní deskových her mají hráči navozený „aktivní, pozorný a nestresující stav mysli“, což je jeden ze znaků hry. Tento stav je ideální k rozvíjení tvořivosti a učení [4].

I proto mohou být deskové hry velmi vhodné k osvojení nových dovedností, ke kterému může docházet buď cíleně během her, které byly přímo navrženy k použití konceptů a postupů IM, ale i spontánně během komerčně prodávaných společenských her. Již jen samotná kontrola, zda deskovou hru hráči hrají přesně podle pravidel a nic neopomenuli, dokáže zapojit tento způsob myšlení [12, s. 66]. V takové chvíli hráči aplikují jakési testování a debugování toho, co pravidla hry dovolí, diskutují nad možnostmi, hledají správná vyjádření, rozkládají problém na části, postupně na něj aplikují abstrakci a vynechají pro danou chvíli nepodstatné detaily, a nakonec porovnávají nastalý problém se vzorovou situací v pravidlech a vyhodnotí, zda je daný tah povolený. Diskuze během her vede k lepšímu uvědomění, co vše se ve hře musí řešit a často jak moc komplexní problémy se v některých strategických či logických hrách musí řešit, aby hráči dospěli k vítězství.

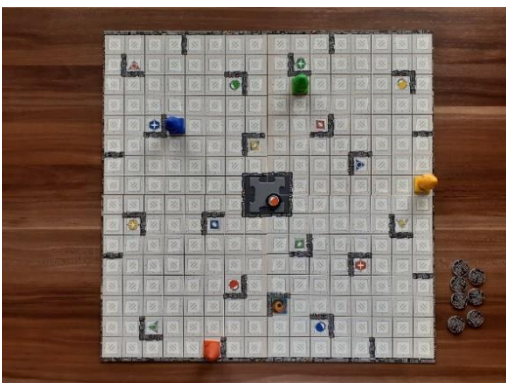
Berland [13, s. 168] vyjádřil názor, že hraní deskových her umožňuje lidem stát se „*odbornějšími informatickými mysliteli*“ a zároveň si tím osvojíme zábavnou formou nový způsob uvažování k řešení problémů, který můžeme využít i v jiných reálných situacích. Myšlení, které je zapojováno právě během hraní logických a strategických her, se vztahuje k IM [13, s. 168].

V rámci bakalářské práce, na jejímž základě vznikl tento příspěvek, jsme zvolili a představili tři komerčně prodávané deskové hry, které nevznikly s primárně edukačním účelem, ale zároveň dokážou zaujmout dospělé jedince – vysokoškolské studenty (nejen) učitelství informatiky, kteří si potřebují nyní více než kdy dříve osvojit prvky IM, aby jej dokázali identifikovat a následně dobře navést své budoucí žáky k jeho efektivnímu rozvoji. Na základě [13] jsou námi vybrané hry logické a strategické. Zvolené deskové hry pracují s vlastní soutěživostí hráčů, ani jedna z nich není nutně kolaborativní (přímo kolaborativním hrám, kdy hráči musí spolupracovat, pokud chtějí vyhrát, a IM v nich aplikovaném se věnovali [12]). Uvedené hry pracují spíše se zvýšenou motivací v rámci porovnání s ostatními hráči, vynecháváme tedy záměrně

nutnou spolupráci hráčů ve hře, ačkoli se v hrách může vyskytovat a může být i přínosná. Pravidla her jsou pochopitelná i pro mladší hráče, než jsou vysokoškolsí studenti, i přesto hry dokážou nabídnout dospělým herní zážitek a zároveň je postavit před problémy a rozvinout jejich schopnost uplatnit IM. Jedná se o hry *Ricochet Robots*, *Gangster City* a *Osadníci z Katanu*.

2.1 Ricochet Robots

Ricochet Robots je hra s nejmenší příběhovou linkou ze zvolených her, zaměřuje se však přímo na představení algoritmů k prohledávání jako efektivního nástroje myšlení. Dala by se nazvat i logickou optimalizační úlohou, i přesto dokáže hráče pohltit. Hledají totiž cestu robota k cíli ve co nejmenším počtu tahů, přičemž mají stanovená pravidla jeho pohybu a cestu hledají v hlavě bez přímého pohybu postavy robota. Herní plocha je zobrazena na Obrázku 1.



Obrázek 1: Hra Ricochet Robots.

Pokud se v Ricochet Robots pokusíme najít alespoň částečně rozvíjené koncepty a postupy IM, určitě zde hráči uplatní a procvičí svou schopnost logického myšlení, stejně tak algoritmické myšlení a případnou aplikaci algoritmů sloužících k prohledávání stavového prostoru. Dá se hovořit o rozpoznávání vzorů pohybu robota po herním plánu. Hráči jsou nuceni vyhodnotit, zda cesta, kterou našli, je skutečně nejkratší, k čemu se nabízí i možnost automatizace, předání tohoto problému k řešení počítači.

2.2 Gangster City

Gangster City je moderní variací na hru *Logik* zasazenou do detektivního prostředí vyšetřování vraždy. Je zaměřena přímo na aplikaci logického myšlení a dedukci. Hráč na základě symbolů na kartách odhaluje, jaká karta byla jemu přiřazena (na kartě je znázorněna spáchaná vražda a symboly odhalují podrobnosti), přičemž jeho domněnky mu potvrzují či vyvrací jeho protihráči. Herní karty s obalem hry jsou ukázány na Obrázku 2.



Obrázek 2: Hra Gangster City.

I tato poměrně jednoduchá hra má strukturovaný tah snadno popsateľný algoritmem. Postupným zjišťováním a získáváním dalších informací hráč zapojuje logické myšlení a zhodnocuje správnost svých odhadů.

2.3 Osadníci z Katanu

Osadníci z Katanu jsou považovány za jednu z deskových her, která k „deskohraní“ přitáhla spoustu lidí z důvodu své skvělé herní mechaniky, zároveň se již od roku 1995 stále těší oblíbenosti, což dokazují i turnaje v ní pořádané. Hráči budují vesnice na herním plánu, obchodují mezi sebou či taktizují získáváním náhodných akčních karet. Prvek náhody zde zajišťují hody kostkou na začátku každého tahu, avšak vliv není tak výrazný jako např. v běžném *Člověče, nezlob se*.

Samotný herní proces hry Osadníci z Katanu má stanovenou strukturu tahů, přičemž každý tah začíná hodem kostkami, které určí, zda hráči získají suroviny na obchodování či stavbu vesnic, nebo jim bude uškozeno. Druhá fáze tahu je strategická a zabývá se obchodem, ať již se samotnou hrou či jinými hráči. Třetí fáze tahu je určena pro budování vesnic, které hráči umísťují na herní plochu. Správné umístění vesnic či měst je klíčové pro stanovení strategie hry, stejně tak zde hraje roli použití speciálních akčních karet, jež mohou být v různých částech hry více či méně účinné. Na Obrázku 3 je rozložený herní plán Osadníků z Katanu s figurkami tří různých hráčů.



Obrázek 3: Hra Osadníci z Katanu.

V této hře hráči musí velmi přemýšlet o svých tazích a uvažovat i do budoucna – již první umístění vesnic do herního plánu by mělo být strategické, jelikož může ovlivnit průběh celé hry. Algoritmický postup je zde zastoupen přesným definováním struktury tahu, kde se využívá i většinou podmínková logika. Kroky tohoto algoritmu se dají snadno zapsat do podoby kódu [13]. Hráči v průběhu hry vyhodnocují, jak se jim daří, a podle toho upravují svou dosavadní strategii a tah po tahu, část hry po další část hry, se postupným vývojem blíží k ukončení hry.

Berland [13] vyjádřil, že v Osadnicích z Katanu hráči dbají na dodržování pravidel, jelikož hra jim poté bude připadat zábavnější; pokud se jim však některá pravidla opravdu nezdají, mohou se dohodnout na jejich úpravě – v tomto případě bude IM společensky posíleno.

Prvky IM částečně rozvíjené v námi zvolených třech hrách jsou uvedeny v Tabulce 1, kde RR znamená Ricochet Robots, GC znamená Gangster City a OzK znamená Osadníci z Katanu. Doplňme ještě, že ve všech strategických či logických deskových hrách je vždy uplatňována jistá míra abstrakce a zároveň hráč vždy rozdělí hlavní problém (výhra ve hře) na menší části, které postupně vyřeší.

Tabulka 1: Prvky IM využívané v deskových hrách.

	RR	GC	OzK
Logické myšlení	✓	✓	
Algoritmy a algoritmické myšlení	✓	✓	✓
Vzory a jejich rozpoznávání	✓		✓
Abstrakce	✓	✓	✓
Hodnocení	✓	✓	✓
Automatizace	✓		
Rozklad problému na části	✓	✓	✓
Vytváření počítačových artefaktů			
Testování a ladění	✓	✓	✓
Postupný vývoj	✓	✓	✓
Spolupráce a kreativita	✓	✓	✓

3 NÁVRH VÝZKUMNÉHO ŠETŘENÍ

Aby byly hry efektivní v rozvoji IM, musí je i někdo hrát. Cílem proběhlého šetření bylo zjistit vztah studentů učitelství informatiky na ZČU k vybraným deskovým hrám a zda jejich hráči v nich dokážou identifikovat prvky IM. Částečně jsme ve výzkumu i testovali znalost pojmu IM.

Zvolili jsme smíšený výzkumný design – nejprve jsme na základě dotazníku oslovili všechny studenty, poté jsme s vybranými adepty (hráči zvolených her), kteří na sebe zanechali kontakt a chtěli se dále průzkumu zúčastnit, uskutečnili polostrukturované rozhovory. Testovacím vzorkem byli pouze studenti informatiky z důvodu alespoň částečného povědomí o pojmu IM, což mělo být zajištěno úvodem získaným z předmětu KVD/ALGV (Algoritmizace ve vzdělávání), kteří všichni v době výzkumného šetření buď již měli splněny či jej plnili.

Dotazníky byly rozeslány elektronicky, rozhovory se uskutečnily on-line videohovorem. Každému respondentovi rozhovoru byly položeny stejné otázky cílené na danou hru, aby nám sám sdělil, co může hra naučit nového, před jaké problémy je hráč postaven a poté, zda dokáže identifikovat prvky IM v hrách uplatňované. Pokud bylo nutné, byly položeny doplňující otázky reagující na respondentova slova.

4 VÝSLEDKY ŠETŘENÍ

Z šetření (návratnost dotazníku byla 47 %) vyplynulo, že více než čtvrtina respondentů pravidelně hraje různé deskové hry a necelé dvě třetiny alespoň příležitostně, což je dobrým předpokladem pro získání zájmu o deskové hry jako prostředku pro rozvoj IM. Strategické a logické hry (tedy hry s největším potenciálem k rozvoji IM) z dotazníkového šetření vzešly jako druhé nejoblíbenější deskové hry a soudě podle výsledků je s největší pravděpodobností více než polovina respondentů hraje.

Zajímavá a pro případný budoucí výzkum užitečná (ale i přesto nepovinná) otázka položená na konci dotazníku se respondentů ptala, zda je samotné napadání nějaká desková hra, která by mohla rozvíjet kompetence k řešení problému, a tím pádem i IM. Respondenti sami od sebe navrhli některé kooperativní hry (Pandemic), komplexní budovatelské či příběhové hry (Mars: Teraformace, Arkham Horror, Pán Prstenů desková hra,

This War of Mine), hry pro jednoho hráče (Club 2 %), které opravdu mohou IM rozvíjet. Někteří respondenti sami své návrhy zdůvodnili, např. u party hry Aktivity respondent zdůraznil, že si musí u pantomimy rozdělit celistvý úkol na menší části, další uvedli obecně strategické hry, které vyžadují „myslet vždy o krok dopředu“.

Na základě zájmu respondentů o další účast v průzkumu byli vybráni tři respondenti. Větší počet nebyl možný, kontakt na sebe zanechalo pouze pět respondentů, přičemž tři z nich byli vybráni tak, aby každý mohl mluvit o jedné z her a zároveň měli lepší povědomí o IM, což jsme usoudili podle odpovědí na otázky týkající se IM v dotazníku.

Všichni zúčastnění rozhovoru se k hrám dostali až na vysoké škole, jednomu z nich byla vybraná desková hra představena přímo jako hra rozvíjející IM. Každý z nich dokázal popsat jistou strategii, kterou je nutné v hrách uplatňovat. Všichni označili obecně deskové hry jako zdroj zábavy. Na doplňující dotaz dokázali všichni samostatně vyjádřit a zdůvodnit některé prvky IM obecně uplatňované v deskových hrách, zároveň se shodují, že hry mohou sloužit jako „trénink“ na situace v reálném životě a prohra v nich je podstatným prvkem sloužící ke změně uvažování. Respondenti dokázali vyjmenovat, v čem by mohly jejich pro rozhovor přidělená hry rozvíjet IM; nevyjmenovali všechny, ale dokázali si je odůvodnit. Dokážou tedy tyto hrané hry vnímat i jako rozvíjející.

5 ZÁVĚR

V článku založeném na bakalářské práci jsme se věnovali deskovým hrám a jejich potenciálu k rozvoji IM, zároveň nás zajímalo, jak jsou na tom se znalostí IM momentální studenti učitelství informatiky na ZČU FPE a zda sami dokážou v hrách prvky IM identifikovat.

Formulovali jsme význam pojmu IM a jeho charakteristické znaky – koncepty a postupy IM, stejně tak jsme přiblížili pojem hra a jeho souvislost s učením a myšlením. Deskové hry jsme představili jako prostředek pro rozvoj IM, jelikož ve všech logických a strategických deskových hrách jsou hráči postaveni před problémem, na který musí aplikovat IM.

Na základě vyjmenování charakteristických znaků IM jsme vybrali tři komerčně prodávané deskové hry, které jsme analyzovali a identifikovali v nich alespoň částečně rozvíjené prvky IM. Každá z her kladla větší důraz na jiné prvky (Ricochet Robots – algoritmické myšlení, Gangster City – logické myšlení, Osadníci z Katanu – aplikace strategie, zhodnocení dané situace, postupný vývoj a zhodnocení výhodnosti případné spolupráce hráčů).

Provedli jsme dotazníkové šetření, zjistili, že podstatná část respondentů deskové hry hraje. Získali jsme návrhy na další rozvíjející deskové hry, přičemž všechny byly podnětné. Naše vybrané hry převážně většina respondentů znala a polovina z nich alespoň jednu z nich hrála. V následných rozhovorech všichni respondenti dokázali správně vyjmenovat prvky IM, které je možné uplatnit a rozvíjet v dané deskové hře. Můžeme tedy shrnout, že je dokáží vnímat jako zábavný prostředek pro rozvoj IM.

Největší limitací a slabinou tohoto šetření je malý vzorek respondentů (celkem zasláno 110 studentům učitelství informatiky na ZČU, návratnost 47 %), stejně tak nízký počet rozhovorů způsobený i nízkým zájmem o další účast v průzkumu. Bylo by vhodné šetření zopakovat globálně na více univerzitách, což může být obsahem dalšího výzkumu.

6 BIBLIOGRAFICKÉ ODKAZY

- [1] MŠMT. *Rámcový vzdělávací program pro základní vzdělávání* [online]. Praha, 2021 [cit. 20. 2. 2023]. Dostupné z: <https://revize.edu.cz/files/rvp-zv-2021.pdf>
- [2] MŠMT. *Rámcový vzdělávací program pro gymnázia* [online]. Praha, 2022 [cit. 20. 2. 2023]. Dostupné z: <https://revize.edu.cz/files/001-rvp-gym-vyznacene-zmeny.pdf>
- [3] SOCHOROVÁ, L. Didaktická hra a její význam ve vyučování. *Metodický portál: Články* [online]. 26. 10. 2011, [cit. 14. 2. 2023]. Dostupné z: <https://clanky.rvp.cz/clanek/c/z/13271/DIDAKTICKA-HRA-A-JEJI-VYZNAM-VE-VYUCOVANI.html>. ISSN 1802-4785.
- [4] SUCHÁNKOVÁ, E. *Hra a její využití v předškolním vzdělávání*. Praha: Portál, 2014. ISBN 978-80-262-0698-9.
- [5] PUGNEROVÁ, M. *Psychologie: pro studenty pedagogických oborů*. Praha: Grada. Pedagogika (Grada), 2019. ISBN 978-80-271-0532-8.
- [6] *Professor Seymour Papert*. [online]. Copyright 2000. MaMaMedia [cit. 14. 2. 2023]. Dostupné z: <http://www.papert.org/>.
- [7] WING, J. M. Computational thinking. *Communications of the ACM* [online], 2006, **49**(3), 33–35 [cit. 15. 2. 2023]. ISSN 0001-0782. Dostupné z: doi:10.1145/1118178.1118215.
- [8] DENNING, P. J., TEDRE, M. *Computational thinking*. Cambridge, Massachusetts: The MIT Press, 2019. MIT Press essential knowledge series. ISBN 978-0-262-53656-1.
- [9] imysleni.cz. Co je infromatické myšlení? *Informatické myšlení*. [online]. Copyright 2018. [cit. 15. 2. 2023]. Dostupné z: <https://www.imysleni.cz/informaticke-mysleni/co-je-informaticke-mysleni>.
- [10] GROVER, S., PEA, R.. Computational thinking: a Competency Whose Time Has Come. In: SENTANCE, S. ed., BARENDSSEN, E. ed., SCHULTE, C. ed. *Computer science education: perspectives on teaching and learning in school*. London: Bloomsbury Academic, 2018, s. 19–39. ISBN 978-1-350-05710-4.
- [11] WANG, P. S. *From computing to computational thinking*. Boca Raton: CRC Press, Taylor & Francis Group, ©2016. ISBN 978-1-4822-1765-1.
- [12] BERLAND, M., LEE, V. R. Collaborative Strategic Board Games as a Site for Distributed Computational Thinking. *International Journal of Game-Based Learning* [online], 2011, vol. 1, iss. 2, 65–81 [cit. 18. 2. 2023]. ISSN 2155-6849. Dostupné z: doi:10.4018/ijgbl.2011040105.
- [13] BERLAND, M. Understanding Strategic Boardgames as Computational- Thinking Training Machines. In: COSTIKYAN, G. ed. a DAVIDSON, D. ed. *Tabletop: analog game design*. Pittsburgh, PA, USA: ETC Press, 2011, s. 167–173. ISBN 978-1-257-87060-8.

Virtuální třída jako nástroj na zdokonalování didaktických dovedností studentů učitelských programů

Virtual classroom as a tool to improvement of teacher programs students' didactical skills

Mgr. Miroslav Zíka
Západočeská univerzita
Klatovská tř. 51
306 14 Plzeň
Česká republika
zikam@kv.d.zcu.cz

ABSTRACT

This article is focused on description and development of the Virtual Classroom model, which we are continuously testing and further developing at the Faculty of Education of the University of West Bohemia in Pilsen. Main modifications are made largely on graphical and user level and we can observe an increasing level of respondents involvement compared to the earlier version. It struggled mostly with the lack of graphical realism and the imperfections of the previous model. We also got positive feedback in the case provide opportunity to try out the application and to realize a short pedagogical output with feedback from the didactician.

Keywords

Immersive virtual reality. Development. Education. Training.

ABSTRAKT

Tento článek se zaměřuje na popis a vývoj modelu Virtuální třídy, který průběžně testujeme a na základě získaných dat dále vyvíjíme na Fakultě pedagogické Západočeské univerzity v Plzni. Úpravy jsou prováděny z velké míry především na grafické i uživatelské rovině, ze získaných šetření můžeme sledovat vzrůstající míru vtažení respondentů oproti dřívější verzi. Ta se potýkala převážně s nedostatečnou grafickou realností a nedokonalostí modelu. Získáváme také pozitivní zpětnou vazbu v podobě vyzkoušení si samotné aplikace a poskytnutí možnosti realizovat krátký pedagogický výstup nanečisto se zpětnou vazbou od přítomného didaktika.

Klíčová slova

Imersní virtuální realita. Vývoj. Vzdělávání. Trénování.

1 ÚVOD

Virtuální třída je aplikace vyvíjená za účelem poskytnutí bezpečného prostředí studentům Pedagogické fakulty Západočeské univerzity v Plzni k cvičné výukové činnosti. Toto uměle vytvořené prostředí jim primárně umožňuje zdokonalovat pedagogické a didaktické dovednosti, nicméně lze aplikaci využít také za účelem osvojení si komunikačních praktik v pozici učitele. Nedílnou součástí je přítomnost učitele-didaktika, který za kooperace menšího počtu studentů poskytuje zpětnou vazbu k samotnému výstupu. Díky variabilnímu prostředí je možné připravit takřka libovolné situace. Na vývoj jsou kladeny podmínky autentičnosti a skutečnosti za účelem dosažení co největší míry zaujetí studenta ve virtuální realitě.

Zdrojem inspirace pro funkcionality a samotné prostředí byly pedagogické projekty simulující výuku v klasické třídě. Jedná se především o projekt TeachLivE, třídu o pěti žácích ve smíšené realitě. Tento projekt je zaměřen na interakce učitel-žák, přičemž jednotliví žáci jsou ovládáni z vedlejší místnosti [1]. Obdobnými projekty, lépe řečeno spíše herní platformou, byly aplikace Second Life a Active Words. Ty nabízely mnohem širší možnosti aktivit a do pedagogických výzkumů byly zapojeny v rámci akademických projektů ([2], [3]).

Oproti zmíněným projektům využívá Virtuální třída imerzní virtuální realitu. Tu mají studenti k dispozici skrze zařízení HTC Vive, které se skládá ze zobrazovacího zařízení v podobě brýlí či helmy (tzv. headsetu), série snímacích zařízení a ručních ovládačů. Pohyb ve virtuální realitě není neomezený, ale je spjat s prostorovými možnostmi místnosti v reálném světě [4]. Dané omezení je možné minimalizovat za využití softwarových doplňků umožňujících nezávislý přesun ve virtuální realitě, např. funkce teleportu či pohybu zástupného avatara za použití ovládačů.

Počátky virtuální reality lze datovat do roku 1966, kdy za pomoci podobné technologie docházelo k výcviku pilotů United States Air Force. Širšímu využití ve veřejnosti došlo až v 90. letech, kdy herní společnosti, např. SEGA a Nintendo, představily první hry s virtuální realitou a vlastní verzí headsetu. Tyto prvotní návrhy se neseťkaly s úspěchem [5]. Pravděpodobně kvůli nedostatečné míře přesvědčivosti a omezeným možnostem starých technologií. Pro využití virtuální reality ve vzdělávání je dle L. A. Diekera [6] důležité, aby vytvořené prostředí umožňovalo odbourání nedůvěry a opakované výstupy vedly k systematickému procesu učení.

V současnosti se simulace a trénování pomocí virtuální reality využívá v mnohých oblastech. Roshan Ganeshan ve svém projektu *Emerging Technologies* [7] kromě využití v armádě (simulátor jízdy tankem, simulace bojů, výuka kultury a jazyka na cizím území, aj.) zvažuje širší nasazení ve vzdělávání, např. v rámci autoškoly či pro muzejní pedagogiku. Virtuální realita své uplatnění nachází také během výcviku jednotek záchranného sboru, blíže u jednotek hasičů, kde je možné simulovat zásah obsahující riziko kontaktu s jedovatou či radioaktivní látkou. Lze takto realizovat těžko připravitelné situace, např. bioteroristický útok [8].

Pomocí Virtuální třídy byly realizovány dva výzkumy. První z nich se věnoval možnostem a praktickému využití v přípravě budoucích učitelů magisterského studijního programu. Účastníci výzkumu ve třech opakovaných výstupech seznamovali žáky na předem vybrané téma z učiva Přírodopisu a Informatiky (látka

byla vybrána pro žáky 2. stupně základní školy). S pomocí přítomných didaktiků a ostatních studentů proběhla po každém výstupu diskuze, která shrnovala problémové prvky proběhlé výuky. V následujícím výstupu, který byl opakován po týdenní lhůtě, byl sledován postup zlepšení a alterace výuky dle připomínek.

Předmětem druhého výzkumu byla komunikace učitele s rodiči studentů, např. na třídní schůzce. Jednalo se o schůzku na předem domluvené téma (zdražení cen obědů, zrušení zájmového kroužku aj.), jehož průběh ale nebyl relativně připravený jako předchozí zmíněný. Problematika konverzace s rodiči spočívala ve změněné roli učitele vůči ovládaných avatarů, kdy obě zúčastněné strany jsou na stejné úrovni.

2 METODOLOGIE

Vývoj Virtuální třídy je realizován konceptem *design-based research*. Tento výzkum v roce 1992 definovala Ann Brown jako *“efektivní intervenci, která by měla být schopna přejít z naší experimentální třídy do průměrných tříd s průměrnými žáky a učiteli, za podpory reálné technologické a personální podpory.”*. Dále ho lze chápat jako interaktivní a participativní výzkum, který přemostňuje propast mezi akademickým a vzdělávací praxí ([9], [10]). Jedná se o výzkum, který slučuje dvě relativně vzdálené prostředí a umožňuje tím velmi širokou spolupráci a nové pohledy na vzdělávací problémy. Štemberge a Cencič člení tento výzkum na následující 4 etapy:

1. etapa

Jedná se o klíčovou etapu, ve které dochází k definování primárního výzkumného problému. To zahrnuje stanovení výzkumných otázek a hypotéz, jejichž validita se bude zkoumat. Nedílnou součástí této fáze je také rešerše literárních zdrojů. Počáteční etapa by se dala pokládat za přípravnou část celého procesu, na kterém se podílí výzkumníci, praktici a dle tematického zaměření výzkumu také odborníci na danou problematiku.

2. etapa

Po definování výzkumného problému a jeho dílčích otázek je nutno vybrat vhodné metody k jejich zkoumání, způsob sběru dat a naplánování jejich využití. Je nutno pečlivě zvážit časovou náročnost, finanční zdroje, případná omezení a znalosti členů výzkumného týmu. Vše vychází z předešlé přípravy literatury. V tomto případě tato fáze zahrnuje také modelování a implementaci rozhraní Virtuální třídy. Obecně lze říci, že výstupem druhé etapy je podrobný harmonogram výzkumu a příprava všech potřebných nástrojů (zahrnující také výběr zkušební vzorku).

3. etapa

Během třetí etapy dochází k realizaci připraveného výzkumu a sběru dat pomocí všech členů pracovního týmu. Může se jednat minimálně o dvě iterace zavádění inovací a vyhodnocování proběhlých změn. Velmi často je ale tento počet navýšen na tři až čtyři iterace, čímž dochází k získání většího objemu dat a rychlejšímu posunu inovace.

4. etapa

V poslední fázi dochází vyhodnocování proběhlého výzkumu. Zde v závislosti na získaných datech může dojít k finálnímu návrhu inovace a završení problému, popřípadě dochází k úpravě

metodiky a opakování výzkumu z počátečních fází. Návrat do dřívějších fází umožňuje alteraci výzkumných otázek, metodiky či dílčích nástrojů a jejich opětovné nasazení do cyklického ověřování.

2.1 Alfa verze Virtuální třídy

Na Virtuální třídě se podílí dvě fakulty – Fakulta pedagogická (Katedra výpočetní a didaktické techniky, Centrum biologie, geověd a envigogiky) a Fakulta strojní (Katedra průmyslového inženýrství a managementu). Tato mezioborová spolupráce, která navázala na předchozí zkušenosti Fakulty strojní s obdobným virtuální projektem, dala vzniknout prostředí umožňující simulovat výuku studentům pedagogických oborů. Tohle prostředí umožnilo vstoupit do předpřipravené třídy a řešit problémy v naplánovaných či improvizovaných situacích. Ve verzi byla k dispozici mapa světa, která umožňovala výuku klíčových bodů kurikula zeměpisu.



Obrázek 1: Alfa verze Virtuální třídy (volný pohled).

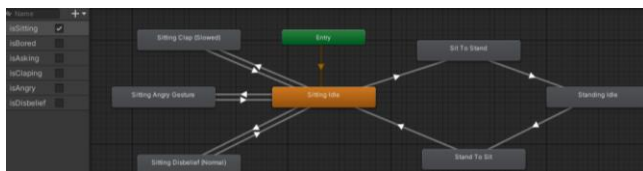
2.2 Vývoj

Vývoj aplikace je realizován v multiplatformním herním enginu Unity. Jedná se vývojové prostředí (může být označováno jako framework), které obsahuje knihovny a nástroje (např. Animator, vykreslování 3D scény aj.), pomocí nichž je značně urychlen a usnadněn proces vývoje. V případě potřeby doplňujících funkcionalit je možné využít placené doplňky. Kromě uzpůsobených nástrojů, které se využívají pomocí grafického rozhraní, jsou v projektu aplikovány i vlastní C# skripty. Vzhledem k časté aktualizaci Unity enginu je využívána verze 2019.4.21.

Způsob programování v herním frameworku Unity lze v případě Virtuální třídy popsat na ovládání avatarů. K tomuto účelu jsou využity následující konstrukty:

Animator

Jedná se o defaultní komponentu, která je ve frameworku Unity k dispozici. Jejím primárním účelem je kontrolovat a ovládat přechod mezi jednotlivými animacemi modelu v závislosti na stavu definovaných proměnných o datovém typu bool. Součástí modulu je grafické rozhraní, které umožňuje měnit hodnotu proměnných. Na obrázku níže jsou znázorněny animace avatarů, jenž v počátečních podmínkách (vstup na scénu) sedí na židličkách. Pro přehlednost jsou názvy proměnných shodné se stavem, kterému odpovídají. Přechod do jiného stavu je inicializován změnou hodnot. Komponenta nabízí širší možnosti úpravy – délku trvání přechodu mezi jednotlivými stavy, plynulost přechodu, zda lze stav opustit kdykoliv, nebo jen na konci přidružené animace aj. Tento nástroj umožňuje spravovat animace získané pomocí webového nástroje Mixano.com.



Obrázek 2: Znárodnění stavů, komponenta Animator.

FinalIK

Pro tvorbu vlastních animací lze využít doplněk FinalIK, který lze zakoupit na Unity Asset Store. Tento nástroj rozšiřuje rozhraní Unity a umožňuje inicializovat animace pomocí skriptů. Za tímto účelem vkládá do modelů vlastní kostru, jejichž klouby lze libovolně umisťovat a měnit jejich orientaci. Ve Virtuální třídě je využit pro animaci hlášení, k čemuž využívá pomocný bod – na obrázku níže je vykreslen jako zelený kruh. FinalIK je použit v kombinaci s manuálně napsanými třídami State Machine, jenž částečně duplikují dříve zmíněný Animator.



Obrázek 3: Kostra FinalIK.

State Machine controller

State Machine je veřejná třída skládající se z množiny stavů. Pomocí ní je možné definovat chování jednotlivých stavů, přičemž rozšiřuje dříve zmíněný Animator. Ten pouze definuje stavové proměnné a přechody, neumožňuje ale nijak měnit hodnoty v proměnných. Každá třída stavu odpovídá následující struktuře:

public void Enter()

První veřejná funkce, která se automaticky volá při přechodu do daného stavu. Jsou v ní inicializované potřebné proměnné a volány soukromé funkce třídy. Nejdůležitější proměnnou je zde animator, do které je pomocí funkce GetComponent() uložena stejnojmenná komponenta.

public void HandleInput()

V této funkci jsou do proměnných ukládány příznaky zmáčknutých kláves. Jedná se tedy o případný listener, který je cyklicky volán. Použité klávesy a jejich pojmenování je v Unity možné spravovat pomocí Input Manageru.

public void LogicUpdate()

Tato cyklicky volána metoda slouží pro změnu stavu objektu stateMachine. Případná změna má za následek volání funkce Exit() současného stavu a volání funkce Enter() stavu, do kterého je na základě stisknuté klávesy přecházeno. Jedná se tedy o skriptem ovládaný přechod mezi stavy.

public void Exit()

Jedná se o funkci, která je volána na konci cyklu třídy. Lze ji využít např. pro změnu proměnné Animatoru.

```
animator.SetBool("isAngry", false);
```

Obrázek 4: Změna stavu objektu Animator.

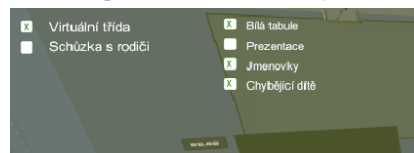
Kromě herního frameworku Unity je využíván repositář GitHub, pomocí něhož je možné sledovat proběhlé změny, případně ho lze využít pro obnovu starší verze projektu. Kromě funkce porovnání jednotlivých variant se verzovací software využívá pro kooperaci práce více programátorů, kdy při každém nahrání změny do sdíleného úložiště je prováděna kontrola na datové konflikty.

Pro správné spuštění aplikací využívající virtuální realitu je nutno mít nainstalovanou platformu Steam a volně dostupný nástroj Steam VR, který je možné stáhnout pomocí Steam klienta. Aplikace je exportována do samostatně spustitelného souboru a přidružených dat. Poslední verze v součtu zabírá 462 MB. Virtuální třídu je možné spouštět na počítačích, jehož komponenty jsou výrobcem označovány jako VR ready. Jedná se především o grafické karty a procesory. Minimální požadavky se liší v závislosti na používaném headsetu a rozlišení jeho obrazu. Pro zařízení HTC Vive, jenž je na Pedagogické fakultě využíváno, je v minimálních požadavcích uvedena grafická karta NVIDIA GTX 1060 (resp. AMD Radeon RTX 480) a procesor Intel Core i5-4590 (resp. AMD FX 8350). Nutná velikost RAM paměti je minimálně 8 GB [11].

2.3 Virtuální třída v současnosti

Z pilotního ověřování bylo získáno mnoho podmětů do dalšího rozvoje aplikace. Patrně největší nedostatek alfa verze byla absence základního grafického rozhraní, pomocí něhož by bylo možné upravovat pomůcky ve třídě či ji navrátit do původní podoby před zahájením výuky. Mezi jednotlivými výstupy studentů bylo nutno aplikaci vypnout a zapnout, aby byly opětovně zajištěny výchozí podmínky. Z toho důvodu došlo k vytvoření více pomocných částí. Současnou verzi lze z hlediska grafického rozhraní rozčlenit na tři navzájem provázané sekce:

- hlavní nabídka,
- nastavení jednotlivých scén s možností úprav,
- pracovní část, prostředí Virtuální třídy (Obrázek 6).



Obrázek 5: Grafické rozhraní sekce Nastavení.

Třídy odpovídají standardům OOP a obsahují pravdivostní proměnné používané k vypínání/zapínání přídatných funkcí a metody zajišťující získávání dalších textových informací. Během spuštění aplikace jsou do složky aplikací (AppData) vytvořeny dočasné soubory formátu .json a následně je s nimi manipulováno pomocí grafického rozhraní (v sekci Nastavení). V rámci vývoje jsou tyto soubory mazány při ukončení aplikace, aby nedocházelo k nekonzistentnosti nabídek. Levá část nastavení zahrnuje globální hodnoty. Jedná se především o výběr rozvržení kamer scénářů, viz níže, a funkci teleportu, která je přístupná pouze v rámci ladění programu. S pomocí grafického rozhraní bylo zajištěna modulárnost i bez přímé úpravy zdrojového kódu, které s sebou nese potenciální riziko vzniku chyby. Sekce

nastavení obsahuje dále modul umožňující volbu pohlaví avatara sedícího na konkrétní pozici ve třídě.

Spuštění vybraného scénáře (Virtuální třída a Schůzka s rodiči) je možné z hlavního menu. V rámci realizace výzkumů a propagačních ukázek jsou připraveny tři zobrazovací režimy, mezi nimiž je možné přepínat v globálním nastavením. Jedná se o rozvržení:

- zobrazení odpovídající pouze pohledu HTC Vive,
- dělené zobrazení pohledu headsetu a zadní kamery,
- zobrazení zahrnující pohled headsetu, zadní kamery a webkamery snímající respondenta.

První z výše uvedených kombinací slouží pouze k zobrazení pohledu respondenta ve Virtuální třídě, zbylé kombinace zahrnují zadní pohled. Jedná se o pohled, který slouží ke dvěma účelům – umožňuje částečně pozorovat pohyb uživatele v prostoru třídy a současně je didaktik, resp. uživatel za počítačem, schopen ovládat avatary (žáky či rodiče). Červený indikátor zobrazuje vybrané avatary, které v případě klávesového vstupu provedou požadovanou akci.



Obrázek 6: Snímek pohledu respondenta a ovládání žáků.

V závislosti na uplatněném nastavení může vyučující respondent využít možnost prezentace a bílou tabuli se 4 barevnými fixy. Tato plocha umožňuje psát libovolné poznámky sloužící jako podpůrné texty k výkladu. Jedinou nevýhodou je nutnost osvojit si ovládání a způsob psaní, který je závislý na interakci modelů fixy a tabule. Z toho důvodu začíná cyklické testování první iterací, v níž si respondent vyzkouší pohyb ve virtuálním prostředí, uchop předmětu, odhození předmětu a psaní. Tohle krátké vyzkoušení slouží také k ověření studentovy reakce na „vstup“ do virtuální reality, během něhož se můžou projevit příznaky virtual sickness.

3 ZÁVĚR

Tento článek je zaměřen na nástroj Virtuální třída, který je vyvíjen na Pedagogické fakultě Západočeské univerzity v Plzni. Stručně představuje hlavní grafické úpravy alfa verze, kdy došlo k implementaci základního uživatelského rozhraní umožňující zásah do používaného modelu bez přímého zásahu do kódu. Vytvořené rozhraní umožňuje široké možnosti rozšíření pomocí jednoduchého skrývání či zobrazování příslušné vrstvy a k ní přiřazených ovládacích prvků.

Proběhlé pedagogické výzkumy prozatím zkoumaly adekvátnost zařazení nástroje jako doplňujícího prostředku před realizací pedagogických praxí. Během realizace zmíněných výzkumů bylo

kromě primárních aspektů vyzozorována také různá míra pohlčení prostředí na sledované respondenty, tj. vyučující studenty učitelství.

Z proběhlých iterací výuky a zpětné vazby od studentů je nejvíce patrná omezená variace chování avatarů. Svým kamenným pohledem působily na některé respondenty nepříjemně a tím docházelo ke snižování celkové dojmu imerze prostředí. Právě z toho důvodu bychom na projektu dále pracovali ve směru návrhu optimálního ovládání avatarů a rozšiřování jejich činností. Velkým přínosem by bylo také umožnění přímého ovládání modelů dětí/dospělých přisedícími studenty. Alternativou k zúčastněnému ovládání by byla ucelená sada činností, kterou by nezávisle na modelu spouštěl aplikaci obsluhující člověk (např. povstání modelu, uchopení předmětu, hození předmětu po učiteli, útek ze třídy aj.).

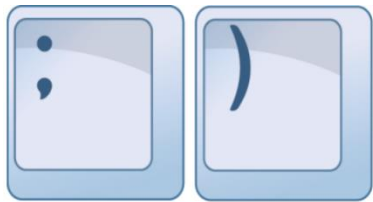
Velmi důležitým cílem nadcházejícího vývoje je také rozšíření aplikace o větší množství prostorů. V plánu je realizace celistvé školy, což zahrnuje specializované učebny pro výuku odborných předmětů, společné prostory nabízející řešení výchovně problematických situací či pouhé kanceláře, v nichž by studenti mohli navícovat přímou komunikaci s rodičem či žákem. Samozřejmě není dostačující pouze rozšiřovat množství využitelných modelů, ale také zvyšovat jejich grafickou realnost a nabídnout široké a přesvědčivé modely lidí. U modelů v současné době evidujeme dva klíčové nedostatky, které bychom v budoucnu rádi vyřešili – strnulost pohledu způsobenou absencí mrkání a nepohybování rtů.

BIBLIOGRAFICKÉ ODKAZY

- [1] M.J Kelley & T. Wenzel, "A Parent-Teacher Reading Conference Project: Using a Virtual Environment (TeachLive™) to Improve Elementary Pre-Service Teachers' Conferencing Skills" *The Reading Professor*, vol. 41, no. 1, Article 22, 2018. Dostupné z: <https://scholar.stjohns.edu/thereadingprofessor/vol41/iss1/22>
- [2] A. Santos "The SimEscuela: An innovative virtual environment for teacher training." *Engaging the Avatar: New Frontiers in Immersive Education*, pp. 365-390, 2012.
- [3] R. Lorenzo-Alvarez, T. Rudolphi-Solero, M.J. Ruiz-Gomez & F. Sendra-Portero, "Game-Based learning in virtual worlds: a multiuser online game for medical undergraduate radiology education within second life." *Anatomical sciences education*, vol. 13, no. 5, pp. 602-617, 2020.
- [4] A.S. Won, J. Bailenson, J. Lee & J. Lanier, "Homuncular flexibility in virtual reality." *Journal of Computer-Mediated Communication*, vol. 20, no. 3, pp. 241-259, 2015.
- [5] S. Kavanagh, A. Luxton-Reilly, B. Wuensche, & B. Plimmer, "A systematic review of virtual reality in education." *Themes in Science and Technology Education*, vol. 10, no. 2, pp. 85-119, 2017.
- [6] L. A. Dieker, J.A. Rodriguez, B. Lignugaris/Kraft, M.Ch. Hynes & C.E. Hughes, "The potential of simulated environments in teacher education: Current and future possibilities." *Teacher Education and Special Education*, vol. 37, no. 1, pp. 21-33, 2014.
- [7] R.K. Ganeshan, "Emerging Technologies - Oculus Rift: The next level of perception." Luton: Computer Department Bedfordshire University, 2015. Dostupné z:

https://www.researchgate.net/publication/276244192_Oculus_Rift_A_Single_Page_Study

- [8] S. Stansfield, D. Shawver, A. Sobel, M. Prasad & L. Tapia "Design and implementation of a virtual reality system and its application to training medical first responders." *Presence: Teleoperators and Virtual Environments*, vol. 9, no. 6, pp. 524-556, 2000.
- [9] A. Brown, "Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings." *Journal of the Learning Sciences*, vol 2, no. 2, pp. 141-178, 1992.
- [10] T. Štemberger & M. Cencič, "Design-Based Research: the Way of Developing and Implementing Educational Innovation", *World Journal on Educational Technology*, vol. 8, 2016. Dostupné z: https://www.researchgate.net/publication/309889224_Design_Based_Research_the_Way_of_Developing_and_Implementing_Educational_Innovation
- [11] RoadToVR, "How to Tell if Your PC is VR Ready" [online; cit. 17.3.2023]. 3.11.2023. Dostupné z: <https://www.roadtovr.com/how-to-tell-pc-virtual-reality-vr-oculus-rift-htc-vive-steam-vr-compatibility-tool/>



WORKSHOPY

Workshop 1

Zuzana Tkáčová

Umelá inteligencia v školskej informatike - Kreatívne multimediálne nástroje

Umelá inteligencia sa ukazuje byť potrebnou súčasťou kurikula školskej informatiky, aby sme boli schopní zodpovedne pripraviť mladú generáciu na výzvy, ktoré umelá inteligencia prináša a bude prinášať do života spoločnosti. Vďaka dostupným nástrojom vhodným pre rôzne vekové skupiny žiakov základných aj stredných škôl je možné vyučovať túto tému zaujímavým a atraktívnym spôsobom. V tejto časti workshopu sa zameriavame na kreatívne multimediálne nástroje využívajúce umelú inteligencia pri práci s grafikou, animáciami a videom.

Workshop 2

Patrik Voštinár

CoSpaces - prostredie na vytvorenie AR a VR aplikácií počas výučby na ZŠ a SŠ

V súčasnosti sa rozšírená a virtuálna realita teší veľkej popularite u rôznych vekových kategórií. V rámci workshopu si predstavíme online prostredie, ktoré umožňuje pomerne rýchlo vytvoriť virtuálne prostredie, v ktorom môžeme programovať pomocou blokov a jazykov Python a TypeScript. V rámci workshopu si vytvoríte aplikáciu pre rozšírenú realitu, 360 stupňovú fotku s doplnenými objektami a virtuálne prostredie s rôznymi pohybujúcimi sa objektami.

Workshop 3

Spoločnosť EDUXE

LEGO Education SPIKE Prime pre 2. stupeň základných a stredných škôl

V rámci workshopu si vyskúšame edukačnú pomôcku - stavebnicu Spike Prime vhodnú pre 2. stupeň ZŠ a SŠ. Menšiu časť workshopu budeme venovať aj stavebniciam Spike Essential pre 1. stupeň a Lego Education BricQ Motion.

Workshop 4

Zuzana Tkáčová

Umelá inteligencia v školskej informatike – Programátorské nástroje

Umelá inteligencia sa ukazuje byť potrebnou súčasťou kurikula školskej informatiky, aby sme boli schopní zodpovedne pripraviť mladú generáciu na výzvy, ktoré umelá inteligencia prináša a bude prinášať do života spoločnosti. Vďaka dostupným nástrojom vhodným pre rôzne vekové skupiny žiakov základných aj stredných škôl je možné vyučovať túto tému zaujímavým a atraktívnym spôsobom. V tejto časti workshopu bude venovaná programátorským nástrojom, ktoré umožňujú žiakom využívať vo svojich programátorských projektoch predtrénované modely alebo vytvárať vlastné prostredníctvom strojového učenia. Zamyslíme sa taktiež nad úlohou, ktorú zohrávajú dáta v procese učenia sa umelej inteligencie a analyzujeme limity a príležitosti umelej inteligencie v rôznych aplikáciách.



S P O N Z O R I

Medzinárodná konferencia DIDINFO 2023 sa koná s podporou:

SOFTIP®

GlobalLogic®
A Hitachi Group Company

NEXTTECH

Continental 
The Future in Motion


ŽELEZIARNE®
PODBREZOVÁ

EDUXE

 American Spaces
Slovakia

Názov: **DidInfo 2023**
Editori: Mgr. Adam Dudáš, PhD., RNDr. Alžbeta Michalíková, PhD.,
PaedDr. Patrik Voštinár, PhD., doc. Ing. Jarmila Škrinárová, PhD.
Vydanie: 1. vydanie
Vydavateľ: Univerzita Mateja Bela, Fakulta prírodných vied v Banskej Bystrici, Slovensko
Rok: 2023
Rozsah: 191 strán
Formát: elektronický, konferenčný zborník
ISBN: 978-80-557-2038-8
EAN: 9788055720388
ISSN: 2454-051X
DOI: <https://doi.org/10.24040/2023.9788055720388>



Táto publikácia je šírená pod licenciou Creative Commons Attribution 4.0 International Licence CC BY.

Title: **Proceedings of conference DidInfo 2023**
Editors: Mgr. Adam Dudáš, PhD., RNDr. Alžbeta Michalíková, PhD.,
PaedDr. Patrik Voštinár, PhD., doc. Ing. Jarmila Škrinárová, PhD.
Edition: First edition
Publisher: Matej Bel University, Faculty of Natural Sciences in Banská Bystrica, Slovakia
Year: 2023
Pages: 191
Format: electronic, conference proceedings
ISBN: 978-80-557-2038-8
EAN: 9788055720388
ISSN: 2454-051X
DOI: <https://doi.org/10.24040/2023.9788055720388>



This publication is distributed by the Licence Creative Commos Attribution 4.0 International Licence CC BY.