

ZÁPADOČESKÁ UNIVERZITA V PLZNI

FAKULTA PEDAGOGICKÁ

KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

**Tvorba distančního kurzu předmětu Algoritmizace
pro vzdělávání na vysokých školách**

DIPLOMOVÁ PRÁCE

Bc. Viktor Chejlava

Učitelství pro základní školy, obor Učitelství informatiky pro základní školy

Vedoucí práce: PhDr. Tomáš Jakeš, Ph.D.

Plzeň 2018

Prohlašuji, že jsem diplomovou práci vypracoval samostatně
s použitím uvedené literatury a zdrojů informací.

V Plzni, 2. července 2018

.....
vlastnoruční podpis

OBSAH

SEZNAM ZKRATEK	4
ÚVOD	5
1 VYMEZENÍ ZÁKLADNÍCH POJMŮ	6
1.1 ALGORITMUS	6
1.2 ALGORITMICKÉ MYŠLENÍ	8
1.3 ALGORITMIZACE	9
2 METODY URČENÉ PRO ROZVOJ ALGORITMICKÉHO MYŠLENÍ	10
3 NÁSTROJE PRO ROZVOJ ALGORITMICKÉHO MYŠLENÍ	14
3.1 APLIKACE PRO ROZVOJ ALGORITMICKÉHO MYŠLENÍ	14
3.1.1 Scratch	14
3.1.2 Kodu.....	15
3.1.3 EasyLogo	17
3.1.4 Hour of Code.....	18
3.1.5 Code combat.....	19
3.2 ROBOTI A ROBOTICKÉ STAVEBNICE PRO ROZVOJ ALGORITMICKÉHO MYŠLENÍ.....	20
3.2.1 Bee-bot	20
3.2.2 Dash a Dot	21
3.2.3 Ozobot	22
3.2.4 Lego mindstorms	23
4 E-KURZ PŘEDMĚTU ALGORITMIZACE VE VZDĚLÁVÁNÍ	25
4.1 ZÁKLADNÍ POPIS KURZU.....	25
4.2 POUŽITÝ LEARNING MANAGEMENT SYSTEM.....	26
4.3 CÍLE KURZU	27
4.4 OBSAH KURZU A JEHO STRUKTURA	28
4.4.1 Použité činnosti a studijní materiály	28
4.4.2 Použití dalších nástrojů.....	29
4.4.3 Použitá multimédia.....	31
4.5 STRUKTURA JEDNOTLIVÝCH KAPITOL.....	31
4.5.1 Úvod	31
4.5.2 Úvod do algoritmizace.....	31
4.5.3 Vývojové diagramy	32
4.5.4 Strukturogramy	32
4.5.5 Rozvoj algoritmizace u dětí a studentů na ZŠ a SŠ	32
4.5.6 Programujeme ve Scratchi.....	33
4.5.7 Programujeme s LEGO Mindstorms	34
4.5.8 Závěr	35
4.6 PŘÍNOS KURZU	35
4.6.1 Náročnost kurzu	35
4.6.2 Srozumitelnost kurzu.....	36
4.6.3 Podrobnost kurzu	36
4.6.4 Dostatečnost multimédií	37
4.6.5 Dostatečnost cvičení v kurzu	38
4.6.6 Přehlednost kurzu.....	39
4.6.7 Zajímavost kurzu.....	40
4.6.8 Použití externích zdrojů.....	41
4.6.9 Další připomínky	42

4.7	NÁVRH VYLEPŠENÍ KURZU	42
5	VÝZKUMNÉ ŠETŘENÍ	44
5.1	METODOLOGIE VÝZKUMU	44
5.2	CHARAKTERISTIKA VÝZKUMNÉHO SOUBORU.....	44
5.3	ČASOVÝ HARMONOGRAM VÝZKUMU	44
5.4	POPIS DOTAZNÍKŮ	45
5.5	CHARAKTERISTIKA RESPONDENTŮ	45
5.5.1	Věk studentů	45
5.5.2	Vystudovaná SŠ.....	46
5.5.3	Obsah hodin.....	46
5.5.4	Programování	47
5.5.5	Předchozí studium na vysoké škole.....	48
5.5.6	Programování a volný čas.....	49
5.5.7	Tvorba vývojových diagramů a strukturogramů	50
5.5.8	Tvorba programu.....	51
5.5.9	Praktická zkušenost s dětskými programovacími prostředími	51
5.5.10	Problémy při programování.....	52
5.5.11	Jste nebo chcete se stát učitelem/učitelkou?	53
5.6	POSTOJE PŘI VSTUPNÍM TESTOVÁNÍ.....	54
5.6.1	Obtížnost testové části	54
5.6.2	Postoje programování a algoritmizace	54
5.6.3	Student oboru Informatika se zaměřením na vzdělávání by měl skvěle programovat 55	
5.6.4	Je programování pro studenty náročné?	56
5.6.5	Rozkládání velkých problémů na podproblémy	57
5.6.6	Vytrvalost studentů	58
5.6.7	Nejdříve algoritmizace a poté programování?	59
5.6.8	Algoritmizace na základní škole.....	60
5.6.9	Algoritmizace a on-line kurz	61
5.7	POSTOJE PŘI VÝSTUPNÍM TESTOVÁNÍ.....	62
5.7.1	Obtížnost testové části	62
5.7.2	Postoje programování a algoritmizace	63
5.7.3	Student oboru Informatika se zaměřením na vzdělávání by měl skvěle programovat 64	
5.7.4	Je programování pro studenty náročné?	64
5.7.5	Rozkládání velkých problémů na podproblémy	65
5.7.6	Vytrvalost studentů	65
5.7.7	Nejdříve algoritmizace a poté programování?	66
5.7.8	Algoritmizace na základní škole.....	66
5.7.9	Algoritmizace a on-line kurz	67
5.8	POROVNÁNÍ VSTUPNÍCH A VÝSTUPNÍCH POSTOJŮ STUDENTŮ.....	67
5.9	VSTUPNÍ ZNALOSTI A DOVEDNOSTI.....	69
5.9.1	Celkové vyhodnocení testu	69
5.9.2	Otázka 1 (upravena z ibobr.cz)	69
5.9.3	Otázka 2 (čerpaná z ibobr.cz)	71
5.9.4	Otázka 3 (čerpaná z ibobr.cz)	72
5.9.5	Otázka 4 (čerpaná z ibobr.cz)	74
5.9.6	Otázka 5 (čerpaná z ibobr.cz)	75

5.9.7	Otázka 6 (zdroj vlastní)	77
5.9.8	Otázka 7 (zdroj vlastní)	78
5.9.9	Otázka 8 (zdroj vlastní)	79
5.9.10	Otázka 9 (zdroj vlastní)	80
5.9.11	Otázka 10 (zdroj vlastní)	82
5.9.12	Otázka 11 (zdroj vlastní)	84
5.9.13	Otázka 12 (zdroj vlastní)	86
5.10	VÝSTUPNÍ ZNALOSTI A DOVEDNOSTI	86
5.10.1	Celkové vyhodnocení testu	87
5.10.2	Otázka 1 (upraveno z ibobr.cz)	88
5.10.3	Otázka 2 (upraveno z ibobr.cz)	89
5.10.4	Otázka 3 (čerpaná z ibobr.cz)	90
5.10.5	Otázka 4 (upravená z ibobr.cz)	91
5.10.6	Otázka 5 (čerpáno z ibobr.cz)	93
5.10.7	Otázka 6 (vlastní zadání).....	94
5.10.8	Otázka 7 (zdroj vlastní)	95
5.10.9	Otázka 8 (zdroj vlastní)	96
5.10.10	Otázka 9 (zdroj vlastní)	97
5.10.11	Otázka 10 (zdroj vlastní)	98
5.10.12	Otázka 11 (zdroj vlastní)	99
5.10.13	Otázka 12 (zdroj vlastní)	101
5.11	POROVNÁNÍ VSTUPNÍCH A VÝSTUPNÍCH DOVEDNOSTÍ STUDENTŮ	102
ZÁVĚR.....		105
RESUMÉ		106
SEZNAM LITERATURY		107
SEZNAM OBRÁZKŮ, TABULEK, GRAFŮ A DIAGRAMŮ		110
PŘÍLOHY		I

SEZNAM ZKRATEK

LMS – Learning Management System – systém pro řízení výuky

pdf – Portable Document Format – přenosný formát dokumentů

doc – přípona souboru pro textové procesory

docx – přípona souboru pro textové procesory

LED - Light-Emitting Diode – svítivá dioda

KVD – Katedra výpočetní a didaktické techniky

ALGV – předmět Algoritmizace ve vzdělávání

URL – Uniform Resource Locator – jednotná adresa zdroje

ZŠ – Základní škola

SŠ – Střední škola

ZČU – Západočeská univerzita v Plzni

GNU - General Public Licence

IT – informační technologie

ÚVOD

*„Everybody in this country should learn how to program a computer...
because it teaches you how to think.“ (Steve Jobs)*

Algoritmizace je v dnešní době velice často skloňovaným pojmem, převážně na základních a středních školách. Velice efektními nástroji pro rozšiřování algoritmického myšlení mohou být dětské blokové programovací jazyky, které primárně neslouží k tomu, aby se žáci, nebo studenti stali experty v komerční sféře, ale právě k lepšímu analyzování problému, využití dostupných prostředků a vymýšlení neefektivnějších řešení.

Hlavním cílem této práce je vytvořit pro studenty bakalářského studia ucelený on-line kurz, ve kterém si udělají přesnější představu o tom, jak rozvíjet algoritmické myšlení u sebe, ale i u žáků a studentů.

Pro potřeby práce je důležité nejdříve vymezit několik základních pojmů, mezi které zařazují algoritmus, algoritmické myšlení a algoritmizaci. Jedním z dalších cílů práce je popsat metody a nástroje, kterými je možné algoritmické myšlení rozšiřovat (zařazeny jsou i aplikace, dětské robotické hračky a stavebnice).

Cílem praktické části je vyhodnocení vstupních znalostí a dovedností studentů v oblasti algoritmizace a také jejich postojů, a na těchto základech vybudovat celý elektronický kurz s přihlédnutím na rozdíly prezenční a kombinované formy studia. V textové části budou také uvedeny stručné informace o systému pro on-line vzdělávání a o tvorbě elektronického kurzu.

Závěrečným cílem práce je otestovat přínos kurzu. Přínos kurzu je testován pomocí porovnání vstupních a výstupních znalostí a dovedností. Následně bylo provedeno šetření zaměřené na evaluaci kurzu. Z těchto poznatků pak vyplývají možnosti, jak kurz v budoucnu vylepšit.

Mezi nejdůležitější zdroje, které byly použity pro účely této práce, zařazují Moodle na Západočeské univerzitě v Plzni, oficiální webové stránky analyzovaných nástrojů a oficiální webové stránky státního pedagogického ústavu na Slovensku.

Tato práce je určena především učitelům a studentům pedagogických fakult, kteří zvažují zapojení algoritmizace nebo on-line vzdělávání do svých předmětů.

1 VYMEZENÍ ZÁKLADNÍCH POJMŮ

1.1 ALGORITMUS

Algoritmus je zjednodušeně řečeno metoda řešení problému krok za krokem. Pšenčíková (2009) definuje algoritmus jako přesný postup, který popisuje řešení daného problému v konečném počtu kroků.

Virus (1995) se na problematiku definování algoritmu dívá z jiného úhlu a argumentuje, že algoritmus je základní matematický pojem, který není možné definovat, a tudíž je třeba se uchýlit k opisu. Algoritmus pak popisuje jako návod, jak provést určitou činnost. V případě programování půjde z pravidla o transformaci množiny vstupních dat na množinu výstupní.

Server techopedia.com provozovaný společností Janalta Interactive Inc. uvádí následující definici a vysvětlení.

Definice:

“An algorithm is a step by step method of solving a problem. It is commonly used for data processing, calculation and other related computer and mathematical operations. An algorithm is also used to manipulate data in various ways, such as inserting a new data item, searching for a particular item or sorting an item.”

Volný překlad: Algoritmus může být chápán, jako metoda postupného řešení problému (krok za krokem), která se obvykle používá pro zpracování dat, výpočty a pro další výpočetní a matematické operace. Algoritmy se také často používají k manipulaci s daty a to různými způsoby, například vkládání nové datové položky do seznamu, vyhledávání položky v seznamu, nebo k řazení položek v seznamu.

Vysvětlení:

„An algorithm is a detailed series of instructions for carrying out an operation or solving a problem. In a non-technical approach, we use algorithms in everyday tasks, such as a recipe to bake a cake or a do-it-yourself handbook. Technically, computers use algorithms to list the detailed instructions for carrying out an operation. For example, to compute an employee’s paycheck, the computer uses an algorithm. To accomplish this task, appropriate data must be entered into the system. In terms of efficiency, various algorithms are able to accomplish operations or problem solving easily and quickly.“

Volný překlad: Algoritmus může být také chápán jako podrobná (elementární) série instrukcí pro provádění určité operace, nebo řešení problému. V netechnickém přístupu používáme algoritmy v každodenních úkonech, jako je například recept na pečení koláče, nebo příručka „do-it-yourself“. Z technického pohledu počítače používají algoritmy k sestavení podrobných pokynů pro provádění určité operace. Například pro výpočet výplaty zaměstnance. Aby byl počítač schopný tento algoritmus vykonat, musí být do systému zadány příslušné údaje. Různé algoritmy jsou schopny úlohu zpracovávat různě efektivně.

Melichar (2006) uvádí, že s algoritmy se setkáváme na každém kroku a zmiňuje celou řadu příkladů z běžného života – např. návody na obsluhu domácích spotřebičů včetně jednoduchých instrukcí jako např. „Při odchodu zhasněte světlo“.

Původ slova algoritmus datuje dostupná literatura už k přelomu 7. a 8. století našeho letopočtu, kdy arabský matematik Abū ‘Abd Allāh Muhammad ibn Mūsā al-Chwārizmī vydal dvě knihy s postupy výpočtů lineárních a kvadratických rovnic a poslední část jeho jména al-Chwārizmī byl v latinském překladu změněn na Algorismi (Harper 2016, Mareš, 2006, Zemánek 1981).

Pšenčíková (2007), Krček a Kreml (1993) a další uvádějí, že každý algoritmus musí splňovat určitá pravidla. Konkrétně zmiňují následující základní vlastnosti algoritmů:

- determinovanost (jednoznačné určení každého kroku, vždy je jasné, co má následovat),
- obecnost (algoritmus není sestaven pro konkrétní problém, ale pro všechny problémy založené na stejném principu),
- rezultativnost (po provedení určitého počtu kroků algoritmus dosáhne správného výsledku),
- konečnost (algoritmus po svém provedení skončí).

Motyčka (1999) zdůrazňuje jako jednu z nutných vlastností algoritmu i jeho srozumitelnost nebo přehlednost, která je předpokladem modifikace a rozšiřování či vylepšování stávajícího algoritmu.

Jiným způsobem se na vlastnosti algoritmů dívá Donald Knuth ve svém díle *The Art of Computer Programming*, který zvažuje:

- konečnost (finitness),
- jednoznačnost (definitness),
- vstup (input),
- výstup (output),
- efektivnost (effectiveness).

Algoritmus zapsaný v některém programovacím jazyce označujeme jako program (Ďuráková, Dvorský, Ochodková, 2002). Podle Hornika (2016) je tedy programování přepis daného algoritmu do daného programovacího jazyka.

1.2 ALGORITMICKÉ MYŠLENÍ

Futschek (2006) algoritmické myšlení chápe jako termín, který se velmi často používá jako jedna z nejdůležitějších kompetencí, kterou lze dosáhnout ve vzdělání v informatice. Algoritmické myšlení můžeme chápat jako soubor schopností, které jsou spojeny s konstrukcí a porozuměním algoritmů:

- schopnost analyzovat problémy,
- schopnost přesně určit o jaký problém se jedná,
- schopnost nalézt elementární akce odpovídající danému problému,
- schopnost vytvořit správný algoritmus k danému problému pomocí těchto elementárních akcí,
- schopnost přemýšlet o všech možných případech, které mohou při vykonávání algoritmu nastat,
- schopnost zvýšit efektivitu algoritmu.

Podobně chápou algoritmus autoři publikace „Examples of Algorithmic Thinking in Programming Education“. Podle nich algoritmické myšlení představuje jeden z hlavních pojmů informatiky. Algoritmické myšlení se ukázalo být všestranným a nepostradatelným nástrojem pro řešení problémů a našlo své uplatnění ve vědě. Autoři také uvádí, že

udržitelné vzdělávání v oblasti informačních technologií má být založeno právě na rozvoji algoritmického myšlení, jakožto na hlavním cíli tohoto vzdělávání.

Algoritmické myšlení je zmiňováno ve vztahu k informační výchově – např. Černochová (2005) uvádí, že obsahem informační výchovy by měly být jednak znalosti a dovednosti z technologií a informatiky, jednak logické, strategické, analyticko-syntetické, virtuální, algoritmické a strukturální myšlení.

1.3 ALGORITMIZACE

Hornik (2016) uvádí, že výuka programování a algoritmizace by (minimálně na základní škole) měla mít za účel objasnění principů a postupů právě algoritmizace, přičemž programování je jen pomůcka k dosažení tohoto cíle. Nejde tedy o schopnost programovat, ve smyslu psát kód, ale o schopnost algoritmizovat, neboli vymýšlet řešení a postupy řešení daného problému. Žáci by měli tedy být především vedeni k tomu, aby dokázali přesně definovat problém, analyzovat ho a vysvětlit ho dalším lidem.

Landa (1974) chápe algoritmizaci jako učení studentů „jak“, namísto „co“. Hlavním přínosem tohoto přístupu je pro něj učit nezávisle přemýšlet. Pohled na nové perspektivy výuky zaměřené na získání nezávislých a obecných schopností pro řešení problémů je nyní velice aktuální. Vědecké disciplíny začínají čerpat z koncepcí kybernetiky, logiky, teorie informací, operačního výzkumu a dalších. Algoritmizace se liší od programování, ale je s programováním slučitelná (můžeme ji označit jako předpoklad). Dále Landa porovnává principy programové výuky (cílem je posílit učení pomocí okamžité zpětné vazby) s výukou algoritmizace, jejímž cílem je organizace myšlení při řešení problémů tak, aby se dosáhlo vysoké spolehlivosti.

Pšenčíková (2009) chápe algoritmizaci jako jednu z nejdůležitějších činností při vytváření software. Uvádí, že je to část dovedností, která nepodléhá času, momentální módě, ani firemním zájmům. Programovací jazyky se mění, jediné co zůstává stejné, je právě algoritmizace. Pokud si uživatelé osvojí algoritmický způsob myšlení, pak se stanou nepostradatelnými odborníky, a to i v případě, že se nebudou zabývat algoritmizací.

2 METODY URČENÉ PRO ROZVOJ ALGORITMICKÉHO MYŠLENÍ

Aktivitami rozvíjející algoritmické myšlení mohou být také činnosti respektující posloupnost kroků, jako je postup dle návodu či řazení vět či objektů (Berki 2016).

Každý učitel a praktický odborník v oblasti informatiky musí do své přípravy zahrnout základy teoretických disciplín. V dnešní době je žákům a studentům důležité dát nejen statické znalosti (definice, věty, důkazy) bez dalších propojení, ale také dovednosti, návyky a postoje.

Habiballa (2004) ve svém příspěvku „DIDACTIC EXPERIMENT IN EDUCATION OF THEORETICAL COMPUTER SCIENCE“ uvádí: *Z praxe učitelů (i studentů) těchto předmětů lze často slyšet, že klasická výuka založená na učení znalostí při použití organizační formy hromadného vyučování vede k memorování bez pochopení principů, demotivaci pro další studium, a tím i formování nesprávných návyků a postojů v informatice u budoucích absolventů.*

Každý žák a student by si měl osvojit kompetenci ke zkoumání řešitelnosti určitých problémů a její efektivnosti.

Vzhledem k tomu, že v českém rámcovém vzdělávacím programu, není zastoupeno téma algoritmicizace, k účelům této práce byl použit slovenský Štátný vzdelávací program, ve kterém je podrobně popsán celek nazvaný algoritmické řešení problémů včetně výkonových a obsahových standardů.

V následujících tabulkách je popsán rozvoj algoritmického myšlení v 6. a 8. třídě nižšího středního vzdělání podle výše zmiňovaného Štátného vzdelávacího programu.

Výkonové standardy jsou vždy popsány: Žák na konci x. ročníku základní školy ví/dokáže:

Tabulka 1: Štátny vzdelávací program - Analýza problému

	Výkonový standard	Obsahový standard
Analýza problému	<ul style="list-style-type: none"> • uvažovat o omezeních, které souvisí s řešením úlohy, • identifikuje problémy, • uvažovat o hraničních případech (na úrovni cyklů), • rozhodnout o pravdivosti nebo nepravdivosti tvrzení (výroku), • vybrat prvky nebo možnosti podle pravdivosti tvrzení, • popsat vztahy mezi informacemi vlastními slovy, • uvést protipříklad, v kterém něco neplatí, nefunguje, • uvažovat o různých řešeních. 	Vlastnosti a vztahy: platí - neplatí, a, nebo, ne (neformálně) Procesy: krokování sekvence a opakování, rozhodnutí o pravdivosti tvrzení

Tabulka 2: Štátny vzdelávací program - Jazyk na zápis řešení

	Výkonový standard	Obsahový standard
Jazyk na zápis řešení	<ul style="list-style-type: none"> • použít jazyk na popis řešení problému - aplikovat pravidla, konstrukce jazyka, • použít matematické výrazy v jazyku pro zápis algoritmů. 	Vlastnosti a vztahy: algoritmus - programovací jazyk, vstup - algoritmus - výsledek, chybný zápis, konstrukce jazyka jako: postupnost příkazů, cyklus s pevným počtem opakování, podmíněný příkaz, pojmenovává postupnost příkazů Procesy: sestavení programu v jazyku na zápis algoritmů, spuštění programu

Tabulka 3: Štátny vzdelávací program: Pomocí posloupnosti příkazů

	Výkonový standard	Obsahový standard
Pomocí posloupnosti příkazů	<ul style="list-style-type: none"> • řešit problém skládáním příkazů do posloupnosti, • aplikovat pravidla konstrukce jazyka pro sestavení posloupnosti příkazů, • interpretovat postupnost příkazů, • hledat chybu v posloupnosti. 	Pojmy: příkaz, parametr příkazu, posloupnost příkazů Vlastnosti a vztahy: jak souvisí příkazy, pořadí příkazů a výsledek, pravidla jazyka pro sestavení sekvence příkazů Procesy: sestavení a upravení příkazu/příkazů, vyhodnocení postupnosti příkazů, úprava sekvence příkazů (přidání, odstranění příkazu, změna pořadí příkazů)

Tabulka 4: Štátny vzdelávací program - Pomocí cyklů

	Výkonový standard	Obsahový standard
Pomocí cyklů	<ul style="list-style-type: none"> rozpoznat opakující se vzory při řešení zadaného problému, rozpoznat, jaká část algoritmu se má vykonat před, při a po skončení cyklu, stanovit počet opakování pomocí hodnoty, řešit problémy, které vyžadují známý počet opakování, zapsat řešení problému s cyklem pomocí jazyka, interpretovat algoritmy s cykly. 	Pojmy: opakování, počet opakování, tělo cyklu Vlastnosti a vztahy: jak souvisí počet opakování s výsledkem Procesy: sestavení, upravení těla cyklu, nastavení počtu opakování

Tabulka 5: Štátny vzdelávací program - Pomocí větvení

	Výkonový standard	Obsahový standard
Pomocí větvení	<ul style="list-style-type: none"> rozpoznat situace a podmínky, kdy je potřeba použít větvení, rozpoznat, jaká část algoritmu se má vykonat před, v rámci a po skončení větvení, sestavit a zapsat podmínku, vyřešit problémy, které vyžadují větvení s jednoduchou podmínkou (bez logických spojek), zapsat řešení problému s větvením pomocí jazyka, interpretovat algoritmy s větvením. 	Pojmy: větvení, podmínka Vlastnosti a vztahy: konstrukce větvení s jednoduchou podmínkou, pravda, nepravda - splněná a nesplněná podmínka Procesy: sestavení, upravení větvení, vytvoření podmínky, vyhodnocení podmínky

Tabulka 6: Štátny vzdelávací program - Pomocí proměnných

	Výkonový standard	Obsahový standard
Pomocí proměnných	<ul style="list-style-type: none"> identifikovat údaje ze zadání úlohy, které musí být zapamatovatelné, resp. se mění, a vyžadují použití proměnných, aplikovat pravidla, konstrukce jazyka pro nastavení a použití proměnné, vyřešit problémy, ve kterých je potřeba zapamatovat a často použít zapamatované hodnoty, zobecnit řešení tak, aby fungovalo nejen s konstantami, interpretovat algoritmy s výrazy a proměnnými. 	Pojmy: proměnná, název proměnné, hodnota proměnné, operace (+, -, *, /) Vlastnosti a vztahy: pravidla jazyka pro použití proměnné, jméno proměnné - hodnota proměnné. Procesy: nastavení hodnoty proměnné, zjištění hodnoty, změna hodnoty, vyhodnocení výrazu s proměnnými, čísla a operace

Tabulka 7: Štátny vzdelávací program - Pomocí nástroje na interakci

	Výkonový standard	Obsahový standard
Pomocí nástroje na interakci	<ul style="list-style-type: none"> rozpoznat situace, kdy je očekávaný vstup, zapsat algoritmus, který reaguje na vstup, interpretovat zapsané řešení, vytvořit hypotézu, jak neznámý algoritmus zpracuje zadaný vstup. 	Vlastnosti a vztahy: prostředky jazyka pro: získání vstupu, zpracování vstupu a zobrazení na výstupu Procesy: čekání na neznámý vstup - vykonání akce - výstup, následný efekt

Tabulka 8: Štátny vzdelávací program - Interpretace zápisu řešení

	Výkonový standard	Obsahový standard
Interpretace zápisu řešení	<ul style="list-style-type: none"> realizovat návod, postup, algoritmus řešení úlohy - interpretovat ho, krokovat řešení, simulovat činnost vykonavatele programu, vyjádřit princip fungování návodu - objevit a popsat vlastními slovy princip fungování jednoduchého algoritmu, najít vztah mezi vstupem algoritmu a výsledkem. 	Vlastnosti a vztahy: jazyk - vykonávání programu Procesy: Krokování, co se děje v počítači v případě chyby v programu

Tabulka 9: Štátny vzdelávací program - Hledání, opravování chyb

	Výkonový standard	Obsahový standard
Hledání, opravování chyb	<ul style="list-style-type: none"> rozpoznat, že program nepracuje správně, hledat chybu ve vlastním nesprávně pracujícím programu a opravit ji, zjistit, pro jaké vstupy, v kterých případech, situacích program nepracuje správně, interpretovat návod, ve kterém je chyba, diskutovat a argumentovat o správnosti řešení (svého i cizího), rozlišit chybu při realizaci od chyby v zápise, diskutovat o různých postupech a výstupech řešení (porovnat řešení konkrétního problému od různých žáků z hlediska délky výsledku, trvání, velikosti kódu/zápisu), doplnit, dokončit, modifikovat rozpracované řešení, navrhnout vylepšení řešení. 	Vlastnosti a vztahy: chyba v postupnosti příkazů (špatný příkaz, chybějící příkaz, vyměněný příkaz nebo příkaz navíc), chyba v algoritmu s cyklem, s větvením a s proměnnými, chyba při realizaci (logická chyba), chyba v zápise (syntaktická chyba) řešení, které lépe splní stanovené kritérium v zadaném problému Procesy: hledání chyby

Z následujících tabulek, je patrné, že na základních školách na Slovensku má algoritmizace v celku informatika velké zastoupení. Důsledkem toho by měli všichni absolventi základních škol na Slovensku mít mnohem lepší úroveň algoritmického myšlení, než v České Republice.

3 NÁSTROJE PRO ROZVOJ ALGORITMICKÉHO MYŠLENÍ

Nástrojů pro rozvoj algoritmického myšlení je celá řada a každým dnem přibývají nové, protože algoritmizace získává stále větší pozornost učitelů a vzdělávacích institucí u nás i v zahraničí. Pro potřeby kurzu, který byl vytvořen v rámci diplomové práce, po pečlivém studiu dostupných zdrojů včetně odkazů pedagogických komunit (např. rvp.cz, GEG ČR, iSen atd.), mnoha diskusích s učiteli z praxe (v rámci konference Učitel In i v rámci pedagogické praxe) i vlastním testování ve výuce na základní škole byly vybrány příklady různých typů nástrojů. Prostor v práci byl věnován jednak pro aplikace rozvíjející algoritmické myšlení (Scratch, Kodu, Easy Logo, Code.org, Code Combat) a v neposlední řadě také pro roboty a robotické stavebnice (Bee-bot, Dash a Dot, Ozobot, Lego Mindstorms). Tyto nástroje jsou dále podrobněji představeny v následujících podkapitolách a jsou také součástí aktivit v kurzu, který byl vytvořen pro předmět Algoritmizace ve vzdělávání (KVD/ALGV), v rámci diplomové práce.

3.1 APLIKACE PRO ROZVOJ ALGORITMICKÉHO MYŠLENÍ

V této kapitole budou představeny aplikace, které je možné využít pro rozvoj algoritmického myšlení.

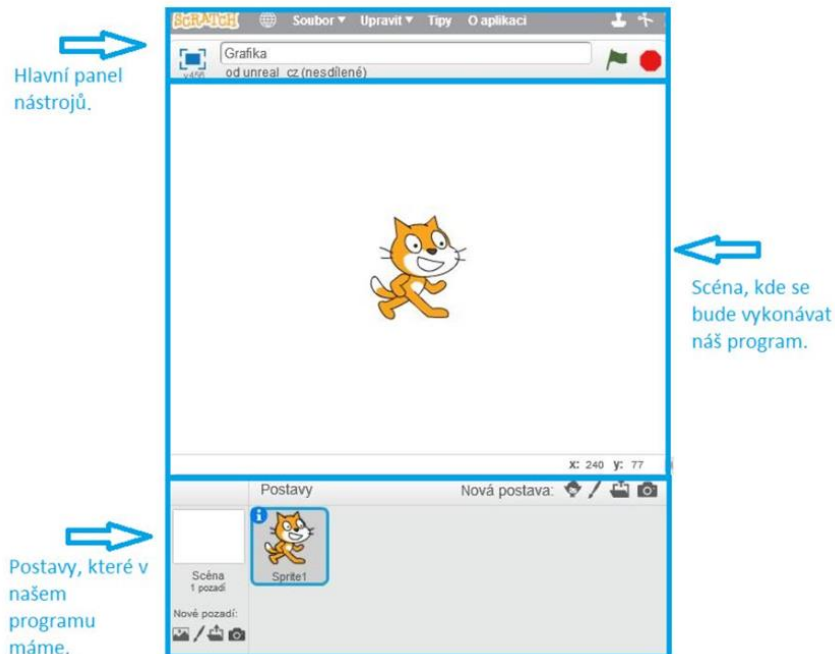
3.1.1 SCRATCH

Programovací jazyk Scratch byl vyvinut v Mediální laboratoři Massachusetského Institutu technologií v roce 2007 firmou Lifelong Kindergarden. Volný překlad názvu firmy znamená „celý život ve školce“.

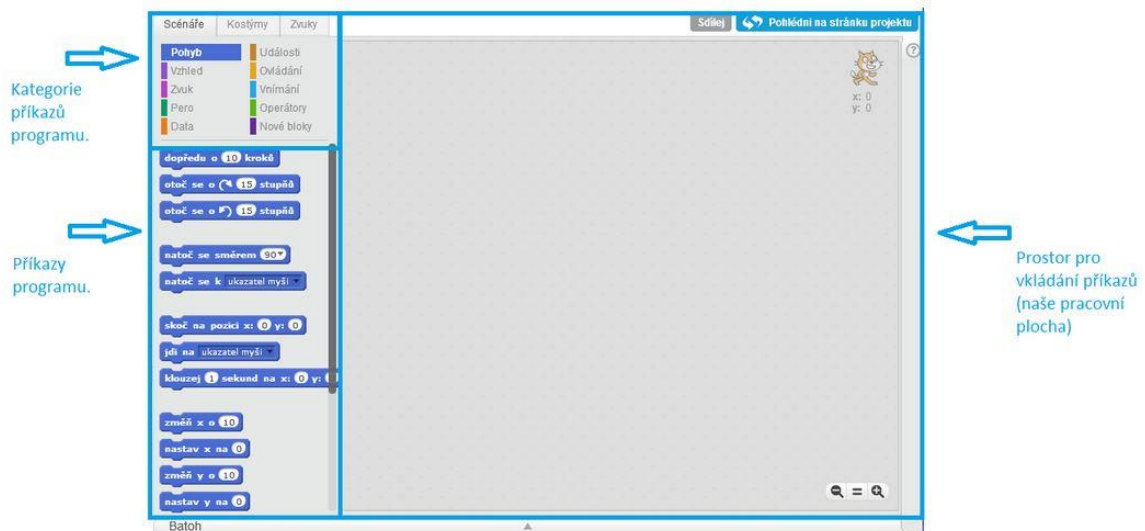
Prostředí programovacího jazyka Scratch je ideální pro studenty, kteří nemají s programováním žádné zkušenosti. Vývojové prostředí je intuitivní s přehledným uživatelským rozhraním, které studenty a žáky k programování láká. V aplikaci je možné programovat úplně od začátku a rozhýbat tak svou postavu (sprite), stejně jako vytvářet zajímavé programy a hry. Vše je možné díky on-line účtu ukládat do cloudu. Také je možné v projektu pokračovat z jakéhokoliv počítače připojeného k internetu.

V levé části je panel s nástroji, který je rozdělený do osmi skupin. Uprostřed je scéna s panelem skriptů, kde dochází ke skládání jednotlivých příkazů. Příkazy se vážou na řízený objekt, zvaný sprite – skřítek. Začátečníci tak snadno poznají příkazy pro pohyb skřítky, pro změnu jeho vzhledu, přehrávání zvuků, zanechání stop, nebo tvorbu vlastních proměnných.

Další velkou skupinou jsou bloky s podmínkami (if, else) a bloky cyklů. Neméně důležité jsou sekce s matematickými funkcemi (sčítání, odečítání, porovnání), generování náhodných čísel, práce s textovými řetězci a vytváření seznamů. Obrázky 1 a 2 znázorňují rozložení programovacího prostředí.



Obrázek 1: Prostředí Scratch



Obrázek 2: Prostředí Scratch

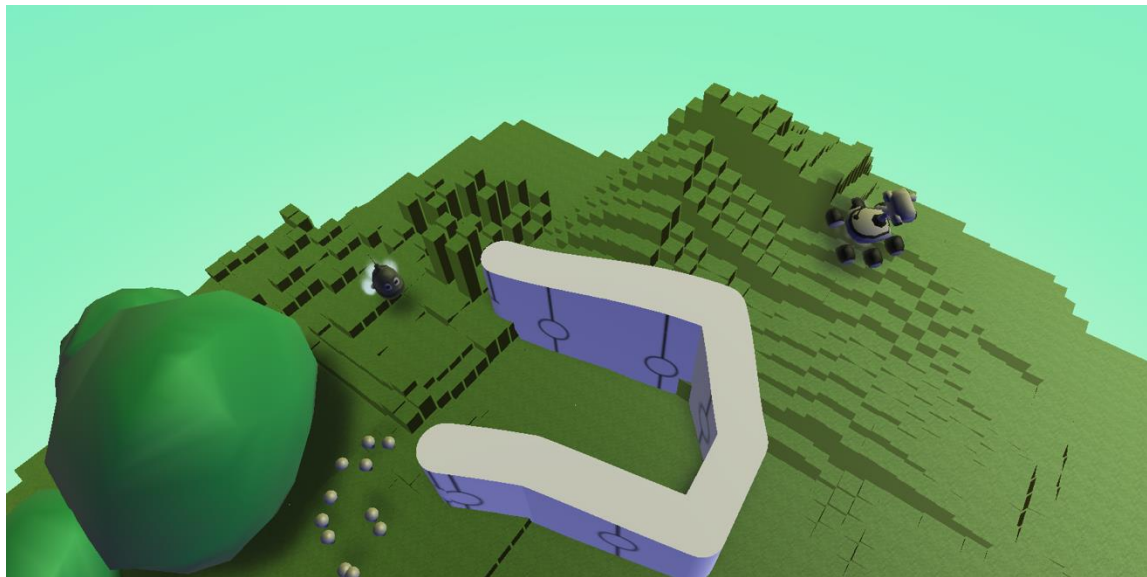
3.1.2 KODU

Kodu je vizuální programovací jazyk od společnosti Microsoft. Je určený pro všechny moderní systémy od společnosti Microsoft včetně herní konzole Xbox 360. Kodu je

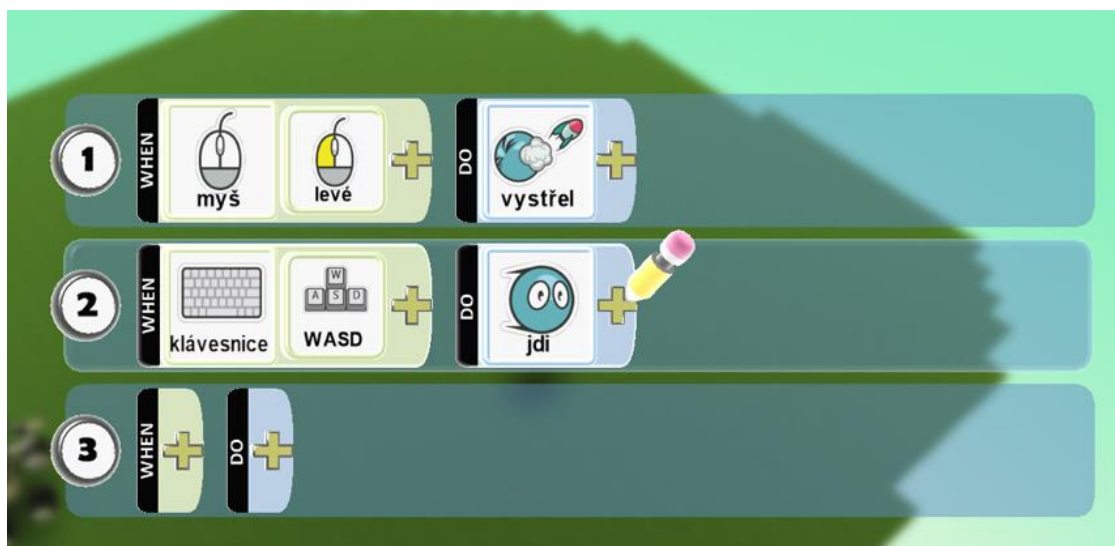
navržený tak, aby jeho ovládání zvládly i menší děti a mohly tak začít vytvářet hry bez předchozích znalostí programování.

Celý virtuální svět v Kodu se odehrává na vzdálené planetě. Pomocí různých nástrojů si žáci a studenti mohou formovat terén. Možností je téměř neomezené množství. Dále je možné do světa přidat různé předměty (sbírání bodů, energie...).

Pokud žáci či studenti dokončí přípravu herního světa, mají za úkol vložit do hry robota pojmenovaného Kodu a další herní charaktery, u kterých je nutné naprogramovat události. Veškerá programovací logika je velice jednoduchá, je založena převážně na událostech WHEN – DO (viz obrázek 4).



Obrázek 3: Vygenerovaný svět v Kodu



Obrázek 4: Programování v Kodu

3.1.3 EASYLOGO

Aplikace EasyLogo byla vytvořena pro uživatele, kteří nemají velké počítačové dovednosti, za účelem zpřístupnit jim programování a zlepšit jejich schopnost řešení problémů. Aplikace je dostupná pro nekomerční použití zdarma i v českém jazyce.

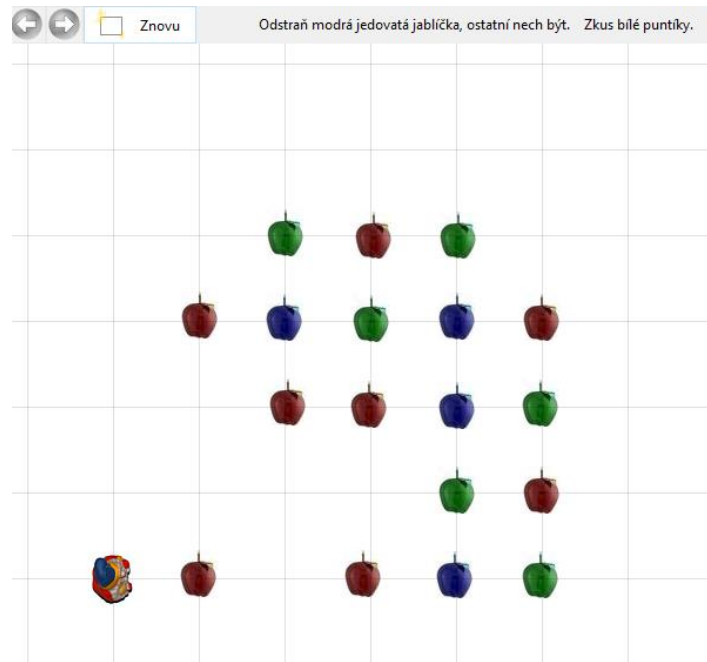
EasyLogo v základní verzi obsahuje 40 úloh pro rozvoj algoritmického myšlení. Učitelé nebo i žáci si mohou vymyslet aktivity vlastní pomocí přiloženého návodu.

Klíčové vlastnosti aplikace EasyLogo:

- EasyLogo pracuje s mřížkou (viz obrázek 5),
- program je zkonstruován z jednotlivých bloků, které uživatel skládá,
- program se provádí automaticky, vždy po přidání bloku.

Vzdělávací cíle různých aktivit:

- ovládat pohyb postavy přímo – kliknutím na tlačítka,
- vytvořit sekvenci příkazů (program) pro ovládání postavy,
- vytvořit sekvenci příkazů, řešící určitý problém,
- rozpoznat opakující se vzory a navrhnout cyklus,
- navrhnout postup k vytvoření obrazců,
- opravit nebo zefektivnit program.



Obrázek 5: Úloha v aplikaci EasyLogo

3.1.4 HOUR OF CODE

Hour of Code, neboli Hodina kódu je projekt, jehož cílem je v jedné hodině představit hravou formou žákům a studentům, že programování nemusí být vždy komplikované. Pokud se žákům Hodina kódu zalíbí, jsou pro ně připraveny další kurzy, ve kterých své algoritmické myšlení mohou zdokonalovat.

V tomto projektu se objevují známé tváře, jako je například Bill Gates (zakladatel firmy Microsoft), Mark Zuckerberg (zakladatel Facebooku) a další, kteří k žákům a studentům skrz videa promlouvají.

Jednotlivé Hodiny kódu jsou vždy tematicky naladěny na populární téma u dětí (Minecraft, Frozen, Angry Birds, Star Wars...). Celé programování pak probíhá pomocí bloků.

Učitelé mají možnost zaregistrovat celou třídu a získat tak přehled o učení žáků a studentů.



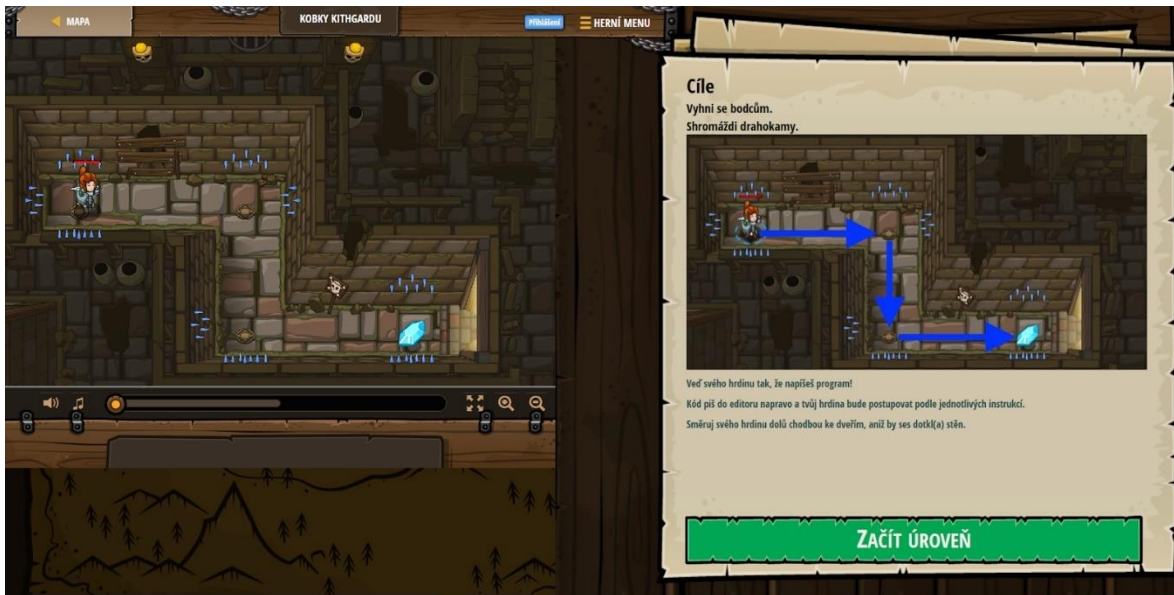
Obrázek 6: Ukázka z Hour of Code

3.1.5 CODE COMBAT

Tato aplikace se od předchozích liší, v tom, že jako jediná z výběru se neprogramuje pomocí bloků, ale pomocí textových příkazů. Tato aplikace si klade za cíl vyučovat programovací jazyk Python, Javascript, Lua a CoffeeScript a to v původní podobě, včetně všech formálních pravidel syntaxe (viz obrázek 8).

Z obrázku je také patrné, že prostředí této aplikace je obdobné jako hry ve stylu RPG (role-playing game). Uživatel ovládá pomocí příkazů svého hrdinu, který během své mise získává zkušenosti.

Projekt Code Combat propaguje systémy pro výuku založené na hře, protože jsou podle výzkumů pro žáky a studenty lepší v porovnání s klasickými metodami. Pomáhají jim udržet si znalosti a lépe se soustředit. Zpětná vazba je studujícím předávána v reálném čase, studenti tak mohou okamžitě přizpůsobovat svá řešení.



Obrázek 7: Prostředí hry Code Combat

```

1 # Přejdi k drahokamu.
2 # Nedotýkej se zdí!
3 # Piš svůj kód níže.
4
5 ✓ hero.moveRight()
6 ✓ hero.moveDown()
7 ▶ hero.moveRight()
8
9

```

Obrázek 8: Kód ve hře Code Combat

3.2 ROBOTI A ROBOTICKÉ STAVEBNICE PRO ROZVOJ ALGORITMICKÉHO MYŠLENÍ

V této kapitole budou představeny robotické hračky a stavebnice, které jsou určeny k rozvoji algoritmického myšlení.

3.2.1 BEE-BOT

Na rozvíjení algoritmického myšlení v nižších ročnících základní školy je Bee-bot ideální volbou. Ovládání Bee-bota (robotické včelky) je velice jednoduché. Na hřbetu Bee-bota jsou směrové šipky a tlačítko pro spuštění programu. Stlačením tlačítka vždy Bee-bot vykoná jeden úkon. Šipkou dopředu popojede o jeden díl, šipkou doprava se natočí o 90 stupňů doprava, šipkou dozadu popojede zpět o jeden díl, šipkou doleva se natočí o 90 stupňů doleva. Programování probíhá vždy v sekvenci stisknutí šipek dle zadání úlohy. Po dokončení zadávání se stiskne tlačítko pro spuštění. Poté se spustí naprogramovaná

sekvence. Nevýhodou je, že při opravení chyby žáci musí bee-bota znovu naprogramovat. Tuto nevýhodu řeší tabletová aplikace, která ji odstraňuje.

Bee-bot svůj program nejčastěji vykonává na rovné podložce. Podložky mohou být v různých provedeních. Mohou být již připravené (koupené z oficiálního e-shopu), nebo je možné si ve třídě vytvořit vlastní podložku z obrázků, k probíranému tématu. Tuto podložku mohou žákům připravit učitelé, nebo si ji žáci mohou připravit sami.



Obrázek 9: Robotická hračka Bee-Bot (zdroj: volné dílo)

3.2.2 DASH A DOT

Dash a Dot jsou roboti od firmy Wonder Workshop. Tito roboti jsou určeni pro děti od 5 let a výše. Dash je větší robot, který má motory pro pohyb a také motor pro natáčení hlavy na rozdíl od Dots, který se nepohybuje. Oba roboti mohou přehrávat zvuky, které si může uživatel vybrat z připravené databáze, nebo si je sám nahrát. Oba roboti také obsahují celou řadu programovatelných LED. Mezi další akce těchto robotů patří: reakce na stisknutí tlačítka, zvukové události (například při zatleskání), události založené na vzdálenosti od určitého objektu a události založené na akcelerometru (při kolizi robota, nebo při zatřesení).

Tito roboti mohou být ovládáni a programováni skrze několik aplikací:

- Aplikace Go: umožňuje přímé ovládání robotů (robot se pak chová jako auto na ovládání),
- Aplikace Xylo: umožňuje hrát na přídavném xylofonu,

- Aplikace Path: ovládání robota Dash pomocí kreslení na displej,
- Aplikace Blockly: umožňuje programovat roboty za pomoci blokového programování (pomocí různých miniher nebo volného programování).

Roboti komunikují s tabletem nebo smartphonem pomocí rozhraní Bluetooth.

3.2.3 OZOBOT

Ozobot je univerzální programovatelná robotická hračka.

Funguje na základě čtení barevných kódů (zelená, červená, modrá a černá) pomocí barevného senzoru. Další jeho vlastností je sledování čáry (šířka čáry musí být alespoň 5 mm). Po správném přečtení a vyhodnocení kódu Ozobot vykoná požadovanou akci (obrázek 10).



Obrázek 10: Ukázka kódu - Ozobot

Aktuálně je možné najít dva typy Ozobotů BIT a EVO.

Ozobot BIT - jednodušší provedení. Zvládá samovolný pohyb po předem připravené cestě, má primitivní typ rozhodování (optický a barevný senzor). Zvládá příkazy podle nakreslených sekvencí na papíře, umí pracovat s cykly, podmínkami, proměnnými, funkcemi a navigací po čáře.

Vylepšená verze EVO má navíc proximální senzor, umí pracovat s intervaly, umí přerušit program a má integrovaný reproduktor. Velkou výhodou je také možnost blokový program nahrát přímo z aplikace (není tedy nutné používat barevný senzor).

Pro Ozobota je možné nakreslit cestu na papír, stejně tak je možné vytvořit cestu v aplikaci na tabletu. Zároveň, pomocí aplikace Ozoblockly se dá vytvořit i vlastní program robota. Aplikace opět využívá blokové programování, to je dokonce uzpůsobeno věku žáka. Po vytvoření se program jednoduše nahraje přečtením barevného kódu, nebo pomocí bezdrátových technologií (pouze u Ozobota EVO). Aplikace funguje na zařízeních s operačním systémem iOS a Android.



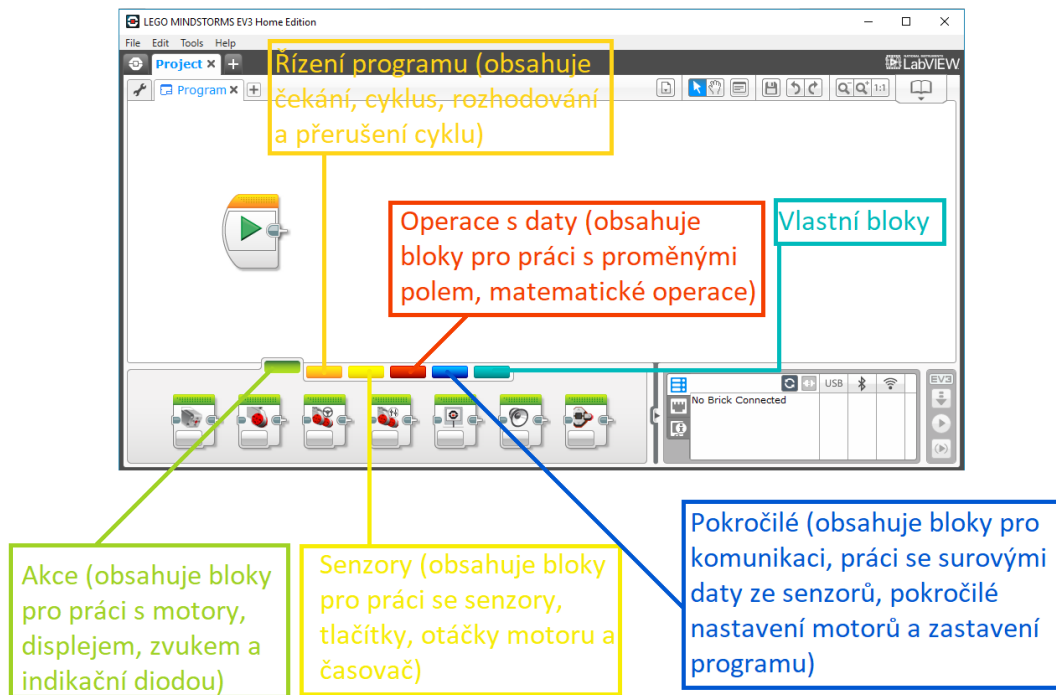
Obrázek 11: Ozobot (zdroj: volné dílo)

3.2.4 LEGO MINDSTORMS

Další zajímavou možností, jak u žáků rozvíjet algoritmické myšlení, jsou robotické stavebnice. Významným zástupcem této kategorie je stavebnice Lego Mindstorms.

Stavebnice je vytvořena tak, aby si mohli studenti svého robota nejdříve sestavit (podle vlastní kreativity, nebo podle přiložených plánek) a následně si ho naprogramovat pomocí vývojového prostředí EV3.

Programování robota je opět řešeno pomocí bloků. Student nebo žák vybírá jednotlivé bloky ze sekcí a tyto bloky modifikuje dle své potřeby a skládá je za sebe. Nástroj EV3 obsahuje bloky pro řízení programu, operace s daty, akce, senzory, pokročilé akce a uživatel si může vlastní bloky také vytvořit (viz obrázek 12).



Obrázek 12: Přehled programovacích bloků EV3

Firma Lego má pro začátečníky připravené tutoriály, které je seznámí s prostředím a jeho základními funkcemi. Velká výhoda je samozřejmě v tom, že díky programování pomocí připravených bloků, zcela odpadá potíže se syntaxí programovacího jazyka. Studenti se tak mohou soustředit pouze na samotný algoritmus a odpadá tak nutnost řešit jeho syntaktický zápis.

4 E-KURZ PŘEDMĚTU ALGORITMIZACE VE VZDĚLÁVÁNÍ

Tato kapitola je věnována představení on-line kurzu k podpoře předmětu Algoritmizace pro vzdělávání (KVD/ALGV), který byl vytvořen v rámci diplomové práce. Hlavním důvodem pro vznik e-kurzu je studentům prezenční formy, kombinované formy a studentům rozšířeného studia informatiky zpřístupnit studijní opory pro zvýšení úspěšnosti učení a také úspěšnosti plnění předmětu.

4.1 ZÁKLADNÍ POPIS KURZU

Tento elektronický kurz podává studentům vhled do základů algoritmizace, ale také do základů programování v dětských programovacích jazycích. Kurz je psaný s ohledem na cílovou skupinu kurzu (studenti prvního ročníku bakalářského studia), je tedy psán srozumitelně, čtivě a je použita velká paleta multimédií.

Kurz byl tvořen na základě vstupního dotazníku. Většina problémových částí byla v kurzu vysvětlena.

Jak již bylo uvedeno, základní cílová skupina kurzu KVD/ALGV jsou studenti prvního ročníku bakalářského studia, kteří zahajují své studium s různými předchozími znalostmi a dovednostmi v oblasti algoritmizace a programování, které si odnášejí z předchozího studia, nebo z vlastních podnětů (Kapitola 5 průzkum). Kurz tedy může být chápán i jako nástroj pro srovnání těchto znalostí a dovedností, které jsou požadovány v navazujících předmětech (Programování 1 pro vzdělávání KVD/PGM1P, Programování 2 pro vzdělávání KVD/PGM2B).

Kurz je časově rozvržen na standardní dobu studia předmětu KVD/ALGV, to v praxi znamená jeden semestr. Je velice žádoucí jednotlivé kapitoly zveřejňovat ještě před probráním jednotlivých témat při kontaktních setkáních, to studentům umožní individuální přípravu na předmět.

Při vytváření kurzu byl kladen velký důraz na to, aby byl obsah kurzu zobrazitelný také na přenosných zařízeních.

Kurz byl vytvořen s přihlédnutím na rozdíly prezenční a kombinované formy studia. Veškerá probíraná látka byla dostatečně vysvětlena tak, aby nebyla potřeba dalších komentářů

učitele. V kurzu bylo také použito několik nástrojů pro komunikaci vzájemně mezi studenty nebo mezi studenty a tutorem kurzu.

4.2 POUŽITÝ LEARNING MANAGEMENT SYSTEM

Název systému Moodle je akronymem pro modulární objektově orientované dynamické vzdělávací prostředí (Modular Object-Oriented Dynamic Learning Environment). Moodle jako systém je softwarový balík pro podporu prezenční, kombinované a distanční výuky s využitím e-learningových kurzů, které jsou dostupné prostřednictvím webového prohlížeče. Nediskutovatelnou výhodou tohoto systému je, že je poskytován bezplatně jako svobodný a otevřený software pod licencí GNU (General Public Licence). To v praxi znamená, že je software chráněn autorským právem, ale i přesto je možné ho bezplatně instalovat, šířit a dokonce i modifikovat (Z knihy Moodle kompletní průvodce tvorbou a správou elektronických kurzů).

Celá verze on-line kurzu je provozována v univerzitním prostředí Moodle, který je využíván již od roku 2007. Největší výhodou tohoto řešení je propojení s univerzitní aplikací STAG, které nabízí automatické propojení předmětu s on-line kurzem.

Studenti, kteří mají předmět zapsaný, mají přístup do kurzu vytvořen automaticky a přihlašují se do něj pomocí webového rozhraní, které je umístěno na adrese www.moodle.zcu.cz svým univerzitním Orion účtem pomocí nástroje WebAuth (obrázek 13).

Přihlášení

Uživatelé Západočeské univerzity se přihlašují svým **univerzitním Orion účtem** přes **WebAuth**. Klikněte na obrázek přihlašovací stránky WebAuthu:

Orion WebAuth

ZÁPADOČESKÁ
UNIVERZITA
V PLZNI

Orion login:

Heslo:

[Nápověda](#) | [Nechci se přihlásit](#)

Kde to jsem? Kam jsem se to zase dostal?
Webový server, na který se snažíte přihlásit, byl zařazen do domény jednotného přihlášení (single sign-on, SSO), a vyžaduje ověření vaší identity platným uživatelským jménem a heslem. Stránka, na které se právě nacházíte, je vstupním bodem k webovým serverům ZČU zařazeným pod systém jednotného přihlašování. To znamená, že svoji identitu prokážete skrze tuto stránku prvnímu serveru, na který chcete přistupovat, a tím jste automaticky přihlášen(a) i k ostatním serverům v doméně.

Výhody
Větší pohodlí pro uživatele (heslo zadávají jen jednou) a technicky vyšší bezpečnost: mezi prohlížečem a webovým serverem se neposílá heslo, ale jen autentizační token. Platnost tokenu je navíc časově omezena.

Důležité upozornění
Nikdy nezadávejte Orion jméno a heslo do webových formulářů, pokud se nejedná o přihlašovací stránku systému WebAuth (tato stránka) nebo oficiální webový nástroj pro změnu hesla. Obě aplikace jsou provozovány na stroji webkdc.zcu.cz!!!

Obrázek 13: Přihlašování do Moodle pomocí WebAuth

LMS Moodle nabízí velké množství nástrojů pro řízení výuky, množství možností publikování studijních článků, nástroje pro komunikaci, nástroje pro zpětnou vazbu studentům (testy, ankety a úkoly...).

4.3 CÍLE KURZU

Podle oficiálních stránek předmětu na Courseware je cílem předmětu představit možnosti rozvoje algoritmického myšlení pomocí různorodých metod, aktivit a programovacích prostředí. Na oficiální cíl předmětu navazují dílčí cíle pro semináře předmětu. Semináře plynule navazují na přednášky předmětu. Jejich cílem je rozvíjet teoreticky získané znalosti z přednášek o praktické dovednosti. Důraz je kladen na rozvoj algoritmického myšlení studentů jako na předpoklad pro jejich celoživotní rozvoj a základ úspěšného zvládnutí navazujících předmětů v oblasti algoritmizace a programování.

Cíl e-kurzu tedy vychází jak z cíle celého předmětu, tak z dílčího cíle semináře. Z pohledu studenta předmětu KVD/ALGV je potřeba získat zejména:

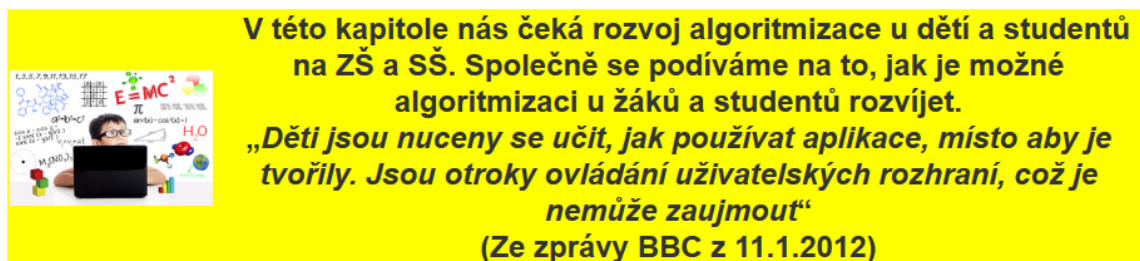
1. obecný základ
 - udělat si přesnější představu o možnostech rozvíjení algoritmického myšlení u sebe, ale i u žáků a studentů,

- seznámit se s hlavními pojmy, principy a zásadami v oblasti algoritmizace,
 - osvojit si základní dovednosti využití ICT při rozvoji algoritmického myšlení;
2. specifické dovednosti
- získat základní dovednosti při práci s aplikacemi určenými pro rozvoj algoritmického myšlení,
 - získat předpoklady pro rozvoj svého algoritmického myšlení,
 - osvojit si dovednosti při práci v LMS Moodle na ZČU v roli studenta.

4.4 OBSAH KURZU A JEHO STRUKTURA

Celý e-kurz je logicky uspořádán do osmi témat (včetně úvodu a závěru). Jednotlivá témata jsou velice výrazně oddělena pomocí podbarvených tabulek s ilustračním obrázkem a motivačním popisem toho, co bude v celém tématu probráno.

Rozvoj algoritmizace u dětí a studentů na ZŠ a SŠ



V této kapitole nás čeká rozvoj algoritmizace u dětí a studentů na ZŠ a SŠ. Společně se podíváme na to, jak je možné algoritmizaci u žáků a studentů rozvíjet.

„Děti jsou nuceny se učit, jak používat aplikace, místo aby je tvořily. Jsou otroky ovládní uživatelských rozhraní, což je nemůže zaujmout“

(Ze zprávy BBC z 11.1.2012)

Obrázek 14: Oddělení jednotlivých témat

4.4.1 POUŽITÉ ČINNOSTI A STUDIJNÍ MATERIÁLY

V této podkapitole je uveden celkový výčet činností a studijních materiálů, které byly při tvorbě kurzu použity:

- **fórum**, tento modul umožňuje studentům asynchronní komunikaci. Ve fóru mohou studenti předmětu KVD/ALGV řešit své otázky a sdílet své náměty a tipy;
- **novinky**, tento modul je zvláštním typem fóra. Do tohoto fóra mohou přispívat pouze tutoři kurzu. Novinky tedy fungují jako nástěnka nebo informační tabule pro studenty. Pokud by chtěl student reagovat na novinku, musí k tomu využít diskusní fórum;

- **URL**, modul URL umožňuje tutorovi kurzu přidávat odkazy na vše, co je volně k dispozici on-line. V kurzu je tento modul používán pro zajímavé materiály na internetu, které souvisí s probíraným tématem;
- **knih**a, tento modul umožňuje vytvořit vícestránkový studijní materiál s obsahem děleným na kapitoly a podkapitoly. V kurzu je tento modul používán pro většinu studijních článků. U tohoto modulu je pro studenty velkou výhodou možnost, si celou knihu vytisknout;
- **dotazník**, modul dotazník umožňuje realizovat dotazníkové šetření u účastníků kurzu. V kurzu KVD/ALGV je tento modul použit v každém tématu pro evaluaci kurzu směřující ke zvýšení kvality jeho obsahu;
- **anketa**, tento modul umožňuje učiteli položit otázku a definovat výběr z více odpovědí. V kurzu je anketa použita až v závěru jako nástroj pro rychlý sběr dat;
- **stránka**, modul nazvaný stránka umožňuje vytváření jednotlivých stránek, které mohou zobrazovat text, obrázky, zvuk, video, webové odkazy a vložený kód. V kurzu jsou stránky použity převážně pro úkoly, které si studenti mohou vyzkoušet a nejsou určené pro hodnocení a odevzdávání;
- **soubor**, tento modul umožňuje tutorovi kurzu přidávat studijní materiály, ve formě souboru např. pdf, doc, docx, ... V kurzu jsou soubory využity pro vkládání pracovních listů;
- **složka**, tento modul umožňuje zobrazit několik souvisejících souborů v jedné složce. V kurzu je tento modul použit pro zdrojový kód a plánek pro sestrojení.

4.4.2 POUŽITÍ DALŠÍCH NÁSTROJŮ

V této podkapitole je uveden celkový výčet dalších nástrojů, které byly při tvorbě kurzu použity:

- **skrytý text**, v kurzu je také pracováno se skrytým textem. Jelikož Moodle sám o sobě tuto funkci neumožňuje, je použit jednoduchý skript (napsán v javascriptu). Jeho použití není velice intuitivní. Celý kód se vkládá do HTML zobrazení a vypadá takto:

```

<script type="text/javascript">
function toggle(obj)
{
  var obj=document.getElementById(obj);
  if (obj.style.display == "block") obj.style.display = "none";
  else obj.style.display = "block";
}
</script>
<a href="javascript: void(0);" onclick="toggle('q1')">PRVNÍ SLOVO,
NA KTERÉ SE BUDE KLIKAT</a>
<div id="q1" style="display:none;">SKRYTÝ TEXT JEDNA</div> <br>
<a href="javascript: void(0);" onclick="toggle('q2')">DRUHÉ SLOVO,
NA KTERÉ SE BUDE KLIKAT</a>
<div id="q2" style="display:none;">SKRYTÝ TEXT DVA</div>

```

Toto řešení je použito nejčastěji u otázek k promyšlení. Další použité řešení je s kombinací, které zahrnuje textové pole tak, aby si mohli studenti porovnat své řešení s řešením modelovým;

- **textová pole**, vkládání textových polí přímo do studijního materiálu v LMS Moodle také není z uživatelského prostředí možné. Nutno podotknout, že data z textového pole se nikam neodesílají, je tedy pouze na studentech, zda své řešení porovnají s modelovým. Pro použití textového pole bylo opět využito vložení značky v HTML kódu, která vypadá takto:

```

<textarea rows="4" cols="50" style="width: 90%; height: 60px;"></textarea>

```

- **sledování plnění činností**, tento nástroj v LMS Moodle slouží studentům, kteří si mohou pomocí checkboxu zaškrtnout, zda daný studijní materiál nebo studijní činnost již absolvovali. Tento nástroj zpřehledňuje studentům průchod kurzem.
- **embedované aplikace**, v kurzu je také využito embedování aplikací v HTML. Tedy vkládání externích webových aplikací přímo do studijních materiálů. Embedování je použito pro externí videa z YouTube, mapy z Google Maps a programy z aplikace Scratch. I přesto, že Moodle umožňuje přidávání videí z externích adres přes vlastní multimediální přehrávač, embedování bylo použito, protože studenti jsou na prostředí YouTube zvyklí a v tomto formátu jsou videa přehratelná téměř na všech zařízeních.

4.4.3 POUŽITÁ MULTIMÉDIA

V této podkapitole je uveden celkový výčet multimédií, která byla při tvorbě kurzu použita:

- **Video**, v kurzu je použito několik videí z externích zdrojů, ale i video vlastní, které představuje základní orientaci v aplikaci pro vytváření vývojových diagramů (Draw.io).
- **Statický obrázek**, v kurzu je použito velké množství obrázků. Obrázky můžeme rozdělit na výstřižky obrazovky z různých aplikací, ilustrační obrázky a obrázky doplněné o popis jednotlivých sekcí.

4.5 STRUKTURA JEDNOTLIVÝCH KAPITOL

Obsah celého e-kurzu vychází z plánů pro přednášky a semináře. Celý e-kurz je logicky uspořádán do devíti témat (včetně úvodu a závěru). V této kapitole bude věnován prostor pro krátké představení každé kapitoly kurzu.

4.5.1 Úvod

První téma nazvané Úvod slouží pro přivítání studentů předmětu KVD/ALGV a stručné představení cílů. Toto téma obsahuje dvě fóra (jedno určené pro novinky a aktuality kurzu, druhé pro studenty a pro jejich otázky, náměty a tipy), dále toto téma obsahuje vstupní dotazník.

4.5.2 ÚVOD DO ALGORITMIZACE

Celé toto téma slouží k pochopení základních náležitostí, které musí algoritmus splňovat, aby mohl být za algoritmus považován.

Toto téma obsahuje jednu knihu, jeden URL odkaz a dotazník pro zpětnou vazbu.

Knihu obsahuje pět kapitol a jednu podkapitulu. V tomto studijním materiálu je vysvětleno, co je to algoritmus, jaké má vlastnosti, příklady algoritmu, jak lze algoritmus zapsat a co je to rekurzivní algoritmus.

Přímo ve studijním materiálu nalezneme několik cvičení na zamyšlení pro studenty (úskalí slovního zápisu algoritmu, vlastní popis vlastností algoritmu, popis reálného algoritmu pomocí jeho vlastností a určení kde se v programu nachází rekurze).

URL odkaz, který je součástí tohoto tématu, vede na webové stránky České encyklopedie algoritmů, kterou mohou studenti využít jako doplňkový zdroj informací.

4.5.3 VÝVOJOVÉ DIAGRAMY

Následující téma je věnované praktické tvorbě vývojových diagramů. Studenti po prostudování tohoto tématu budou znát základní značka a umět pracovat s aplikací Draw.io.

Toto téma obsahuje jednu knihu, tři stránky s cvičeními a dotazník pro zpětnou vazbu.

Kniha obsahuje dvě kapitoly a čtyři podkapitoly. V tomto studijním materiálu je popsáno, jakou normou se grafická podoba zápisu algoritmů řídí, základní značky a řídicí struktury vývojových diagramů, přehled všech značek a řídicích struktur v jednotném přehledu, jak vytvářet vývojové diagramy v prostředí draw.io včetně krátkého videotutoriálu.

Jednotlivá cvičení vždy obsahují i modelové řešení. První a druhé cvičení (Collatzův problém a člověče nezlob se) slouží k procvičení práce v prostředí draw.io, ale také k vlastnímu tvoření požadovaného algoritmu. Třetí cvičení nazvané „Je to algoritmus?“ slouží pro procvičení základních vlastností algoritmu, kdy se studenti snaží odhalit, zda nedošlo k jejich porušení.

4.5.4 STRUKTUROGRAMY

Třetí téma je věnované praktické tvorbě strukturogramů. Studenti po prostudování tohoto tématu budou znát základní značky a také budou schopni pracovat s aplikací Structorizer.

Toto téma obsahuje jednu knihu, dvě stránky s cvičeními a dotazník pro zpětnou vazbu.

Kniha obsahuje tři kapitoly a dvě podkapitoly. Studijní text je věnován popsání strukturogramů, porovnání strukturogramu a vývojového diagramu, základním značkám, popisu aplikace Structorizer a porovnání verzí pro počítač a mobilní zařízení se systémem Android, základnímu nastavení aplikace a základní práci v tomto prostředí.

Toto téma obsahuje dvě cvičení. Obě cvičení mají vždy vytvořené modelové řešení. První cvičení slouží k procvičení práce v prostředí Structorizer, kdy mají studenti za úkol převést vývojový diagram na strukturogram, a porovnat výhody obou zápisů. V druhém cvičení mají studenti za úkol vytvořit strukturogram pro řešení kvadratické rovnice.

4.5.5 ROZVOJ ALGORITMIZACE U DĚTÍ A STUDENTŮ NA ZŠ A SŠ

V pořadí čtvrté téma je věnováno rozvoji algoritmizace u dětí a studentů na ZŠ a SŠ. Studenti po prostudování tohoto tématu získají široký přehled o používaných aplikacích

a robotických hračkách a stavebnicích, které slouží k rozvoji algoritmizace u žáků a studentů.

Toto téma obsahuje dvě knihy, pět URL odkazů a dotazník pro získání zpětné vazby.

První kniha s názvem Robotika pro rozvoj algoritmizace u dětí a studentů obsahuje čtyři kapitoly. Studijní texty doplněné o infografiku a videa jsou věnována robotům Bee-Bot, Dash a Dot, Ozobot a robotické stavebnici LEGO Mindstorms.

Druhá kniha s názvem „Aplikace pro rozvoj algoritmizace u dětí a studentů“ obsahuje čtyři kapitoly. Studijní text zahrnují seznámení s aplikacemi code.org, Scratch, Code Combat a Kodu.

První URL odkaz vede na Metodický portál RVP a studenti se v něm dozví, jak je možné využít code.org na prvním stupni základní školy. Následující odkaz vede na Google Doodle ze 4. prosince 2017, který oslavuje 50 let dětského programování, tento Doodle obsahuje velice hezky vytvořenou aktivitu blokového programování. V pořadí třetí odkaz vede opět na metodický portál a představuje aktivitu pro studenty středních škol, kteří si sami programují aplikace pro mobilní telefon s operačním systémem Android. Čtvrtý odkaz vede na webové stránky soutěže Bobřík informatiky, kde si studenti mohou vyzkoušet jednotlivé úkoly. Poslední odkaz vede na projekt „Umíme programovat“, ve kterém mohou studenti načerpat základní programovací dovednosti.

4.5.6 PROGRAMUJEME VE SCRATCHI

Následující téma je celé věnováno jednomu dětskému programovacímu prostředí Scratch. Po prostudování tohoto tématu budou studenti seznámeni s vývojem aplikací v tomto prostředí a získají manuál všech programových bloků. Studenti se také seznámí se základními principy a technikami, které se používají i ve standardních programovacích jazycích. Dále se studenti seznámí s vytvářením programů želví grafiky v různých prostředích a v neposlední řadě získají i praktické dovednosti s tvorbou želví grafiky v prostředí Scratch včetně rekurzivního volání programu.

Toto téma obsahuje dvě knihy, jeden soubor, jednu stránku s cvičením a dotazník pro zpětnou vazbu.

První kniha s názvem Představení aplikace Scratch obsahuje 12 jednotlivých kapitol, které se věnují základnímu ovládnutí aplikace, rozvržení scén a postav, základním programovým

sekcím (pro události, pohyb, zvuk, vzhled, pero, data, ovládání, vnímání a operátory) a také vytvářením nových bloků.

Následující kniha nazvaná Želví grafika obsahuje pět kapitol, které jsou věnovány představení želví grafiky v různých programovacích prostředích (Umíme programovat, EasyLogo a Scratch), dále je zde uvedena infografika pro inicializaci módu kreslení ve Scratchi, příklady pro vykreslení základních obrazců (čtverec, kružnice, obrazce ze čtverců a kružnic, spirála a domeček jedním tahem) a následují dva příklady s využitím rekurze (Kochova křivka a Pythagorův strom).

Soubor, který je v tomto tématu dostupný (ve formátu pdf), umožňuje studentům získat informace o tom, jak probíhá programátorská soutěž Scratch Cup, a vyzkoušet si jednotlivé příklady určené pro žáky základních škol.

Na stránce nazvané Atari Breakout je pro studenty připraveno praktické cvičení v programovacím prostředí Scratch. Toto cvičení je také doplněno o rozšiřující zadání úkolu.

4.5.7 PROGRAMUJEME S LEGO MINDSTORMS

Předposlední kapitola je věnována robotické stavebnici LEGO Mindstorms. Studenti se po prostudování této kapitoly budou orientovat v možnostech této stavebnice, senzorech, které je možné ke stavebnici připojit a také se seznámí s materiály, které je možno využít při výuce na základní škole.

Toto téma obsahuje jednu knihu, dva soubory (ve formátu pdf), jednu složku a URL odkaz.

Kniha nazvaná „Představení vývojového prostředí EV3 a LEGO Digital Designer“ obsahuje 4 kapitoly. V těchto čtyřech kapitolách je stručně představeno vývojové prostředí EV3, kostka a její možnosti připojení, přehled motorů a senzorů, které je možno ke kostce připojit, a práce v prostředí LEGO Digital Designer.

Následují dva pdf soubory, ve kterých jsou uvedeny příklady (jednoduchý robot a houpačka) a jedna složka se zdrojovým souborem a plánkem sestavení závory.

URL odkaz vede na oficiální stránky LEGA na ZČU, kde studenti mohou načerpat mnoho námětů, inspirace a návodů.

4.5.8 ZÁVĚR

V této kapitole jsou pro studenty dostupné zdroje, z kterých bylo při vytváření kurzu čerpáno a také závěrečný dotazník pro zjištění výstupních znalostí, dovedností a postojů studentů. V tomto dotazníku byla také zařazena evaluace celého kurzu.

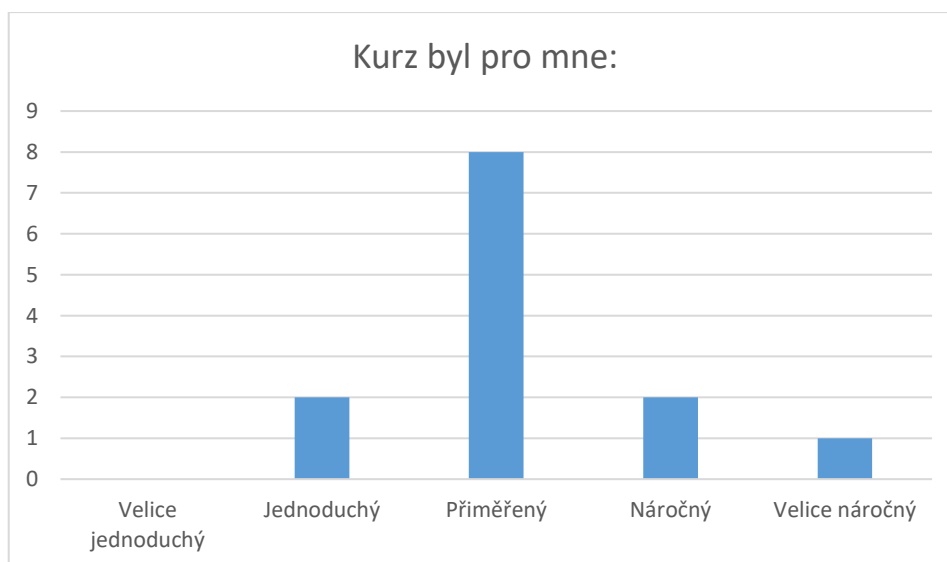
4.6 PŘÍNOS KURZU

Pro zjištění, zda je kurz srozumitelný a příjemný pro vysokoškolské studenty, byl vytvořen evaluační dotazník, ve kterém se studenti mohli vyjádřit k náročnosti kurzu, srozumitelnosti kurzu, podrobnosti kurzu, multimédiím, cvičením, přehlednosti, zajímavosti a použití externích zdrojů. Na závěr evaluačního dotazníku byly zařazeny dvě otázky s volnou odpovědí („Je něco, co byste v kurzu doplnili, vypustili nebo změnili?“ a „Další připomínky“).

V následujících podkapitolách je zpracováno vyhodnocení výsledků evaluačního dotazníku.

4.6.1 NÁROČNOST KURZU

Naprostá většina respondentů (62 %) považuje kurz za přiměřeně náročný. Pouze několika respondentům se zdá kurz spíše náročný (Náročný 15 % a Velice náročný 8 %). 2 respondenti (15 %) označili kurz jako jednoduchý.

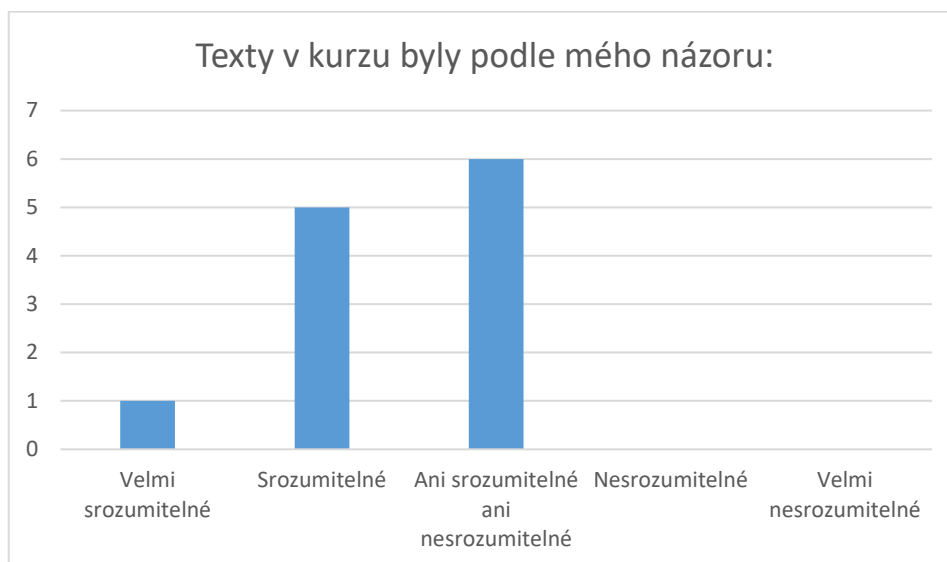


Graf 1: (Evaluace kurzu) Náročnost kurzu

Podle respondentů byla náročnost kurzu vhodně zvolena, kurz pro většinu respondentů nebyl ani příliš jednoduchý, ani velice náročný. Při obou variantách by mohlo dojít ke ztrátě motivace studentů. Náročnost kurzu je tedy vhodné zachovat.

4.6.2 SROZUMITELNOST KURZU

6 respondentů (46 %) označilo možnost ani srozumitelné/ani nesrozumitelné. 5 studentů (38 %) označilo odpověď srozumitelné a pro jednoho respondenta byl kurz velice srozumitelný.

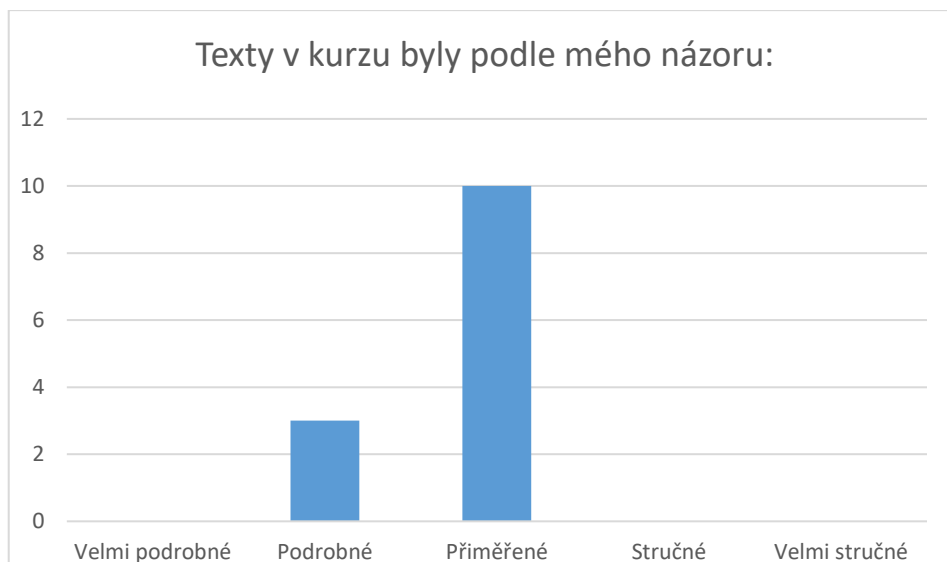


Graf 2: (Evaluace kurzu) Srozumitelnost kurzu

I přesto, že v dotazníkovém šetření nikdo neuvedl, že je kurz nesrozumitelný nebo velmi nesrozumitelný, je zde zastoupeno největší procento těch, kteří měli ke srozumitelnosti kurzu nějaké výhrady (6 respondentů). Proto by bylo vhodné kurz ještě lépe strukturovat a použít více názorných příkladů.

4.6.3 PODROBNOST KURZU

Menší části respondentů označilo texty za podrobné (23 %). Zbýlých 77 % studentů uvedlo, že texty v kurzu byly přiměřené k probírané látce.

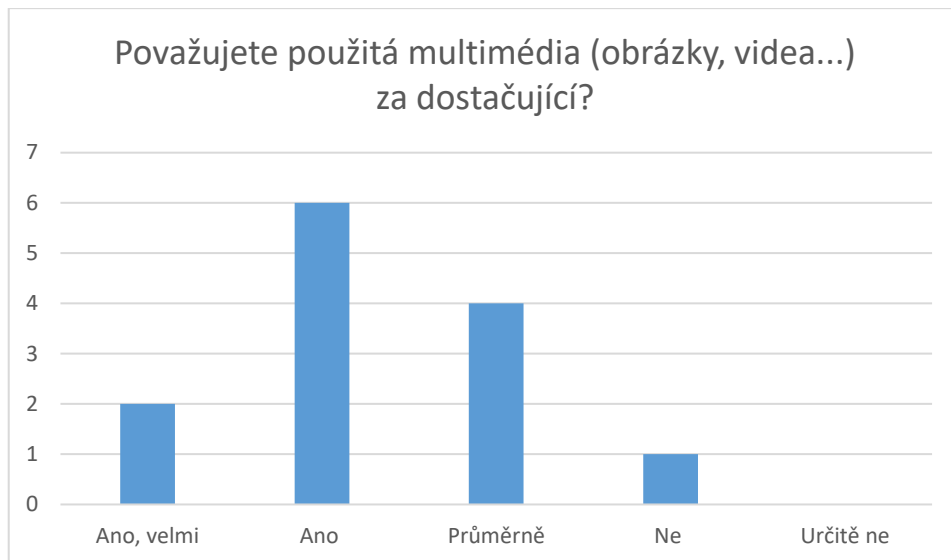


Graf 3: (Evaluace kurzu) Podrobnost kurzu

Podrobnost textů v kurzu většina respondentů označila za přiměřenou. Texty v kurzu tedy není potřeba více rozepisovat.

4.6.4 DOSTATEČNOST MULTIMÉDIÍ

Větší část respondentů je s použitými multimédii spokojena (Ano 46 %, Ano velmi 15 %). Spíše průměrné zastoupení multimédií uvádí 31 % respondentů. Jeden respondent je s multimédií nespokojený.

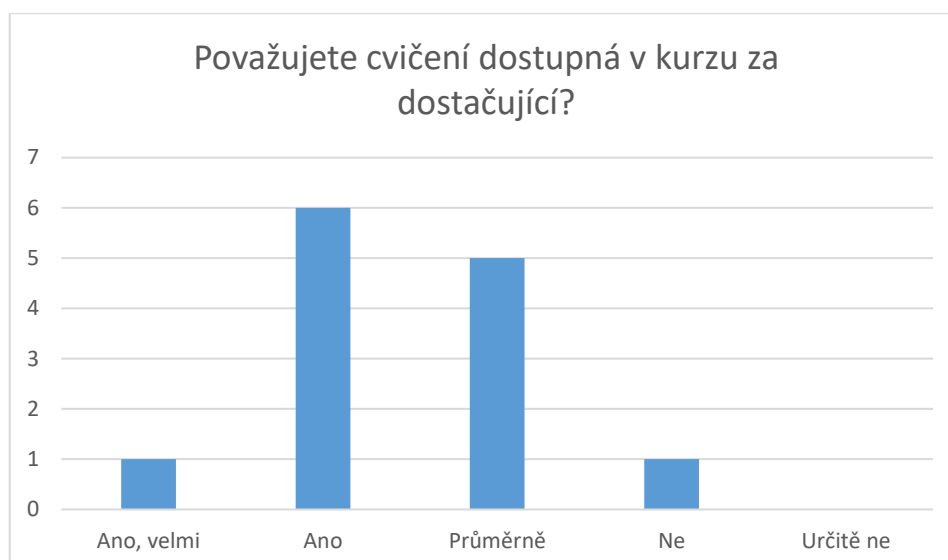


Graf 4: (Evaluace kurzu) Dostatečnost multimédií

Zastoupení multimédií v kurzu je podle respondentů dostatečné. Z průzkumu také vyplývá, že přidání dalšího multimediálního materiálu by kurzu prospělo a studenti by ho poté vnímali lépe.

4.6.5 DOSTATEČNOST CVIČENÍ V KURZU

Velice podobný případ jako v případě multimédií (Graf 4). Studenti jsou většinou s množstvím cvičení spokojeni, nebo je označují za průměrné. Opět jeden respondent uvádí, že s cvičeními není spokojen.

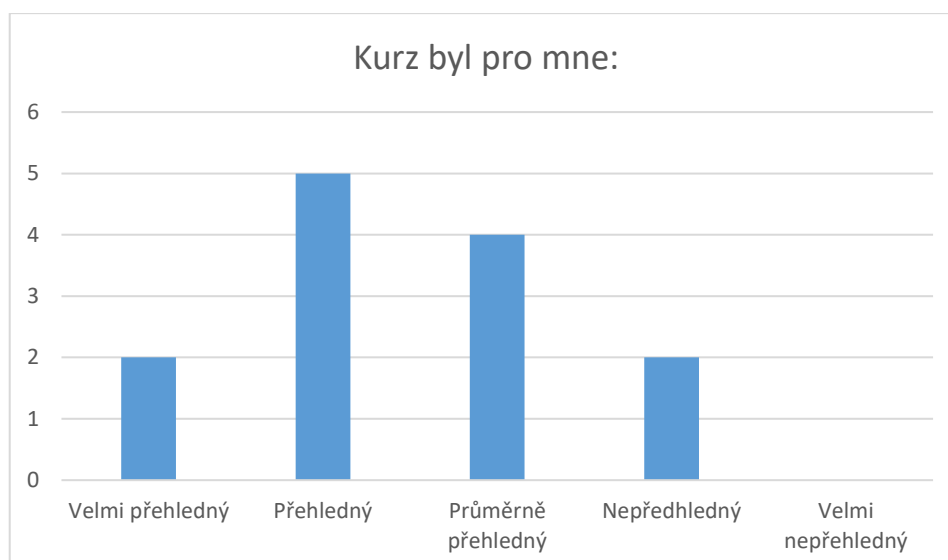


Graf 5: (Evaluace kurzu) Dostatečnost cvičení v kurzu

Další možností vylepšení kurzu je, přidání dalších cvičení k jednotlivým kapitolám, které by uspokojilo i část respondentů, kteří považují cvičení v kurzu za průměrné, nebo s ní nejsou spokojeni.

4.6.6 PŘEHLEDNOST KURZU

Studenti předmětu KVD/ALGV nejčastěji uvádějí, že je pro ně kurz přehledný (38 %), velmi přehledný je kurz pro 15 % respondentů. Průměrnou přehlednost uvádí 31 % respondentů. Nepřehlednost kurzu uvádí 15 % respondentů.

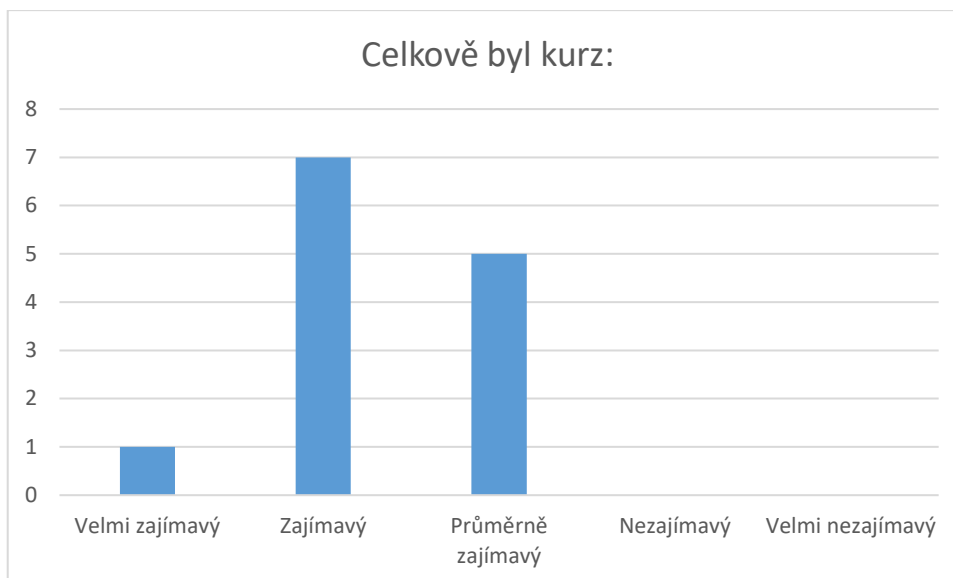


Graf 6: (Evaluace kurzu) Přehlednost kurzu

Při tvorbě kurzu byl na přehlednost kladený velký důraz. Jednotlivé kapitoly byly od sebe výrazně odděleny a struktura jednotlivých částí kurzu byla logicky řazena do modulu s názvem Kniha. Respondenti, kteří označili kurz jako nepřehledný, nebo průměrně přehledný nemuseli mít zkušenost s prostředím LMS Moodle, který je na ZČU velice rozšířený a práce v něm nemusí přijít studentům příliš intuitivní. Do kurzu by bylo vhodné přidat kapitolu nazvanou „Základní orientace v prostředí LMS Moodle“, která by studentům mohla v orientaci v kurzu pomoci.

4.6.7 ZAJÍMAVOST KURZU

Z evaluace kurzu vyplývá, že kurz byl pro studenty spíše zajímavý (jeden respondent hodnotil kurz jako velice zajímavý, 54 % respondentů jako zajímavý). Zbýlých 38 % poté hodnotilo kurz jako průměrně zajímavý.

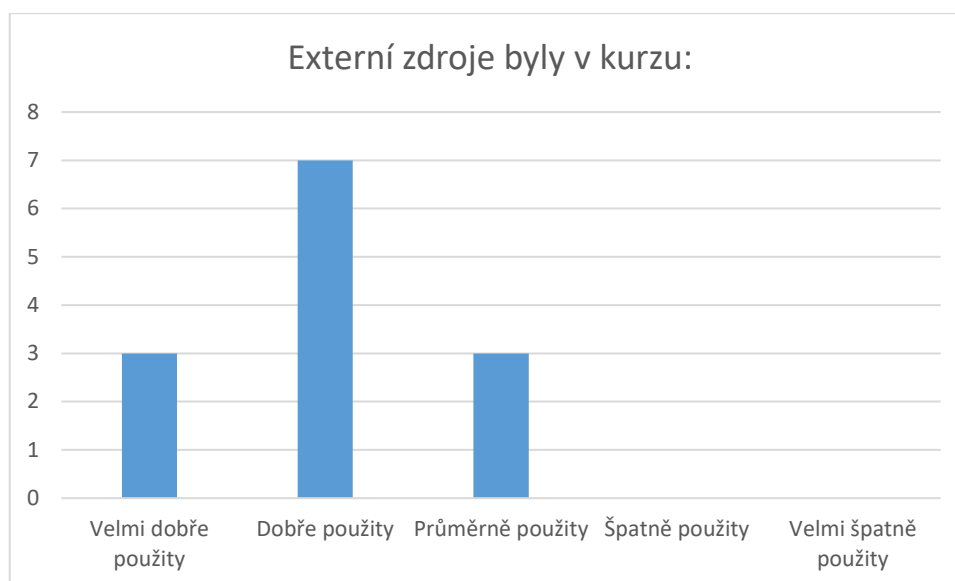


Graf 7: (Evaluace kurzu) Zajímavost kurzu

Celkově studenti hodnotili kurz v oblasti zajímavosti velice kladně. Aby byl kurz ještě zajímavější, bylo by možné přidat více multimédií, cvičení a zajímavostí z reálné praxe z oblasti algoritmizace.

4.6.8 POUŽITÍ EXTERNÍCH ZDROJŮ

Celkem 77 % respondentů použití externích zdrojů hodnotí velice kladně (54 % externí zdroje byly dobře použity a 23 % externí zdroje byly velmi dobře použity). Zbýlých 23 % respondentů hodnotí, že zdroje v kurzu byly použity průměrně.



Graf 8: (Evaluace kurzu) Použití externích zdrojů

Nedílnou součástí on-line kurzu bylo použití externích zdrojů pro individuální prostudování materiálů, pro studenty, kteří by se rádi v probíraných tématech ponořili do větší hloubky. Z průzkumu vyplývá, že studenti tuto možnost velice oceňují.

4.6.9 DALŠÍ PŘIPOMÍNKY

Na závěr evaluace kurzu byl pro studenty připraven prostor, ve kterém se mohli vyjádřit, co by v kurzu doplnili, vypustili nebo změnili. K této otázce se vyjádřili 3 respondenti. Odpovědi se týkaly: více příkladů na procvičení, více návodů a vypuštění videí, které prý respondenty pouze zdržují.

V dalších připomínkách jeden respondent uvedl, že během studia předmětu KVD/ALGV neměl vůbec potřebu kurz navštěvovat a stačila mu prezenční výuka předmětu.

4.7 NÁVRH VYLEPŠENÍ KURZU

Kurz byl upravován již při běhu předmětu KVD/ALGV. Nejdříve byla opravena kvalita obrázků, která byla díky špatně zvolené kompresi pro potřebu kurzu nedostatečná.

Dále byly v kurzu opraveny zdroje na externí webové stránky, které v průběhu kurzu přestaly fungovat. Tato skutečnost byla nahlášena studentkou předmětu.

V kurzu byly také upraveny některé texty.

Mezi další možné návrhy vylepšení kurzu patří:

- přidání více multimédií,

- přidání více cvičení z reálné praxe,
- přidání kapitoly základní orientace v LMS Moodle,
- jazyková korektura textů,
- rozšíření kurzu o další kapitolu věnující se teorii grafů,
- přidání cvičení, které by mohl učitel hodnotit,
- zařazení autotestů po každé kapitole.

5 VÝZKUMNÉ ŠETŘENÍ

V rámci výzkumného šetření proběhla práce se dvěma dotazníky. Jednalo se o dotazník poskytnutý účastníkům kurzu před spuštěním výuky a po jejím skončení. Tyto dotazníky byly zaměřeny na znalosti, dovednosti v oblasti algoritmizace a postoje studentů vůči výroky o programování a algoritmizaci.

5.1 METODOLOGIE VÝZKUMU

Výzkum byl realizován pomocí testového dotazníku vlastní konstrukce v digitální podobě (pomocí nástroje Google formuláře). V dotaznících byly využity tyto typy otázek: jedna odpověď, více odpovědí, stručná odpověď a lineární stupnice.

5.2 CHARAKTERISTIKA VÝZKUMNÉHO SOUBORU

Respondenty šetření byli studenti 1. ročníku oboru Informatika se zaměřením na vzdělávání. Cílová skupina byla oslovena vyučujícím kontaktně na prvních seminářích předmětu KVD/ALGV a poté i pomocí nástroje v LMS Moodle (Novinky – nástěnka pro aktuality), tento nástroj zprávu také doručil na všechny e-mailové adresy studentů. Dotazník vyplnilo pouze 15 respondentů z celkového počtu 23 zapsaných do kurzu. Takto nízká návratnost je nejspíše způsobena tím, že v průběhu semestru několik studentů své studium přerušilo, ukončilo, nebo jim bylo ukončeno pro neplnění studijních povinností.

Pouze jeden dotazník byl vyhodnocen jako neplatný, a to z toho důvodu, že odpověď byla obdržena až 17. 3. 2018, v žádném případě se tak nemohlo jednat o testování vstupních znalostí, dovedností a postojů.

Výstupního testování se zúčastnilo 13 respondentů.

5.3 ČASOVÝ HARMONOGRAM VÝZKUMU

První část výzkumu (vstupní dotazník) byla realizována v časovém intervalu od 9. 2. 2018 do 21. 2. 2018. Velký důraz byl kladen na co nejrychlejší sběr dat (ideálně ještě před začátkem výuky).

Druhá část výzkumu proběhla v termínu od 21. 5. do 20. 6. 2018.

5.4 POPIS DOTAZNÍKŮ

Vstupní testování obsahovalo 12 znalostních a dovednostních otázek ze širokého spektra algoritmizace. Následovalo 9 otázek na určení aktuálních postojů studentů. V poslední části dotazníkového šetření byly otázky zaměřeny na demografické údaje o respondentovi (celkem 14 otázek, z nichž dvě závisely na předchozích odpovědích studentů). Tyto otázky byly záměrně umístěny až na konec dotazníku, aby respondent nebyl od vyplňování odrazen. Celý vstupní dotazník je anonymní.

Výstupní testování obsahovalo 12 znalostních a dovednostních otázek ze širokého spektra algoritmizace. Následovalo 9 otázek na určení aktuálních postojů studentů. Poslední část testu obsahovala evaluaci celého on-line kurzu.

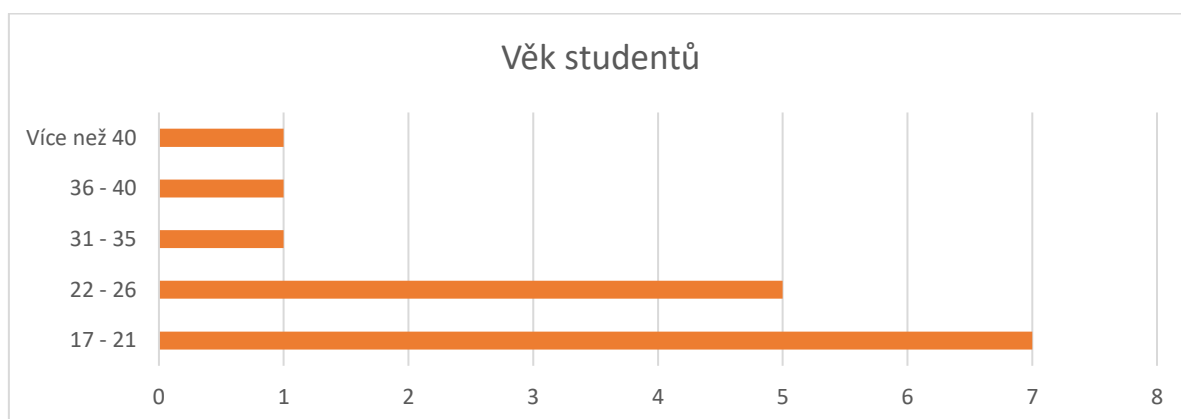
Zpracování jednotlivých odpovědí bylo provedeno analýzou a následným tabulkovým a grafickým znázorněním.

5.5 CHARAKTERISTIKA RESPONDENTŮ

V této kapitole jsou představeny základní informace o respondentech výzkumného šetření.

5.5.1 VĚK STUDENTŮ

Z grafu vyplývá, že největší část studentů je ve věku 17–21 let (46,67 %). Následuje rozmezí 22–26 (33,33 %). Další hodnoty (31–35, 36–40 a více než 40) obsáhlo vždy po (6,67 %).



Graf 9: (Charakteristika respondentů) Věk studentů

Největší část studentů předmětu KVD/ALGV spadá do kategorie 17–21. Nejčastěji se tedy jedná o čerstvé absolventy středních škol, nebo studenty, kteří přešli z jiné vysoké školy, nebo fakulty.

5.5.2 VYSTUDOVANÁ SŠ

Z grafu vyplývá, že největší část studentů předmětu KVD/ALGV absolvovalo střední odbornou školu zaměřenou na IT (40 %). Střední průmyslovou školu, stejně tak jako gymnázium v průzkumu vystudovalo vždy 20 % studentů. Střední odbornou školu elektrotechnickou, obchodní akademii a konzervatoř studoval vždy jeden respondent ze vzorku (6,67 %).



Graf 10: (Charakteristika respondentů) Vystudovaná střední škola

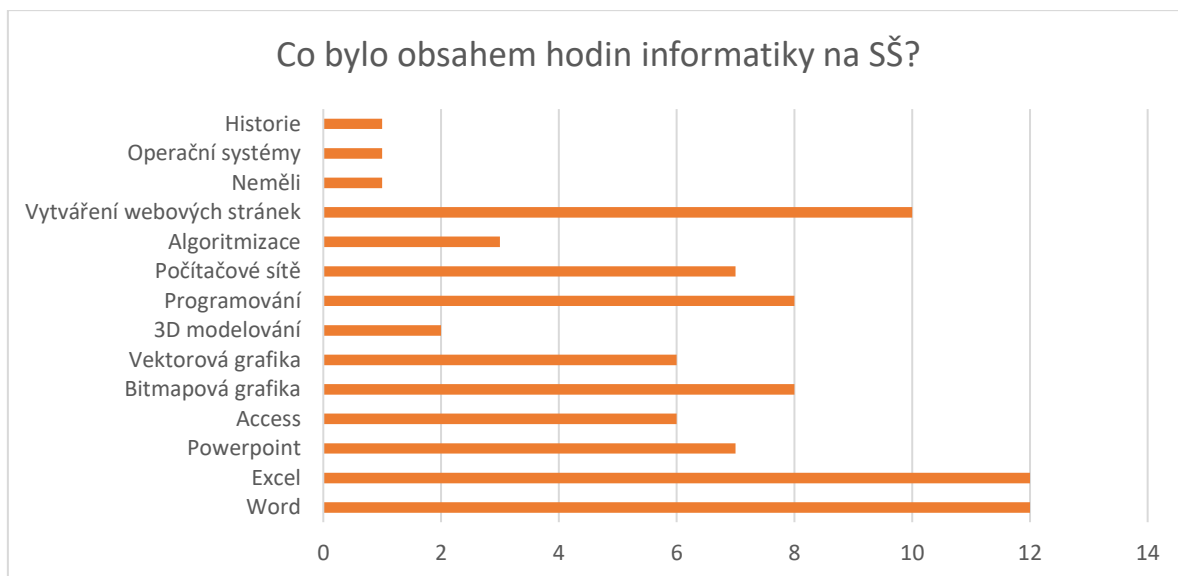
Z průzkumu je patrné, že velká většina respondentů absolvovala školu, ve které měla možnost seznámit se s programováním.

5.5.3 OBSAH HODIN

Z třetího grafu je patrné, že celých 80 % respondentů během svého studia na střední škole přišlo do kontaktu s balíčkem Microsoft Office (Word – 80 %, Excel – 80 %, Powerpoint – 46,7 %, Access 40 %). Vytvářením webových stránek se na střední škole zabývalo 66,7 % studentů.

Programování jako předmět bylo vyučováno u nadpoloviční většiny (53,4 % respondentů). Algoritmizace byla vyučována u 20 % respondentů. Na středních školách respondentů byla vyučována ve značné míře i počítačová grafika (bitmapová 53,4 %, vektorová 40 %). Počítačové sítě pak 46,7 % studentů. 3D modelování uvádí 13,3 % respondentů.

Jeden z respondentů uvádí, že na střední škole byla probíraná látka z historie počítačů. Další se mimo jiné na střední škole zabíral operačními systémy. Jeden respondent uvádí, že informatiku na střední škole vůbec neměl.



Graf 11: (Charakteristika respondentů) Obsah hodin informatiky

Z tohoto grafu je názorně vidět, že většina respondentů přichází ze škol, ve kterých je výuka informatiky na vyšší úrovni (viz Graf 10: Vystudovaná střední škola). V předmětu KVD/ALGV je tedy velké procento studentů, kteří mají dobré předpoklady pro úspěšné studium tohoto předmětu. Na druhou stranu je zde téměř poloviční zastoupení studentů, kteří s programováním nikdy do styku nepřišli. Proto je zařazení předmětu Algoritmizace velice žádoucí a může tak sloužit jako opakovací předmět pro studenty, kterým se na střední škole dostalo kvalitního vzdělání v oblasti informačních a komunikačních technologiích, ale i jako pomyslný odrazový můstek pro ty, kterým se vzdělání v takovéto míře nedostalo.

5.5.4 PROGRAMOVÁNÍ

V předešlém grafu je vidět (viz Graf 11: Obsah hodin informatiky), že téměř poloviční menšina se na střední škole nesetkala s programováním (46,7 %). Tato otázka si kladla za cíl získat od studentů přehled, ve kterých programovacích jazycích výuka probíhala. Nejčastěji zmiňovaný programovacím jazykem u respondentů se stala objektově orientovaná Java (33,4 % z celku). Dále byl často zastoupen skriptovací jazyk Javascript (26,7 % z celku). Shodný počet respondentů programoval v jazycích php, C, C++ a C# (20 %). Pascal uvedlo 13,3 % studentů.



Graf 12: (Charakteristika respondentů) Programování na střední škole

Je pozitivní, že studenti předmětu KVD/ALGV, kteří mají předchozí zkušenosti s programovacími jazyky, většinou programovali v jazycích, které jsou dobře a často využívané v praxi. Výukové jazyky, mezi které můžeme zařadit Pascal (13,3 %) a Object Pascal (0 %) byly ve výuce studentů zařazeny minimálně. Python, který má poměrně jednoduchou syntaxi také nebyl na žádné škole z uvedeného vzorku použit.

5.5.5 PŘEDCHOZÍ STUDIUM NA VYSOKÉ ŠKOLE

Jedna třetina studentů odpověděla, že se nejedná o první vysokou školu, na které kdy studovali.

Studenti, kteří již na některé vysoké škole studovali, uvádějí také na kterých: Univerzita Karlova, Univerzita Palackého v Olomouci, Západočeská univerzita v Plzni (Fakulta pedagogická, Fakulta elektrotechnická, Fakulta strojní). Jeden respondent dokonce uvádí dvě předchozí studia na Fakultě elektrotechnické a Fakultě strojní.

Většina respondentů (75 %), kteří již v minulosti studovali na jiné vysoké škole, se v rámci tohoto studia setkali s programováním nebo algoritmizací.

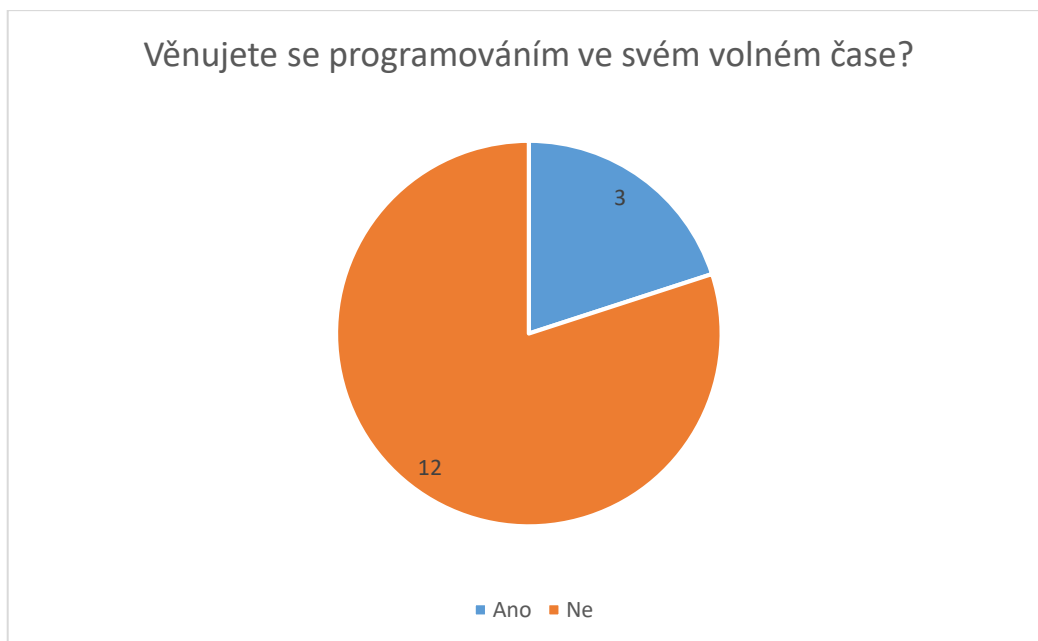


Graf 13: (Charakteristika respondentů) Předchozí studium na vysoké škole

Celkem velké procento respondentů (33,3 %) uvádí, že již na některé vysoké škole studovalo, přechod na FPE může mít mnoho příčin, mezi které může patřit např. velká obtížnost učiva, neztotožnění se s přípravou na budoucí zaměstnání, nebo potřeba rozšířit svou aprobaci pro vyučování na základní nebo střední škole.

5.5.6 PROGRAMOVÁNÍ A VOLNÝ ČAS

Z následujícího grafu vyplývá, že pouze 20 % respondentů se věnuje programování ve svém volném čase.

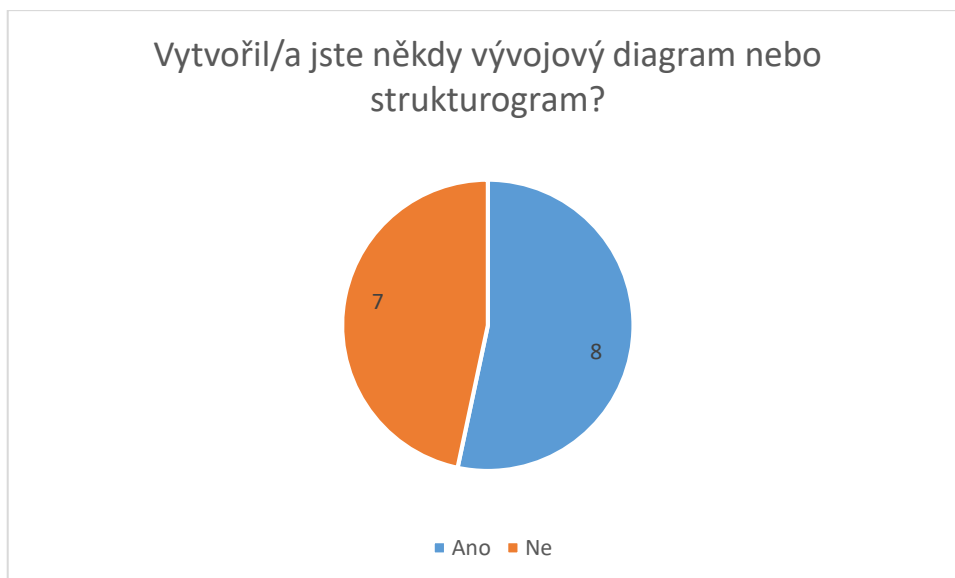


Graf 14: (Charakteristika respondentů) Programování a volný čas

Pokud vezmeme v potaz, že pouze 53,4 % studentů se s programováním setkalo na střední škole (viz Graf 11), není toto číslo zanedbatelné.

5.5.7 TVORBA VÝVOJOVÝCH DIAGRAMŮ A STRUKTUROGRAMŮ

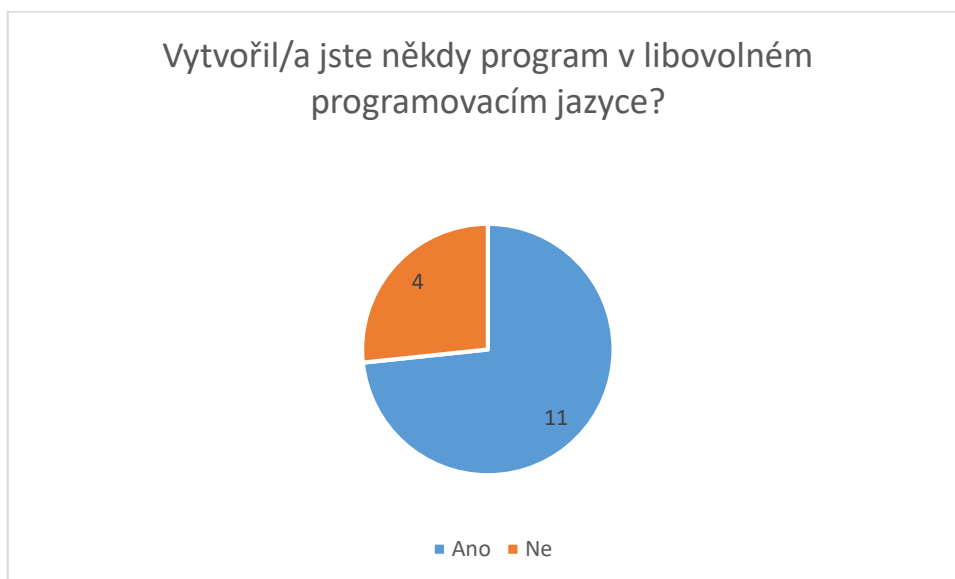
U této otázky větší část (53,4 %) respondentů uvádí, že se s tvorbou vývojového diagramu, nebo strukturogramu už v minulosti setkala.



Graf 15: (Charakteristika respondentů) Tvorba vývojových diagramů a strukturogramů

5.5.8 TVORBA PROGRAMU

Drtivá většina respondentů (73,4 %) uvádí, že někdy ve svém životě již vytvořili nějaký program



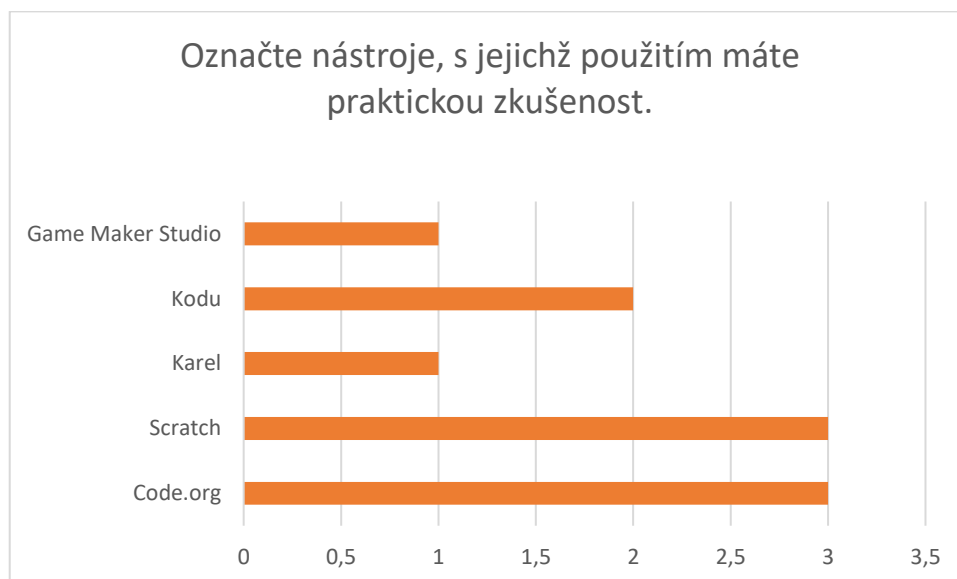
Graf 16: (Charakteristika respondentů) Tvorba programu

Vysvětlením takto vysokých hodnot může být např. použití jednoduchých výukových prostředí pro tvorbu programů.

5.5.9 PRAKTICKÁ ZKUŠENOST S DĚTSKÝMI PROGRAMOVACÍMI PROSTŘEDÍMI

Mezi studenty předmětu KVD/ALGV panuje největší povědomí o programovacím prostředí Scratch (20 %) a výukovém portálu Code.org (20 %). S prostředím, které bylo vytvořeno firmou Microsoft – Kodu se setkala a prakticky si vyzkoušelo 13,3 % respondentů. Se

starším prostředím Karel přišel do kontaktu jeden respondent (6,67 %), stejně tak jako s prostředím pro vývoj her Game Maker Studio (6,67 %).

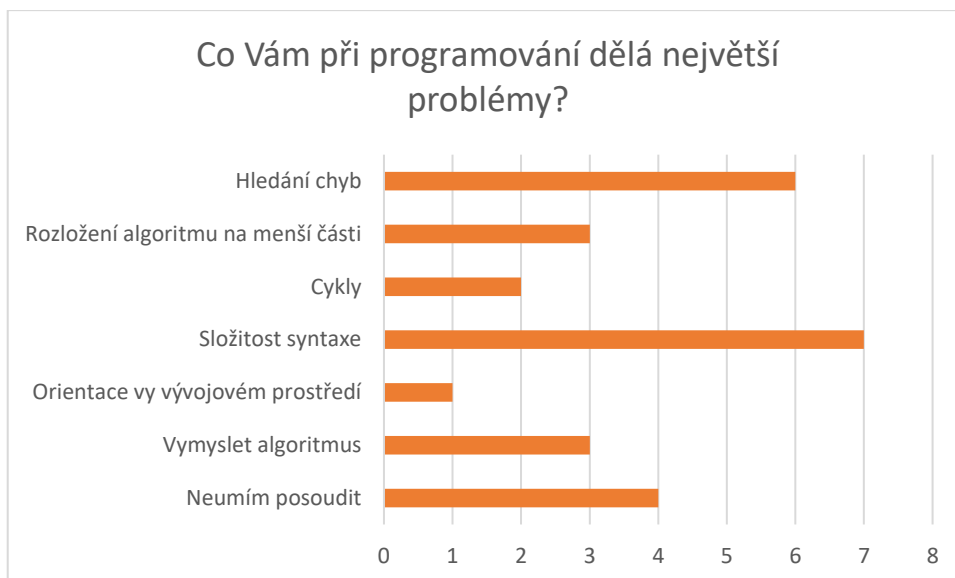


Graf 17: (Charakteristika respondentů) Praktická zkušenost s dětskými programovacími prostředími

Code.org a Scratch patří v dnešní době mezi nejoblíbenější prostředí pro základní výuku programování, rozvoj logického a algoritmického myšlení. Dalo se tedy předpokládat, že si studenti práci v těchto prostředí již vyzkoušeli.

5.5.10 PROBLÉMY PŘI PROGRAMOVÁNÍ

Tato otázka by měla zodpovědět, co způsobuje studentům největší problém při programování. Jako nejvíce problematická část studentům přijde složitost syntaxe programovacího jazyka (46,7 %). Téměř stejný počet respondentů (40 %) má při programování problém s hledáním chyb již v hotovém kódu. Rozložení algoritmu na menší části, tak aby se dal zapsat do programovacího jazyka, činí problém 20 % respondentů. Stejnému počtu respondentů (20 %) dělá problém algoritmus vymyslet. 13,3 % studentů při programování dělají problém cykly. Jeden respondent (6,7 %) se obtížně orientuje ve vývojovém prostředí. 26,7 % studentů neumí tuto otázku posoudit.

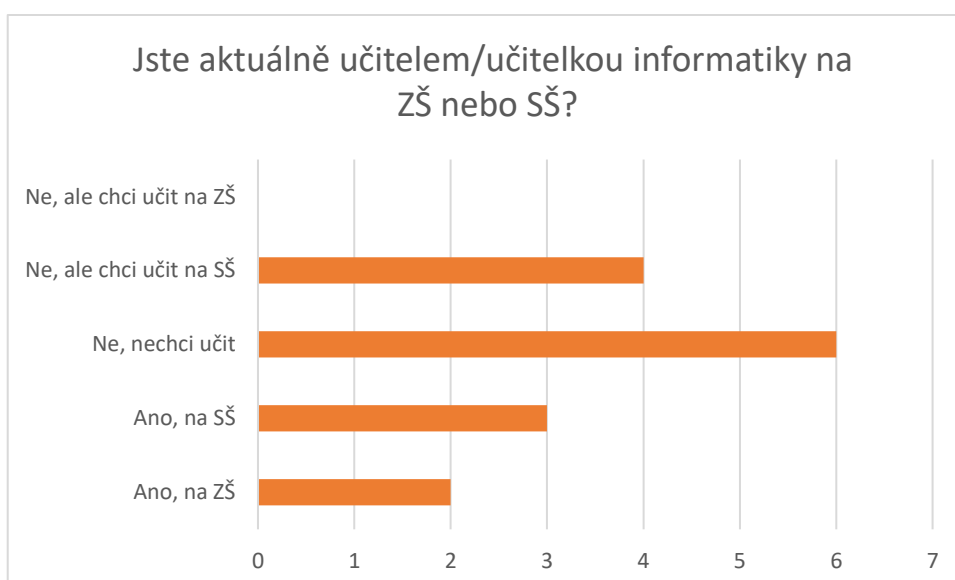


Graf 18: (Charakteristika respondentů) Nejčastější problémy při programování

Jako nejčastější problém se jeví složitost syntaxe jednotlivých příkazů programovacího jazyka. Proto je vhodné i na vysoké škole nejdříve pracovat s blokovým programováním, nebo volit jazyk, který má syntaxi jednodušší.

5.5.11 JSTE NEBO CHCETE SE STÁT UČITELEM/UČITELKOU?

Ve výzkumném vzorku se nachází 33,3 % učitelů (na ZŠ 13,3%, na SŠ 20%). Respondenti, kteří se nechtějí stát učitelem, jsou zastoupeni 40 %. 26,6 % odpovídajících by chtěli po absolvování učit na střední škole.



Graf 19: (Charakteristika respondentů) Zastoupení učitelů mezi studenty

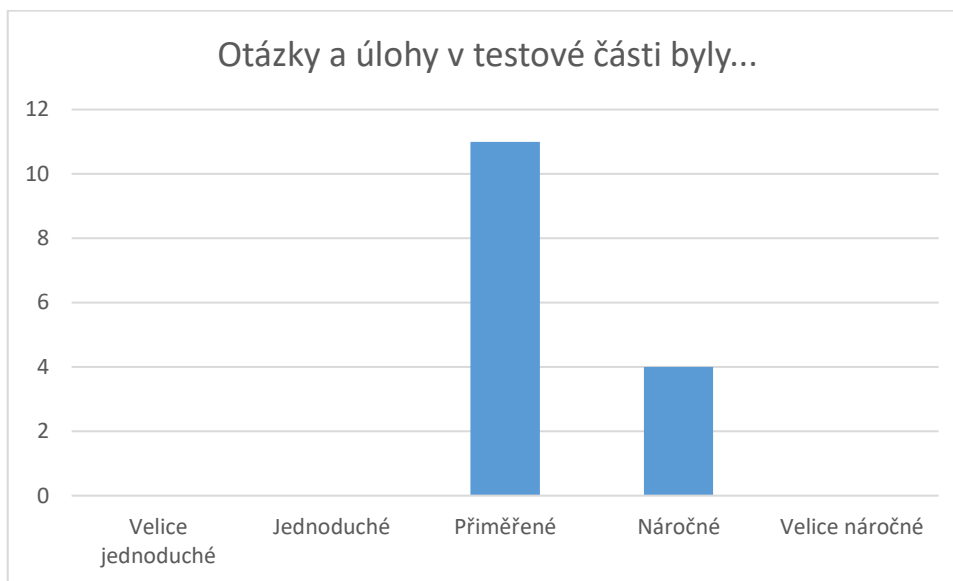
Velkým překvapením, je že nikdo z dotazovaných nehodlá učit na základní škole, toto rozhodnutí se může u studentů během studia ještě změnit.

5.6 POSTOJE PŘI VSTUPNÍM TESTOVÁNÍ

V následujících podkapitolách je provedena analýza jednotlivých výroků, které byly pro vstupní testování použity.

5.6.1 OBTÍŽNOST TESTOVÉ ČÁSTI

Více než dvě třetiny respondentů (73,4 %) uvedlo, že testové otázky a úlohy byly přiměřené. Zbýlý počet respondentů (26,7 %) uvedl, že otázky a úlohy byly náročnější.

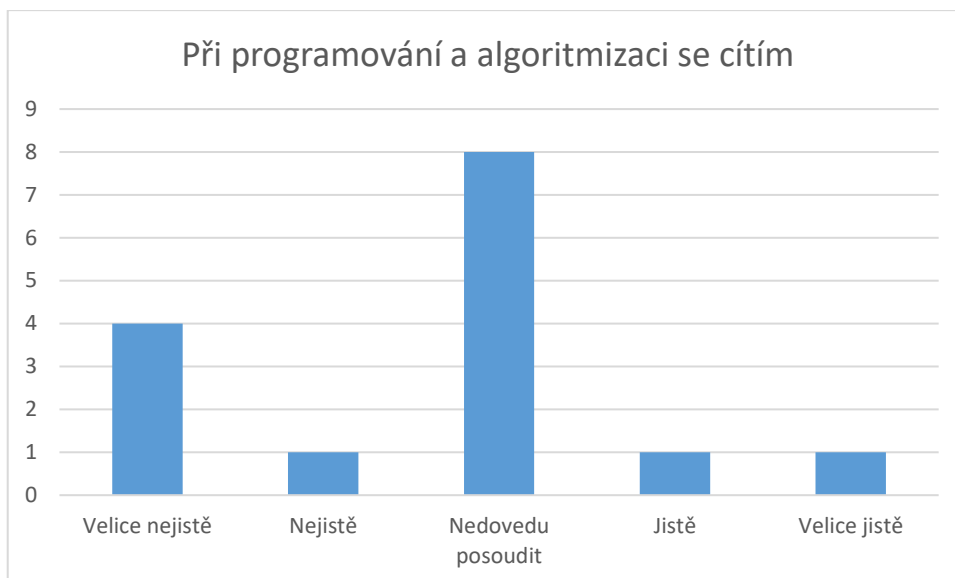


Graf 20: (Vstupní postoje) Obtížnost testové části

Podle postojů studentů byla testovací část vhodně připravena. Otázky, které byly do testování zařazeny, se studentům nejevily ani jako velice jednoduché (až triviální), ani příliš obtížné (až téměř neřešitelné).

5.6.2 POSTOJE PROGRAMOVÁNÍ A ALGORITMIZACE

Nejčastější odpověď (53,4 %) u druhé otázky, která testuje postoje studentů je, že nedovedou posoudit, jak se při programování nebo algoritmizaci cítí. Na ostatní otázky (nejistě, jistě a velice jistě) v průzkumu kladně odpověděl vždy jeden respondent (6,7 %).

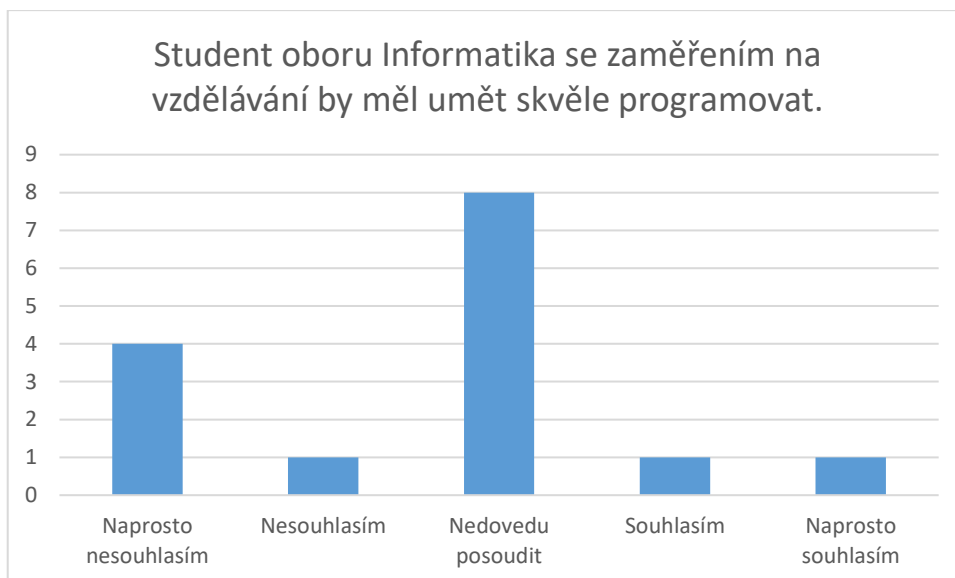


Graf 21: (Vstupní postoje) Postoje při programování a algoritmizaci

Vysoká hodnota u odpovědi „nedovedu posoudit“ je do značné míry ovlivněna otázkou z následující části dotazníku (informace o studentovi Graf 11), ze kterého je názorně vidět, že 46,7 % studentů na SŠ programování ani algoritmizaci neměli. Z toho důvodu je možné, že se s algoritmizací a základy programování v kurzu KVD/ALGV setkali poprvé.

5.6.3 STUDENT OBORU INFORMATIKA SE ZAMĚŘENÍM NA VZDĚLÁVÁNÍ BY MĚL SKVĚLE PROGRAMOVAT

Z největší části (53,4 %) studenti opět odpovídali číslem 3 na lineární stupnici (nedovedu posoudit). S výrokem „naprosto nesouhlasím“ a „nesouhlasím“ se ztotožnila jedna třetina dotazovaných (33,4 %). Možnost „souhlasím“ a „naprosto souhlasím“ označili dva respondenti (13,3 %).

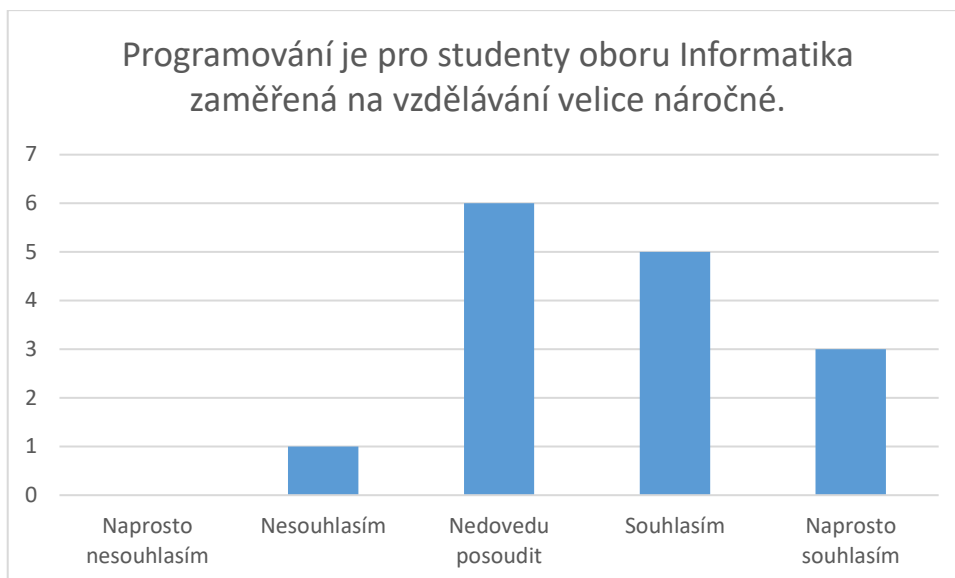


Graf 22: (Vstupní postoje) Studenti oboru informatika se zaměřením na vzdělávání by měli umět skvěle programovat

Větší část studentů se přiklání k tvrzení, že pro studenty daného oboru není nutná programátorská příprava na vyšší úrovni.

5.6.4 JE PROGRAMOVÁNÍ PRO STUDENTY NÁROČNÉ?

U čtvrté otázky určené pro zjištění studentských postojů se studenti spíše shodují na tom, že programování je velice náročné (souhlasím 33,4 %, naprosto souhlasím 20 %). S tvrzením se neztotožňuje pouze jeden respondent (6,7 %). Nerozhodných respondentů je 40 %.

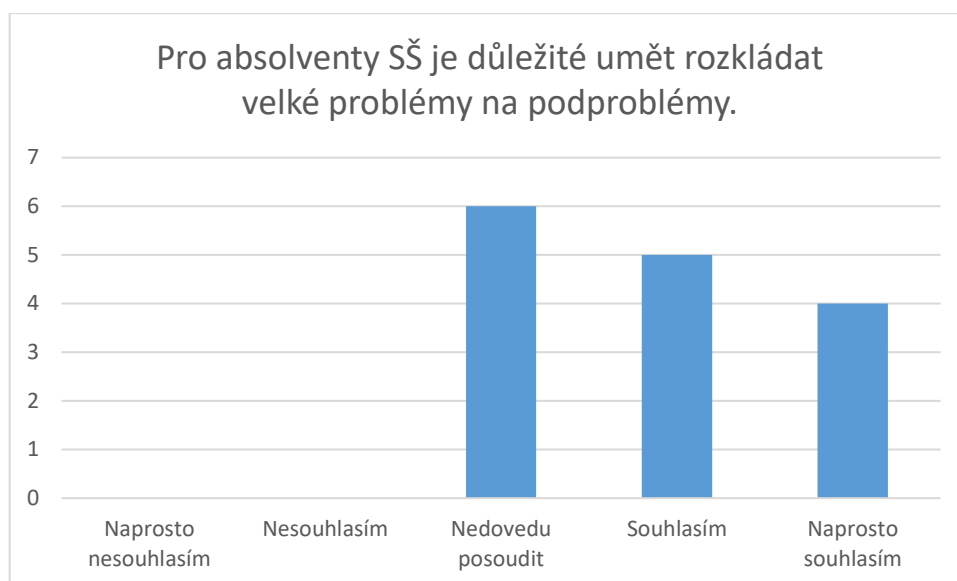


Graf 23: (Vstupní postoje) Je programování pro studenty náročné?

Studenti předmětu KVD/ALGV se přiklánějí k tomu, že je programování velice náročné. Největší problémy při programování shledávají ve složitosti syntaxe, hledání chyb již v hotovém programu, vymyšlení algoritmu a rozložení složitějšího algoritmu na menší části.

5.6.5 ROZKLÁDÁNÍ VELKÝCH PROBLÉMŮ NA PODPROBLÉMY

Respondenti považují za velice důležité umění rozkládat velké problémy na menší podproblémy u absolventů středních škol (možnost „souhlasím“ označilo 33,4 % respondentů a „naprosto souhlasím“ 26,7 % respondentů). 40 % studentů se nedovedlo rozhodnout ani pro jednu variantu (Graf 16: Pro absolventy SŠ je důležité umět rozkládat velké problémy na podproblémy.).

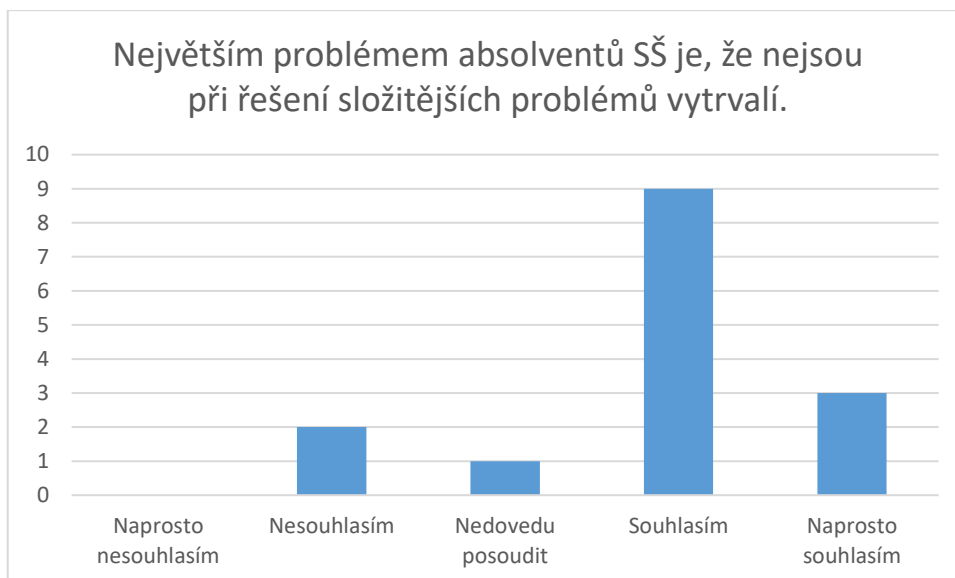


Graf 24: (Vstupní postoje) Rozkládání velkých problémů na podproblémy

Z odpovědí mimo jiné vyplývá, že studenti hlásící se na VŠ by tuto schopnost již měli ovládat.

5.6.6 VYTRVALOST STUDENTŮ

U této otázky se většina respondentů shoduje, že největším problémem středoškoláků je, že nejsou při řešení složitějších problémů vytrvalí („souhlasím“ 60 %, „naprosto souhlasím“ 20 %).

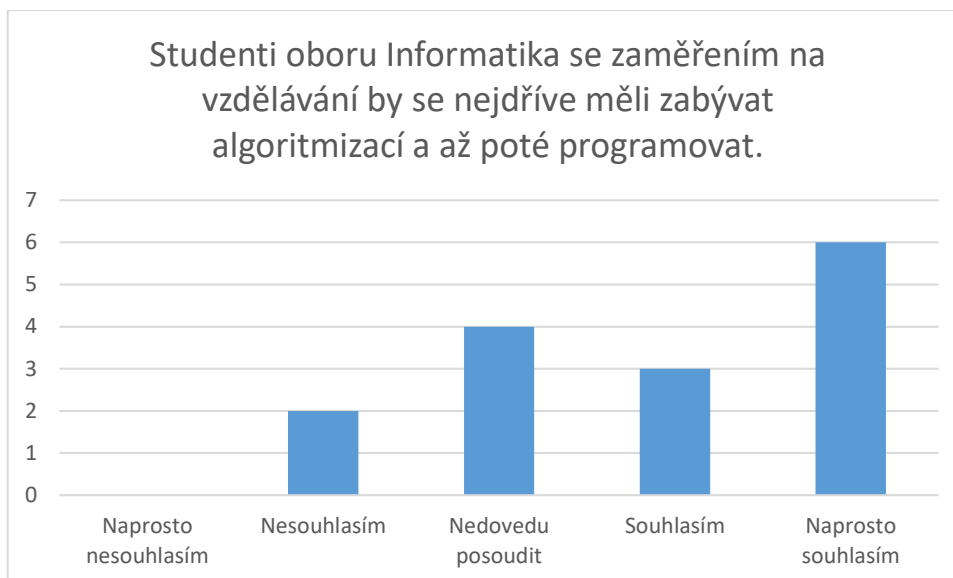


Graf 25: (Vstupní postoje) Vytrvalost studentů při řešení složitějších problémů

Studenti, kteří se zúčastnili vstupního testování, se ztotožnili s tím, že největším problémem po střední škole pro ně je vytrvat při řešení složitějších problémů. Tato otázka může být chápána jak ve smyslu učení se programovat, tak i např. při studiu.

5.6.7 NEJDŘÍVE ALGORITMIZACE A POTÉ PROGRAMOVÁNÍ?

Studenti se v této otázce vyjádřili, že je pro ně vhodné se nejdříve věnovat algoritmizaci a až poté programovat (možnost „naprosto souhlasím“ označilo 40 % respondentů a možnost „souhlasím“ 20 % respondentů). Pouze 13,3 % respondentů si myslí, že toto uspořádání není pro studenty vhodné.

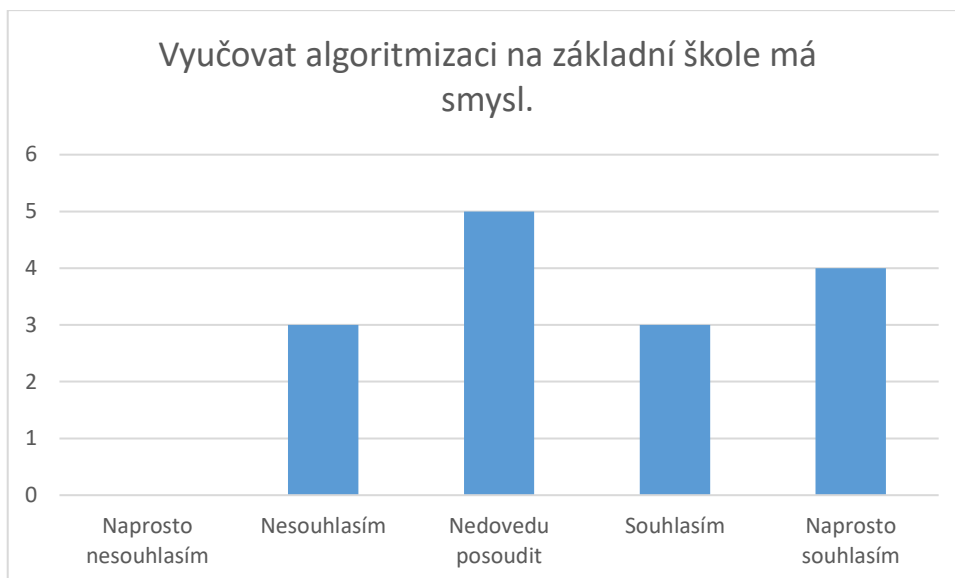


Graf 26: (Vstupní postoje) Nejdříve algoritmizace a poté programování?

Dle výsledků z dotazníku je velice vhodné zařazení předmětu KVD/ALGV do studijního plánu před předměty, ve kterých se vyučuje programování.

5.6.8 ALGORITMIZACE NA ZÁKLADNÍ ŠKOLE

U této otázky se postoje studentů poněkud rozcházejí. Pro vyučování algoritmizace na ZŠ je 20 % respondentů, kteří označili volbu „souhlasím“ a 26,7 % respondentů, kteří označili volbu „naprosto souhlasím“. S tvrzením nesouhlasí 20 % respondentů. Jedna třetina respondentů situaci neumí posoudit.

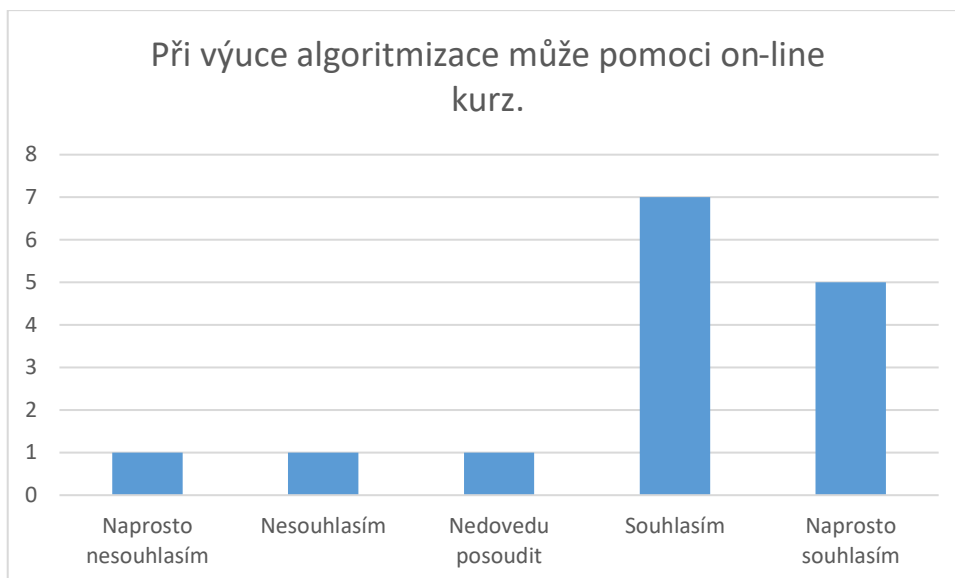


Graf 27: (Vstupní postoje) Algoritmizace na základní škole

Tato otázka si kladla za cíl zjistit postoje studentů k vyučování algoritmizace na základní škole. Pozitivním zjištěním je, že většina studentů se k vyučování algoritmizace na ZŠ přiklání a chtěli by v této oblasti žáky vzdělávat.

5.6.9 ALGORITMIZACE A ON-LINE KURZ

Většina studentů chápe on-line kurz jako pomoc při studiu (46,7 % s tvrzením souhlasí, 33,4 % naprosto souhlasí). On-line kurz jako zbytečnost při studiu chápe 13,3 % respondentů.



Graf 28: (Vstupní postoje) Algoritmizace a on-line kurz

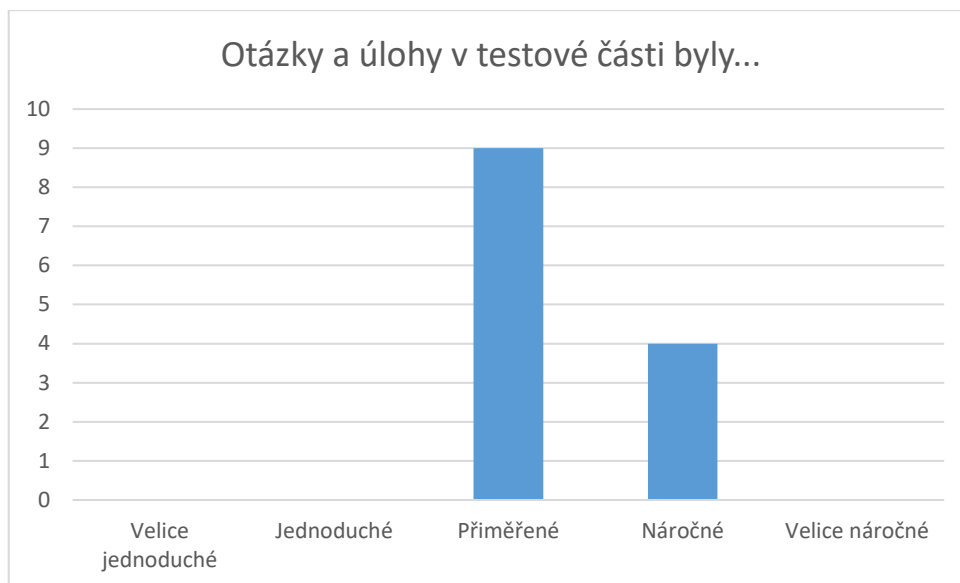
V této otázce, bylo důležité zjistit, jestli studenti chtějí při svém vzdělávání používat i jiné materiály, mezi které patří i on-line kurz.

5.7 POSTOJE PŘI VÝSTUPNÍM TESTOVÁNÍ

V následujících podkapitolách je provedena analýza jednotlivých výroků, které byly pro výstupní testování použity.

5.7.1 OBTÍŽNOST TESTOVÉ ČÁSTI

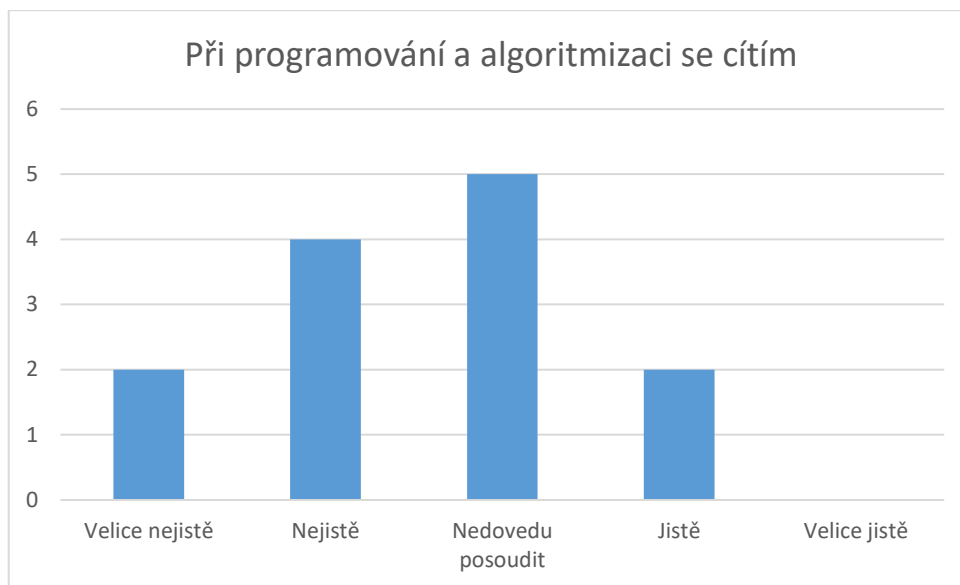
Většině respondentů přišly otázky a úlohy v testové části přiměřené (69 %). Zbýlých 31 % respondentů hodnotí otázky jako náročné.



Graf 29: (Výstupní postoje) Obtížnost testové části

5.7.2 POSTOJE PROGRAMOVÁNÍ A ALGORITMIZACE

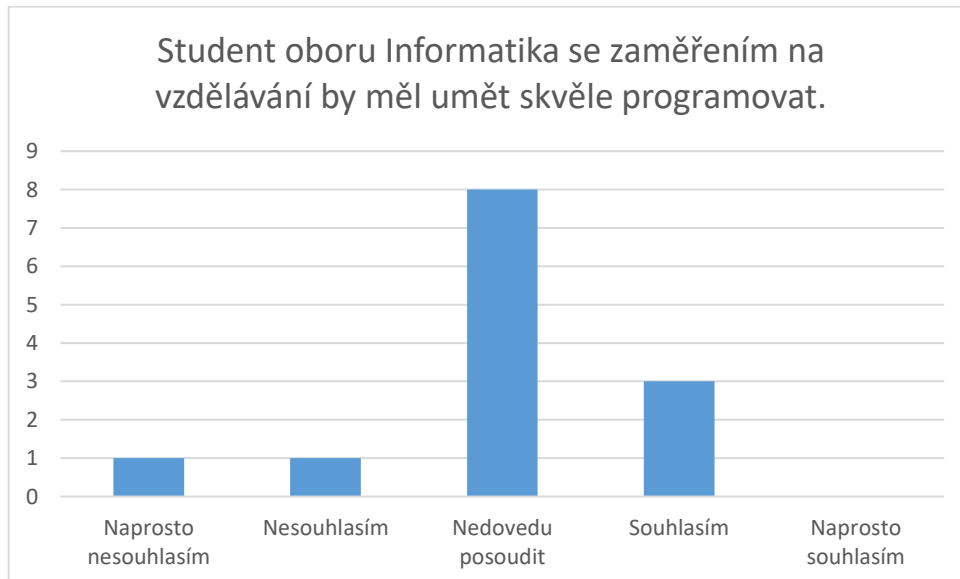
U této otázky nejvíce respondentů (38 %) volilo možnost 3 na lineární stupnici, tedy že stále nedovedou posoudit, zda se při algoritmizaci cítí spíše jistě nebo nejistě. Nejistě se i po absolvování on-line kurzu i předmětu KVD/ALGV cítí 31 % respondentů (velice nejistě 15 % respondentů). Jistě se cítí 15 % respondentů.



Graf 30 (Výstupní postoje) Postoje při programování a algoritmizaci

5.7.3 STUDENT OBORU INFORMATIKA SE ZAMĚŘENÍM NA VZDĚLÁVÁNÍ BY MĚL SKVĚLE PROGRAMOVAT

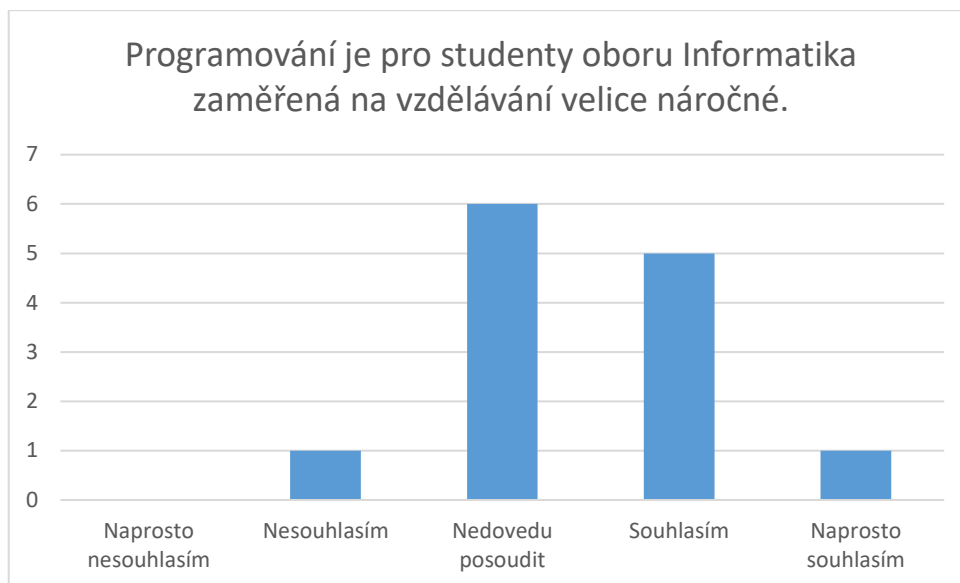
U této otázky nejčastěji odpovídali studenti možností „nedovedu posoudit“ (62 %). S tvrzením souhlasí 23 % respondentů a proti tvrzení 15 % respondentů (7,5 % nesouhlasí a 7,5 % naprosto nesouhlasí).



Graf 31: (Výstupní postoje) Studenti oboru informatika se zaměřením na vzdělávání by měli umět skvěle programovat

5.7.4 JE PROGRAMOVÁNÍ PRO STUDENTY NÁROČNÉ?

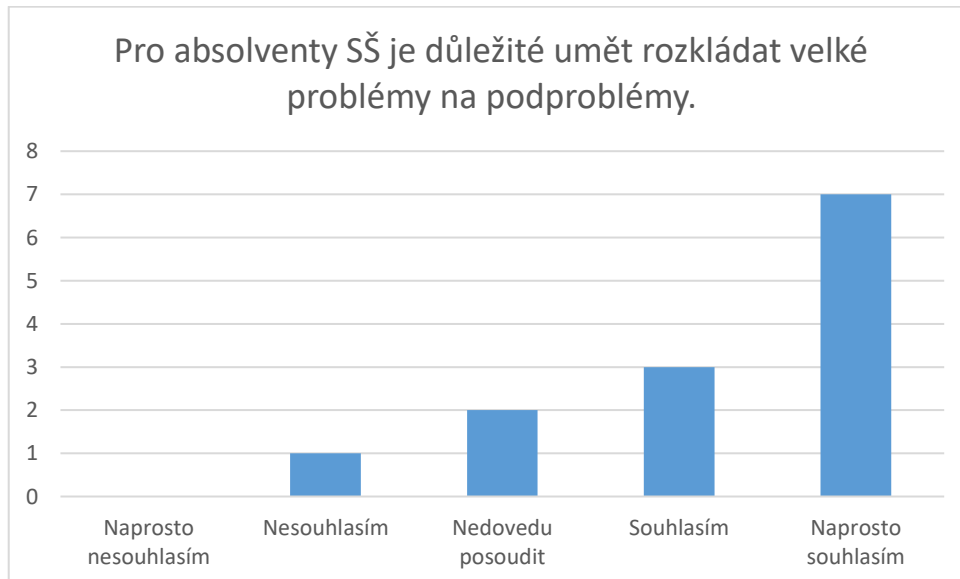
Respondentů, kteří výrok nedovedou posoudit je 46 %. S tvrzením souhlasí 46 % respondentů (8 % naprosto souhlasí). Pouze jeden respondent (8 %) s tvrzením nesouhlasí.



Graf 32: (Výstupní postoje) Je programování pro studenty náročná?

5.7.5 ROZKLÁDÁNÍ VELKÝCH PROBLÉMŮ NA PODPROBLÉMY

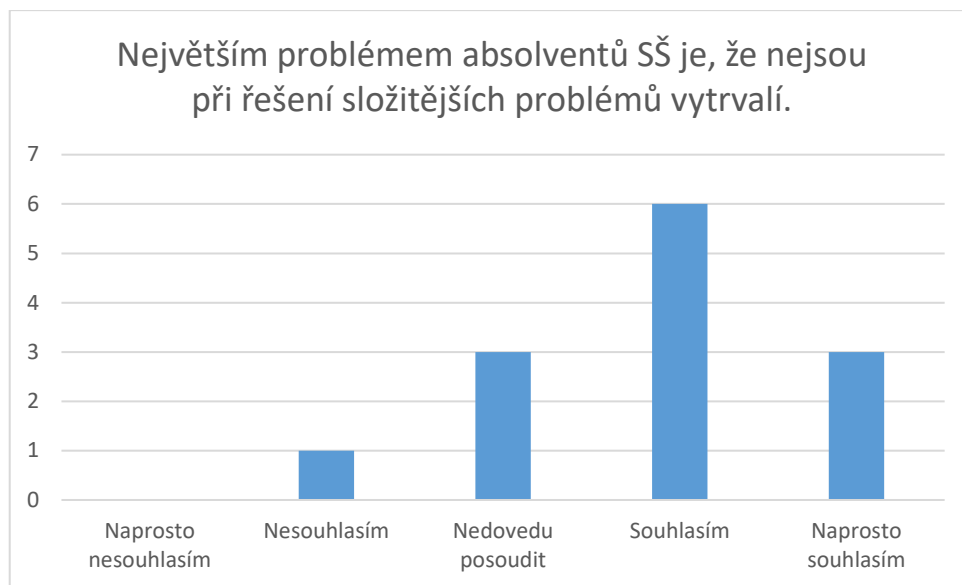
Větší část respondentů (77 %) se přiklání k tomu, že rozkládání velkých problémů na podproblémy je pro absolventy velice důležité (54 % Naprosto souhlasí, 23 % souhlasí). Pouze jeden respondent nesouhlasí.



Graf 33: (Výstupní postoje) Rozkládání velkých problémů na podproblémy

5.7.6 VYTRVALOST STUDENTŮ

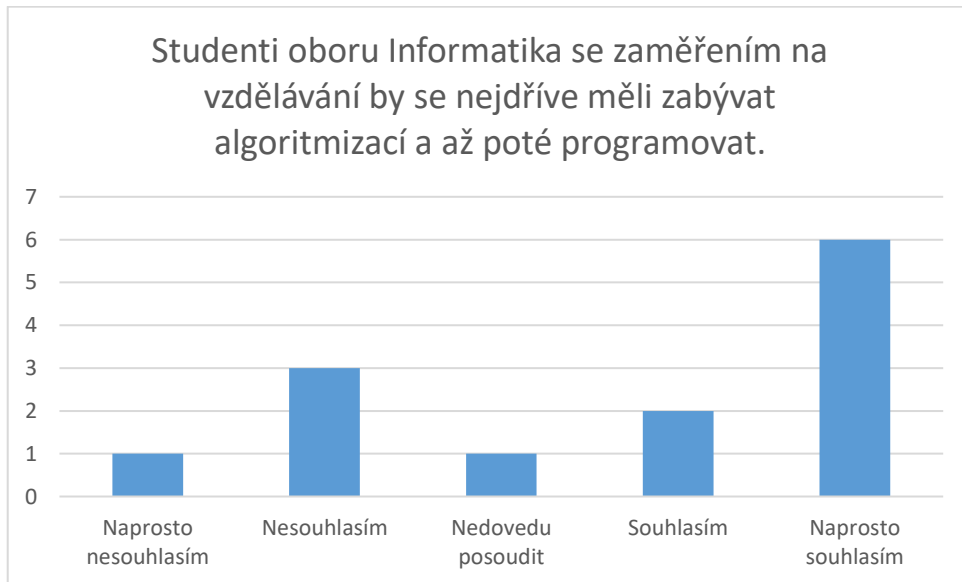
Nejvíce respondentů (46 %) označilo možnost „souhlasím“. Dále 23 % studentů uvádí, že s výrokem naprosto souhlasí, stejný počet studentů se pak nepřiklonil ani k jedné z nabízených variant. Pouze 1 respondent s tímto tvrzením nesouhlasí.



Graf 34: (Výstupní postoje) Vytrvalost studentů při řešení složitějších problémů

5.7.7 NEJDŘÍVE ALGORITMIZACE A POTÉ PROGRAMOVÁNÍ?

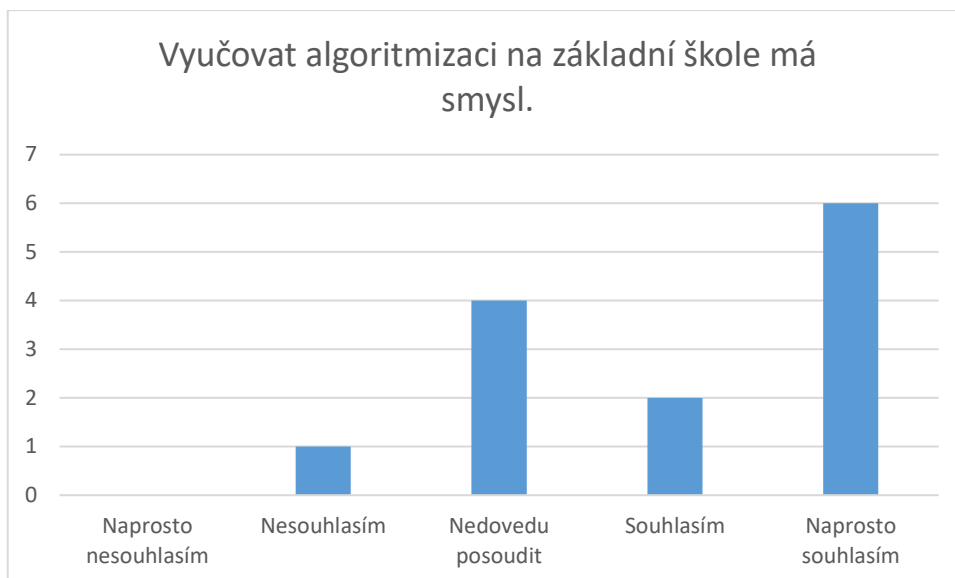
Nejvíce respondentů se s tímto tvrzením ztotožňuje (naprosto souhlasím 46 % a souhlasím 15 %). Na druhé straně studentů, kteří s tímto výrokiem nesouhlasí či naprosto nesouhlasí, se v šetření nachází 31 % (naprosto nesouhlasí 1 respondent).



Graf 35: (Výstupní postoje) Nejdříve algoritmizace a poté programování?

5.7.8 ALGORITMIZACE NA ZÁKLADNÍ ŠKOLE

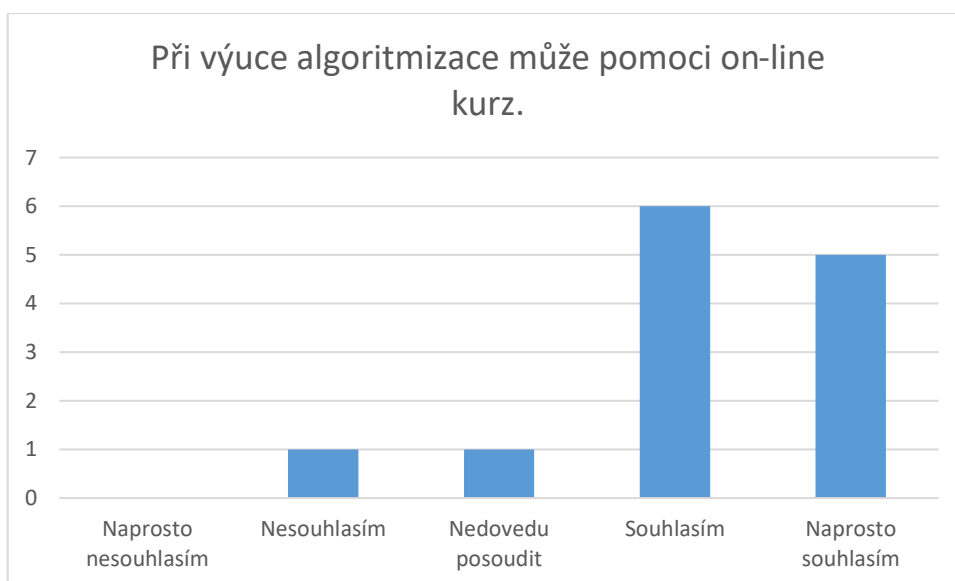
S názorem, že vyučovat algoritmizaci na základní škole má smysl souhlasí 62 % respondentů (46 % naprosto souhlasím a 15 % souhlasí). Odpověď „nedovedu posoudit“ označilo 31 % respondentů. Pouze 1 respondent s tvrzením nesouhlasí.



Graf 36: (Výstupní postoje) Algoritmizace na základní škole

5.7.9 ALGORITMIZACE A ON-LINE KURZ

Pouze jeden respondent s tvrzením nesouhlasí a jeden situaci nedovede posoudit. Zbytek respondentů s tvrzením souhlasí (46 % souhlasím, 38 % naprosto souhlasím).



Graf 37: (Výstupní postoje) Algoritmizace a on-line kurz

5.8 POROVNÁNÍ VSTUPNÍCH A VÝSTUPNÍCH POSTOJŮ STUDENTŮ

V této podkapitole je uvedeno porovnání změn v postojové rovině studentů před a po absolvování kurzu.

Mezi vstupními a výstupními postoji studentů nejsou tak markantní rozdíly, většinou se jedná pouze o jemné nuance. Proto v tomto porovnání budou uvedeny pouze výroky, u kterých je prokazatelná změna.

Výrok: Pro absolventy SŠ je důležité umět rozkládat velké problémy na podproblémy.

Tabulka 10: Porovnání postojů 1

	Naprosto nesouhlasím	Nesouhlasím	Nedovedu posoudit	Souhlasím	Naprosto souhlasím
Vstupní postoje	0 %	0 %	40 %	33 %	27 %
Výstupní postoje	0 %	8 %	15 %	23 %	54 %

Z tabulky je patrné, že po absolvování kurzu s výrokem naprosto souhlasí o polovinu více respondentů. Je tedy možné, že se studenti během svého studia s většími problémy setkali a vztahují tak tento výrok na sebe.

Výrok: Vyučovat algoritmizaci na základní škole má smysl.

Tabulka 11: Porovnání postojů 2

	Naprosto nesouhlasím	Nesouhlasím	Nedovedu posoudit	Souhlasím	Naprosto souhlasím
Vstupní postoje	0 %	20 %	33 %	20 %	27 %
Výstupní postoje	0 %	8 %	31 %	15 %	46 %

Po absolvování kurzu KVD/ALGV se počet respondentů, kteří jsou naprosto přesvědčeni o tom, že vyučovat algoritmizaci na základní škole, má smysl vzrostl o 19 %, zároveň o 12 % klesl počet respondentů, kteří s tímto výrokiem nesouhlasí. Tento posun v postojích studentů pravděpodobně přinesl obsah e-kurzu a předmětu ALGV.

U ostatních výroků jsou výsledky téměř shodné.

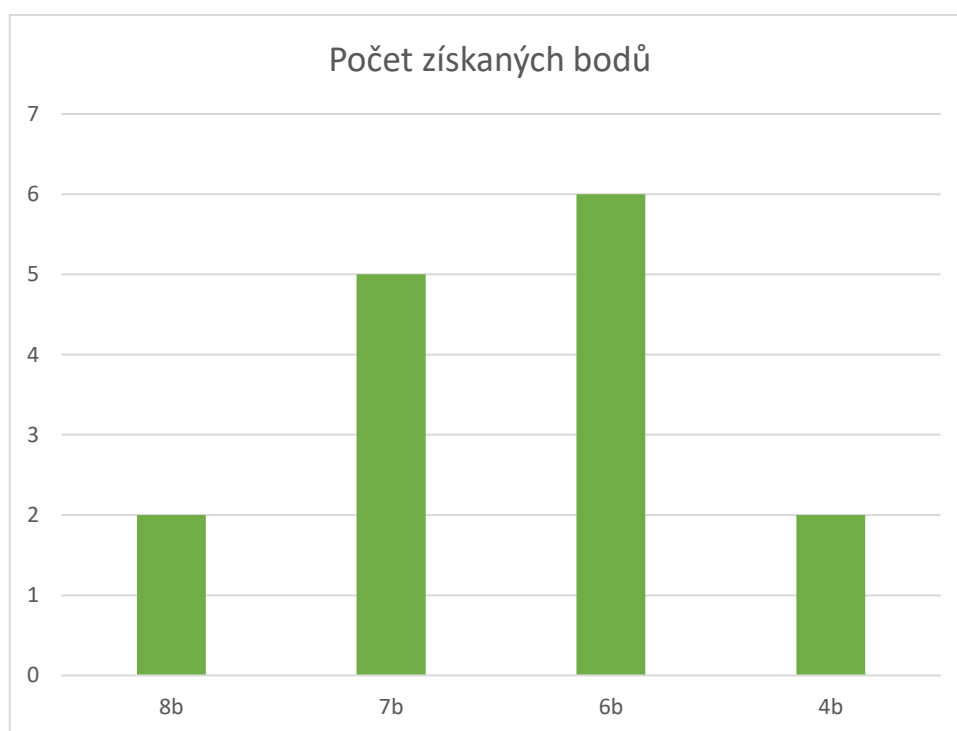
5.9 VSTUPNÍ ZNALOSTI A DOVEDNOSTI

Cílem této kapitoly je představit otázky, které byly použity při vstupním testování studentů a tyto otázky vyhodnotit.

U každé otázky je vypsáno zadání, ilustrační obrázek (pokud je dostupný), typ otázky, správná odpověď, popis řešení úlohy, požadavky na předchozí znalosti studentů a vyhodnocení úspěšnosti studentů.

5.9.1 CELKOVÉ VYHODNOCENÍ TESTU

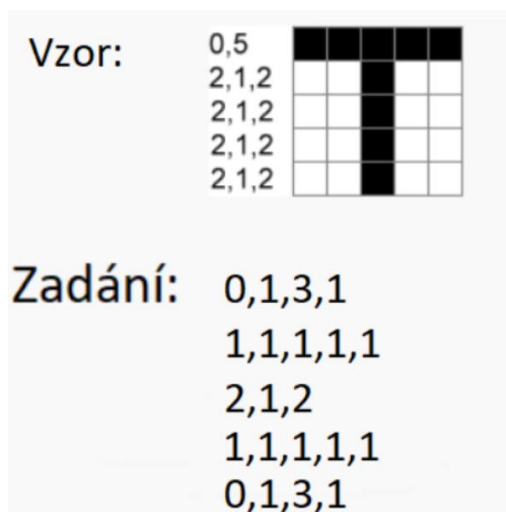
Zastoupení respondentů, kteří v testové části dosáhli minima (4 body) bylo 13 %. Stejný procentuální vzorek respondentů dosáhl maxima (8 bodů). Sedm bodů získalo 33 % respondentů. Šest bodů pak získalo 40 % respondentů. Bodový průměr byl 6,33 bodu.



Graf 38: (Vstupní znalosti) Počet získaných bodů

5.9.2 OTÁZKA 1 (UPRAVENA Z IBOBR.CZ)

Zadání úlohy: Podle vzoru zjistěte, které písmeno je popsáno následujícími čísly (zadání).



Obrázek 15: Zadání úlohy 1

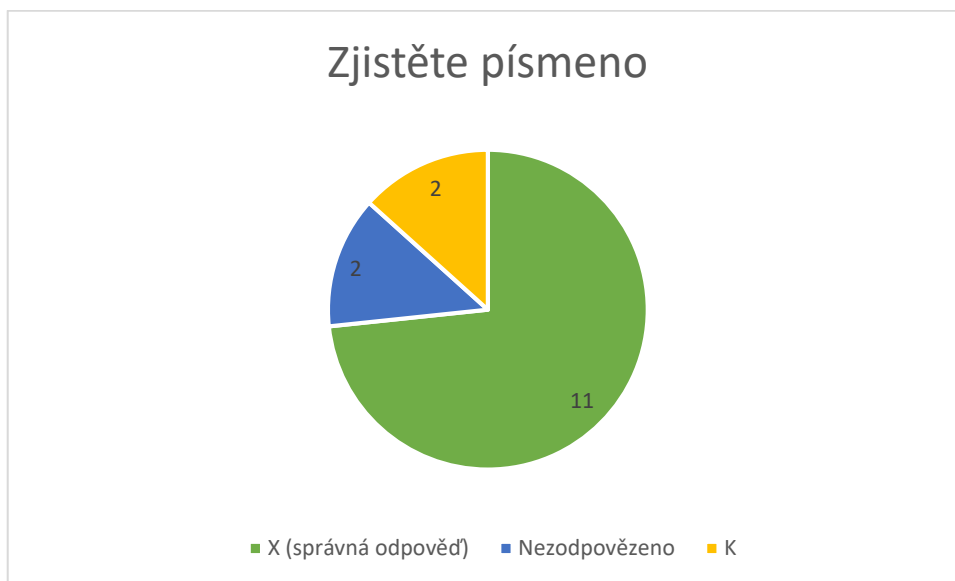
Typ otázky: otázka se stručnou odpovědí

Správná odpověď: x, X

Popis úlohy: V této úloze je důležité, aby si studenti (podle vzoru) uvědomili vztah černých a bílých bodů, pomocí číselného vyjádření. V tomto případě vždy první číslo udává počet bílých bodů, následované počtem černých bodů.

Předchozí znalosti studentů: Ne, studenti mohou správně odpovědět bez použití znalostí z předchozího studia.

Úspěšnost studentů:



Graf 39: (Vstupní znalosti) Otázka 1

S touto úlohou studenti neměli větší problémy, 73,4 % studentů odpovědělo správně. Pouze ve dvou (13,3 %) případech došlo k záměně písmena X za K. Dva respondenti (13,3 %) na otázku neodpověděli.

5.9.3 OTÁZKA 2 (ČERPANÁ Z IBOBR.CZ)

Zadání úlohy: V buňce C2 je napsán vzorec. Jestliže zkopírujeme tuto buňku do buňky D3, jaké číslo bude v buňce D3?

	A	B	C	D
1	1	5		
2	3	3	69	
3	4	2		

Formula v buňce C2: =A1+2\$E\$37*

Obrázek 16: Zadání úlohy 2

Typ otázky: jedna odpověď

Možnosti: 69; 71; 73; Nelze určit

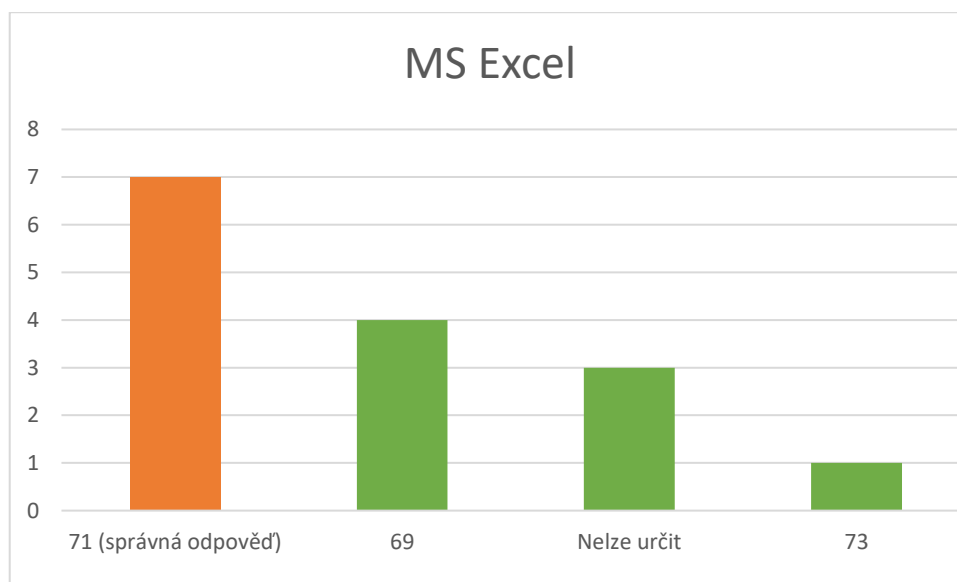
Správná odpověď: 71

Popis úlohy: Při řešení této úlohy potřebují studenti ovládat základy práce v kancelářském balíku MS Office, v aplikaci Excel. Po prostudování vzorce a vyhodnocení absolutních a relativních odkazů je patrné, že po zkopírování buňky se změní A1 na B2 (relativní odkaz)

a odkaz zadaný absolutně zůstane po zkopírování buňky stejný. Tedy jediná změna proběhne v přičítání buňky B2, která je v našem případě 3.

Předchozí znalosti studentů: Ano, nutná znalost absolutních a relativních odkazů v aplikaci MS Excel.

Úspěšnost studentů:



Graf 40: (Vstupní znalosti) Otázka 2

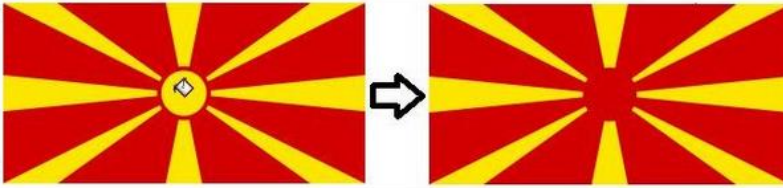
Správnou odpověď u této otázky zvolilo 46,7 % respondentů. Problémem, který při řešení této úlohy mohl nastat, byla neznalost absolutního a relativního odkazování v prostředí MS Excel.

5.9.4 OTÁZKA 3 (ČERPANÁ Z IBOBR.CZ)

Zadání úlohy:

Nástroj *Plechovka* v bitmapovém grafickém editoru pracuje tak, že vybarví celou jednobarevnou plošku, do které klikneme (viz obrázky vpravo).

Petr má vlajku Filipín (dole) a chce ji přebarvit na vlajku České republiky. Může použít nástroj *Plechovka* a nástroj pro výběr barvy, žádný jiný.



Kolik nejméně použití nástroje *Plechovka* stačí k přebarvení?

Poznámka: Počítá se každé kliknutí do obrázku při zapnutém nástroji *Plechovka*.



Obrázek 17: Zadání úlohy 3

Typ otázky: jedna odpověď

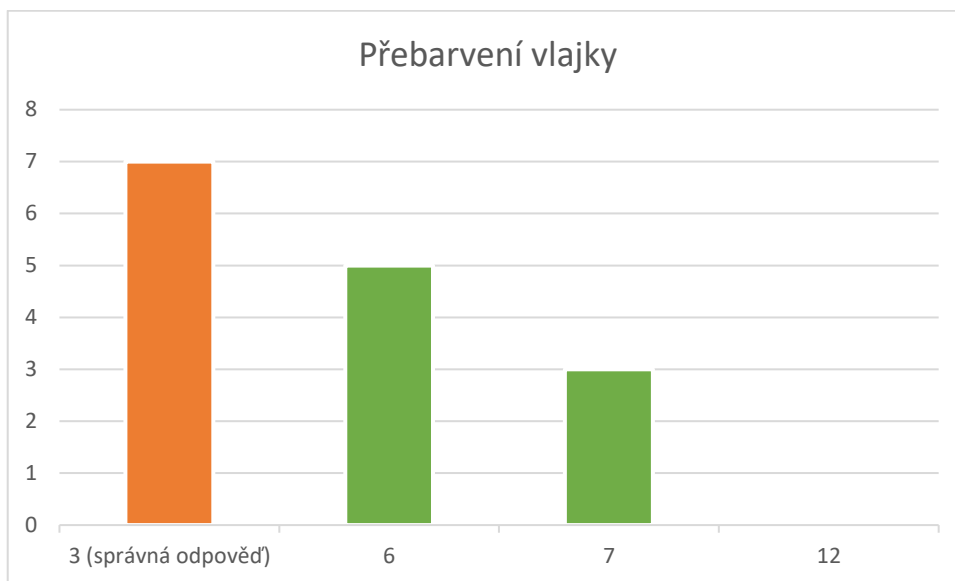
Možnosti: 3; 12; 7; 6

Správná odpověď: 3

Popis úlohy: V této úloze je nutné, aby si studenti uvědomili, jak jednoduchý nástroj *Plechovka* funguje, a optimalizovali své původní řešení. Student by si měl uvědomit, že všechny znaky vlajky Filipín na bílém podkladu jsou ve stejné barvě, proto tedy stačí vyplnit bílý podklad žlutě, následně přebarvit modrý podklad vlajky na bílou a následně vzniklý žlutý trojúhelník na modrou.

Předchozí znalosti studentů: Ne, studenti mohou správně odpovědět bez použití znalostí z předchozího studia. Všechny důležité informace studenti naleznou v zadání úlohy.

Úspěšnost studentů:

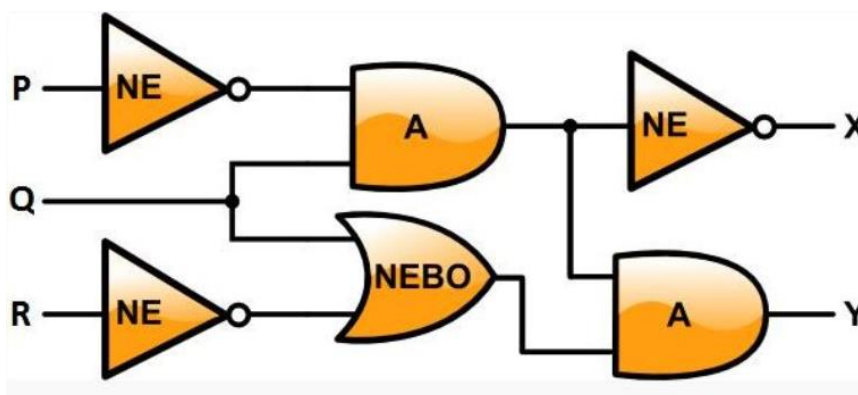


Graf 41: (Vstupní znalosti) Otázka 3

Na tuto otázku správně odpovědělo 7 respondentů (46,7 %). Jedna třetina respondentů vybralo odpověď 6, i přesto, že na 6 tahů není možné vlajku podle zadání přebarvit. Tři respondenti označili nutný počet tahů číslem 7, přebarvení vlajky je na takovýto počet tahů možný, jen není nejkratší.

5.9.5 OTÁZKA 4 (ČERPANÁ Z IBOBR.CZ)

Zadání úlohy: Jakých hodnot budou nabývat výstupy X a Y? P=0 (VYP) Q=1 (ZAP) a R=1 (ZAP), když platí: NE (NOT) - Když je vstup ZAP, výstup je VYP, Když je vstup VYP, výstup je ZAP; NEBO (OR) - Výstup je ZAP kromě případu, kdy oba vstupy jsou VYP; A (AND) - Výstup je ZAP, pouze pokud oba vstupy jsou ZAP.



Obrázek 18: Zadání úlohy 4

Typ otázky: jedna odpověď

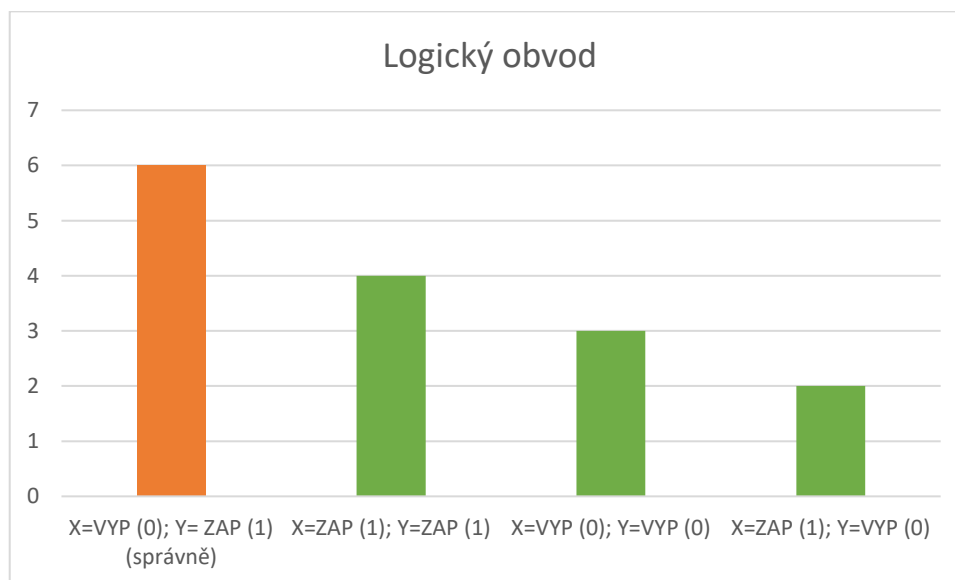
Možnosti: X=VYP (0); Y=VYP (0); X=ZAP (1); Y=ZAP (1); X=ZAP (1); Y=VYP (0); X=VYP (0); Y=ZAP (1)

Správná odpověď: X=VYP (0); Y= ZAP (1)

Popis úlohy: V této úloze byl zadán jednoduchý logický obvod, včetně vysvětlení, jak jednotlivé logické členy obvodu pracují. Studentům bez předchozí znalosti logických obvodů stačilo pouze sledovat návod, podle kterého by měli být schopni zjistit výstupy X a Y.

Předchozí znalosti studentů: Ne, studenti mohou správně odpovědět bez použití znalostí z předchozího studia. Všechny důležité informace studenti naleznou v zadání úlohy.

Úspěšnost studentů:

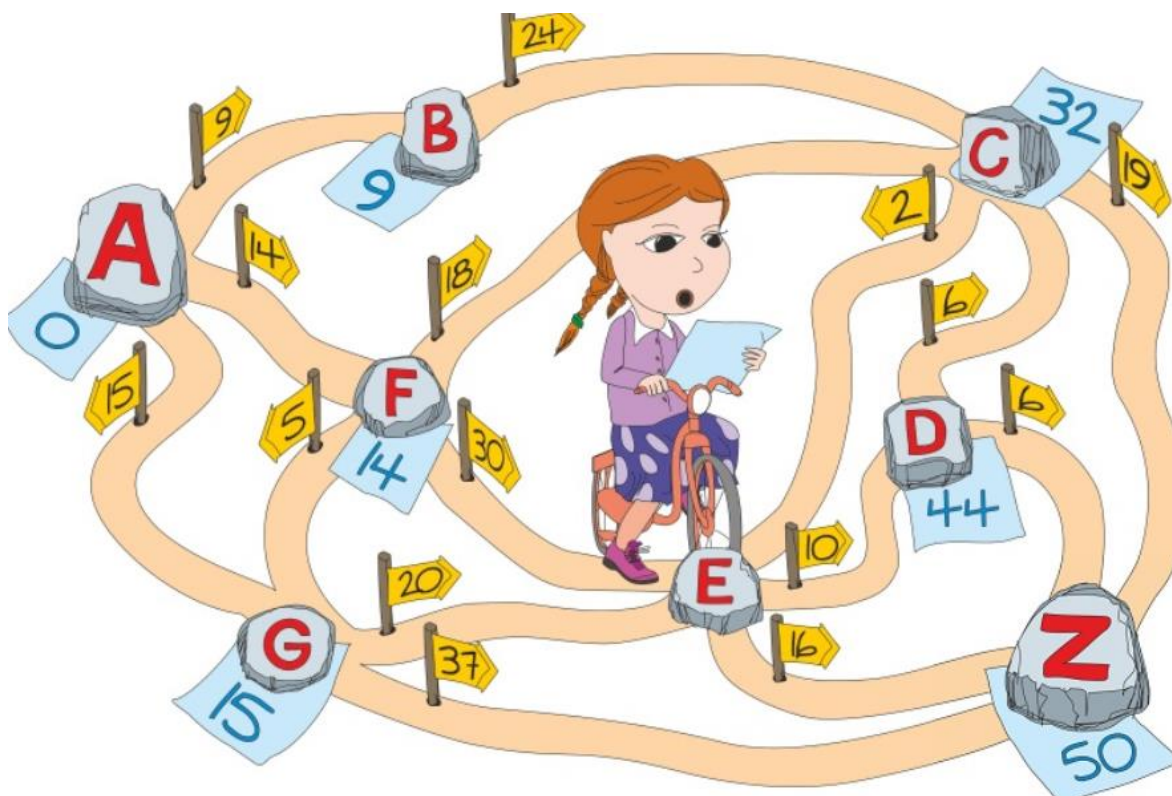


Graf 42: (Vstupní znalosti) Otázka 4

Tato otázka i přes nápovědu v zadání byla pro studenty oříškem. Správnou odpověď u čtvrté otázky vybralo pouze 40 % studentů. Tato otázka má jednu z nejhorších úspěšností v celém testu.

5.9.6 OTÁZKA 5 (ČERPANÁ Z IBOBR.CZ)

Zadání úlohy: Jaké číslo bude u křižovatky označené písmenem E? (Všechny cyklostezky jsou pouze jednosměrné.)



Obrázek 19: Zadání úlohy 5

Typ otázky: jedna odpověď

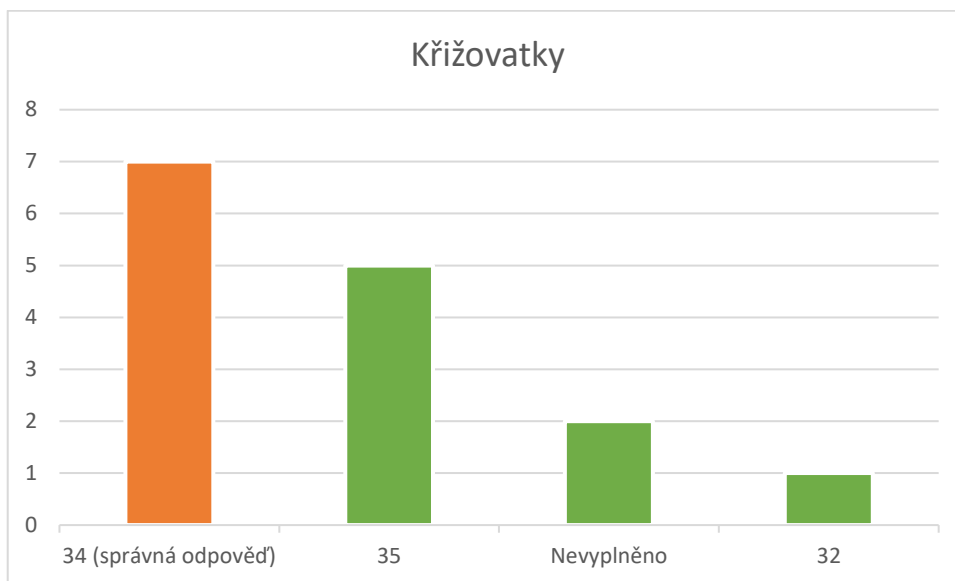
Možnosti: 32; 34; 35; 44

Správná odpověď: 34

Popis úlohy: Studenti potřebují odpozorovat pravidla, podle kterých se čísla na papírech u každé křižovatky chovají (nejkratší možná cesta). Na křižovatku E je možno dojet z křižovatek C, F nebo G. Nejkratší vzdálenost je tedy $32 + 2$, 34 je tedy správná odpověď.

Předchozí znalosti studentů: Ne, studenti mohou správně odpovědět bez použití znalostí z předchozího studia. Všechny důležité informace studenti naleznou v zadání úlohy.

Úspěšnost studentů:

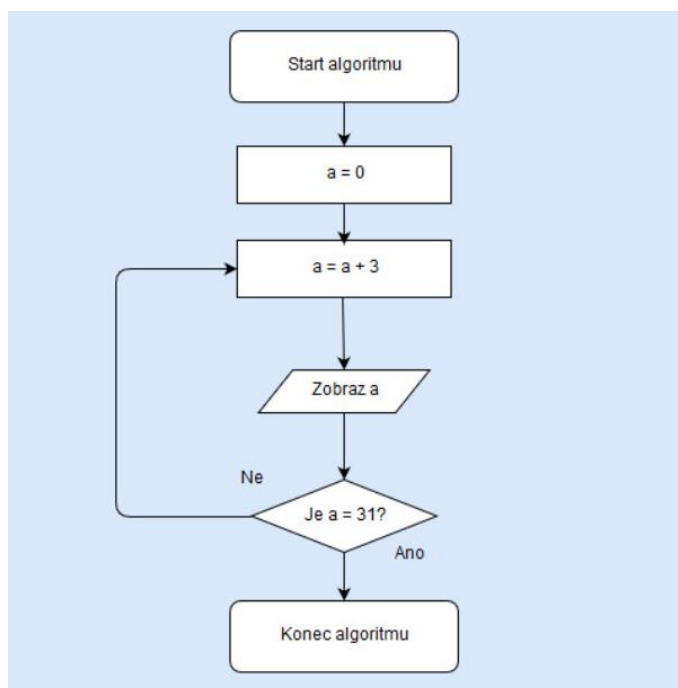


Graf 43: (Vstupní znalosti) Otázka 5

V této otázce bylo 46,7 % úspěšných studentů. Dva respondenti (13,3 %) na otázku vůbec neodpověděli.

5.9.7 OTÁZKA 6 (ZDROJ VLASTNÍ)

Zadání úlohy: Jedná se o algoritmus?



Obrázek 20: Zadání úlohy 6

Typ otázky: jedna odpověď

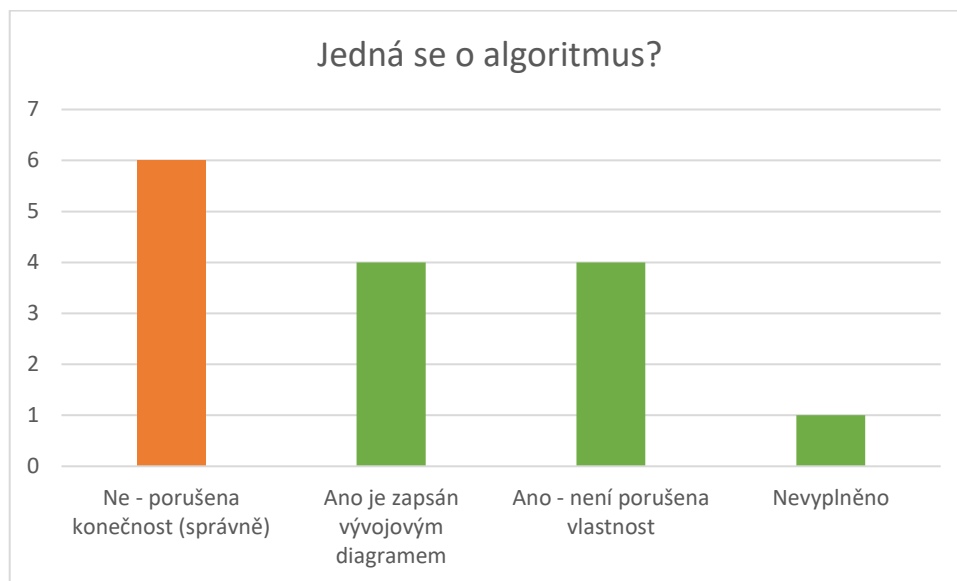
Možnosti: Ano. Je zapsán vývojovým diagramem; Ano. Je zapsán vývojovým diagramem a není zde porušena žádná základní vlastnost algoritmu; Ne. Je porušena základní vlastnost algoritmu (konečnost); Ne. Algoritmus může být reprezentován pouze slovním popisem.

Správná odpověď: Ne. Je porušena základní vlastnost algoritmu (konečnost).

Popis úlohy: V této úloze byl zadán jednoduchý vývojový diagram, u kterého měli studenti určit, zda byla porušena jedna ze základních vlastností algoritmu. V tomto případě byla porušena konečnost algoritmu (postupným přičítáním čísla 3 nikdy nemůže být výsledek 31).

Předchozí znalosti studentů: Ano, pro správné vyřešení úlohy je nutné, aby studenti znali základní vlastnosti algoritmů.

Úspěšnost studentů:



Graf 44: (Vstupní znalosti) Otázka 6

Správnou odpověď zvolilo 40 % respondentů. Dalších 27 % respondentů odpovědělo, že se o algoritmus jedná, protože je zapsán vývojovým diagramem. Dalších 27 % respondentů odpovědělo, že se jedná o algoritmus a že nebyla porušena žádná základní vlastnost. Jeden respondent na otázku neodpověděl.

5.9.8 OTÁZKA 7 (ZDROJ VLASTNÍ)

Zadání úlohy: Můžeme postup pro uvaření dobrého čaje považovat za algoritmus?

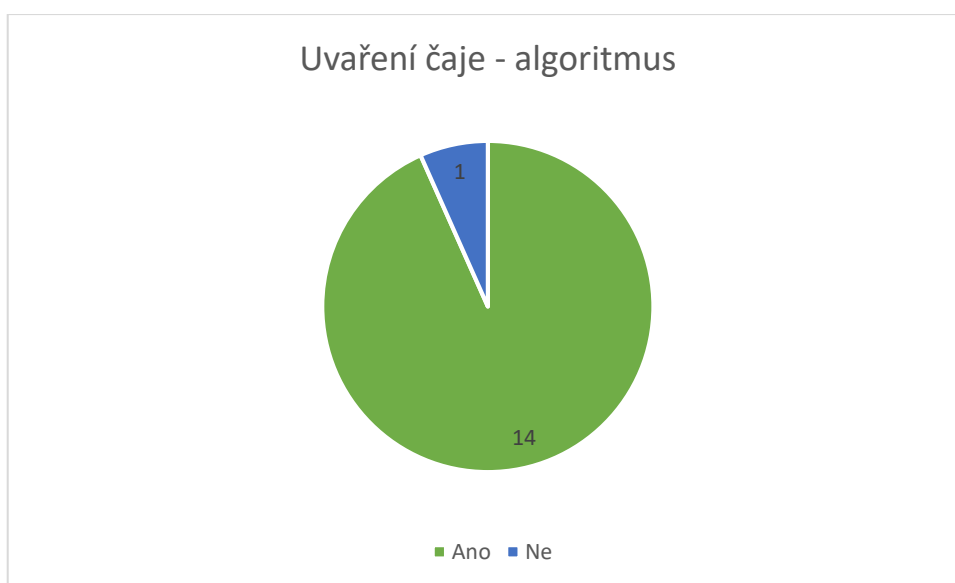
Typ otázky: ano/ne

Správná odpověď: ano

Popis úlohy: Tato průzkumná otázka je zařazena v testu proto, abychom pochopili, co studenti chápou pod pojmem algoritmus, zda si pod tím představí spíše pracovní postup, nebo pracovní postup zpracovávaný počítačem.

Předchozí znalosti studentů: Ano, pro správné vyřešení úlohy je nutné, aby studenti znali základní vlastnosti algoritmů.

Úspěšnost studentů:



Graf 45: (Vstupní znalosti) Otázka 7

Téměř všichni respondenti (93 %) považují postup pro uvaření čaje za algoritmus. Pouze jeden respondent označil možnost ne.

5.9.9 OTÁZKA 8 (ZDROJ VLASTNÍ)

Zadání úlohy: Jaké 3 druhy cyklů rozeznáváme?

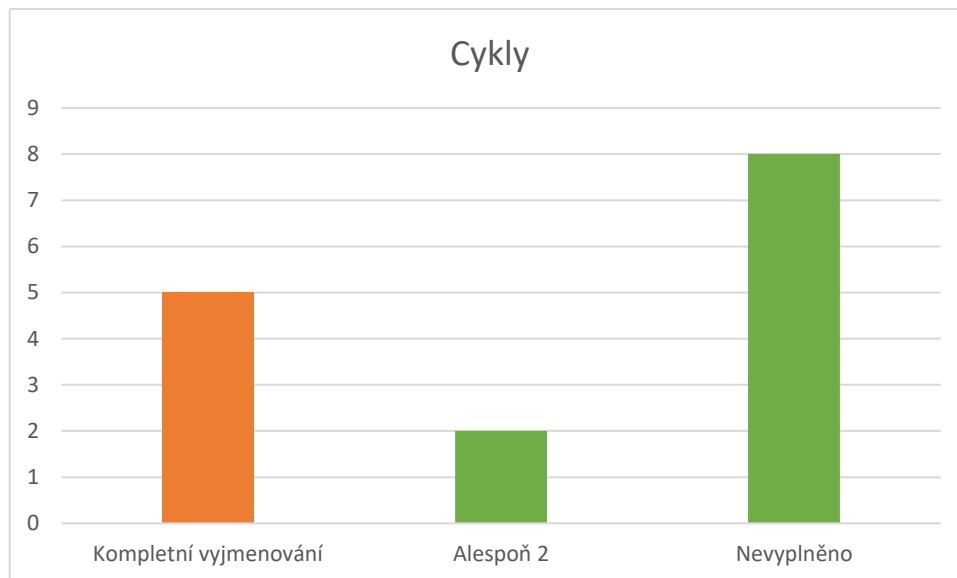
Typ otázky: stručná odpověď

Správná odpověď: Cyklus s podmínkou na začátku, cyklus s podmínkou na konci, cyklus s pevným počtem opakování.

Popis úlohy: Jedna ze znalostních odpovědí, kdy správnou odpověď budou znát převážně studenti, kteří se již ve svém životě setkali s programováním nebo algoritmizací.

Předchozí znalosti studentů: Ano, pro správné zodpovězení otázky je nezbytná znalost základů programování nebo algoritmizace.

Úspěšnost studentů:

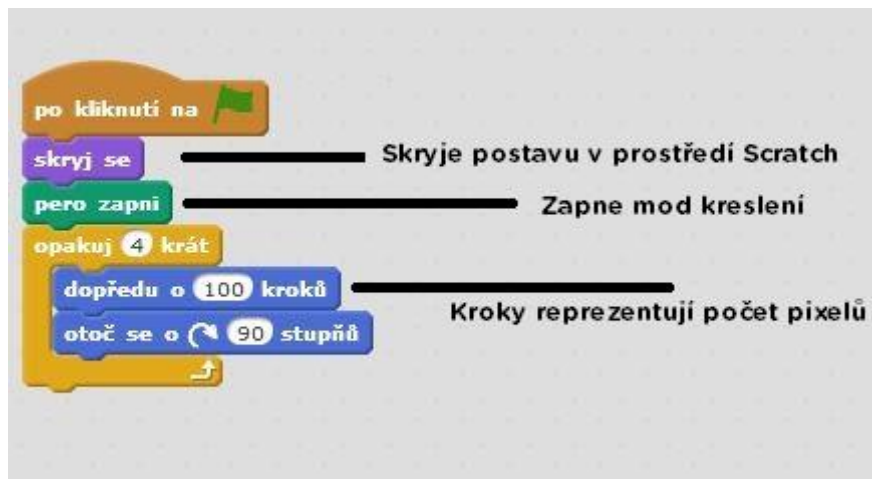


Graf 46: (Vstupní znalosti) Otázka 8

Všechny typy cyklů správně vyjmenovalo 33 % respondentů, 13 % respondentů správně uvedlo 2 typy cyklů. Zbýlých 53 % studentů neodpovědělo. Tato otázka byla pro studenty velice obtížná a to nejspíše z toho důvodu, že pro správné zodpovězení byly potřeba předchozí znalosti a také proto, že se jednalo o otázku s otevřenou odpovědí.

5.9.10 OTÁZKA 9 (ZDROJ VLASTNÍ)

Zadání úlohy: Co bude výstupem tohoto jednoduchého programu?



Obrázek 21: Zadání úlohy 9

Typ otázky: jedna odpověď

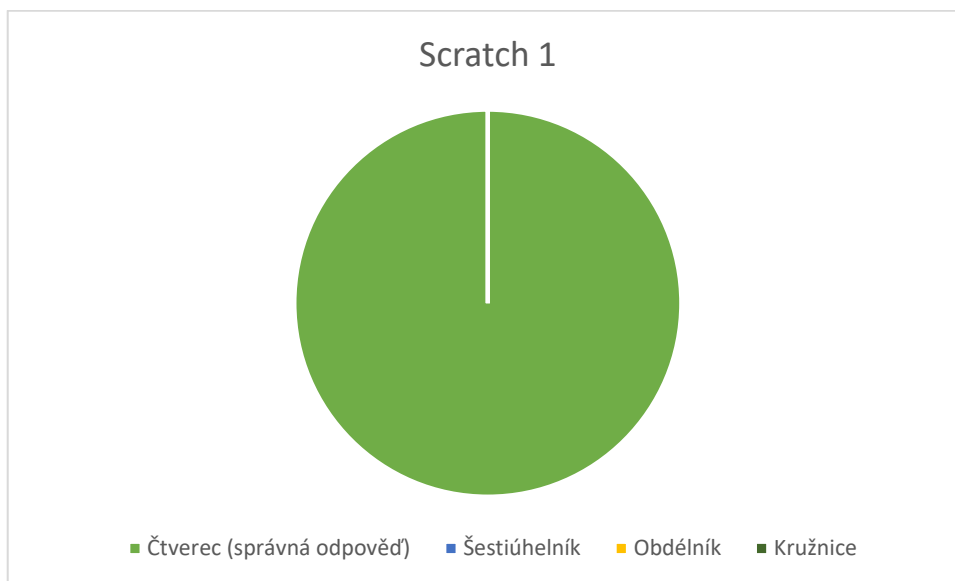
Možnosti: Kružnice; Obdélník; Čtverec; Šestiúhelník

Správná odpověď: Čtverec

Popis úlohy: Úvodní úloha, která zároveň studentům slouží jako seznámení s blokovým programováním v prostředí Scratch. V programu je nejdříve blok pro skrytí postavy, následně blok, který zapne mód kreslení. Na tyto bloky je napojený cyklus s pevným počtem opakování (4), ve kterém jsou umístěny dva bloky (dopředu o 100 kroků a otoč se vpravo o 90 stupňů).

Předchozí znalosti studentů: Ne, studenti mohou správně odpovědět bez použití znalostí z předchozího studia. Všechny důležité informace studenti naleznou v zadání úlohy.

Úspěšnost studentů:

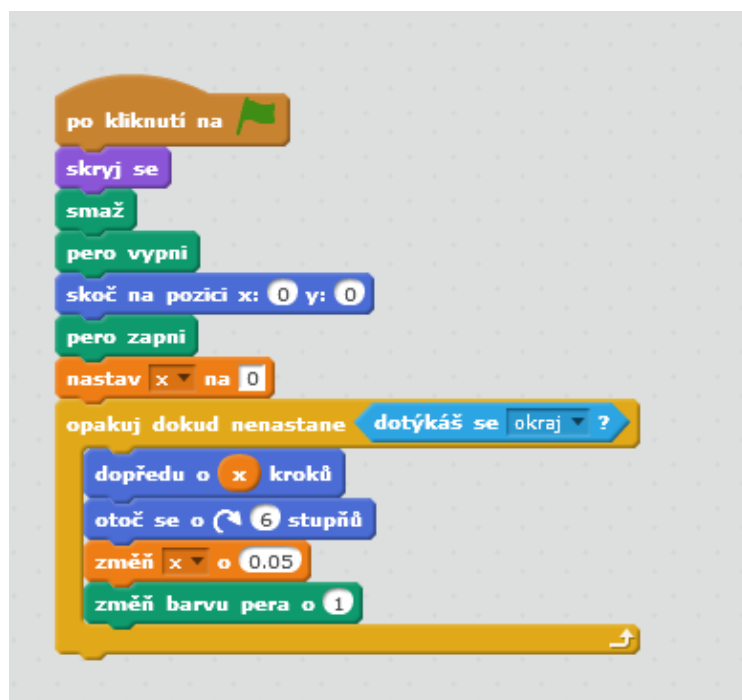


Graf 47: (Vstupní znalosti) Otázka 9

Jednoduchá otázka, která studentům nedělala žádné problémy. Všichni studenti odpověděli správně.

5.9.11 OTÁZKA 10 (ZDROJ VLASTNÍ)

Zadání úlohy: Co bude výstupem tohoto jednoduchého programu?



Obrázek 22: Zadání úlohy 10

Typ otázky: jedna odpověď

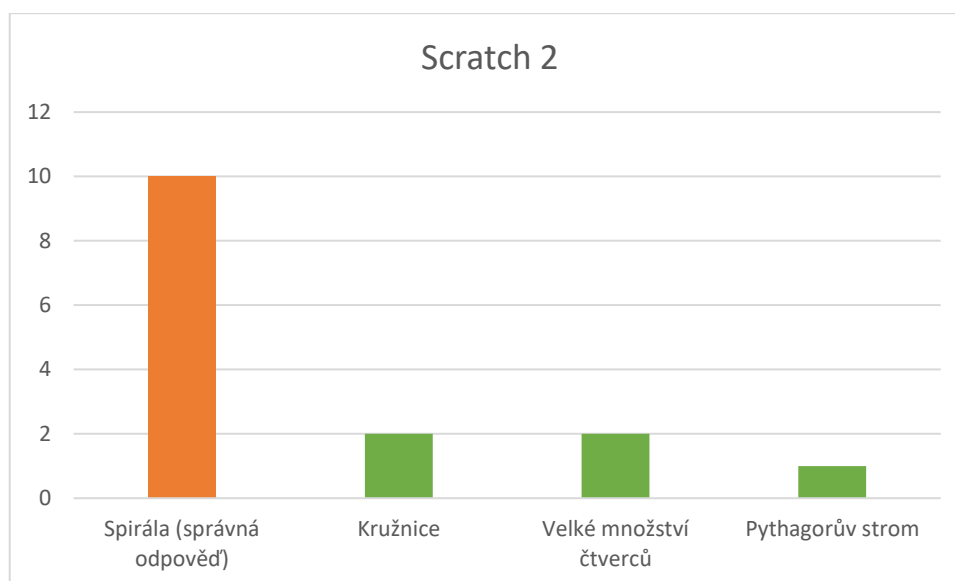
Možnosti: Pythagorův strom; Spirála; Kružnice; Velké množství čtverců, které dohromady vypadají jako diamant

Správná odpověď: Spirála

Popis úlohy: Složitější úloha z prostředí blokového programování v prostředí Scratch. V programu nejdříve nalezneme bloky pro inicializaci želví grafiky (skryj se, smaž, pero vypni, skoč na pozici x: 0 a y: 0, pero zapni, nastav x na 0) a cyklus pro vykreslení spirály (cyklus s podmínkou na začátku), který skončí, až když se Sprite dotkne okraje obrazovky. V této úloze je také zahrnuta práce s proměnnou x, která se v těle cyklu inkrementuje vždy o 0,05. V programu je také zahrnuta změna barvy pera (při každém průchodu se změní barva pera o 1).

Předchozí znalosti studentů: Ne, studenti mohou správně odpovědět bez použití znalostí z předchozího studia. Všechny důležité informace studenti naleznou v zadání úlohy.

Úspěšnost studentů:

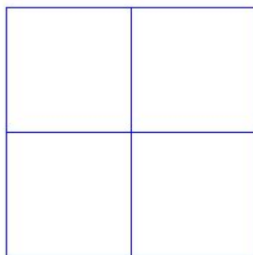


Graf 48: (Vstupní znalosti) Otázka 10

Ani s další otázkou z prostředí Scratch studenti neměli větší problémy. Dvě třetiny respondentů odpověděly na otázku správně. 13 % respondentů vybralo možnost kružnice a stejný počet respondentů vybral možnost „velké množství čtverců“. Pythagorův strom zvolil jeden respondent.

5.9.12 OTÁZKA 11 (ZDROJ VLASTNÍ)

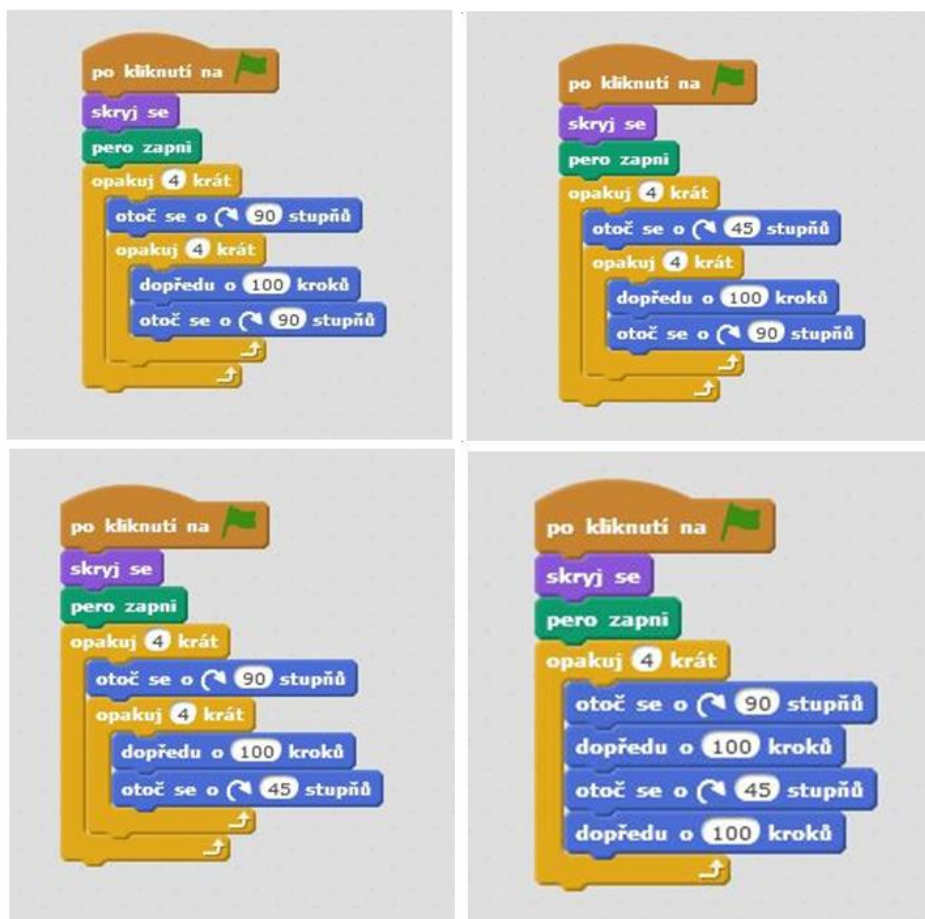
Zadání úlohy: Vyberte správné řešení, které vykreslí čtyři čtverce tak, jak je vidět na obrázku.



Obrázek 23: Zadání úlohy 11

Typ otázky: jedna odpověď

Možnosti:



Obrázek 24: Možnosti úlohy 11

Správná odpověď:

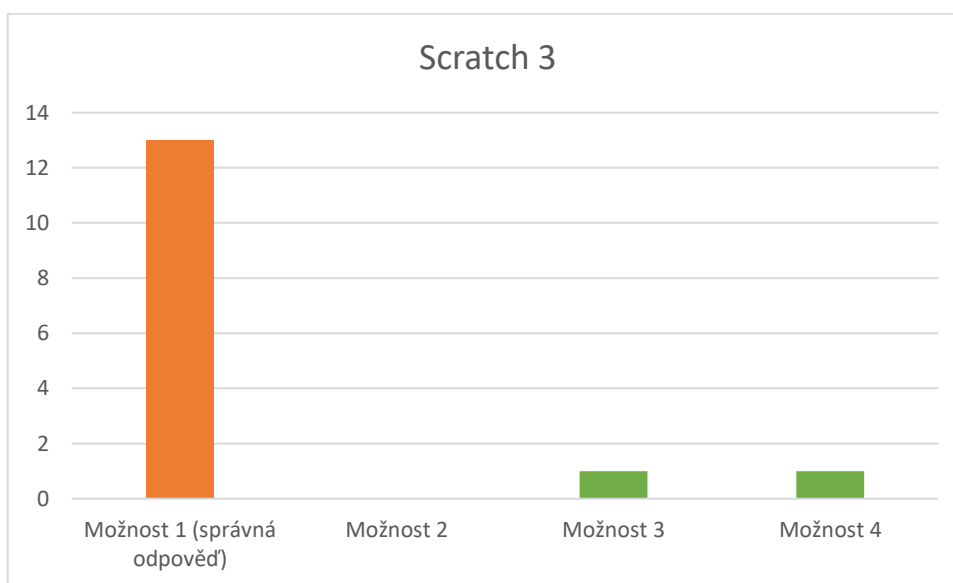


Obrázek 25: Řešení úlohy 11

Popis úlohy: Tato úloha je také zaměřena na blokové programování v prostředí Scratch. Zvláštnost této úlohy je v tom, že studenti rozhodují, který blok programu vykreslí obrazec, který je v zadání. Pro získání správného řešení si studenti musí uvědomit, jak pracují dva vnořené cykly.

Předchozí znalosti studentů: Ne, studenti mohou správně odpovědět bez použití znalostí z předchozího studia. Všechny důležité informace studenti naleznou v zadání úlohy.

Úspěšnost studentů:



Graf 49: (Vstupní znalosti) Otázka 11

Ani tato úloha pro studenty nebyla obtížná. Správně odpověděli téměř všichni respondenti.

5.9.13 OTÁZKA 12 (ZDROJ VLASTNÍ)

Zadání úlohy: Podívejte se na video a odpovězte na následující otázku. Určete, který typ algoritmu řazení je ve videu prezentován.

Typ otázky: jedna odpověď

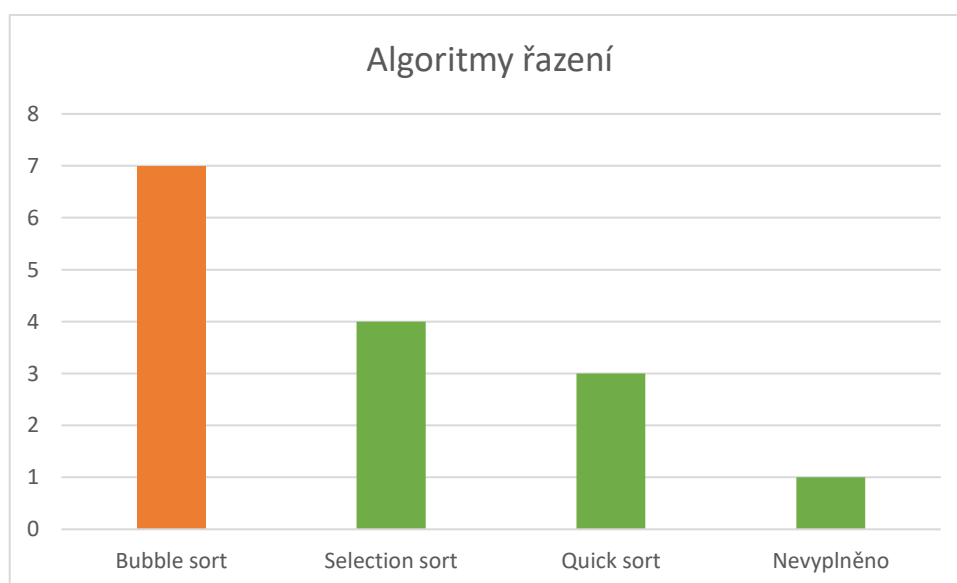
Možnosti: Quick sort; Bubble sort; Shell sort; Selection sort

Správná odpověď: Bubble sort

Popis úlohy: V této úloze mají studenti za úkol shlédnout video a poté podle svých předchozích znalostí posoudit, jaký algoritmus řazení je na videu prezentován.

Předchozí znalosti studentů: Ano, pro správné zodpovězení otázky je nezbytné orientovat se v problematice řadících algoritmů.

Úspěšnost studentů:



Graf 50: (Vstupní znalosti) Otázka 12

Na tuto otázku správně odpovědělo 54 % respondentů.

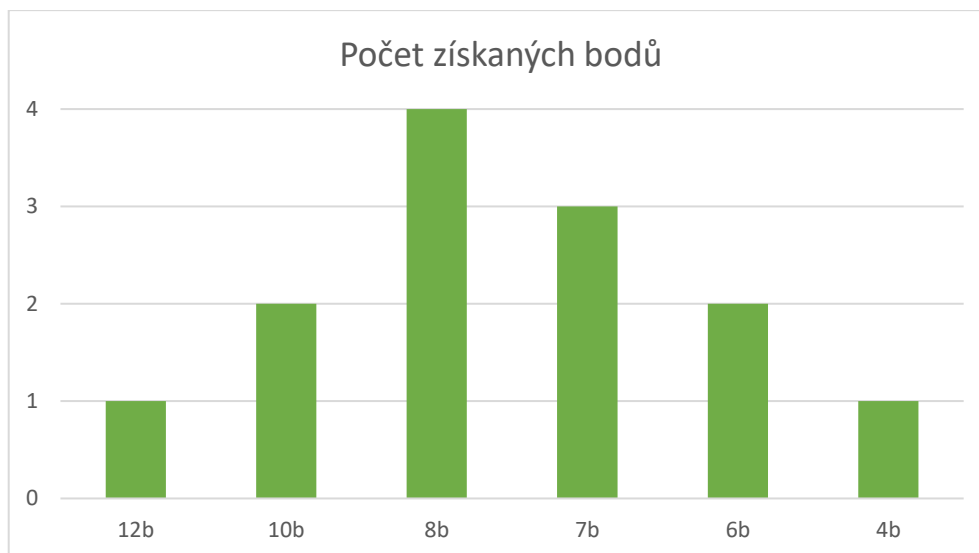
5.10 VÝSTUPNÍ ZNALOSTI A DOVEDNOSTI

Ve výstupním dotazníku bylo použito několik otázek ze vstupního dotazníku (jmenovitě: 3, 5, 8, 12) pro přímé porovnání. U několika otázek byly upraveny hodnoty (jmenovitě: 1, 2, 4, 6, 7, 9, 10, 11).

Cílem této kapitoly je představit otázky, které byly použity při výstupním testování studentů a tyto otázky vyhodnotit.

5.10.1 CELKOVÉ VYHODNOCENÍ TESTU

Zastoupení respondentů, které v testové části dosáhlo minima (4 body) bylo 8 %. Stejný procentuální vzorek respondentů dosáhl maxima (12 bodů). Deset bodů získalo 15 % respondentů. Osm bodů pak získalo 31 % respondentů. Na hranici sedmi bodů dosáhlo 23 % respondentů. Šest bodů získalo 15 % respondentů. Bodový průměr byl 7,77 bodu.



Graf 51: (Výstupní znalosti) Počet získaných bodů

5.10.2 OTÁZKA 1 (UPRAVENO Z IBOBR.CZ)

Zadání úlohy: Podle vzoru zjistěte, které písmeno je popsáno následujícími čísly (zadání).

Vzor:

0,5	■	■	■	■	■
2,1,2	■	■	■	■	■
2,1,2	■	■	■	■	■
2,1,2	■	■	■	■	■
2,1,2	■	■	■	■	■

Zadání:

0,1,2,1,1
 0,1,0,1,2
 0,2,3
 0,1,0,1,2
 0,1,2,1,1

Obrázek 26: Zadání úlohy 1 - Výstup

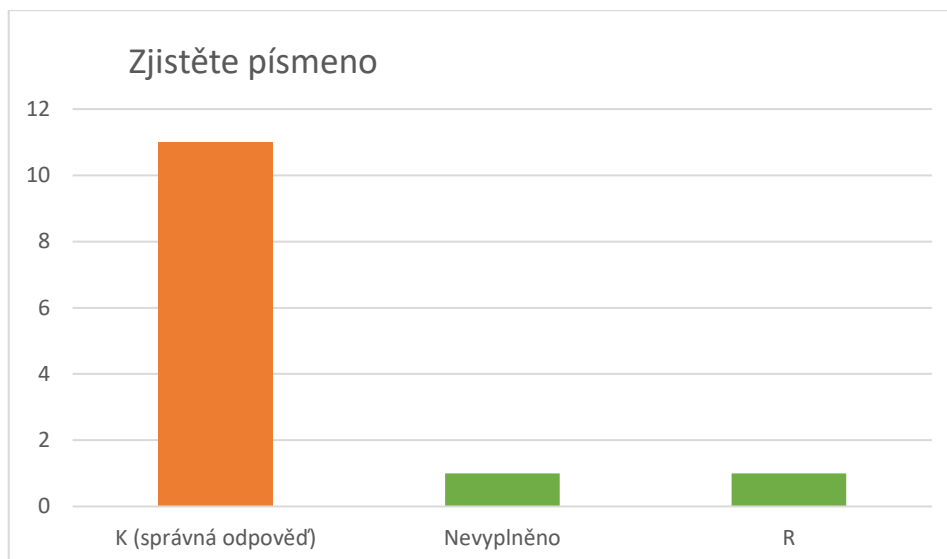
Typ otázky: stručná odpověď

Správná odpověď: K

Popis úlohy: viz kapitola 5.9

Předchozí znalosti studentů: viz kapitola 5.9

Úspěšnost studentů:



Graf 52: (Výstupní znalosti) Otázka 1

Studenti odpovídali ve většině případů (85 %) správně. Pouze jeden respondent otázku nevyplnil a jeden jako správnou odpověď napsal R.

5.10.3 OTÁZKA 2 (UPRAVENO Z IBOBR.CZ)

Zadání úlohy: V buňce C2 je napsán vzorec. Jestliže zkopírujeme tuto buňku do buňky D3, jaké číslo bude v buňce D3?

	A	B	C	D
1	2	5		
2	3	4	70	
3	8	4		
4				

Formula v buňce C2: $=A1+2*\$E\20

Obrázek 27: Zadání úlohy 2 - Výstup

Typ otázky: jedna odpověď

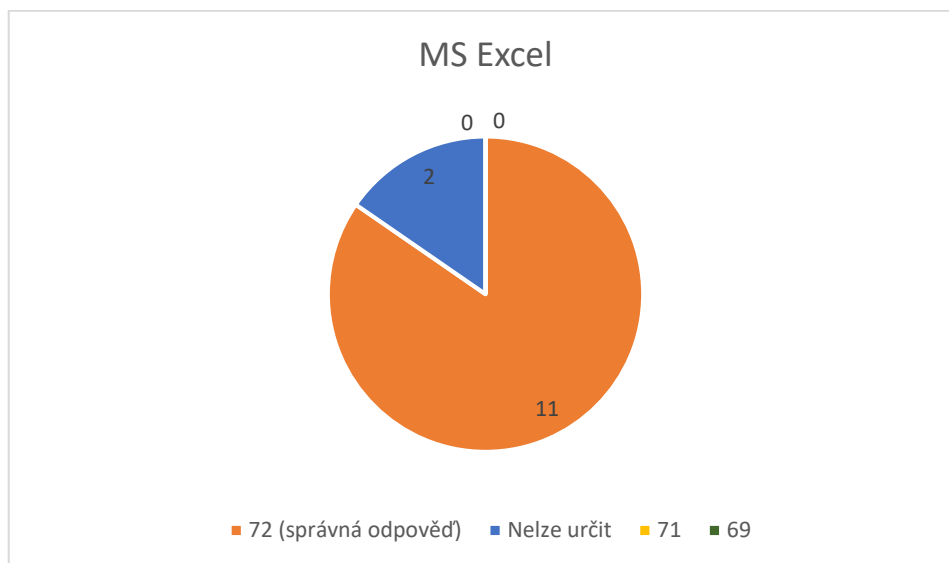
Možnosti: 72; 71; 69; Nelze určit

Správná odpověď: 72

Popis úlohy: viz kapitola 5.9

Předchozí znalosti studentů: viz kapitola 5.9

Úspěšnost studentů:



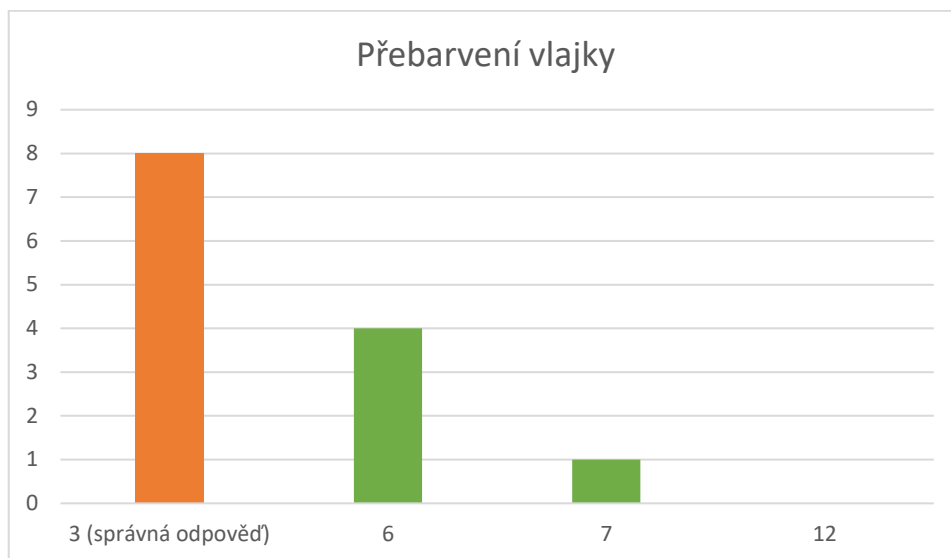
Graf 53: (Výstupní znalosti) Otázka 2

I u této otázky dosáhli studenti velice dobrých výsledků (85 % odpovídalo správně). Pouze dva respondenti vybrali odpověď „nelze určit“.

5.10.4 OTÁZKA 3 (ČERPANÁ Z IBOBR.CZ)

Stejně zadání viz kapitola 5.9

Úspěšnost studentů:

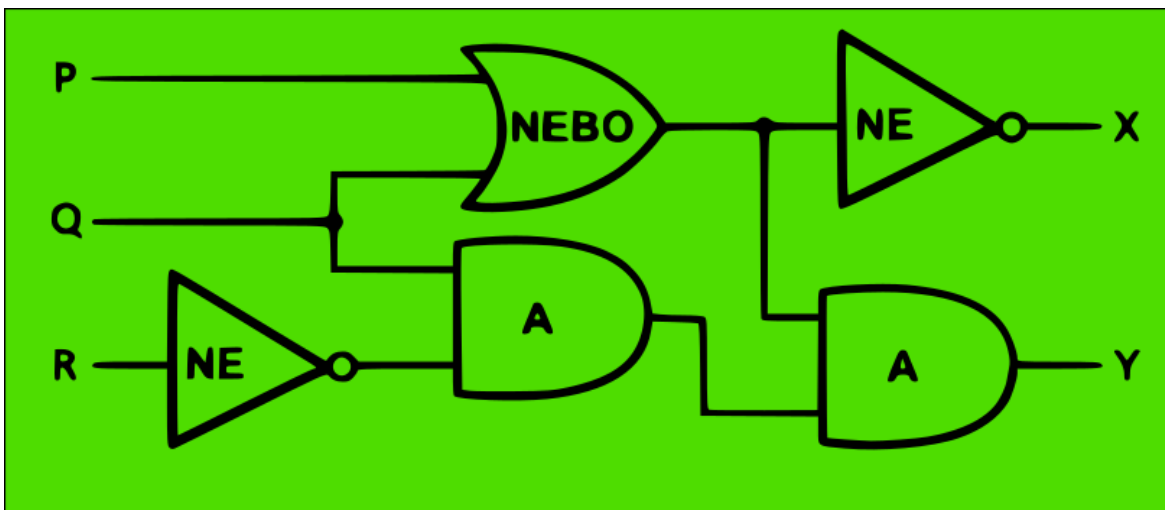


Graf 54: (Výstupní znalosti) Otázka 3

O něco málo hůře, než v předchozích dvou otázkách si studenti vedli u této otázky. Správnou odpověď v tomto případě volilo 62 % respondentů. Dalších 31 % volilo možnost 6. Jeden respondent zvolil možnost 7.

5.10.5 OTÁZKA 4 (UPRAVENÁ Z IBOBR.CZ)

Zadání úlohy: Jakých hodnot budou nabývat výstupy X a Y? $P=0$ (VYP) $Q=1$ (ZAP) a $R=0$ (VYP) pokud platí: NE (NOT) - Když je vstup ZAP, výstup je VYP, Když je vstup VYP, výstup je ZAP; NEBO (OR) - Výstup je ZAP kromě případu, kdy oba vstupy jsou VYP; A (AND) - Výstup je ZAP, pouze pokud oba vstupy jsou ZAP.



Obrázek 28: Zadání úlohy 4 - Výstup

Typ otázky: jedna odpověď

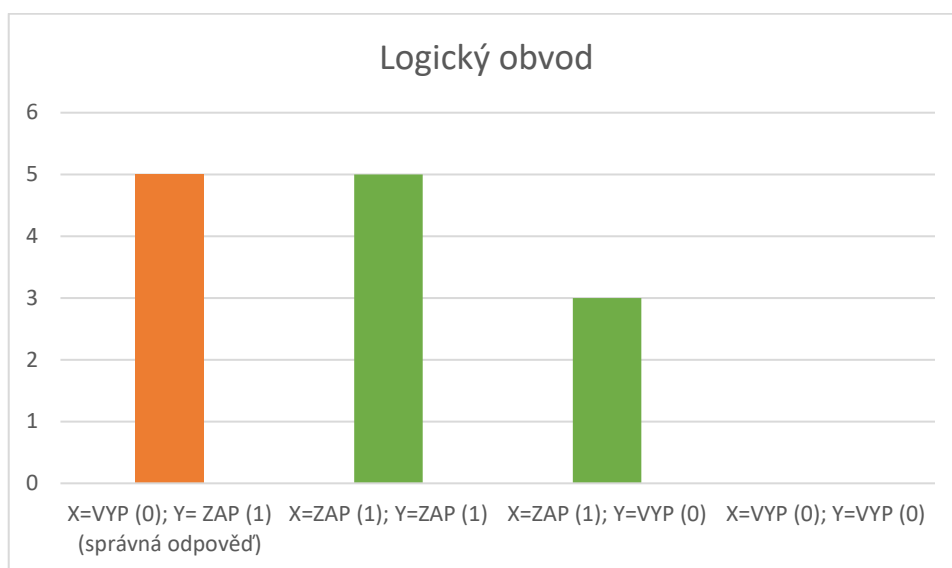
Možnosti: X=VYP (0); Y=VYP (0); X=ZAP (1); Y=ZAP (1); X=ZAP (1); Y=VYP (0); X=VYP (0); Y=ZAP (1)

Správná odpověď: X=VYP (0); Y= ZAP (1)

Popis úlohy: viz kapitola 5.9

Předchozí znalosti studentů: viz kapitola 5.9

Úspěšnost studentů:



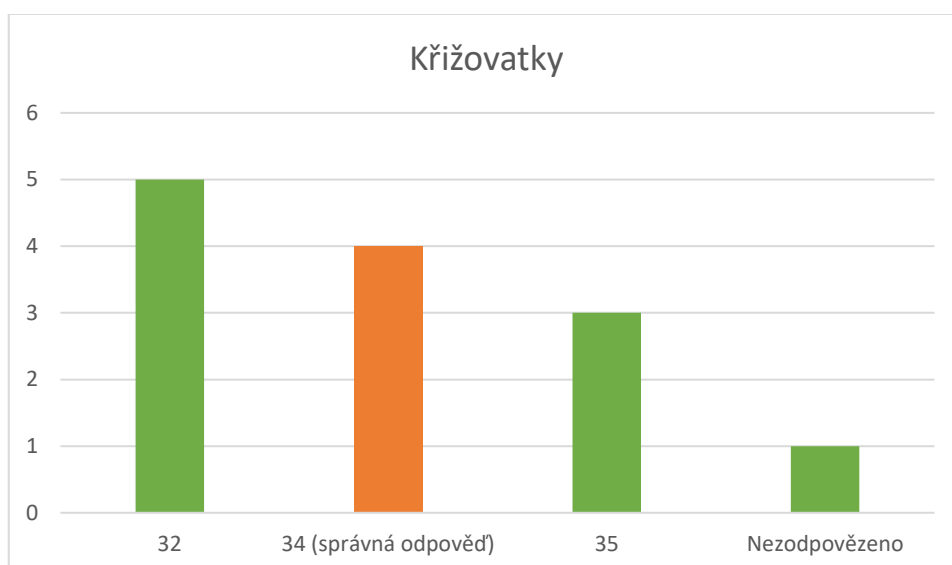
Graf 55: (Výstupní znalosti) Otázka 4

Obdobná otázka při vstupním testování dělala studentům veliký problém. I modifikovaná úloha byla pro studenty oříškem. Správně odpovědělo pouze 38 % respondentů.

5.10.6 OTÁZKA 5 (ČERPÁNO Z IBOBR.CZ)

Stejně zadání viz kapitola 5.9

Úspěšnost studentů:

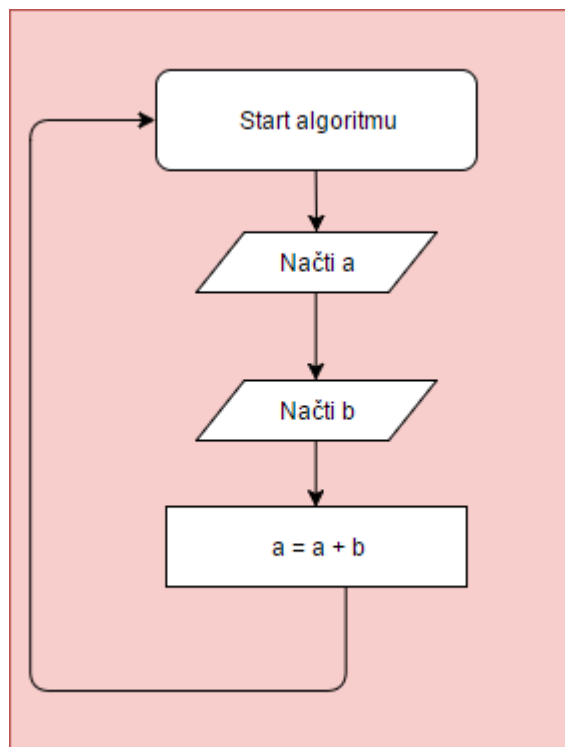


Graf 56: (Výstupní znalosti) Otázka 5

V této úloze, studenti dosáhli o poznání horších výsledků než při vstupním testování. Správnou odpověď označilo pouze 31 % respondentů.

5.10.7 OTÁZKA 6 (VLASTNÍ ZADÁNÍ)

Zadání úlohy: Jedná se o algoritmus?



Obrázek 29: Zadání úlohy 6 - Výstup

Typ otázky: jedna odpověď

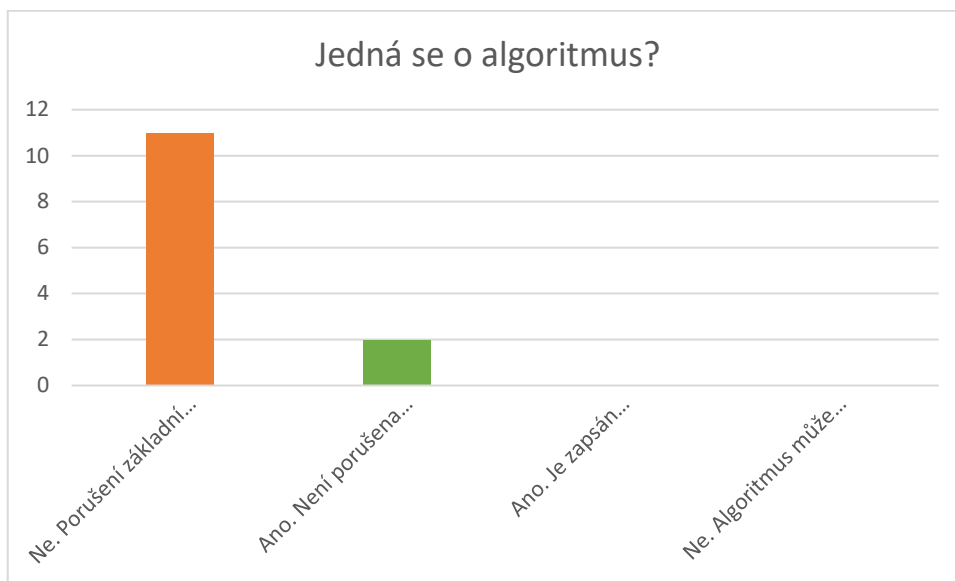
Možnosti: Ano. Je zapsán vývojovým diagramem; Ano. Je zapsán vývojovým diagramem a není zde porušena žádná základní vlastnost algoritmu; Ne. Je porušena základní vlastnost algoritmu; Ne. Algoritmus může být reprezentován pouze strukturogramem.

Správná odpověď Ne. Je porušena základní vlastnost algoritmu.

Popis úlohy: viz kapitola 5.9

Předchozí znalosti studentů: viz kapitola 5.9

Úspěšnost studentů:



Graf 57: (Výstupní znalosti) Otázka 6

Většina studentů na tuto otázku odpověděla správně (85 %). Pouze dva studenti odpovídali nesprávně.

5.10.8 OTÁZKA 7 (ZDROJ VLASTNÍ)

Zadání úlohy: Můžeme postup pro uvaření dobrého čaje považovat za algoritmus?

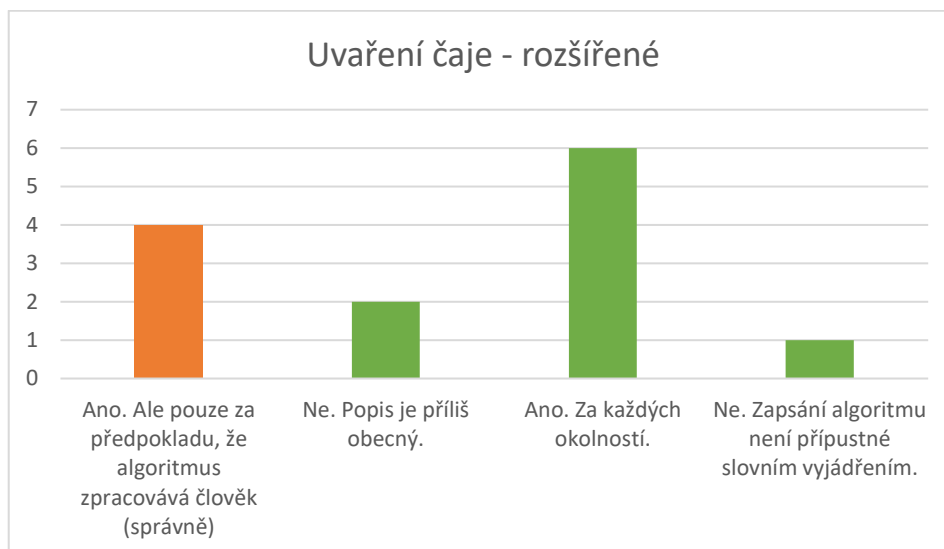
Možnosti: Ano. Ale pouze za předpokladu, že algoritmus zpracovává člověk; Ne. Popis je příliš obecný; Ano. Za každých okolností; Ne. Zapsání algoritmu není přípustné slovním vyjádřením.

Správná odpověď: Ano. Ale pouze za předpokladu, že algoritmus zpracovává člověk.

Popis úlohy: viz kapitola 5.9

Předchozí znalosti studentů: viz kapitola 5.9

Úspěšnost studentů:



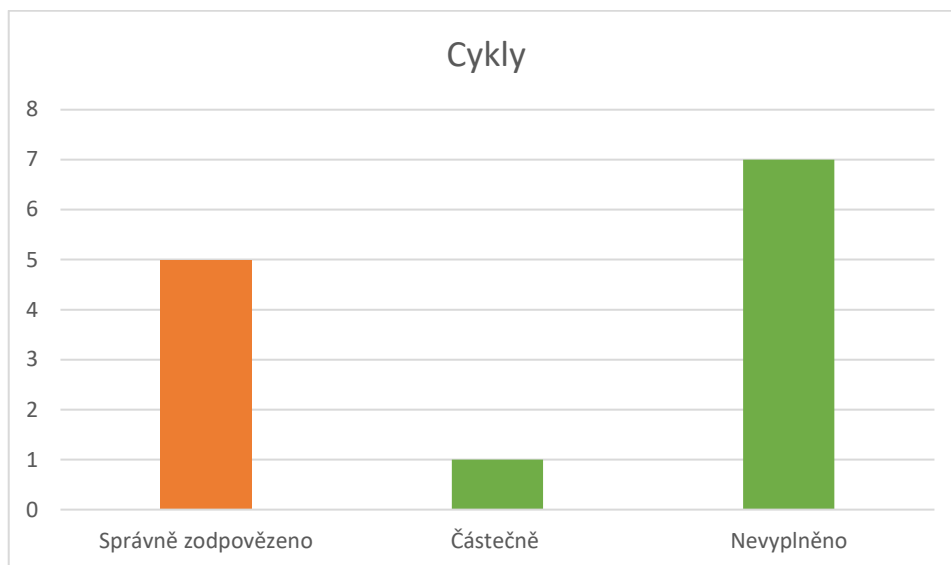
Graf 58: (Výstupní znalosti) Otázka 7

Modifikovaná otázka, která přímo vycházela z první kapitoly kurzu, nedopadla úplně dle očekávání. Správnou odpověď zvolilo pouze 31 % respondentů. Nejvíce respondentů (46 %) volilo odpověď „Ano. Za každých okolností.“ bohužel tato odpověď nemůže být brána za správnou, protože počítač by algoritmu pro uvaření čaje nerozuměl.

5.10.9 OTÁZKA 8 (ZDROJ VLASTNÍ)

Stejně zadání viz kapitola 5.9

Úspěšnost studentů:



Graf 59: (Výstupní znalosti) Otázka 8

U této náročnější znalostní otázky správně odpovídalo 38 % respondentů. 1 respondent uvedl dva typy cyklů. 54 % studentů otázku nevyplnilo.

5.10.10 OTÁZKA 9 (ZDROJ VLASTNÍ)

Zadání úlohy: Jednou ze základních vlastností algoritmu je determinovanost – vyberte vhodnou definici.

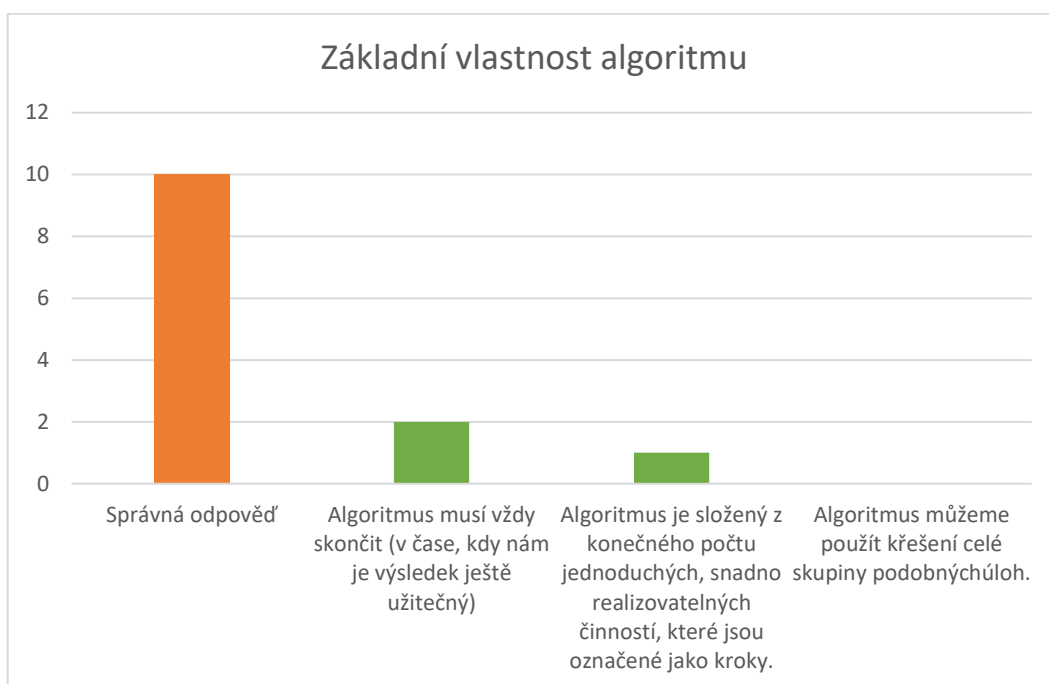
Možnosti: Algoritmus je složený z konečného počtu jednoduchých, snadno realizovatelných činností, které jsou označené jako kroky; Po každém kroku lze určit, zda popisovaný proces skončil, pokud neskončil, musí být definováno, kterým krokem se má pokračovat; Algoritmus musí vždy skončit (v čase, kdy nám je výsledek ještě užitečný); Algoritmus můžeme použít k řešení celé skupiny podobných úloh.

Správná odpověď: Po každém kroku lze určit, zda popisovaný proces skončil, pokud neskončil, musí být definováno, kterým krokem se má pokračovat.

Popis úlohy: Znalostní úloha, ve které mají studenti vybrat správnou definici základní vlastnosti algoritmu.

Předchozí znalosti studentů: Ano, pro správné vyřešení úlohy je nutné, aby studenti znali základní vlastnosti algoritmů.

Úspěšnost studentů:

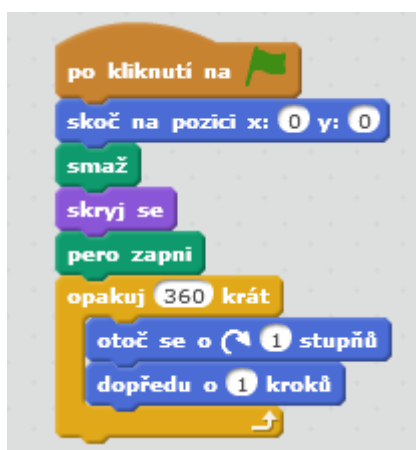


Graf 60: (Výstupní znalosti) Otázka 9

Odpoověď: „Po každém kroku lze určit, zda popisovaný proces skončil, pokud neskončil, musí být definováno, kterým krokem se má pokračovat,“ volila většina respondentů (77 %). Zbýlých 23 % studentů odpovědělo špatně.

5.10.11 OTÁZKA 10 (ZDROJ VLASTNÍ)

Zadání úlohy: Co bude výstupem tohoto jednoduchého programu?



Obrázek 30: Zadání úlohy 10 - Výstup

Typ otázky: jedna odpověď

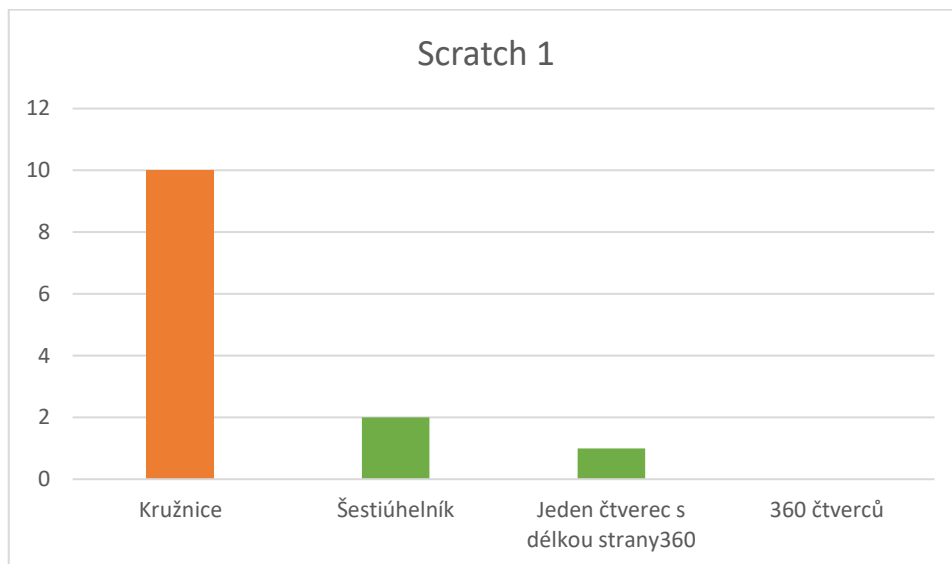
Možnosti: 360 čtverců; Kružnice; Jeden čtverec s délkou strany 360; Šestiúhelník

Správná odpověď: Kružnice

Popis úlohy: viz kapitola 5.9

Předchozí znalosti studentů: viz kapitola 5.9

Úspěšnost studentů:



Graf 61: (Výstupní znalosti) Otázka 10

Úkoly z vývojového prostředí Scratch studentům nedělaly problémy ani při vstupním testování. Zde volilo správnou odpověď 77 % respondentů.

5.10.12 OTÁZKA 11 (ZDROJ VLASTNÍ)

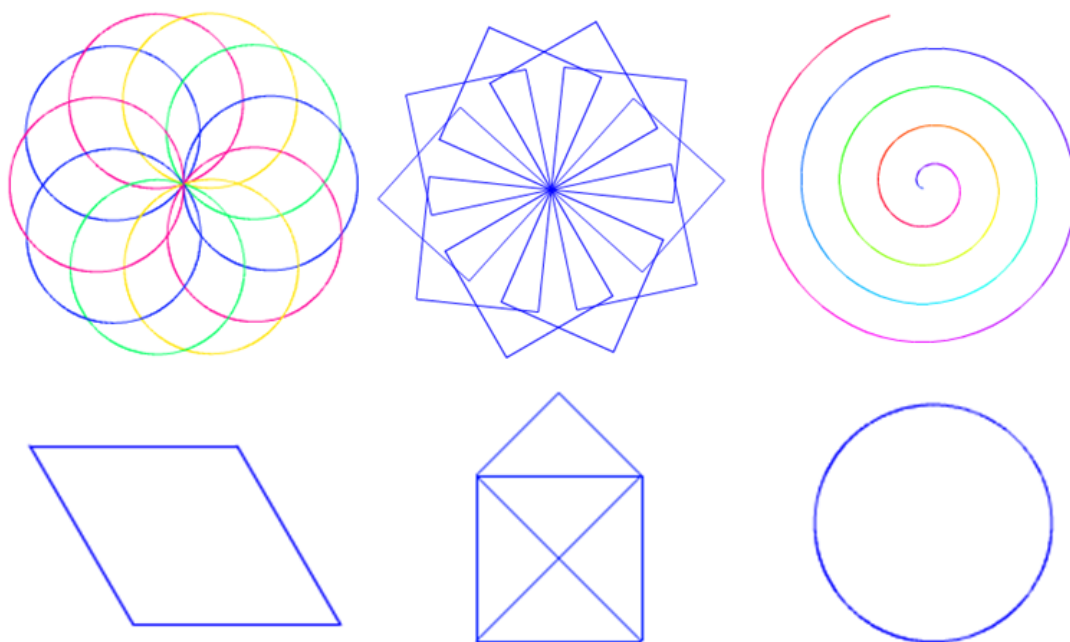
Zadání úlohy: Co bude výstupem tohoto jednoduchého programu?



Obrázek 31: Zadání úlohy 11 - Výstup

Typ otázky: jedna odpověď

Možnosti:



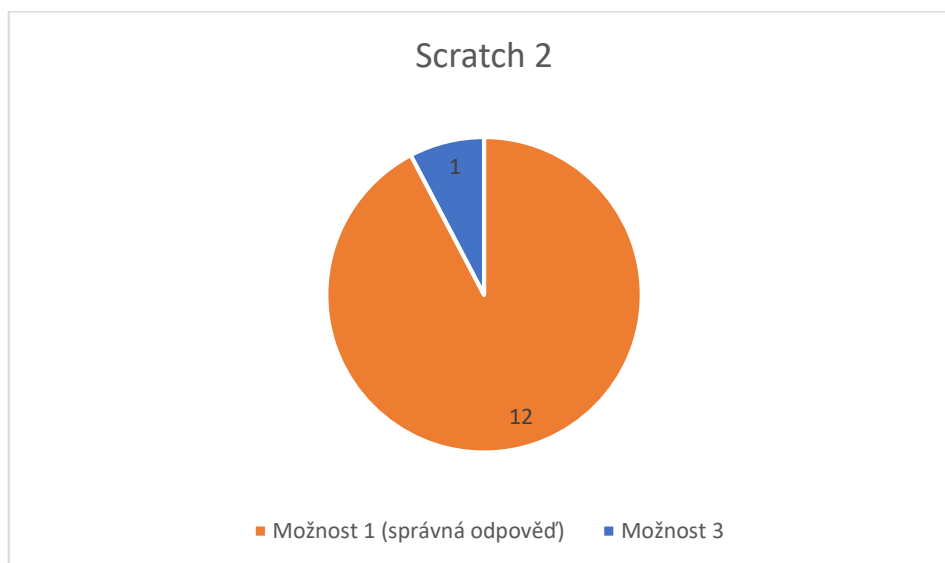
Obrázek 32: Možnosti úlohy 11 - Výstup

Správná odpověď: Možnost 1

Popis úlohy: viz kapitola 5.9

Předchozí znalosti studentů: viz kapitola 5.9

Úspěšnost studentů:



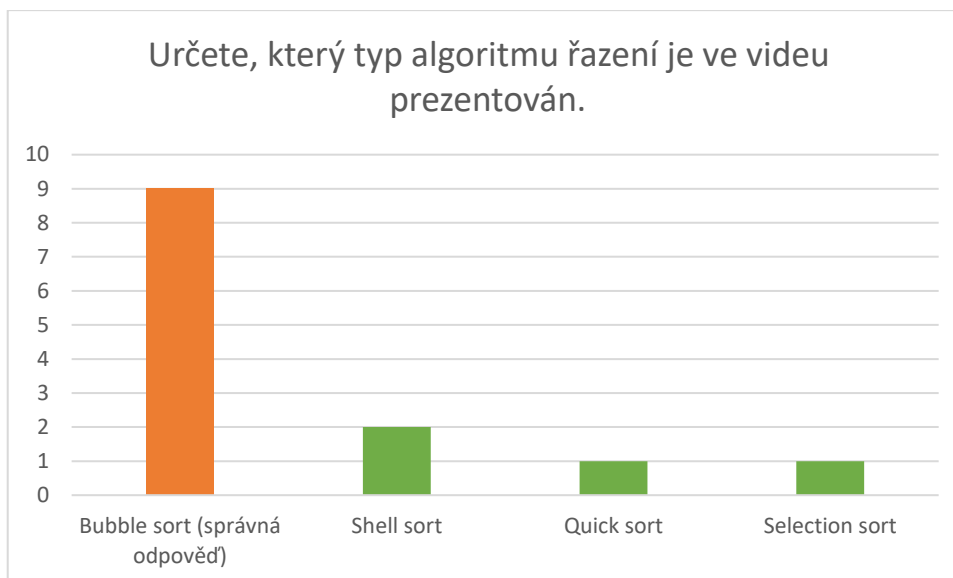
Graf 62: (Výstupní znalosti) Otázka 11

Ani u této úlohy studenti neměli sebemenší problémy. Správně odpovědělo 92 % respondentů. Pouze jeden respondent neodpověděl správně.

5.10.13 OTÁZKA 12 (ZDROJ VLASTNÍ)

Stejně zadání viz kapitola 5.9

Úspěšnost studentů:



Graf 63: (Výstupní znalosti) Otázka 12

Na otázku týkající se algoritmu řazení správně odpovídá 69 % respondentů.

5.11 POROVNÁNÍ VSTUPNÍCH A VÝSTUPNÍCH DOVEDNOSTÍ STUDENTŮ

V této podkapitole jsou uvedeny změny ve znalostní a dovednostní rovině studentů před a po absolvování kurzu.

Celkové vyhodnocení testu

Rozptyl získaných bodů se při vstupním a výstupním testování velice liší. Při vstupním testování byl rozptyl hodnot 4–8, při výstupním 4–12. Při výstupním testování tedy jeden respondent získal v testu plný počet bodů.

Aritmetický průměr výsledků při vstupním testování byl 6,33 bodu, při výstupním 7,77 bodu. Průměrné zlepšení po absolvování kurzu je tedy téměř o 1,5 bodu s tím, že je nutno zohlednit, že testové otázky byly pro respondenty náročnější.

Během analýzy vstupních a výstupních znalostí a dovedností studentů byly zjištěny 2 extrémy. U některých otázek se studenti po absolvování kurzu výrazně zlepšili, u některých se výrazně zhoršili.

Otázka 1 – Zjistěte písmeno

U této otázky proběhlo u studentů mírné zlepšení. Při vstupním testování odpovídalo správně 73 % respondentů. U výstupního testování 85 % respondentů. Otázka byla při výstupním testování modifikována.

Otázka 2 – MS Excel

U druhé otázky v testové části je velmi viditelné zlepšení studentů. Při vstupním testování správně odpovědělo 54 % respondentů. Na druhé straně, při výstupním testování správně odpovědělo 73 % respondentů. Otázka byla při výstupním testování modifikována.

Otázka 3 – Přebarvení vlajky

Třetí otázka přinesla pouze mírné zlepšení na výstupu (62 %) oproti (47 %) na vstupu. Otázka byla shodná při vstupním i výstupním testování.

Otázka 4 – Logický obvod

Čtvrtá otázka, která byla pro studenty velice náročná, měla velice podobnou úspěšnost při vstupním testování (40 %), jako při testování výstupním (38 %). Otázka byla při výstupním testování modifikována.

Otázka 5 – Křížovky

U páté otázky byl výsledek horší u výstupního testování (32 % respondentů odpovědělo správně), než u vstupního (47 % respondentů odpovědělo správně). Otázka byla shodná při vstupním i výstupním testování.

Takto špatný výsledek může být způsobený tím, že v elektronickém kurzu nebyla věnována pozornost algoritmům pro hledání nejkratší cesty a při vstupním testování měli studenti větší štěstí při náhodném výběru odpovědi.

Otázka 6 – Jedná se o algoritmus?

U této otázky se po absolvování kurzu projevilo velké zlepšení studentů. Při výstupním testování na otázku odpovědělo správně 92 % respondentů oproti 46 % při vstupním testování. Otázka byla při výstupním testování modifikována.

Otázka 7 – Uvaření čaje

V pořadí sedmá otázka, u které byl největší pokles úspěšnosti. Při vstupním testování správnou odpověď zvolilo 92 % respondentů, při výstupním testování správnou odpověď zvolilo pouze 31 % respondentů. Otázka byla při výstupním testování modifikována.

V tomto případě byla otázka položena stejným způsobem, pouze možnosti výběru byly jiné, u vstupního testování byla pouze možnost Ano/Ne, u výstupního testování byly možnosti

4. Správnou odpovědí na otázku „Můžeme postup pro uvaření dobrého čaje považovat za algoritmus?“ u výstupního testování bylo: „Ano. Ale pouze za předpokladu, že algoritmus zpracovává člověk.“ Studenti nejčastěji volili, že tento postup může být považován za algoritmus za každých okolností.

Otázka 8 – Cykly

U otázky věnující se cyklům bylo dosaženo podobné úspěšnosti při vstupním (33 % respondentů odpovědělo správně) i výstupním testování (38 % respondentů odpovědělo správně). Otázka byla shodná při vstupním i výstupním testování.

Otázka 9 – Úvodní otázka Scratch a základní vlastnosti algoritmu

Při vstupním testování bylo vhodné dát studentům popis bloků pro inicializaci želví grafiky v prostředí Scratch. Při výstupním testování již úvodní otázky nebylo třeba, proto byla nahrazena otázkou týkající se základních vlastností algoritmu.

Na úvodní otázku týkající se želví grafiky ve Scratchi odpověděli správně všichni respondenti.

Na otázku týkající se základních vlastností algoritmu odpovědělo správně 77 % respondentů.

Otázka 10 – Scratch

U desáté otázky byly výsledky na vstupu (67 % správně odpovídajících) přibližně stejné jako na výstupu (77 % správně odpovídajících). Otázka byla při výstupním testování modifikována.

Otázka 11 – Scratch 2

V pořadí jedenáctá otázka měla podobnou úspěšnost při vstupním (87 % správně odpovídajících) i výstupním testování (92 % správně odpovídajících). Otázka byla při výstupním testování modifikována.

Otázka 12 – Algoritmy řazení

U poslední otázky bylo zaznamenáno mírné zlepšení na výstupu (69 % správných odpovědí), oproti vstupnímu testování (47 % správných odpovědí). Otázka byla shodná při vstupním i výstupním testování.

ZÁVĚR

V této diplomové práci byly nejprve představeny základní pojmy. V následující kapitole byl prostor pro popsání metod pro rozvoj algoritmického myšlení, které byly inspirovány slovenským Štátným vzdelávacím programom. V pořadí třetí kapitola se věnuje nástrojům pro rozvoj algoritmického myšlení, mezi které byly zařazeny robotické hračky a stavebnice (Bee-Bot, Ozobot, Dash a Dot a Lego Mindstorms) a aplikace (Scratch, Kodu, EasyLogo, Hour of Code a Code combat).

V následující kapitole byla popsána metodika tvorby elektronického kurzu v prostředí LMS Moodle, která vycházela ze vstupního testování studentů. V této kapitole je také zahrnut popis použitých činností a studijních materiálů a multimédií. V této části byly představeny také cíle kurzu. Následně byl kurz prakticky otestován a výsledky byly analyzovány. Z analýzy výsledků pak vycházely návrhy pro vylepšení kurzu.

V pořadí pátá kapitola byla věnována výzkumnému šetření mezi studenty KVD/ALGV. První část šetření podala ucelenou charakteristiku výzkumného vzorku. Následně proběhlo testování vstupních a výstupních postojů studentů z oblasti algoritmizace a jejich porovnání, které se ve většině případů příliš nelišilo vyjma postojů k výrokům („Pro absolventy SŠ je důležité umět rozkládat velké problémy na podproblémy.“ a „Vyučovat algoritmizaci na základní škole má smysl.“), s kterými se po absolvování kurzu ztotožnilo mnohem více respondentů. Třetí část této kapitoly byla věnována vyhodnocení vstupních a výstupních znalostí a dovedností studentů. Studenti po absolvování předmětu dosáhli vyššího průměrného bodového ohodnocení než při testování vstupním.

Cíle vytyčené v úvodu práce byly splněny. Na tuto diplomovou práci je možné navázat vylepšením on-line kurzu, do kterého by bylo vhodné zařadit více multimédií, cvičení a autotestů. Dále by bylo vhodné, opakovat a analyzovat zlepšení i v dalších ročnících běhu předmětu.

RESUMÉ

Tato diplomová práce pojednává o problematice výuky algoritmizace na vysoké škole a tvorbě on-line kurzu k předmětu Algoritmizace ve vzdělávání. V teoretické části je věnován prostor pro vymezení základních pojmů, mezi které patří algoritmus, algoritmické myšlení a algoritmizace. Dále jsou v teoretické části popsány metody a nástroje určené pro rozvoj algoritmického myšlení.

V praktické části jsou zjištěny vstupní a výstupní znalosti a dovednosti studentů bakalářského studia Informatika zaměřená na vzdělávání a jejich postoje před a po absolvování kurzu. Na základě vstupního šetření je vytvořen on-line kurz, který je následně evaluován.

This diploma thesis deals with problems associated with teaching algorithms at university level, as well as creating an on-line course for the subject "Algorithms in education". In theoretical part, space is devoted to the definition of basic terms, among which belong algorithms, algorithmic thinking, and creating algorithms. The theoretical part then describes the methods and tools intended for the development of algorithmic thinking.

In the practical part, the input and output knowledge and skills of Informatics Bachelors degree students and their approach before and after the course are analysed. An on-line course is created on the basis of the input research, and then the course is evaluated.

SEZNAM LITERATURY

1. BERKI, J., 2016. Projektované, realizované a dosažené ICT kurikulum na základních školách. České Budějovice. Disertační práce. Jihočeská univerzita v Českých Budějovicích, Pedagogická fakulta. [online] Dostupné z: <<https://theses.cz/id/y5olfe/>> [cit. 2018-06-22].
2. BROMOVÁ, J., 2012. Výuka algoritmizace na základní škole - aktuální stav. Master's thesis. University of South Bohemia in České Budějovice, Pedagogical Faculty, České Budějovice. Thesis supervisor doc. PaedDr. Jiří Vaníček, Ph.D. [online] Dostupné z: <<https://theses.cz/id/p527u4/>> [cit. 2018-06-22]. ČERNOCHOVÁ, Miroslava, 2005. K východiskům didaktiky informační výchovy. In: Pavel BENEŠ, Vladimír RAMBOUSEK a Irena FIALOVÁ (Eds.). Vzdělávání pro život v informační společnosti I. Praha: Univerzita Karlova v Praze, Pedagogická fakulta. s. 171–192. ISBN 80-7290-198-2.
3. ČERNOCHOVÁ, M., 2005. K východiskům didaktiky informační výchovy. In: Pavel BENEŠ, Vladimír RAMBOUSEK a Irena FIALOVÁ (Eds.). Vzdělávání pro život v informační společnosti I. Praha: Univerzita Karlova v Praze, Pedagogická fakulta, s. 171–192. ISBN 80-7290-198-2.
4. ČERNÝ, M. Výukoví roboti: nástroj pro rozvoj algoritmického myšlení. [online] Dostupné z: <https://clanky.rvp.cz/clanek/c/g/19905/VYUKOVI-ROBOTI-NASTROJ-PRO-ROZVOJ-ALGORITMICKEHO-MYSLENI.html/>
5. DRLÍK, Martin. *Moodle: kompletní průvodce tvorbou a správou elektronických kurzů*. Brno: Computer Press, 2013. ISBN 978-80-251-3759-8.
6. ĎURÁKOVÁ, D., Dvorský, J., a Ochodková, E., 2002. Základy algoritmizace. Scriptum. VŠB Technická univerzita Ostrava, Katedra informatiky, Ostrava, 289 s.
7. FUTSCHEK, G. Algorithmic Thinking: The Key for Understanding Computer Science. [online] Dostupné z: <https://publik.tuwien.ac.at/files/PubDat_140308.pdf>
8. HABIBALLA, H. Využití programovacích metod při výuce teoretické informatiky (disertační práce). FPV UKF Nitra. 19.2.2004.146 s.
9. HARPER, Douglas. (2016). Algorithm [online]. Online Etymology Dictionary. Lancaster [cit. 2018-06-21]. Dostupné z: <https://www.etymonline.com/word/algorithm>

10. HORNIK, Tomáš. Možnosti rozvíjení algoritmického myšlení s využitím projektů Hour of Code a Scratch [online]. Hradec Králové, 2016 [cit. 2018-06-22]. Dostupné z: <<https://theses.cz/id/jf3lnn/>>. Diplomová práce. Univerzita Hradec Králové, Pedagogická fakulta. Vedoucí práce PhDr. Michal Musílek, Ph.D..
11. HROMKOVIČ, J., Kohn, T., Komm, D., Serafini, G., 2016. Examples of Algorithmic Thinking in Programming Education. [online] Dostupné z: <http://ioinformatics.org/oi/pdf/v10_2016_111_124.pdf>
12. KNUTH, D. The Art of Computer Programming [online]. [cit. 2018-06-20]. Dostupné z: http://broiler.astrometry.net/~kilian/The_Art_of_Computer_Programming%20-%20Vol%201.pdf
13. LANDA, L. N. Algorithmization in learning and instruction. Englewood Cliffs, N.J.: Educational Technology Publications, [1974]. ISBN 0877780633.
14. KRČEK, B., KREML, P., 1993. ALGORITMIZACE A PROGRAMOVÁNÍ V JAZYKU PASCAL. 1. vyd. Ostrava: VŠB – Technická univerzita.
15. MAREŠ, M., 2006. Slova, která se hodí, aneb jak si povídat o matematice, kybernetice a informatice. Praha: Academia, 352 s. ISBN 80-200-1445-4
16. MELICHAR, J., 2006. ALGORITMY NA 1. STUPNI ZÁKLADNÍ ŠKOLY. [online] Dostupné z: <http://pf.ujep.cz/files/KMA_poznamkydidamat04.pdf> [cit 2011-02-15].
17. MOODLE česky [online]. [cit. 2018-07-01]. Dostupné z: <https://moodle.org/course/view.php?id=17227>
18. MOTYČKA, A., 1999. Algoritmizace. Brno: Konvoj, 75 s. Scriptum. ISBN 80-85615-80-0.
19. OPAVA, Z., 1989. Matematika kolem nás. 1. vyd. Praha: Albatros, 368 s. ISBN: 13-781-89.
20. PELANEK, R., 2012. Programátorská cvičebnice: [algoritmy v příkladech]. Brno: Computer Press. ISBN 978-80-251-3751-2.
21. PŠENČÍKOVÁ, J., 2009. ALGORITMIZACE. 2. vyd. Kralice na Hané: Computer Media. ISBN 978-80-7402-034-6.
22. Python. [online] Dostupné z: <<https://www.python.org/>>

23. SALANCI, L. EasyLogo. [online] Dostupné z:
<<http://edi.fmph.uniba.sk/~salanci/EasyLogo/index.html>>
24. SALANCI, L. Easy Logo – discovering basic programming concepts in a constructive manner. [online] Dostupné z: <http://edi.fmph.uniba.sk/~salanci/EasyLogo/Paper.pdf>
25. Scratch. [online] Dostupné z: <https://scratch.mit.edu/about>
26. Univerzitní Moodle, Autorská a tuteurská práce v Moodle, [online]. [cit. 2018-07-01]. Dostupné po přihlášení z: <https://phix.zcu.cz/moodle/course/view.php?id=1863>
27. VIRIUS, M., 1995. Základy algoritmizace. Scriptum. Praha: ČVUT, 179 s. ISBN 80-01-01346-4.
28. VIRIUS, M., 2008. Základy algoritmizace. Vyd. 2., přeprac. Praha: Česká technika - nakladatelství ČVUT, 265 s. ISBN 978-80-01-04003-4.
29. Wonder workshop. [online] Dostupné z:
<<https://www.makewonder.com/robots/dashanddot>>
30. WRÓBLEWSKI, P., 2015. Algoritmy. 2. vydání. Brno: Computer Press. ISBN 978-80-251-4126-7.
31. ZEMANEK, H., 1981. Al-Khorezmi - His Background, His Personality, His Work and His Influence. In: Ershov, A. P., Knuth, D. E. (eds.): Algorithms in Modern Mathematics and Computer Science. Lecture Notes in Computer Science, Vol. 122. Berlin-Heidelberg-New York: Springer, p. 1–81.

SEZNAM OBRÁZKŮ, TABULEK, GRAFŮ A DIAGRAMŮ

Obrázek 1: Prostředí Scratch	15
Obrázek 2: Prostředí Scratch	15
Obrázek 3: Vygenerovaný svět v Kodu	16
Obrázek 4: Programování v Kodu	17
Obrázek 5: Úloha v aplikaci EasyLogo	18
Obrázek 6: Ukázka z Hour of Code	19
Obrázek 7: Prostředí hry Code Combat	20
Obrázek 8: Kód ve hře Code Combat	20
Obrázek 9: Robotická hračka Bee-Bot (zdroj: volné dílo)	21
Obrázek 10: Ukázka kódu - Ozobot	22
Obrázek 11: Ozobot (zdroj: volné dílo)	23
Obrázek 12: Přehled programovacích bloků EV3	24
Obrázek 13: Přihlašování do Moodle pomocí WebAuth	27
Obrázek 14: Oddělení jednotlivých témat	28
Obrázek 15: Zadání úlohy 1	70
Obrázek 16: Zadání úlohy 2	71
Obrázek 17: Zadání úlohy 3	73
Obrázek 18: Zadání úlohy 4	74
Obrázek 19: Zadání úlohy 5	76
Obrázek 20: Zadání úlohy 6	77
Obrázek 21: Zadání úlohy 9	81
Obrázek 22: Zadání úlohy 10	82
Obrázek 23: Zadání úlohy 11	84
Obrázek 24: Možnosti úlohy 11	84
Obrázek 25: Řešení úlohy 11	85
Obrázek 26: Zadání úlohy 1 - Výstup	88
Obrázek 27: Zadání úlohy 2 - Výstup	89
Obrázek 28: Zadání úlohy 4 - Výstup	92
Obrázek 29: Zadání úlohy 6 - Výstup	94
Obrázek 30: Zadání úlohy 10 - Výstup	98
Obrázek 31: Zadání úlohy 11 - Výstup	100
Obrázek 32: Možnosti úlohy 11 - Výstup	100
Tabulka 1: Štátny vzdelávací program - Analýza problému	11
Tabulka 2: Štátny vzdelávací program - Jazyk na zápis řešení	11
Tabulka 3: Štátny vzdelávací program: Pomocí posloupnosti příkazů	11
Tabulka 4: Štátny vzdelávací program - Pomocí cyklů	12
Tabulka 5: Štátny vzdelávací program - Pomocí větvení	12
Tabulka 6: Štátny vzdelávací program - Pomocí proměnných	12
Tabulka 7: Štátny vzdelávací program - Pomocí nástroje na interakci	13
Tabulka 8: Štátny vzdelávací program - Interpretace zápisu řešení	13
Tabulka 9: Štátny vzdelávací program - Hledání, opravování chyb	13
Tabulka 10: Porovnání postojů 1	68
Tabulka 11: Porovnání postojů 2	68
Graf 1: (Evaluace kurzu) Náročnost kurzu	35

Graf 2: (Evaluace kurzu) Srozumitelnost kurzu	36
Graf 3: (Evaluace kurzu) Podrobnost kurzu	37
Graf 4: (Evaluace kurzu) Dostatečnost multimédií	38
Graf 5: (Evaluace kurzu) Dostatečnost cvičení v kurzu	39
Graf 6: (Evaluace kurzu) Přehlednost kurzu	40
Graf 7: (Evaluace kurzu) Zajímavost kurzu	41
Graf 8: (Evaluace kurzu) Použití externích zdrojů	42
Graf 9: (Charakteristika respondentů) Věk studentů	45
Graf 10: (Charakteristika respondentů) Vystudovaná střední škola	46
Graf 11: (Charakteristika respondentů) Obsah hodin informatiky	47
Graf 12: (Charakteristika respondentů) Programování na střední škole	48
Graf 13: (Charakteristika respondentů) Předchozí studium na vysoké škole	49
Graf 14: (Charakteristika respondentů) Programování a volný čas	50
Graf 15: (Charakteristika respondentů) Tvorba vývojových diagramů a strukturogramů ..	51
Graf 16: (Charakteristika respondentů) Tvorba programu	51
Graf 17: (Charakteristika respondentů) Praktická zkušenost s dětskými programovacími prostředími	52
Graf 18: (Charakteristika respondentů) Nejčastější problémy při programování	53
Graf 19: (Charakteristika respondentů) Zastoupení učitelů mezi studenty	53
Graf 20: (Vstupní postoje) Obtížnost testové části	54
Graf 21: (Vstupní postoje) Postoje při programování a algoritmizaci	55
Graf 22: (Vstupní postoje) Studenti oboru informatika se zaměřením na vzdělávání by měli umět skvěle programovat	56
Graf 23: (Vstupní postoje) Je programování pro studenty náročné?	57
Graf 24: (Vstupní postoje) Rozkládání velkých problémů na podproblémy	58
Graf 25: (Vstupní postoje) Vytrvalost studentů při řešení složitějších problémů	59
Graf 26: (Vstupní postoje) Nejdříve algoritmizace a poté programování?	60
Graf 27: (Vstupní postoje) Algoritmizace na základní škole	61
Graf 28: (Vstupní postoje) Algoritmizace a on-line kurz	62
Graf 29: (Výstupní postoje) Obtížnost testové části	63
Graf 30 (Výstupní postoje) Postoje při programování a algoritmizaci	64
Graf 31: (Výstupní postoje) Studenti oboru informatika se zaměřením na vzdělávání by měli umět skvěle programovat	64
Graf 32: (Výstupní postoje) Je programování pro studenty náročné?	65
Graf 33: (Výstupní postoje) Rozkládání velkých problémů na podproblémy	65
Graf 34: (Výstupní postoje) Vytrvalost studentů při řešení složitějších problémů	66
Graf 35: (Výstupní postoje) Nejdříve algoritmizace a poté programování?	66
Graf 36: (Výstupní postoje) Algoritmizace na základní škole	67
Graf 37: (Výstupní postoje) Algoritmizace a on-line kurz	67
Graf 38: (Vstupní znalosti) Počet získaných bodů	69
Graf 39: (Vstupní znalosti) Otázka 1	71
Graf 40: (Vstupní znalosti) Otázka 2	72
Graf 41: (Vstupní znalosti) Otázka 3	74
Graf 42: (Vstupní znalosti) Otázka 4	75
Graf 43: (Vstupní znalosti) Otázka 5	77
Graf 44: (Vstupní znalosti) Otázka 6	78
Graf 45: (Vstupní znalosti) Otázka 7	79
Graf 46: (Vstupní znalosti) Otázka 8	80
Graf 47: (Vstupní znalosti) Otázka 9	82
Graf 48: (Vstupní znalosti) Otázka 10	83

Graf 49: (Vstupní znalosti) Otázka 11	85
Graf 50: (Vstupní znalosti) Otázka 12.....	86
Graf 51: (Výstupní znalosti) Počet získaných bodů	88
Graf 52: (Výstupní znalosti) Otázka 1	89
Graf 53: (Výstupní znalosti) Otázka 2.....	90
Graf 54: (Výstupní znalosti) Otázka 3.....	91
Graf 55: (Výstupní znalosti) Otázka 4.....	93
Graf 56: (Výstupní znalosti) Otázka 5.....	93
Graf 57: (Výstupní znalosti) Otázka 6.....	95
Graf 58: (Výstupní znalosti) Otázka 7.....	96
Graf 59: (Výstupní znalosti) Otázka 8.....	97
Graf 60: (Výstupní znalosti) Otázka 9.....	98
Graf 61: (Výstupní znalosti) Otázka 10.....	99
Graf 62: (Výstupní znalosti) Otázka 11	101
Graf 63: (Výstupní znalosti) Otázka 12.....	102

PŘÍLOHY

Volitelně se zde mohou nacházet přílohy.