

Univerzita Karlova

Pedagogická fakulta

Katedra informačních technologií a technické výchovy

DIPLOMOVÁ PRÁCE

Možnosti rozvoje informatického myšlení u žáků 5. třídy s využitím prostředí
Scratch

Possibilities of developing computational thinking of 5th grade pupils using
Scratch

Bc. Marie Fejtková

Vedoucí práce: PhDr. Petra Vaňková, Ph.D.

Studijní program: N7504 Učitelství pro střední školy

Studijní obor: Učitelství VVP pro ZŠ a SŠ – informační a komunikační technologie

Odevzdáním této diplomové práce na téma Možnosti rozvoje informatického myšlení u žáků 5. třídy s využitím prostředí Scratch potvrzuji, že jsem ji vypracovala pod vedením vedoucího práce samostatně za použití v práci uvedených pramenů a literatury. Dále potvrzuji, že tato práce nebyla využita k získání jiného nebo stejného titulu.

Praha 10. 7. 2024

Na tomto místě bych chtěla poděkovat vedoucí práce PhDr. Petře Vaňkové Ph.D. za její cenné rady, velkou trpělivost, ochotu a veškerou pomoc při vedení této práce.

ABSTRAKT

Hlavním cílem této diplomové práce je analyzovat informatické myšlení z hlediska základních konceptů a konkrétní cílové skupiny, žáků 5. ročníku, a na základě získaných poznatků sestavit soubor na sebe navazujících aktivit a úloh se zvyšující se obtížností v programovacím prostředí Scratch, které respektují a rozvíjí základní koncepty informatického myšlení.

V teoretické části jsou definovány cíle práce a využívané výzkumné metody. Tato část je věnována nejen analýze informatického myšlení z hlediska základních konceptů, ale i analýze vzdělávacích materiálů pro oblast Informatiky pro žáky 5. ročníku. Dále se zabývá analýzou nástrojů pro výuku algoritmizace a programování, zejména programovacího prostředí Scratch.

V rámci praktické části byl vytvořen soubor úloh, který se skládá z osmi na sebe navazujících úloh s postupně se zvyšující náročností ověřených v praxi metodou akčního výzkumu. Výsledkem šetření je vznik uceleného souboru úloh včetně doplňujících materiálů (ukázek řešení úloh, prezentací a pracovního listu) a metodických pokynů. Výstupem práce je i webová stránka určená žákům, na které jsou publikované úlohy a další materiály.

KLÍČOVÁ SLOVA

Informatické myšlení, Scratch, programování, výukové materiály, 5. třída ZŠ, soubor úloh

ABSTRACT

The main objective of this thesis is to analyze computational thinking in terms of the key concepts and specific target group, 5th grade pupils, and based on the findings to produce a set of interrelated activities and tasks with increasing difficulty in the Scratch programming environment that respect and develop the basic concepts of computational thinking.

The theoretical part is dedicated to both the analysis of computational thinking in terms of the key concepts and the analysis of educational materials for the field of Computer Science for 5th grade students. Furthermore, it inquires into the analysis of tools for teaching algorithmization and programming, in particular of Scratch programming environment.

As a part of the practical part, a set of tasks has been produced, which consists of 8 consecutive tasks with increasing difficulty, verified in practice by the method of action research. The result of the investigation is the production of a comprehensive set of tasks, including supplementary material (sample solutions, presentations and a worksheet) and methodological instructions. Another result of this thesis is a website for pupils on which the tasks and other materials are published.

KEYWORDS

Computational thinking, Scratch, programming, educational materials, 5th grade, set of tasks

Obsah

1	Úvod.....	7
2	Teoretická část.....	8
2.1	Vymezení cílů a výzkumného pole	8
2.1.1	Výzkumný problém.....	8
2.1.2	Hlavní cíl.....	8
2.1.3	Dílčí cíle a úkoly	8
2.1.4	Metodika.....	9
2.2	Informatické myšlení.....	11
2.2.1	Východiska pro vznik IM.....	11
2.2.2	Definice pojmu informatické myšlení.....	12
2.2.3	Koncepty IM	18
2.3	Informatické myšlení v rámci předmětu Informatika v 5. ročníku.....	24
2.3.1	Revize Rámcového vzdělávacího programu	24
2.3.2	Výukové materiály pro vzdělávací oblast Informatiky pro 5. ročník	27
2.3.3	Nástroje pro výuku oblasti „Algoritmizace a programování“	34
2.3.4	Scratch.....	39
3	Praktická část.....	43
3.1	Cíl práce.....	43
3.2	Rozbor úloh	43
3.2.1	Z hlediska informatického myšlení	43
3.2.2	Z hlediska pedagogického	44
3.2.3	Z hlediska RVP ZV	45
3.2.4	Obecné metodické pokyny ke všem úlohám.....	47

3.2.5	Metody a formy výuky	50
3.3	Výzkumný soubor.....	53
3.4	Soubor úloh.....	55
3.4.1	Úloha č. 1: Vaření s robotem	55
3.4.2	Úloha č. 2: Kreslení schodů s robotem	60
3.4.3	Úloha 3: Pokládání sklenic.....	67
3.4.4	Úloha č. 4: Pokládání sklenic do poliček rozdílné délky	73
3.4.5	Úloha č. 5: Pokládání sklenic na podložku	79
3.4.6	Úloha č. 6: Hladová myš	85
3.4.7	Úloha č. 7: Akvárium	90
3.4.8	Úloha č. 8: Had a padající jablka	96
3.5	Shrnutí poznatků akčního výzkumu	103
4	Závěr.....	105
	Seznam použitých informačních zdrojů	107
	Seznam příloh.....	116

1 Úvod

Od 1. 9. 2023 musely všechny základní školy na 1. stupni povinně zavést výuku informatiky podle revidovaného RVP ZV. Výuková oblast „Informační a komunikační technologie“, která byla zaměřená na používání technologií, byla nahrazena výukovou oblastí „Informatika“, která se zaměřuje na rozvoj informatického myšlení. Na mnoha školách vyučují na 1. stupni informatiku buď učitelé s aprobací pro 1. stupeň bez aprobace pro informatiku, nebo učitelé s aprobací pro informatiku, kteří ovšem nemají aprobaci pro 1. stupeň. Výuka informatiky na 1. stupni může proto přinést specifické výzvy pro učitele.

Tato diplomová práce se proto zabývá rozvojem informatického myšlení u žáků 5. ročníku. Jejím hlavním cílem je analyzovat informatické myšlení z hlediska základních konceptů a konkrétní cílové skupiny, žáků 5. ročníku, a na základě získaných poznatků sestavit soubor na sebe navazujících aktivit a úloh se zvyšující se obtížností v programovacím prostředí Scratch, které respektují a rozvíjí základní koncepty informatického myšlení.

Teoretická část práce je věnována samotnému zmapování problematiky Informatického myšlení, analýze vzdělávacích materiálů, programovacímu prostředí Scratch a odůvodnění jeho volby. Výsledky jednotlivých analýz jsou shrnuty do přehledných tabulek.

V rámci praktické části bude vytvořen soubor úloh, který se skládá z osmi na sebe navazujících úloh s postupně se zvyšující náročností. Jeho ověření v praxi metodou akčního výzkumu proběhne ve dvou cyklech vždy po 3 skupinách žáků. Výstupem šetření bude ucelený soubor úloh včetně doplňujících materiálů (ukázek řešení úloh, prezentací a pracovního listu) i metodických pokynů. Výstupem celé práce je také žákům určená webová stránka, na které budou publikovány úlohy a další materiály.

2 Teoretická část

2.1 Vymezení cílů a výzkumného pole

2.1.1 Výzkumný problém

Hlavním výzkumným problémem, kterým se práce zabývá, je otázka, jaké jsou možnosti rozvoje informatického myšlení u žáků 5. ročníku. Na jeho základě byly stanoveny následující dílčí výzkumné problémy.

- Dílčí výzkumný problém 1: Jaká jsou teoretická východiska informatického myšlení ve vzdělávání?
- Dílčí výzkumný problém 2: Jakými způsoby lze realizovat rozvoj informatického myšlení při respektování jeho konceptů během výuky informatiky pro žáky 5. ročníku?

2.1.2 Hlavní cíl

Hlavním cílem této diplomové práce je analyzovat informatické myšlení z hlediska základních konceptů a konkrétní cílové skupiny, žáků 5. ročníku, a na základě získaných poznatků sestavit soubor na sebe navazujících aktivit a úloh se zvyšující se obtížností v programovacím prostředí Scratch, které respektují a rozvíjí základní koncepty informatického myšlení.

Teoretická část je věnována analýze a komparaci různých pojetí pojmu „informatické myšlení“ a jeho konceptům. Dále se zabývá rozborem předpokladů zařazení informatického myšlení do výuky předmětu informatika na ZŠ.

Praktická část je věnována tvorbě uceleného souboru úloh pro žáky 5. ročníku, jeho ověření v praxi, a zahrnuje metodické pokyny pro učitele.

2.1.3 Dílčí cíle a úkoly

Dílčí cíl 1: Analyzovat pojem „informatické myšlení“.

Úkoly:

- Zmapovat pojem „informatické myšlení“ metodou komparace.
- Zmapovat základní koncepty informatického myšlení.

Dílčí cíl 2: Zmapovat předpoklady pro rozvoj informatického myšlení žáků 5. ročníku ZŠ.

Úkoly:

- Analyzovat RVP.

- Analyzovat učebnice informatiky pro žáky 5. ročníku.
- Analyzovat nástroje pro rozvoj informatického myšlení pro vybranou oblast předmětu Informatika.

Dílčí cíl 3: Sestavit ucelený soubor úloh pro žáky 5. ročníku, který rozvíjí informatické myšlení se zaměřením na jeho koncepty.

Úkoly:

- Navrhnout ucelený soubor úloh rozvíjející informatické myšlení se zaměřením na jeho koncepty v souladu s RVP.
- Vytvořit pro tyto úlohy zadání, řešení, metodické pokyny pro učitele (včetně popsání možných komplikací během výuky a návrhu jejich řešení), podpůrné materiály k úlohám.

Dílčí cíl 4: Prostřednictvím výzkumného šetření v praxi ověřit soubor úloh na několika skupinách žáků 5. ročníku.

Úkoly:

- Prostřednictvím akčního výzkumu ověřit úlohy a jejich metodiku.
- Během ověřování reflektovat získané poznatky a případně následně upravit úlohy i podpůrné materiály.
- Doplnit metodické podklady pro učitele na základě reflexe vzniklé během akčního výzkumu.
- Realizovat přístup k vytvořeným výukovým materiálům prostřednictvím webové stránky.

2.1.4 Metodika

V diplomové práci bylo využito jak empirických, tak teoretických výzkumných metod. V teoretické části jsou použity metody komparace a obsahové analýzy. Praktická část je postavena na empirické metodě akčního výzkumu.

Komparativní metoda

Komparativní metoda je forma zkoumání založená na sledování rozdílů a podobností mezi zkoumanými objekty. (Jandourek 2001, s. 158) Komparativní metoda zahrnuje tyto kroky: specifikaci objektu srovnávání, vymezení srovnávaných znaků, posouzení komparability,

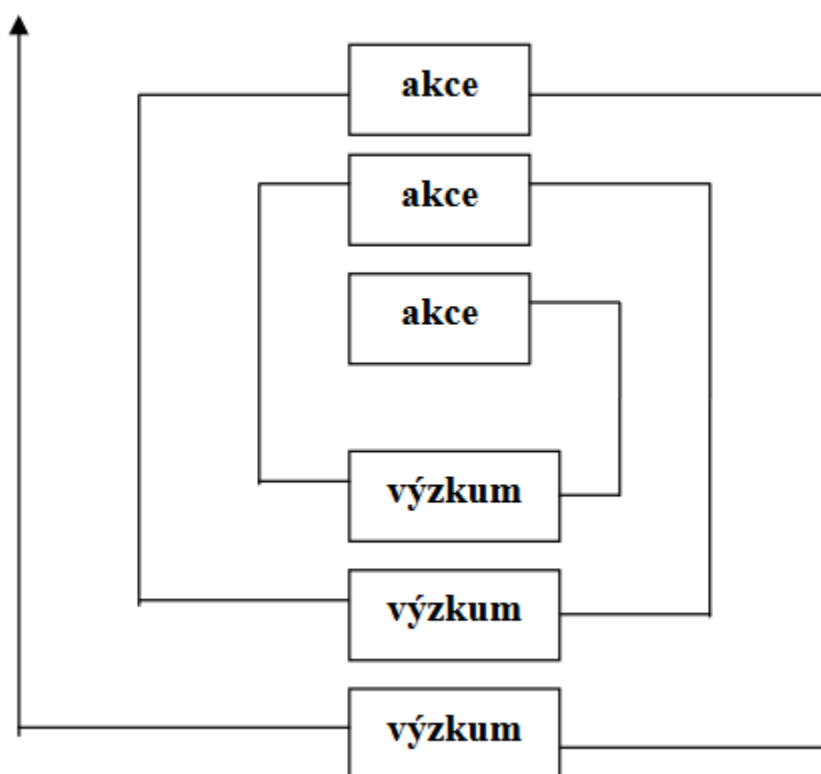
určení konkrétních technik srovnávání, určení způsobu zhodnocení získaných informací a systematiku výstupů. (Nešpor 2017a)

Obsahová analýza

Obsahová analýza patří mezi analytické metody „směřující k uspořádání a odhalování latentní informace v písemných materiálech nejrůznějšího druhu. Je postavena na objektivním, systematickém a kvantitativním popisu zjevného obsahu sdělení (podle B. R. Berelsona, O. Holstiho aj.)“. (Nešpor 2017b) „Jednotlivé analyzované elementy se pak seskupují do předem stanovených kategorií. Kategorizovaná data získaná obsahovou analýzou se kvalitativně i kvantitativně zpracovávají a srovnávají.“ (Nešpor 2017c)

Akční výzkum

Zvláštní místo v pedagogické praxi má akční výzkum. „Akční výzkum se chápe jako nástroj, který učitelům pomáhá řešit problémy školní praxe a zároveň uskutečňovat v této praxi inovace. Učitel se zde objevuje v obtížné dvojroli výzkumníka a aktivního účastníka akce zároveň“. (Janík 2003, s. 4)



Obrázek 1: Průběh akčního výzkumu (Janík 2003, s. 9)

Cílem akčního výzkumu je získávání nových poznatků a změna vlastního vyučování. Na předchozím obrázku je znázorněn cyklus průběhu akčního výzkumu. Výzkumník po provedení akce vyhodnotí její výsledky a na jejich základě pokračuje ve výzkumu. Výzkumné otázky se odvíjí z potřeb učitele a mohou se během výzkumu měnit. I samotný výzkumný vzorek může být nereprezentativní a nemusí být zajištěna jeho konzistentnost. (Janík 2003, s. 7-8). Akční výzkum je „považován za legitimní formu výzkumu za předpokladu, že zpráva je publikována a může být použita i v jiném akčním výzkumu“. (Trna 2011, s. 6) Vzhledem ke svým specifickým má akční výzkum široké uplatnění v pedagogické praxi, protože ve škole dochází k absencím žáků a jejich přechodu mezi školami, odpadlým hodinám, změnám kurikula...

Pro ilustraci je v příloze uvedena nejen tabulka porovnávací klasický a akční výzkum podle T. Janíka (2003), ale i tabulka podle Reevese, J. Trny (2011), která porovnává typologie výzkumných cílů v oborových didaktikách. (Příloha 1, Příloha 2, Příloha 3)

2.2 Informatické myšlení

2.2.1 Východiska pro vznik IM

Samotný pojem „informatické myšlení“ poprvé použil S. Papert již v 80. letech, v publikaci „Mindstorms Children, Computers, and Powerful Ideas“, ale s pojmem dále nepracoval. Informatické myšlení, jako pojem, se dostalo do povědomí veřejnosti až díky článku „Computational Thinking“, který v roce 2006 publikovala Jeannette Wingová. (Mannila a Settle et al. 2014, s. 2)

Snahy o zařazení metod a aktivit, které napomáhají rozvoji informatického myšlení do výuky, byly již před vznikem samotného pojmu. A. Yadav a E. N. Caeli (2019) ve své publikaci věnované historii informatického myšlení uvádí, „že již od 60. let odborníci debatují nad způsobem výuky informatiky“. (Yadav a Caeli 2019, s. 29) Ve své práci charakterizují dva hlavní přístupy k možnostem výuky informatiky v dané době. Jedni zastávali názor, že každý by se měl naučit programovat. Zatímco druzí považovali za stěžejní, aby studenti pochopili pojmy i princip tvorby algoritmů.

Profesor Alan Perlis již v roce 1960 zastával názor, že by se studenti měli učit chápat počítač spíše jako univerzální nástroj pro řešení problémů než jako speciální nástroj pro řešení pouze konkrétního typu problémů. (Yadav a Caeli 2019) Tuto myšlenku dále rozvinul a v roce 1962 tvrdil, že by všichni vysokoškolští studenti měli v rámci svého studia studovat i „theory of

computation“. Prvním krokem měla být výuka programování. Tvrdil, že programování je zkoumání procesů, což je téma, které se týká všech, a že automatizované provádění procesů strojem změní vše. (Guzdial 2008, s. 25)

Ve stejné době dánský profesor Peter Naur prosazoval myšlenku, že informatiku je zapotřebí učit již na základních školách. Informatiku, dánsky „datalogi“, chápal „jako mezipředmětovou dovednost vedoucí k řešení problémů“. (Yadav a Caeli 2019) V Dánsku byla proto v 70. a 80. letech na základních školách „datalogi“ zaváděna jako průřezové téma. Žáci se učili, jak pracovat s daty a jak tvořit algoritmy, a to bez použití počítače. (Yadav a Caeli 2019)

V roce 1967 profesor Massachusettského technologického institutu Seymour Papert představil první programovací jazyk cíleně vytvořený pro děti – Logo. Hlavní postavou je želva, která se pohybuje v písku a zvedáním a pokládáním ocásku kreslí. Děti programují pohyb želvy a kreslení do písku. (What is Logo? 2011) Programovací jazyk Logo měl přinést do škol nový typ myšlení, který S. Papert pojmenoval „procedurální“. Z pohledu konceptu procedurálního myšlení je znalost chápána jako „instrukce zapsaná v programovacím jazyce“. Měl to být protiklad k předchozímu typu myšlení, který S. Papert pojmenoval jako „výrokové“. Ve „výrokovém myšlení“ jsou znalosti chápány jako tvrzení, které může být buď pravdivé nebo nepravdivé. (Papert 2006)

Na konferenci Mathematics Education and Digital Technologies v Hanoji v roce 2006 S. Papert uvedl názor, že v 80. letech bylo příliš brzy zavádět počítače do škol, protože společnost na to ještě nebyla připravená. Zároveň uvedl, že jeho programovací jazyk Logo a koncept procedurálního myšlení byly prvními kroky vedoucí k obsáhlejšímu konceptu „informatického myšlení“. (Lodi a Martini 2021; Papert 2006)

2.2.2 Definice pojmu informatické myšlení

Definice informatického myšlení (computational thinking) od Jeannette Wingové

Svou vizi informatického myšlení (dále uváděné pod akronymem IM) Jeannette Wingová (2006) představila ve článku „Computational thinking“, který publikovala v roce 2006. J. Wingová považuje IM za základní dovednost, kterou by měli ovládat všichni a považuje ji za stejně důležitou jako schopnost číst, psát a počítat. V práci však neuvádí konkrétní definici, IM charakterizuje pouze prostřednictvím jeho vlastností. Vysvětluje, jaké dovednosti jsou pro IM podstatné a uvádí příklady využití v každodenním životě.

Vlastnosti IM podle J. Wingové:

- „IM je o navrhování řešení, ne o programování. „Computing“ je více než umět programovat, vyžaduje abstraktní uvažování na více úrovních.
- IM je základní dovednost potřebná pro život každého jedince v moderní společnosti, nejedná se pouze o rutinní činnosti.
- IM staví na lidském uvažování. Je to způsob, jakým lidé řeší problémy, a nejedná se o naučení člověka přemýšlet jako počítač. Nejedná se o mechanickou dovednost, jedná se o způsob uvažování.
- IM rozšiřuje a spojuje matematické a technické myšlení. Informatika čerpá poznatky z matematického a technického myšlení, její základy stojí na matematice. Jestliže omezená výpočetní kapacita počítačů nutí uživatele přemýšlet více informaticky než matematicky, virtuální systémy umožňují modelovat systémy za hranicemi fyzikálních možností.
- Pro IM jsou důležité nápady, nikoliv výtvořiny – důležitý není jen hardware a software, který vytvoříme, ale také naše schopnost využít informatické myšlení k řešení každodenních problémů.
- IM je obecná dovednost určená pro všechny, nejen pro omezenou skupinu odborníků.“

¹(Wing 2006, s. 35)

J. Wingová ve spolupráci s J. Cunym a L. Snyderem posléze v roce 2010 konkretizovala termín IM jako „soubor myšlenkových procesů, které se využívají pro formulaci problémů a jejich řešení tak, že postup řešení je vyjádřen v podobě, kterou lze efektivně zpracovat agentem zpracovávajícím informace.“² (Wing 2010, s. 1)

J. Wingová se svými spolupracovníky J. Cunym a L. Snyderem předpokládá, že IM umožní lidem:

- „Pochopit, které části problému může řešit počítač.
- Zhodnotit zda, případně jaká, výpočetní technika je vhodná k využití řešení daného problému.
 - Rozumět omezením a schopnostem výpočetní techniky a jejího využití.
- Využití nebo přizpůsobení výpočetní techniky pro nové účely.
- Rozpoznání nových příležitostí pro využití výpočetní techniky.

¹ vlastní překlad

² vlastní překlad

- Využití inforatických postupů jako „rozděl a panuj“ v jakémkoliv oboru.“³ (Wing 2010, s. 3)

Definice inforatického myšlení pro potřeby edukace

Vymezit definici pojmu IM není elementární, protože IM je relativně nový pojem, který nemá jednoznačné a pevně ukotvené pojetí. (Stewart a Baek 2023) Někteří odborníci dokonce argumentují, že IM vzhledem ke svému širokému záběru ani jednoznačnou definici nepotřebuje. (Selby a Woollard 2013, s. 2; Guzdial 2011) Tato nejednoznačnost bývá předmětem kritiky. (Denning 2017) Samotná definice je však důležitá pro potřeby školního vzdělávání, protože teprve na jejím základě může být IM zařazeno do kurikula. (Grover a Pea 2013; Selby a Woollard 2013; Shute a Sun et al. 2017)

Pro zjednodušení implementace IM do vzdělávání různé organizace (např. ISTE a CSTA, Královská společnost, MŠMT) vytvořily vlastní definice uzpůsobené pro potřeby tvorby vzdělávacích plánů pro základní a střední školy, proto se v této kapitole se podíváme na definice IM z hlediska edukace. Prameny byly vybrány tak, aby zahrnuly názory různých didaktiků informatiky na pojem IM.

V českém prostředí se Dan Lessner, spoluautor učebnice „Základy informatiky pro ŠS“ (Lessner a Lána et al. 2020), již v roce 2014 rozhodl přiblížit pojem IM odborné veřejnosti. Ve svém článku „Analýza pojmu Computational Thinking“ (Lessner 2014) představil různá pojetí IM. Lessner ovšem není jediný, kdo se u nás věnoval této problematice. I Bořivoj Brdička, významný odborník na využívání informačních technologií ve vzdělávání, přeložil do českého jazyka definici IM, kterou v roce 2014 vydala ISTE. (Brdička 2014) Oba tyto překlady definice se od sebe mírně liší, protože vnikly nezávisle na sobě a následně byly upraveny pro potřeby českého prostředí. Autorka proto pro upřesnění pojmu vycházela z originálních definic v anglickém jazyce a vytvořila vlastní překlad tak, aby se lépe porovnával s dalšími definicemi a pracemi publikovanými v anglickém jazyce.

MŠMT

V České republice bylo IM do vzdělávání zaváděno v rámci „Strategie vzdělávání do roku 2020“. Aby mohlo být IM implementováno do RVP, i u nás museli didaktici informatiky vytvořit definici IM. V dokumentu „Strategie vzdělávání do roku 2020“ byl pojem IM vysvětlen následovně:

³ vlastní překlad

„Informatické myšlení (computational thinking)–způsob uvažování, který používá informatické metody řešení problémů, a to včetně problémů komplexních či nejasně zadaných. Rozvíjí schopnost žáků analyzovat a syntetizovat, zevšeobecňovat, hledat vhodné strategie řešení problémů a ověřovat je v praxi. Vede k přesnému vyjadřování myšlenek a postupů a jejich zaznamenání ve formálních zápisech, které slouží jako všeobecný prostředek komunikace. Pracuje se základními univerzálními pojmy, které přesahují současné technologie: algoritmus, struktury, reprezentace informací, efektivita, modelování, informační systémy, principy fungování ICT.“ (MŠMT 2014, s. 48)

Imyšlení

Asi nejdostupnějším a nejvýznamnějším zdrojem informací o IM v ČR je webová stránka imysleni.cz. Tato stránka vznikla v rámci projektu PRIM (Podpora rozvoje informatického myšlení), který si klade za cíl *„podporovat změnu orientace školského předmětu informatika z uživatelského ovládání výpočetních technologií směrem k základům informatiky jako oboru“*. (O nás 2018) V rámci projektu je IM chápáno jako *„způsob myšlení, který se zaměřuje na popis problému, jeho analýzu a hledání efektivních řešení“*. (Co je to informatické myšlení 2018)

Nabízí nám sadu nástrojů a postupů. Když se s nimi seznámíme a naučíme se je používat, budeme je moci uplatňovat opakovaně a v různých situacích.

„Budeme umět například:

- *Systematicky posoudit různá řešení, vybrat to nejvhodnější pro danou situaci,*
- *rozdělit velký problém na několik menších, snáze řešitelných,*
- *plánovat a řídit činnosti,*
- *vytvářet a pečlivě popisovat postupy, které spolehlivě vedou k nějakému cíli, i když je vykonává někdo jiný,*
- *vybírat, které aspekty problému jsou podstatné pro jeho řešení a které lze zanedbat,*
- *uspořádat i velké a nesourodé soubory dat tak, abychom je mohli dále využít,*
- *používat jazyky, kterými se domluvíme s počítači, roboty a umělou inteligencí.“ (Co je to informatické myšlení 2018)*

Tento popis je pravděpodobně nejsrozumitelnější ze všech vybraných definic a na rozdíl od definice MŠMT je určen nejen školám, ale i rodičům a žákům, kteří by se chtěli blíže seznámit s novým pojetím výukové oblasti „Informatika“.

International Society for Technology in Education (ISTE) a Computer Science Teachers Association (CSTA)

ISTE je nezisková organizace, která se věnuje technologiím ve vzdělávání. Mezi její činnosti patří i vydávání „ISTE standardů“. ISTE ve spolupráci s CSTA, americkým profesním sdružení učitelů informačních technologií (CSTA), již roce 2011 zveřejnila svoji definici IM, která zahrnuje nejenom schopnost řešit problémy, ale zohledňuje i osobnost řešitele.

„Informatické myšlení je postup řešení problémů. Proces řešení zahrnuje (mimo jiné) tyto znaky:

- Problém je popsán způsobem, který umožňuje využití počítače a dalších nástrojů na jeho řešení,
- data jsou logicky organizována a analyzována,
- data jsou reprezentována pomocí abstrakce jako jsou modely a simulace,
- řešení je pomocí algoritmického myšlení automatizováno jako řada po sobě jdoucích kroků,
- hledání nejefektivnější kombinace činností a využití zdrojů prostřednictvím nalezení, analyzování a vyzkoušení různých způsobů řešení,
- zobecnění postupu řešení konkrétního problému a jeho využití pro řešení dalších problémů.

IM je dále ovlivněno a podpořeno řadou povahových vlastností řešitele. Mezi ně patří:

- Sebejistota při řešení složitých problémů,
- vytrvalost při práci na složitých problémech,
- schopnost práce s nejednoznačným zadáním a údaji,
- schopnost řešit problémy s otevřeným koncem,
- schopnost řešit otevřené problémy,
- schopnost komunikovat a spolupracovat s ostatními pro dosažení společného cíle nebo řešení.“⁴ (ISTE 2011)

Tuto definici nadále ISTE využívá ve standardu „ISTE Computational Thinking Competencies“ (Computational Thinking Competencies, nedatováno), ale na webových stránkách, které jsou určeny veřejnosti, je však definice značně zjednodušena:

⁴ vlastní překlad

„IM je základní gramotnost určená pro všechny žáky, která propojuje čtyři pilíře:

- rozkládání problému,
- hledání vzorů,
- abstrakci,
- algoritmizaci.

Zahrnuje automatizaci procesů pomocí zápisu řešení problémů jako řady po sobě jdoucích kroků.“⁵ (Computational thinking in classroom, nedatováno)

Královská společnost

Královská společnost v roce 2012 analyzovala v publikaci „Shut down or restart? The way forward for computing in UK schools“ stav výuky informatiky v britských školách. Autoři příručky navrhli další směry vývoje výuky informatiky ve Velké Británii a tento směr zahrnoval i rozvoj IM u žáků. (Královská společnost 2012)

„Proces rozpoznávání informaticky zpracovatelných vlastností objektů ve světě kolem nás a použití nástrojů a technik informatiky pro pochopení a analýzu jak v přírodních, tak i v uměle vytvořených systémech a procesech.“⁶ (Královská společnost 2012, s 29)

Souhrn kapitoly

Z množství definic je patrné, že vymezení pojmu IM není ustálené, ale všechny definice se shodují v tom, že pojímají IM jako způsob řešení problému. Většina definic dále zahrnuje i využití automatizace. Zatímco pojetí IM J. Wingové z roku 2012 je zaměřeno na efektivní využití výpočetní techniky při řešení problému, tak pojetí IM určené pro potřeby edukace se více zaměřuje na samotný proces řešení problému. Efektivní využití výpočetní techniky je chápáno spíše jako důsledek osvojení nového způsobu uvažování. Jako příklad mohou sloužit tyto definice:

- IMyšlení: „Používat jazyky, kterými se domluvíme s počítači, roboty a umělou inteligencí.“
- MŠMT: „Pracuje se základními univerzálními pojmy, které přesahují současné technologie.“

⁵ vlastní překlad

⁶ vlastní překlad

- ISTE – „Problém je popsán způsobem, který umožňuje využití počítače a dalších nástrojů na jeho řešení.“

Jedním z dílčích cílů této diplomové práce je vytvoření sady úloh pro rozvoj IM, proto musela autorka zvolit jedno pojetí IM. Autorka se přiklání k definici, ze které vycházejí odborníci z projektu PRIM a uvádějí ji na webových stránkách Imyšlení. Na IM je nahlíženo jako na „*způsob myšlení, který se zaměřuje na popis problému, jeho analýzu a hledání efektivních řešení*“ (Co je to informatické myšlení 2018) Definici považuje za stručnou, ale vystihující. Myslí si, že osvojení toho způsobu myšlení je pro děti přínosem.

2.2.3 Koncepty IM

Jak již bylo uvedeno výše, pro potřeby výuky a tvorby vzdělávacích plánů je třeba jasně vymezit pojem IM, má mít mezipředmětový přesah, a proto je potřeba upřesnit konkrétní dovednosti, které je třeba rozvíjet. Pro vybrané dovednosti a postupy se používají různé názvy. V odborné literatuře se pro vybrané dovednosti a postupy můžeme setkat například s výrazy: pilíře (Computational thinking in classroom, nedatováno), metody (Exploring Computational Thinking, nedatováno), základní dovednosti a koncepty (Bocconi a Chiocciariello et al. 2016, s. 16) anebo jenom koncepty (Jak rozvíjet informatické myšlení 2020). V českém prostředí se převážně používá termín „koncepty“, proto bude použit i v této práci. (Jak rozvíjet informatické myšlení 2020)

Vzhledem k tomu, že IM stále není zcela ukotvený termín, tak i při vymezení jeho konceptů nevládne mezi odborníky shoda. Jako jeden z mnoha příkladů nám může sloužit týdenní kurz pro studenty učitelství v rámci semináře „Learning and Motivation“. V rámci tohoto kurzu byly koncepty IM definovány takto: rozpoznání a dekompozice úkolu, abstrakce, logické myšlení, algoritmizace a debugging (Yadav a Zhou 2011), ale například C. Selby a J. Woollard (2013 s. 4) zavrhl koncept „logické myšlení“ jako nepřesný pojem.

C. Selby a J. Woollard

C. Selby a J. Woollard (2013), profesori Southamptonské univerzity, v roce 2013 publikovali výsledek své rozsáhlé práce. Autoři vyhledali z různých zdrojů velké množství definic IM. Tyto definice následně chronologicky seřadili a porovnali je. Z jednotlivých definic vybrali nejčastější pojmy a obsahově podobné pojmy přiřadili k sobě. Výsledky své práce poté zapsali do přehledné tabulky. Tabulku uvádím v originále, protože překladem by se mohl posunout význam pojmů:

Tabulka 1: Porovnání konceptů IM podle C. Shelby a J. Woollarda (C. Selby a J. Woollard 2013, s. 5)

A thought process	Include	Consensus found in the literature
Abstraction	Include	Consensus found in the literature
Decomposition	Include	Consensus found in the literature
Logical thinking	Exclude	Broad term, not-well defined
Algorithmic thinking	Include	Well-defined across multiple disciplines
Problem solving	Exclude	Broad term, evidences the use of skills; develops acquisition of skills
Evaluation	Include	Well-defined across multiple disciplines
Generalization	Include	Well-defined concept, although the term may not be familiar
Systems design	Exclude	Evidences the use of skills
Automation	Exclude	Evidences the use of skills
Computer science content	Exclude	Evidences the use of skills
Modeling, simulation, and	Exclude	Evidences the use of skills in their creation; manipulation develops acquisition of skills

Společné výzkumné středisko (JRC) Evropské komise

Pro potřeby Evropské komise vznikla publikace „Developing Computational Thinking in Compulsory Education“ (Bocconi a Chiocciariello et al. 2016), ve které skupina autorů porovnala základní koncepty IM. Autoři vybrali z prací věnovaných IM práce, které považovali za stěžejní. Kritéria výběru prací byla: vysoká míra citovanosti, jejich odkazování se na různorodé další studie a v neposlední řadě množství rozličných úhlů pohledu z hlediska oblastí výzkumu a mezinárodních pracovních skupin. (Bocconi a Chiocciariello et al. 2016, s. 17) Porovnání odborných publikací přineslo tyto výsledky:

Tabulka 2 Porovnání konceptů IM (Bocconi a Chiocciariello et al. 2016 s. 17)

Barr & Stephenson, 2011	Lee et al.,	Grover & Pea, 2013	Selby & Woollard, 2013	Angeli et al., 2016
Abstraction	Abstraction	Abstractions and pattern generalizations	Abstraction	Abstraction
Algorithms & procedures		Algorithmic notions of flow of control	Algorithmic thinking	Algorithms (including Sequencing and Flow of control)
Automation	Automation			
	Analysis			
		Conditional logic		
Problem Decomposition		Structured problem decomposition (modularizing)	Decomposition	Decomposition
		Debugging and systematic error detection		Debugging
		Efficiency and performance constraints	Evaluation	
			Generalizations	Generalizations
		Iterative, recursive, and parallel thinking		
Parallelization				
Simulation				
		Symbol systems and representations		
		Systematic processing of information		

Autoři, na základě srovnání vybraných publikací s článkem J. Wingové, ustanovili šest konceptů, kterým přiřadili tyto definice:

- **„Abstrakce (Abstraction):** Abstrakce je proces, po kterém se po odstranění zbytečných detailů stává objekt srozumitelnější. (Csizmadia et al., 2015, s. 7 citovaný v Bocconi a Chiocciariello et al. 2016 s. 18).
- **Algoritmické myšlení (Algorithmic thinking):** Algoritmické myšlení je způsob, kterým dospějeme k řešení problému prostřednictvím jasně definovaných kroků. (Csizmadia et al., 2015, s. 7 citovaný v Bocconi a Chiocciariello et al. 2016)

- **Automatizace (Automatization):** Automatizace je proces šetřící pracovní sílu, v jehož rámci počítač vykoná sled opakujících se příkazů tak, že je v porovnání s lidmi rychlejší a efektivnější. V tomto ohledu jsou počítačové programy „automatizací abstrakce“. (Lee, 2011, s. 33 citovaný v Bocconi a Chiocciariello et al. 2016 s. 18)
- **Dekompozice (Decomposition):** Dekompozice je způsob uvažování o objektech a problémech jako o souboru jejich jednotlivých částí. Jednotlivým částem lze snadněji porozumět, řešit je, rozvíjet je a lze je hodnotit odděleně. To značně usnadňuje hledání řešení komplexních problémů a práci s novými a velkými systémy. (Csizmadia et al., 2015, s. 8 v citovaný v Bocconi a Chiocciariello et al. 2016 s. 18).
- **Debugging:** Debugging je využití analýzy a vyhodnocování pro predikování a ověřování výsledků. Využívají se pro to postupy jako testování, trasování, logické uvažování. (Bocconi a Chiocciariello et al. 2016 s. 18)
- **Zobecnování (Generalization):** Zobecnování je postup zahrnující hledání vzorů, společných vlastností a souvislostí mezi objekty a problémy. Nalezené společné vlastnosti jsou dále využity. Umožňuje rychle řešit problémy na základě předchozích zkušeností s podobnými problémy. Algoritmy vytvořené pro jednu konkrétní úlohu se pak dají upravit pro celou skupinu úloh. Podobnosti se hledají jak ve využitých datech, tak v samotných postupech. (Bocconi a Chiocciariello et al. 2016 s. 18)⁷ (Bocconi a Chiocciariello et al. 2016 s. 18)

PRADA – koncepty pro edukační praxi

V rámci edukační praxe se často sekáváme s dalším pojetím konceptů IM. Tyto koncepty jsou převážně využívané v materiálech určených pro žáky základních škol a jejich učitele. Koncepty „PRADA“ se často využívají při tvorbě výukových materiálech pro žáky (např. Code.org (Computational Thinking, nedatováno)), ve vzdělávacích materiálech určených pro učitele (ISTE (Computational thinking in classroom, nedatováno), v kurzu „Jak rozvíjet inforatické myšlení“ (Kurzy Eduskop 2019)), využívají ho i vysoké školy (např. University of York (What is computational thinking?, nedatováno)). Uplatňují se i při zařazování IM do studia neinformatických oborů (Susanti a Taufik 2021; Anderson 2016), ale i v komerční sféře (např. Microsoft (Preview, nedatováno)).

Koncepty PRADA jsou:

- „Pattern Recognition (= hledání vzorů)

⁷ vlastní překlad

- Abstraction (= abstrakce)
- Decomposition (dekompozice)
- Algorithms (= algoritmizaci).“ (Computational thinking in classroom, nedatováno)

Tato čtveřice bývá někdy doplňována o „debugging“ nebo evaluaci.

Kdo přesně definoval tento model je obtížné dohledat. Nejstarší datovanou definici tohoto modelu se autorce podařilo dohledat ve vymezení IM společností Google. (*Exploring Computational Thinking 2010*) Samotný název „PRADA“ se poprvé objevil v článku „PRADA: A Practical Model for Integrating Computational Thinking in K-12 Education“ (Dong a Catete et al. 2019), který napsala skupina amerických vysokoškolských pedagogů a učitelů. Dnes i ISTE zakomponovala tyto koncepty do své definice IM. (Computational thinking in classroom, nedatováno)

Čím se tedy tento model liší od modelu vytvořeného odbornou pracovní skupinou pro Evropskou komisi?

- *Hledání vzorů* (Pattern recognition) lze považovat za modifikaci *zobecnování* (Generalization). Při srovnání těchto termínů vidíme, že *hledání vzorů* je pouze část *zobecnování*. V této zjednodušené formulaci se žáci zpočátku zaměří na hledání podobností na jednoduchých příkladech (např. cykly v programování). Dalším krokem toho principu je vyhledávání podobností při řešení zadané úlohy s úlohami již vyřešenými. *Hledání vzorů* je tedy možné vnímat jako „generalizaci pro děti“. Děti se napřed naučí hledat podobnosti na konkrétních pro ně známých věcech a postupně jsou vedeny k hledání podobností při volbě řešení problému.
- *Automatizace* (Automatization) jako „koncept“ v systému PRADA úplně chybí. Pokud však budeme s *automatizací* pracovat podle definice od ISTE, tak je *automatizace* zahrnuta do konceptu *algoritmizace*, protože „zahrnuje automatizaci procesů pomocí zápisu řešení problémů jako řady po sobě jdoucích kroků.“ (ISTE, nedatováno) Dále je možné předpokládat, že model PRADA bývá využíván při tvorbě výukových materiálů určených pro děti, ve kterých se žáci seznamují s novým stylem uvažování. Využití konceptu *automatizace* tak, jak jej pojala definice Evropské komise, vyžaduje porozumění možnostem a limitacím výpočetních technologií, pro které ještě děti nemusí mít potřebné znalosti.
- *Debugging* jako „koncept“ bývá v modelu PRADA často vynechán. Je však těžké si představit práci s *algoritmizací* a *abstrakcí* bez zpětné vazby, a tedy hledání chyb a

úpravy řešení. Zde je důležitá role učitele, aby žáky vedl k práci s chybou a hledání optimálního řešení.

Shrnutí kapitoly

V této práci vychází autorka z modelu PRADA, protože ho často využívají učitelé v praxi. Setkáváme se s ním na mnoha školeních i při různých aktivitách rozvíjejících infromatické myšlení. V neposlední řadě se při bližším prozkoumání tolik neliší od definice Evropské komise. Lze ho dokonce považovat za její zjednodušení. Následovně je interpretován model PRADA pro potřeby této práce:

- Hledání sekvencí a vzorů
 - Hledání sekvencí a vzorů je využito nejen při hledání vzorů v rámci jednoho řešení, ale i při hledání podobných způsobů řešení pro další problémy. Žáci si kladou otázky: „Opakuje se zde něco? Mohu zjednodušit řešení použitím cyklů? Viděl jsem již podobný problém? Jak jsem řešil podobný problém?“
- Dekompozice
 - Dekompozici lze chápat jako rozdělení složitého problému na dílčí podproblémy, které lze jednodušeji vyřešit. Žáci si kladou otázky: „Z jakých částí se skládá tento problém? Co už umím vyřešit? K čemu patří tato část problému?“
- Algoritmizace
 - Algoritmizace je způsob řešení problému, při kterém je jeho řešení popsáno jako posloupnost přesně daných kroků. Žáci si kladou například tyto otázky: „Jaké kroky/činnosti potřebuji k vyřešení problému? Jaké budou mít tyto kroky/činnosti pořadí?“
- Abstrakce
 - Abstrakce je proces odstraňování zbytečných detailů tak, aby zůstala jen podstata problému. Žáci si kladou například tyto otázky: „Potřebuji tuto věc vždy při řešení tohoto problému? Je tato věc součástí tohoto problému? Co se stane, když tuto věc odstraním?“

2.3 Informatické myšlení v rámci předmětu Informatika v 5. ročníku

2.3.1 Revize Rámcového vzdělávacího programu

„Rámcové vzdělávací programy (RVP) tvoří obecně závazný rámec pro tvorbu školních vzdělávacích programů škol. Do vzdělávání v České republice byly zavedeny zákonem č. 561/2004 Sb., o předškolním, základním, středním, vyšším odborném a jiném vzdělávání (školský zákon).“ (Rámcové vzdělávací program, 2023) Rámcový vzdělávací program pro základní vzdělávání (dále RVP ZŠ) je kurikulární dokument, který mimo jiné definuje pojetí a cíle základního vzdělávání, klíčové kompetence, které mají být rozvíjeny ve všech předmětech, a vzdělávací oblasti. RVP ZŠ dále určuje obsah výuky, vymezený očekávanými výstupy a časovou dotaci pro jednotlivé předměty. (Rámcový vzdělávací program pro základní vzdělávání 2023)

V roce 2021 vydalo MŠMT opatření č. j.: MSMT-40117/2020-4, kterým změnilo RVP ZŠ. Cílem změny bylo reflektovat vývoj informačních technologií a jejich využití ve společnosti.

Mezi cíle základního vzdělávání je nyní zařazen i cíl, zaměřený na smysluplné ovládnutí digitálních technologií. „Pomáhat žákům orientovat se v digitálním prostředí a vést je k bezpečnému, sebejistému, kritickému a tvořivému využívání digitálních technologií při práci, při učení, ve volném čase i při zapojování do společnosti a občanského života.“ (Rámcový vzdělávací program pro základní vzdělávání 2023, s. 9) Z tohoto cíle pak vychází i nová klíčová kompetence „kompetence digitální“. Mezi digitální kompetence patří vytváření a úprava digitálního obsahu, obsluha a výběr digitálních zařízení a aplikace, práce s informacemi a bezpečnost práce s digitálními zařízeními a daty. (Rámcový vzdělávací program pro základní vzdělávání 2023, s. 13)

Větší důraz na výuku informatiky reflektuje i navýšení minimální hodinové dotace pro obor Informatika. V Rámcovém učebním plánu je navýšen počet hodin na první stupni z jedné hodiny na dvě hodiny a na druhém stupni se nyní počítá se čtyřmi vyučovacími hodinami místo jedné.

Revize RVP přinesla i nový pohled na výukovou oblast, která se věnuje informatice a informačním technologiím. Změnil se nejen obsah, ale i samotný název vzdělávací oblasti. Výukovou oblast „Informační a komunikačních technologií“ nahradila oblast „Informatika“. Nyní se „vzdělávací oblast Informatika zaměřuje především na rozvoj informatického myšlení a na porozumění základním principům digitálních technologií. Je založena na aktivních

činnostech, při kterých žáci využívají infromatické postupy a pojmy.“ (Rámcový vzdělávací program pro základní vzdělávání 2023 s. 38) Žáci se učí pracovat s daty tak, aby byli schopni zvolit nejvhodnější konkrétní postup řešení problému a vybrat nejvhodnější technologii. Neučí se tedy pouze používat konkrétní nástroje, ale učí se přistupovat k problému komplexně, aby sami byli schopni volit vhodná technologická řešení v dnešním neustále technologicky se rozvíjejícím světě. Tomu by měly odpovídat i metody výuky. Podle RVP si žáci na prvním stupni *„prostřednictvím her, experimentů, diskusí a dalších aktivit vytvářejí první představy o způsobech, jakými se dají data a informace zaznamenávat“* (Rámcový vzdělávací program pro základní vzdělávání 2023, s. 39). Na druhém stupni žáci *„tvoří, experimentují, prověřují své hypotézy, objevují, aktivně hledají, navrhuji a ověřují různá řešení.“* (Rámcový vzdělávací program pro základní vzdělávání 2023, s. 39)

Vzdělávací obsah oboru Informatika je rozdělen do čtyř tematických okruhů:

- data, informace a modelování,
- algoritmizace a programování,
- informační systémy,
- digitální technologie.

Každý okruh má definovány očekávané výstupy, včetně minimálních výstupů pro žáky s IVP, a učivo.

Data, informace a modelování

Očekávané výstupy na konci 5. ročníku:

- *„I-5-1-01 uvede příklady dat, která ho obklopují a která mu mohou pomoci lépe se rozhodnout; vyslovuje odpovědi na základě dat*
- *I-5-1-02 popíše konkrétní situaci, určí, co k ní již ví, a znázorní ji*
- *I-5-1-03 vyčte informace z daného modelu*

Učivo

- *data, informace: sběr (pozorování, jednoduchý dotazník, průzkum) a záznam dat s využitím textu, čísla, barvy, tvaru, obrazu a zvuku; hodnocení získaných dat, vyvozování závěrů*
- *kódování a přenos dat: využití značek, piktogramů, symbolů a kódů pro záznam, sdílení, přenos a ochranu informace*

- *modelování: model jako zjednodušené znázornění skutečnosti; využití obrazových modelů (myšlenkové a pojmové mapy, schémata, tabulky, diagramy) ke zkoumání, porovnávání a vysvětlování jevů kolem žáka“ (Rámcový vzdělávací program pro základní vzdělávání 2023, s. 40)*

Algoritmizace a programování

Očekávané výstupy na konci 5. ročníku:

- *„I-5-2-01 sestavuje a testuje symbolické zápisy postupů*
- *I-5-2-02 popíše jednoduchý problém, navrhne a popíše jednotlivé kroky jeho řešení*
- *I-5-2-03 v blokově orientovaném programovacím jazyce sestaví program; rozpozná opakující se vzory, používá opakování a připravené podprogramy*
- *I-5-2-04 ověří správnost jím navrženého postupu či programu, najde a opraví v něm případnou chybu*

Učivo

- *řešení problému krokováním: postup, jeho jednotlivé kroky, vstupy, výstupy a různé formy zápisu pomocí obrázků, značek, symbolů či textu; příklady situací využívajících opakovaně použitelné postupy; přečtení, porozumění a úprava kroků v postupu, algoritmu; sestavení funkčního postupu řešícího konkrétní jednoduchou situaci*
- *programování: experimentování a objevování v blokově orientovaném programovacím prostředí; události, sekvence, opakování, podprogramy; sestavení programu*
- *kontrola řešení: porovnání postupu s jiným a diskuse o nich; ověřování funkčnosti programu a jeho částí opakovaným spuštěním; nalezení chyby a oprava kódu; nahrazení opakujícího se vzoru cyklem“ (Rámcový vzdělávací program pro základní vzdělávání 2023, s. 40)*

Informační systémy

. Očekávané výstupy na konci 5. ročníku:

- *„I-5-3-01 v systémech, které ho obklopují, rozezná jednotlivé prvky a vztahy mezi nimi*
- *I-5-3-02 pro vymezený problém zaznamenává do existující tabulky nebo seznamu číselná i nečíselná data*

Učivo

- *systemy: skupiny objektů a vztahy mezi nimi, vzájemné působení; příklady systémů z přírody, školy a blízkého okolí žáka; části systému a vztahy mezi nimi*
- *práce se strukturovanými daty: shodné a odlišné vlastnosti objektů; řazení prvků do řad, číslovaný a nečíslovaný seznam, víceúrovňový seznam; tabulka a její struktura; záznam, doplnění a úprava záznamu“ (Rámcový vzdělávací program pro základní vzdělávání 2023, s. 41)*

Digitální technologie

Očekávané výstupy na konci 5. ročníku:

- *„I-5-4-01 najde a spustí aplikaci, pracuje s daty různého typu*
- *I-5-4-02 propojí digitální zařízení, uvede možná rizika, která s takovým propojením souvisejí*
- *I-5-4-03 dodržuje bezpečnostní a jiná pravidla pro práci s digitálními technologiemi*

Učivo

- *hardware a software: digitální zařízení a jejich účel; prvky v uživatelském rozhraní; spouštění, přepínání a ovládání aplikací; uložení dat, otevírání souborů*
- *počítačové sítě: propojení technologií, (bez)drátové připojení; internet, práce ve sdíleném prostředí, sdílení dat*
- *bezpečnost: pravidla bezpečné práce s digitálním zařízením; uživatelské účty, hesla“ (Rámcový vzdělávací program pro základní vzdělávání 2023 s. 41)*

2.3.2 Výukové materiály pro vzdělávací oblast Informatiky pro 5. ročník

Výukové materiály

Mezi nejběžnější výukové materiály ve škole patří učebnice, které jsou *„druhem knižní publikace uzpůsobené k didaktické komunikaci svým obsahem a strukturou“*. (Průcha a Walterová 2008, s. 272) Na učebnice se lze dívat jako na součást kurikula, *„vymezují a prezentují vybrané obsahy vzdělávání“* (Janko, nedatováno, s. 3), nebo také jako na didaktický prostředek, *„usměrňují učení žáků a výukovou činnost učitele; zdroj obsahu vzdělávání pro žáky“*. (Janko, nedatováno, s. 3) Posuzování kvality učebnic se věnuje *„analýza učebnic“*, v rámci které se posuzuje struktura didaktického textu, jeho obsah, rozsah, obtížnost, didaktická vymezenost. (Průcha a Walterová 2008, s. 272) Kromě klasických učebnic se ve škole stále

častěji setkáváme i s dalšími druhy didaktických textů, protože za didaktické texty lze považovat všechny tištěné a elektronické materiály, které jsou uzpůsobeny pro výuku. (Průcha a Walterová 2008, s. 81)

Pro následnou analýzu vzdělávacích materiálů byly vybrány učební materiály, které jsou dostupné široké veřejnosti, mohou sloužit nejen učitelům pro výuku, ale i žákům pro domácí přípravu. Učebnice a online portály byly vyhledávány na webových stránkách MŠMT, nakladatelství zaměřených na učebnice a vyhledáváním učebních materiálů na internetu. Byly zařazeny pouze ucelené materiály.

Byla provedena analýza všech tištěných učebnic dostupných na trhu a vybraných elektronických materiálů. Analyzováno bylo celkem 12 materiálů. Pro snadnější orientaci byly tištěné výukové materiály zařazeny do skupin podle nakladatelství, protože učebnice tvoří jednotnou řadu. Elektronické výukové materiály byly rozděleny podle webu, na kterém byly publikované. Jedná se o výukové materiály vzniklé během projektu „Podpora rozvíjení informatického myšlení“ (PRIM) a materiály z www.umimeinformatiku.cz. Výukové materiály publikované na stránkách Imyšlení vzniklé během projektu „Podpora rozvíjení informatického myšlení“ (PRIM), na kterém spolupracovaly české pedagogické fakulty, byly brány za jeden celek.

Analyzované výukové materiály

- **Computer Media:** Informatika pro 5. ročník od Pavla Navrátila (2024)
- **Fraus:** Hybridní pracovní učebnice Informatika 2 Uffi a Uffi (Agh, 2023) a Hybridní pracovní sešit Hledá se Puffy (Agh, 2022) od Petra Agha
- **H-Edu:** Programování s Emilem 1. díl – pracovní sešit a Programování s Emilem 2. díl – pracovní sešit kolektivu autorů Ivana Kalaše, Andreje Blaha, Milana Moravčíka (Informatika s Emilem a Robotika s Emou, 2023)
- **iMyšlení:**
 - Robotika s LEGO WeDo pro 1. stupeň základní školy od autorů Josefa Procházky, Jakuba Lapeše a Daniela Tocháčka (2020)
 - Základy programování ve Scratch pro 5. ročník základní školy od autorů Ivana Kalaše a Karolíny Míkové (2020)
 - Práce s daty pro 5. až 7. ročník základní školy od autorů Zbyňka Filipiho, Denise Mainze A Jana Fadrhonce (2020)
 - Základy informatiky pro 1. stupeň ZŠ od autorů Jana Berkiho a Jindry Drábkové (2020)

- **Nová škola Duha:** Informatika 5 učebnice pro 5. ročník základní školy od Jany Morbacherové (2023)
- **Taktik:** Informatika v pohodě 5 kolektivu autorů Marcely Hrnčířkové, Davida Bally, Pavly Enevové a Miroslava Hynka (Hrnčířková a Balla et al. 2023)
- **Umíme informatiku:** autory úlohy jsou Tomáš Kebert, Tomáš Effenberger a Adéla Štěpková (Autoři projektu)

Analýza materiálů a sledovaná kritéria

Výukové materiály byly analyzovány ze tří hledisek: formálního hlediska, obsahového hlediska a ekonomického hlediska.

Sledovaná kritéria:

1. Formální hledisko:
 - obrazová složka – ilustrace a přehlednost,
 - verbální složka – přizpůsobení slovní zásoby a délky vět cílové skupině.
2. Obsahové hledisko
 - RVP:
 - pokryté oblasti,
 - metody výuky odpovídají RVP – žáci se vzdělávají prostřednictvím her, experimentů, diskusí a dalších aktivit. (Rámcový vzdělávací program pro základní vzdělávání 2023 s. 38)
 - IM :
 - vhodnost didaktických úloh a textů pro rozvoj IM.
3. Ekonomické hledisko:
 - pořizovací cena a další náklady spojené s výukou,
 - doplňující materiály – metodická příručka, modelové ŠVP, doplňující online úlohy a další materiály.

Výsledky analýzy

Na základě prostudovaných vybraných materiálů byla po posouzení jednotlivých hledisek vytvořena přehledná tabulka.

Formální hledisko

Formální hledisko se zaměřuje na přehlednost a srozumitelnost materiálu pro cílovou skupinu – žáky 5. ročníku.

Tabulka 3: Rozbor výukových materiálů z formálního hlediska

Formální hledisko		
	Obrazová složka	Verbální složka
Computer Media	<ul style="list-style-type: none">• přehledná (velmi přehledně zadané úkoly)• velká míra použití tabulek	<ul style="list-style-type: none">• přiměřená délka vět a slovní zásoba
Fraus	<ul style="list-style-type: none">• velmi přehledná• jednotná grafická úprava	<ul style="list-style-type: none">• přiměřená délka vět a slovní zásoba• vysvětlení odborných pojmů (Puffy: str. 32 „Může čára klesnout – jde dolů“)
H-Edu	<ul style="list-style-type: none">• přehledná• výrazné barvy	<ul style="list-style-type: none">• přiměřená slovní zásoba a délka vět
iMyšlení	<ul style="list-style-type: none">• přehledná• ilustrace – především funkční	<ul style="list-style-type: none">• slovní zásoba obsahuje i složitější pojmy, které jsou vysvětleny pomocí názorných ilustrací
Nová škola Duha	<ul style="list-style-type: none">• přehledná• výrazné barvy	<ul style="list-style-type: none">• přiměřená délka vět a slovní zásoba
Taktik	<ul style="list-style-type: none">• přehledná• některé ilustrační obrázky neodpovídají textu (str. 25 uživatelské rozhraní Scratch)	<ul style="list-style-type: none">• slovní zásoba neodpovídá věku žáků (str. 26 „Pro lepší orientaci jsou rozděleny podle svých charakteristik do barevných sekcí.“), dlouhá souvětí• faktické chyby při použití odborných termínů (např. str. 22 záměna pojmu „číslice“ a „číslo“)
Umíme informatiku	<ul style="list-style-type: none">• přehledná	<ul style="list-style-type: none">• některé úlohy mohou mít z pohledu žáků 1. stupně nejasně formulované zadání• některá cvičení obsahují příliš širokou slovní zásobu (např. úloha „společné vlastnosti slova“)

Obsahové hledisko

Obsahové hledisko se zabývá obsahem nejen z hlediska RVP (pokrytí a vhodnost úloh), ale i z hlediska vhodnosti pro rozvoj IM.

Tabulka 4: Rozbor výukových materiálů z obsahového hlediska – RVP

Obsahové hledisko: RVP						
	Data, informace, modelování	Algoritmy a programování	Informační systémy	Digitální technologie	Výuka prostřednictvím her, experimentů, diskusí a dalších aktivit	Doložka MŠMT
Computer Media	ano	částečně	ano	okrajově	ano	ne
Fraus	ano	ne	ano	částečně	ano – i desková hra	ano
H-Edu	okrajově – v rámci algoritmy a programování	ano	okrajově – v rámci algoritmy a programování	okrajově – v rámci algoritmy a programování	ano	ano
iMyšlení	ano	ano	částečně	okrajově	ano	ano
Nová škola Duha	ano	ano	ano	ne	ano – především diskuze	ne
Taktik	ano	ano	ano	okrajově	ne – většinou výklad	ne
Umíme informatiku	ano	ano	ano	okrajově	ano – především hry a experimenty	ne

Tabulka 5: Rozbor výukových materiálů z obsahového hlediska – Rozvoj IM

Obsahové hledisko: rozvoj IM		
	Deklarace	Naplnění
Computer Media	ne	většina úloh vede k rozvoji IM
Fraus	ano (Puffy)	naplňuje
H-Edu	ano (stránky nakladatelství)	naplňuje
iMyšlení	ano (imysleni.cz/ucebnice)	naplňuje
Nová škola Duha	ne	<p>částečně naplňuje</p> <ul style="list-style-type: none"> • některé úlohy jsou totožné s úlohami z iMyšlení, mají pouze rozdílnou grafiku (např. str. 13/cv. 6 převzato z „Práce s daty“, cvičení „Hudební nástroje“ – nástroje nahrazeny květinami; str. 17/cv. 2 převzato ze Základy informatiky str. 33) • skoro celá kapitola „Algoritmizace a programování“ parafrázována ze „Základy práce v Scratch“ ze „Základy programování ve Scratch pro 5. ročník základní školy“.
Taktik	ne – deklaruje zaměření především na informační technologie	většina učebnice obsahuje především výklady a návody
Umíme informatiku	ano (úvodní video)	naplňuje

Ekonomické hledisko

V neposlední řadě je hodnoceno hledisko ekonomické, tedy náklady spojované s pořízením materiálů. Dále také posuzuje i to, zda jsou k dispozici další rozšiřující materiály (např. metodické příručka, rozšiřující úlohy, ...).

Tabulka 6: Rozbor výukových materiálů z ekonomického hlediska

Ekonomické hledisko		
	Dostupné materiály	Celkové náklady
Computer Media	učebnice, metodická příručka; materiály do hodiny ke stažení (videa, pracovní listy k tisku) – nutno vlastnit příručku	vše nutné zakoupit
Fraus	učebnice, metodická příručka, materiály ke stažení, rozšiřující materiály, v učebnici odkazy na online úlohy k procvičení, desková hra v učebnici, modelové ŠVP	placená pouze učebnice
H-Edu	učebnice, metodická příručka, kurz, SW a pracovní sešit	vše placené (možné zakoupit až po absolvování placeného kurzu)
iMyšlení	učebnice, metodická příručka, pracovní listy, soubory Scratch k zadáním, modelový ŠVP	vše zdarma a online
Nová škola Duha	učebnice, klíče ke cvičení, soubory k úlohám Scratch	placená pouze učebnice
Taktik	učebnice, v učebnic jsou QR kódy s odkazy na web s doplňujícími a rozšiřujícími informacemi, soubory k úlohám ve Scratch	placená pouze učebnice
Umíme informatiku	soubor online cvičení, stručný výklad k tématu	zdarma: omezený počet úloh za den; zpoplatněno: neomezený přístup, sledování výsledků žáků, ...

Shrnutí kapitoly

Z pohledu autorky se zdá, že většina učebnic může najít své uplatnění v hodinách informatiky, pouze učebnice od Taktiku se vymyká svým zpracováním. Autorka je toho názoru, že už jen slovní zásoba a prezentace učiva může být pro žáky 5. ročníku náročná sama o sobě. Učební materiály nakladatelství H-Edu bohužel neměla autorka možnost prostudovat do hloubky. Samotná myšlenka, že pedagog bude proškolen pro práci s učebnicí, se však autorce velmi zamlouvá. Bohužel finanční náročnost celého produktu může školy odradit od volby tohoto materiálu. Pro učitele, kteří dávají přednost tištěnému materiálu, učebnice od nakladatelství Fraus a Computer Media jistě najdou své uplatnění v hodinách. Samotná sada materiálů z iMyšlení je však schopna plně pokrýt požadavky na výuku informatiky pro 5. ročník. Materiály jsou přístupné zdarma na webových stránkách nejen učitelům, ale i rodičům a dětem, a to nejen ve škole, ale i doma.

2.3.3 Nástroje pro výuku oblasti „Algoritmizace a programování“

Jeden ze stěžejních konceptů IM je algoritmizace. Je jí věnovaná i celá oblast RVP „Algoritmizace a programování“. Algoritmizace je úzce spojena s programováním, které lze dobře využít i pro rozvoj ostatních konceptů IM, proto se žáci již na 1. stupni učí programovat.

Programování je způsob zadávání pokynů elektronickým zařízením k provádění úkolů, k řešení problémů a ke komunikaci s lidmi. Základním principem programování je rozdělení problému na malé části, které počítač může interpretovat a zpracovat. (Bebbington, nedatováno) Pro zapsání programu je potřeba použít programovací jazyk, který můžeme definovat jako „*umělý jazyk určený pro zápis počítačového programu. Syntaxe a sémantika jsou přesně definovány*“. (Vitovský, 2006)

Výuka programování žáků základní školy, zvláště žáků 1. stupně, přináší svá specifika. Velmi důležitým krokem je nejen samotný výběr vhodného programovacího jazyka, ale i programovacího prostředí.

Programovací jazyky používané v praxi (C, C++, Java, ...) se příliš nehodí pro výuku na 1. stupni, protože mívají složité vývojové prostředí, většinou vyžadují instalaci různých komponent (kompilátor, ...) a podpora češtiny je omezená. Jen samotné ovládnutí programovacího prostředí může být pro žáky velmi obtížné. Tento proces může být tak náročný, že se může stát hlavní náplní výuky místo výuky samotné algoritmizace.

Otázkou volby vhodného nástroje pro programování se zabývá i „pionýr“ blokových programovacích jazyků pro děti A. Repenning. (Prof. Dr. Alexander Repenning, nedatováno)

A. Repenning uvádí, že jedním z problémů při výuce programování je, že programování je pro děti obtížné a nudné, což je zejména spojeno s běžnými programovacími jazyky. Děti chtějí programovat zejména hry nebo roboty, programování matematických úloh je pro ně nezajímavé. Náročnost tvorby programu lze snížit využitím blokových programovacích jazyků. V blokovém programování se kód skládá z bloků, které nahrazují psaný kód. Program je tvořen za pomoci bloků jako puzzle. V jiných formách vizuálního programování se dětem velmi obtížně pracuje a často nelze vytvářet abstraktnější koncepty (parametry, funkce, opakované využití již vytvořeného kódu, ...). (Repenning 2017, s. 68-72)

A. Repenning uvedl 4 hlavní koncepty pro blokový programovací jazyk:

- „Uživatel může bloky skládat do programu.
- Uživatel může bloky upravovat (bloky mají nastavitelné parametry).
- Bloky lze vnořit do jiných bloků.
- Syntax je tvořen lineárním uspořádáním bloků.“ (Repenning 2017, s. 74)

V neposlední řadě A. Repenning rozlišuje nástroje pro programování (programming tools) a nástroje pro informatické myšlení (computational thinking tools). Nástroje pro programování se hodí pro profesionální využití a obecně se hodí pro složitější úlohy, ale nehodí se pro jednoduché úlohy, které lze naopak využít pro výuku IM ve škole. Nástroje pro IM se hodí právě pro tyto jednoduché úkoly s grafickým výstupem, ale obtížně se jimi řeší komplexní úlohy v profesionální sféře. (Repenning 2017, s. 82 – 85)

Mezi oběma typy nástrojů existuje překryv, vhodně zvolený nástroj však usnadní práci jak učitelům, tak žákům, a proto je třeba výběr nástroje pečlivě zvažovat. Výběr nástroje by měl vycházet ze vzdělávacích cílů, potřeb žáků a možností školy. Pro potřeby této práce proto autorka stanovila následující kritéria pro volbu vhodného nástroje.

Sledovaná kritéria:

- Dostupnost:
 - minimální finanční náročnost pro školu,
 - česká lokalizace,
 - vytvořené úlohy budou dostupné žákům i pedagogům bez nutnosti se registrovat,
 - žáci mohou nástroj využít i mimo výuku, nejlépe i na svém mobilním zařízení.
- Pedagogické požadavky:
 - vhodný pro rozvoj IM,

- atraktivnost pro žáky,
- dává prostor pro kreativitu žáků i učitelů,
- umožňuje tvorbu „zábavných“ projektů (her, animací).
- Přiměřená složitost:
 - žáci nemusí věnovat několik vyučovacích hodin studiu programovacího prostředí a jazyku, než budou schopni vytvořit „viditelný“ výstup,
 - lze vytvořit výstup během jedné vyučovací hodiny.

Výsledky srovnání vybraných nástrojů

Na základě analýzy potřeb žáků a typů nástrojů pro výuku algoritmizace byl výběr zúžen na blokové programovací jazyky. Do výběru nebyly zahrnuty robotické hračky ani jiný programovatelný hardware. Programovatelné robotické hračky a roboti mohou být dobrým rozšířením vizuálních programovacích jazyků, mají však značná omezení v dostupnosti. Jsou s nimi spojené potenciálně značné finanční náklady (podle počtu žáků), je nutné zabránit ztrátám a poškození hardwaru. Údržba a správa mohou být další překážkou pro školu. Proto byla tato kategorie vyřazena.

Vybrané nástroje:

- **Programování s Emilem:** Software Programování s Emilem je součástí výukových materiálů pro výuky informatiky na 1. stupni vydaných nakladatelstvím H-Edu. Materiály mají doložku MŠMT a dle informací na e-shopu nakladatelství napomáhají rozvoji IM. (Informatika s Emilem a Robotika s Emou, 2023) Všechny materiály i aplikace jsou placené a je možné je zakoupit až po absolvování placeného školení, proto není možné software blíže prostudovat. Autorka dále vychází z informací uvedených na webových stránkách nakladatelství.
- **MakeCode:** Jedná se o blokový programovací jazyk vytvořený společností Microsoft. Bývá spojován s hardwarovou platformou micro:bit a Minecraft for Education, ale lze jej využít i bez nutnosti pořizování speciálního softwaru nebo hardwaru. (Microsoft MakeCode, nedatováno)
- **Scratch:** Je blokový programovací jazyk pro výuku programování, vytvořený Massachusettským technologickým institutem. (O Scratchi, nedatováno) Jedná se o nejpopulárnější blokový programovací jazyk, v žebříčku TIOBE se umístil na 15. místě. (TIOBE Index for March 2024)

- **mBlock:** Je nadstavba Scratch vytvořená čínskou společností Makeblock. MBlock umožňuje programovat robotické hračky od stejné společnosti (Codey Rocky, mBot, ...) nebo i jiných společností (LEGO Mindstorm, Arduino, ...), samostatné hry a animace. Podporuje nejen blokové programování ale i Python. (MBlock, nedatováno)
- **Snap:** Je další nadstavba Scratch s různorodější sadou příkazů, včetně příkazů napodobujících syntax běžně používaných programovacích jazyků. Poskytuje větší možnosti při tvorbě kódu, včetně tvorby vlastních bloků. (Snap! Build Your Own Blocks, nedatováno)

Dostupnost:

- **Programování s Emilem:** Software je nutné zakoupit, ale výhodou je, že aplikace je v českém jazyce.
- **MakeCode:** MakeCode je zdarma a v české lokalizaci, ale pro využití s platformou micro:bit nebo v Minecraft for Education je zapotřebí zakoupit příslušné zařízení nebo licenci softwaru. Žáci mohou vytvářet vlastní hry za použití nástroje Arcade Games, který je dostupný zdarma online a nevyžaduje žádný další hardware. Bohužel tento nástroj není přeložený do českého jazyka.
- **Scratch:** Je zdarma, dostupný ve verzi online i offline. Obě dostupné verze obsahují českou lokalizaci.
- **mBlock:** Nástroj není nutné kupovat, má verzi online i offline a má českou lokalizaci. Přestože mBlock je rozšíření Scratch, mají rozdílný překlad do českého jazyka.
- **Snap:** Nástroj je dostupný zdarma, nemá úplný český překlad.

Atraktivnost pro žáky:

- **Programování s Emilem:** Dle ukázek na webových stránkách nakladatelství se jedná o programování postavy. Pravděpodobně neumožňuje tvorbu her a animací. Lze předpokládat, že pro děti nebude tak atraktivní jako ostatní nástroje.
- **MakeCode:** Micro:bit i Minecraft for Education nabízí možnost vytváření zajímavých úloh. V Minecraft for Education navíc žáci programují činnosti, které si mohou sami vyzkoušet a mohou pro ně mít smysl. Minecraft je však pro některé žáky až příliš atraktivní a žáci bez zájmu o programování mají tendenci si hrát, nepracovat a rušit ostatní žáky.

- **Scratch:** Scratch je zaměřen na tvorbu animací a her. Zahrnuje vývojové prostředí a sadu obrázků, které lze využít jako grafiku. Žáci si mohou zahrát i již vytvořené hry a prohlédnout si jejich kód, což může žáky motivovat k vlastní tvorbě her.
- **mBlock:** Umožňuje programovat velké množství robotických hraček a programovatelného hardwaru. Lze programovat hry a animace bez dalších hardwarových platforem.
- **Snap:** Umožňuje programování her a animací. Oproti Scratch a mBlock neobsahuje předpřipravené grafické prvky pro tvorbu animací a her. Žáci budou méně motivováni k vlastní tvorbě.

Náročnost pro žáky

- **Programování s Emilem:** Nelze ověřit.
- **MakeCode:** MakeCode je oproti dalším vybraným nástrojům flexibilnější, umožňuje tvorbu detailnějšího kódu. Okamžitá dostupnost mnoha funkcí a nástrojů však může vést k problémům s pochopením vytvářeného kódu a vést k použití metody pokus-omyl bez hlubšího pochopení problematiky. Dalším nedostatkem je způsob organizace kódu. Zatímco ve Scratch je základní jednotkou objekt, ke kterému se přiřadí kód, v MakeCode je pouze jedna pracovní plocha, ve které se tvoří kód a za jeho pomoci se přidávají objekty (postavy, pozadí, ...). Pro žáky 1. stupně se může jednat o náročnější nástroj.
- **Scratch:** Bloky jsou připraveny tak, že v nich nelze udělat syntaktické chyby. Základní jednotka je jeden objekt (postava nebo pozadí), ke kterému se vztahuje kód. Rozdělení na programovatelné objekty je pro žáky zpravidla jednodušší na pochopení a na rozdíl od Make:Code nedochází k vzniku „neexistujících objektů“.
- **mBlock:** MBlock má v porovnání se Scratch větší množství příkazů (např. „open url“), postav a pozadí. Rozdílné je i grafické rozhraní. Nejspíše proto, že mBlock je více zaměřen na programování robotů, tak má větší plochu pro kód na úkor prostoru pro ukázkou vytvořené aplikace, postav a pozadí. Více se hodí pro práci s robotickými hračkami, pro které má aplikace i samostatnou záložku. Pokud chceme učit děti programovat bez použití robotických hraček, tak mBlock obsahuje velké množství postav, pozadí a dalších příkazů, které by mohly odvádět pozornost dětí.
- **Snap:** Patří mezi náročnější nástroje (viz výše), ale velkou výhodou je možnost omezit zobrazované příkazy pro vypracování projektu.

Vhodnost pro rozvoj IM

- **Programování s Emilem:** Dle informací na e-shopu nakladatelství produkt napomáhá rozvoji IM. (Programování s Emilem 1. díl – pracovní sešit, 2023)
- **MakeCode:** Jedná se o nástroj s úzkým zaměřením. Existuje více variant, každá je specializovaná na konkrétní hardwarovou nebo softwarovou platformu (micro:bit, Minecraft, hry v MakeCode Arcade, ...). Možnosti rozvoje IM se u různých variant liší. Obecně se více zaměřuje na vytvoření řešení úlohy než na pochopení problému, ale pro rozvoj IM je pochopení prioritou.
- **Scratch:** Pro rozvoj IM je z hlediska autorky vhodný koncept bloků Scratch. Dokonce někteří pedagogové uvádějí, že Scratch je více zaměřený na výuku inforatického myšlení než na výuku programování. (Comparison Between Different Block-based Programming Platforms 2021) Rozdělení programu na objekty, postavy a pozadí vede žáky intuitivně k rozdělení větších problému na dílčí části. Scratch je obecně zaměřený na tvorbu her a animací, žáky vede k rozboru úlohy a prostřednictvím analýzy dostupných bloků k jejímu řešení.
- **mBlock:** MBlock je možné využít pro rozvoj IM podobně jako Scratch. Více se zaměřuje na práci s robotickými hračkami.
- **Snap:** Podobně jako Scratch, ze kterého vychází, Snap je vhodným nástrojem pro rozvoje IM, ale vzhledem k neúplné české lokalizaci a značné robustnosti není vhodným nástrojem pro výuku na prvním stupni. Autoři sami uvádí, že jejich nástroj je vhodný pro výuku studentů na střední a vysoké škole. (Mönig a Harvey 2020)

Shrnutí kapitoly

Na základě stanovených kritérií byly vybrány nástroje „Scratch“ a „mBlock“, které jsou dostupné, jednoduché pro pochopení, umožňují vytvářet vlastní hry a dávají velký prostor kreativitě. Z těchto nástrojů byl upřednostněn Scratch. Svým jednodušším uživatelským rozhraním a menším množstvím předpřipravených grafických objektů byl vyhodnocen jako vhodnější pro žáky 1. stupně.

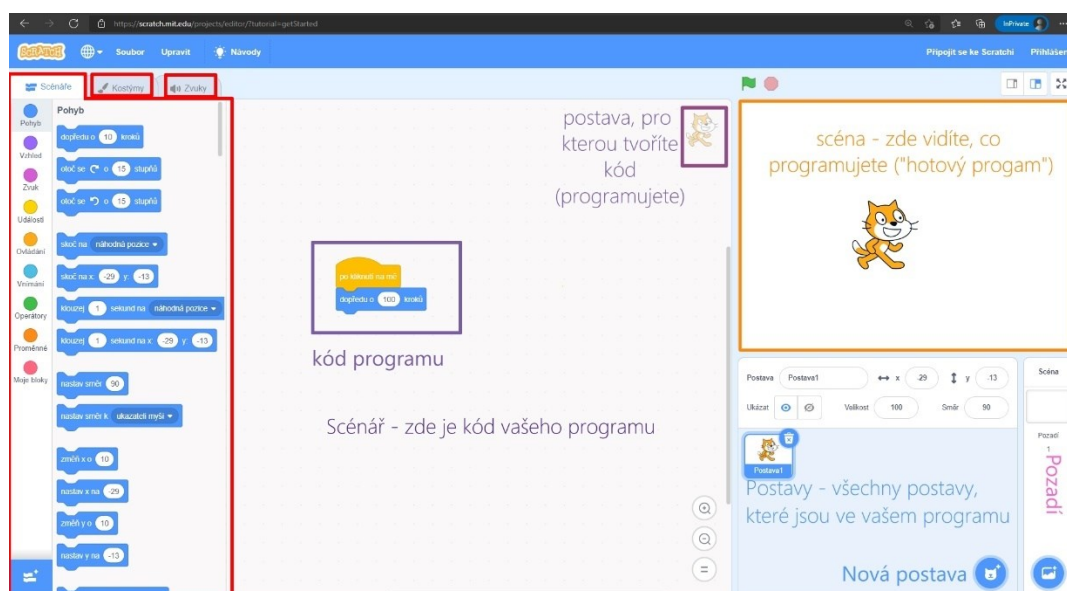
2.3.4 Scratch

Scratch je vizuální programovací jazyk určený především pro výuku programování dětí a mládeže. Byl vytvořen v Media Lab Massachusettského technologického institutu a jeho vznik financovala Národní vědecká nadace Spojených států amerických. (Programování pro děti:

naučte se programovat při tvorbě skvělých her 2013, s. 9) Umožňuje žákům tvořit animace a jednoduché hry. Lze jej spustit jak na počítači, tak na tabletu. (O Scratchi, nedatováno)

Programovací prostředí Scratch je zdarma dostupné jak v online podobě, tak jako desktopová aplikace. Scratch v online verzi umožňuje uživatelům si založit vlastní účet a ukládat si projekty do cloudového prostředí. Učitelé proto snadno mohou s žáky sdílet úkoly a monitorovat jejich práci, ale všechny profily jsou veřejné a nelze sdílet projekty pouze s vybranými uživateli. Scratch sice umožňuje založit z učitelského účtu „třídní skupinu“, ale i ta je veřejná. Proto by učitel měl zvážit volbu online a offline verzí podle věku, mentálních a sociálních dovedností žáků.

Programovací prostředí



Obrázek 2: Uživatelské rozhraní Scratch – obrázková příloha pro žáky

Scénář

Pojem „scénář“ je ve Scratch označení pro kód programu. Scénář je tvořen prostřednictvím bloků, které se spojují za sebe. Scénář je vždy vázán na danou postavu.

Scéna

Samotný program se žákům zobrazí v tzv. „scéně“. Scéna je tvořena postavami a pozadím. Bloky ve scénáři určují chování jednotlivých postav a pozadí ve scéně. Žáci si mohou pozadí vytvořit sami nebo použít některé z „pozadí“, která nabízí Scratch. Scénu lze zvětšit přes celou obrazovku. To je výhodné zejména při výuce mladších žáků. Těm je někdy zapotřebí zdůrazňovat, že program musí být vytvořen tak, aby fungoval po spuštění programu, nikoliv jen po kliknutí na vhodný blok ve scénáři.

Prostřednictvím symbolu „zelené vlaječky“ žáci program spouštějí a za pomoci „červeného puntíku“ program ukončují.

Postava

Objekty, které žáci programují ve Scratch, se nazývají postavy. Umístění postavy ve scéně je určeno kartézským systémem souřadnic. Každá postava má vlastní kód, který není možné mezi postavami sdílet. Žáci mohou pouze kopírovat kódy mezi postavami – nový kód je nezávislý na původním kódu. Postavy však mohou sdílet informace pomocí globálních proměnných a vysílat dalším postavám požadavek („zprávu“) na spuštění podprogramu.

Postavy mohou měnit vzhled pomocí „kostýmů“. Kostýmy lze vybrat z nabídky Scratch, vytvořit vlastní nebo nahrát z počítače.

Programovací jazyk

Jeden blok ve Scratch představuje jeden příkaz. Jednotlivé bloky jsou pro lepší orientaci rozděleny do skupin. Každá skupina má svou funkci a od ostatních se vizuálně liší barvou (např. skupina „Pohyb“ je modrá). Aby žáci snadněji rozlišili účel bloků, mají bloky i různé tvary (např. „špičaté“ bloky slouží jako podmínky pro příkaz „když“, s „kulatými“ bloky lze pracovat jako s proměnnými, ...).

Scratch žákům umožňuje využívat všechny základními konstrukty a principy programování (cykly, podmínky, události, proměnné, funkce, ...). Scratch umožňuje pracovat i se souřadnicemi a s rotací postavy jakožto proměnou – to lze využít pro matematické úlohy. Žáci také mohou pomocí blokových příkazů měnit vzhled postavy, pozadí a pracovat se zvuky.

Kategorie bloků v Scratch:

- Pohyb – bloky pro pohyb postavy na scéně (otáčení, dopředu, směr...); obsahuje proměnnou „směr“ a souřadnice („x“, „y“) pro danou postavu,
- Vzhled – bloky umožňující měnit vzhled pozadí a postav; nachází se zde proměnné „číslo kostýmu postavy“, „číslo kostýmu pozadí“ a „velikost postavy“,
- Zvuk – bloky pro práci se zvuky a proměnnou „hlasitost“,
- Události – bloky pro počáteční události a vysílání zpráv mezi postavami,
- Ovládání – bloky pro práci s cykly, podmínky, klonování postavy a ukončení kódu,
- Vnímání – bloky pro získávání informací ze scény nebo od uživatele; slouží pro zjištění polohy kurzoru, dotyku postavy s jiným objektem nebo barvou; většinu těchto bloků mohou žáci využít v podmínce „když“ nebo jako proměnnou,

- Operátory – bloky umožňují logické a matematické operace, slouží pro práci s proměnnými a seznamy,
- Proměnné – bloky umožňují práci s proměnnými a seznamy; žáci mohou tvořit globální proměnné („pro všechny postavy“) nebo lokální proměnné („jen pro tuto postavu“); rozlišení globální a lokální proměnné občas dělá mladším žákům problém a chybují při výběru typu proměnné,
- Moje bloky – bloky umožňující vytváření procedur a podprogramů; „Moje bloky“ se programují pro každou postavu zvlášť; není možné využívat jeden podprogram pro více postav, ale je možné je jednoduše zkopírovat pro jiné postavy.

3 Praktická část

3.1 Cíl práce

Cílem praktické části diplomové práce je sestavit ucelený soubor úloh pro žáky 5. ročníku, který rozvíjí infromatické myšlení se zaměřením na jeho koncepty, a prostřednictvím výzkumného šetření v praxi ověřit tento soubor úloh na několika skupinách žáků 5. ročníku.

Praktická část diplomové práce by mohla být použita jako doplněk učebnic a přinést pedagogům větší rozmanitost při výběru úloh pro žáky. Autorka se při tvorbě úloh cíleně zaměřila na „praktické činnosti“, se kterými se žáci běžně setkávají (vaření, pokládání sklenic, plavání, ...), a naopak se příliš nezaměřovala na geometrické úlohy, kterým se věnují například učebnice z iMyšlení (Kalaš a Miková, 2020) a Nová škola Duha (Morbacherová, 2023).

3.2 Rozbor úloh

3.2.1 Z hlediska infromatického myšlení

Úlohy rozvíjí IM jako celek. Všechny úlohy jsou stavěné tak, aby žáci museli popsat problém, analyzovat ho, následně úlohu vyřešit a zapsat řešení prostřednictvím programovacího jazyka. Žáci takto vytvářejí řešení, které za ně zpracuje stroj. Úlohy záměrně cílí nejen na rozvoj IM jako celku, ale i na rozvoj jeho jednotlivých konceptů. Koncepty prakticky není možné rozvíjet samostatně, proto se úlohy nezaměřují pouze na jeden koncept, ale postupně rozvíjí více konceptů současně.

Tabulka 7: Rozbor úloh z pohledu IM

Úloha	Algoritmizace	Hledání sekvencí a vzorů	Dekompozice	Abstrakce
1	<ul style="list-style-type: none">scénář – tvorba kódu			
2	<ul style="list-style-type: none">scénář – tvorba kódupořadí bloků v cyklu (např. délka a barva) při kreslení schodů	<ul style="list-style-type: none">opakující se vzor pro kresbu schodů (různé barvy a tvary schodů)		
3	<ul style="list-style-type: none">scénář – tvorba kódupořadí bloků v cyklu („dopředu o“ a „otiskni se“)začátek algoritmu (události)	<ul style="list-style-type: none">opakující se vzor pro pokládání sklenice a zaplnění políčky	<ul style="list-style-type: none">rozložení zadání na dílčí úkoly: položení sklenice a zaplnění políčky	

4	<ul style="list-style-type: none"> viz úloha 3 	<ul style="list-style-type: none"> viz úloha 3 nalezení řešení stejného problému v předchozích úlohách 	<ul style="list-style-type: none"> rozložení zadání na dílčí úkoly: položení sklenice a zaplnění poličky rozložení zadání na činnosti, které žáci umí a které se budou učit 	<ul style="list-style-type: none"> úvaha: Jak poznám konec poličky? Jaké informace jsou podstatné? („podle barvy pozadí“)
5	<ul style="list-style-type: none"> viz úloha 3 	<ul style="list-style-type: none"> viz úloha 4 	<ul style="list-style-type: none"> rozložení zadání na činnosti, které žáci umí a které se budou učit 	
6	<ul style="list-style-type: none"> scénář – tvorba kódu pořadí příkazů v kódu (přesun jídla, otisknout se, ...) začátek algoritmu (události) 	<ul style="list-style-type: none"> nalezení řešení stejného problému v předchozích úlohách nekonečný cyklus pohybu za kurzorem 	<ul style="list-style-type: none"> rozdělení práce na dílčí část: pozadí a dvě postavy s různým kódem rozložení zadání na činnosti, které žáci umí a které se budou učit 	
7	<ul style="list-style-type: none"> scénář – tvorba kódu pořadí příkazů v kódu nekonečný cyklus nekončí – kód za cyklem neproběhne 	<ul style="list-style-type: none"> vlastní blok – Co mají všechny ryby společné? (Plavou.) 	<ul style="list-style-type: none"> rozdělení práce na dílčí část: pozadí a dvě postavy s různým kódem rozdělení kódu ryby na plavání a klonování 	<ul style="list-style-type: none"> plavání – vlastní blok s parametrem klonování
8	<ul style="list-style-type: none"> scénář – tvorba kódu pořadí příkazů v kódu větvení pomocí podmínek 	<ul style="list-style-type: none"> pád jablek nalezení řešení stejného problému v předchozích úlohách 	<ul style="list-style-type: none"> rozdělení práce na dílčí část: pozadí a dvě postavy s různým kódem 	<ul style="list-style-type: none"> vlastní bloky s parametrem – náhodná čísla

3.2.2 Z hlediska pedagogického

Jednotlivé úlohy se skládají ze zadání, vzorového řešení, metodiky a podpurných materiálů. Cílem autorky bylo, aby úlohy mohli využít i učitelé s aprobací pro první stupeň, na které může nyní připadnout i výuka informatiky, s níž nemají zkušenosti.

Úlohy jsou koncipovány tak, aby na konci každé hodiny žáci dosáhli předem daného cíle a zažili pocit úspěchu. Proto mají všechny úlohy dílčí cíle, aby i slabší žáci během hodiny dosáhli alespoň dílčího úspěchu. Pro rychlejší žáky jsou připraveny buď rozšířená zadání, nebo učitel může využít metodu vzájemného učení žáků.

Pro osvojení konceptů informatického myšlení není nutné, aby žáci úlohu dokončili v časovém limitu. Důležité je, aby uvažovali nad řešením úlohy. Úlohy na sebe navazují a postupně rozvíjí nejen koncepty informatického myšlení, ale i samotného programování ve Scratch. S jednotlivými úlohami roste komplexnost úlohy a vyžadovaná míra samostatnosti žáků při tvorbě scénáře a postav. Cyklus úloh postupně žáky seznamuje s novými bloky v návaznosti na použití podobných bloků v předchozích úlohách. Úlohy jsou však na sobě nezávislé, lze je použít i jednotlivě.

Úlohy mají žáky postupně vést k větší samostatnosti při práci ve Scratch. Od úlohy č. 6 již žáci nemají předpřipravené zadání ve Scratch. To slouží k motivaci žáků, kteří postupně uvidí, že zvládají více a více sami. Na konci cyklu úloh si žáci vytvoří vlastní hru, včetně nakreslení postavy, a vlastní bloky, které předtím dostávali předpřipravené.

3.2.3 Z hlediska RVP ZV

Úlohy naplňují RVP ZV 2021 pro vzdělávací oblast Informatika, tematický celek „Algoritmizace a programování“ pro 5. ročník a odpovídají modelovým ŠVP „Opatrně vpřed“ (Modelový ŠVP: OPATRNE VPŘED, 2022, str. 12) a ŠVP „Nebojácně vpřed“ (Modelový ŠVP: NEBOJACNE VPŘED, 2022, str. 11), které jsou uvedeny na stránkách iMyšlení (Modelové školní vzdělávací programy pro základní vzdělávání, 2022). Oproti modelovým ŠVP soubor úloh neobsahuje posílání zpráv mezi postavami. Pokud by bylo zapotřebí, lze ho zařadit do souboru jednoduchou modifikací úloh 6, 7 nebo 8, ve kterých na sebe reagují postavy.

Využití jednotlivých bloků a objektů ve Scratch je zaznamenáno v tabulce 8. Barevně je vyznačeno, zda žáci s bloky a objekty již umí pracovat nebo se s nimi setkávají poprvé. Oranžově jsou vyznačeny bloky, se kterými se žáci teprve seznamují. Modře jsou vyznačeny bloky, které rozšiřují předchozí znalosti. Zeleně jsou označeny bloky, které žáci ovládají z přechozích cvičení.

Tabulka 8: Využití bloků v jednotlivých úlohách

Čísl	Pohyb	Události	Cykly	Podmínky	Pero	Vzhled	Postavy	Pozadí	Klonování	Operátory	Moje bloky
1	dopředu o, otočit se	pasivně – zelená vlajka					úprava kostýmu (rozšiřující)				pasivně – schované
2	dopředu o, otočit se	pasivně – zelená vlajka, po stisknutí klávesy	cyklus s konečným počtem opakování		pasivně – pero zapnout/vypnout, aktivně – barvy						pasivně – schované
3	dopředu o, otočit se	aktivně – po stisknutí klávesy	cyklus s konečným počtem opakování		otiskni se	další kostým, nastav kostým na					pasivně – viditelné ve scénáři
4	dopředu o, otočit se a skoč na (rozšiřující)	aktivně – zelená vlajka	opakuji dokud nenastane	podmínka v cyklu	otiskni se	další kostým, nastav kostým na					pasivně – viditelné ve scénáři
5	dopředu o, otočit se	aktivně – zelená vlajka	cyklus s konečným počtem opakování	příkaz když	otiskni se	nastav kostým na					pasivně – viditelné ve scénáři
6	dopředu o, nastav směr na/skoč na	aktivně – zelená vlajka	nekonečný cyklus	příkaz když/podmínka v cyklu	otiskni se		vytvořit postavu	vytvořit pozadí			
7	dopředu o/klouzej, odrazení od okraje	aktivně – zelená vlajka	cyklus s konečným počtem opakování			další kostým	vytvořit postavu	vytvořit pozadí	klonovat sebe	náhodná čísla	aktivně – vytvořit „plavání“ (s parametrem)
8	dopředu o, souřadnice	aktivně – po stisknutí klávesy, zelená vlajka	nekonečný cyklus, cyklus s konečným počtem opakování, cyklus s podmínkou	podmínka v cyklu	otiskni se (nebo klonování)	další kostým/nastav kostým na, bublina	vytvořit postavu – vlastní kostýmy	vytvořit pozadí – nahrát z PC	klonovat sebe (nebo otisknout se)	náhodná čísla	vytvořit – „padání jablek“ (s parametrem – náhodná čísla)

3.2.4 Obecné metodické pokyny ke všem úlohám

Úlohy mají rozdílnou obtížnost i časovou náročnost. Všechny úlohy, s výjimkou osmé, lze vypracovat během jedné vyučovací hodiny. V časové dotaci je počítáno i s prostorem pro výklad učitele a rozbor úlohy. V následující tabulce jsou přehledně shrnuty metodické pokyny pro jednotlivé úlohy.

Tabulka 9: Metodický rozbor úloh

Úloha	Název	Časová dotace	Materiály:	Formy/metody výuky	Náročnost
1	Vaření s robotem	30 minut výklad a příprava + 15 minut řešení úlohy	<ul style="list-style-type: none"> Scratch soubor pro žáky prezentace 	samostatná práce žáků	velmi lehké
2	Kreslení s robotem	30 minut výklad a příprava + 15 minut řešení úlohy	<ul style="list-style-type: none"> Scratch soubor pro žáky pracovní list/zápis do sešitu 	práce ve dvojici/samostatná práce žáků; vzájemné učení	lehké (bez pracovního listu středně obtížné)
3	Pokládání sklenic	30 minut výklad a příprava + 15 minut řešení úlohy	<ul style="list-style-type: none"> Scratch soubor pro žáky prezentace video s ukázkou cíle úlohy 	samostatná práce žáků; vzájemné učení	středně obtížné
4	Pokládání sklenic do různých políček	30 minut výklad a příprava + 15 minut řešení úlohy	<ul style="list-style-type: none"> Scratch soubor pro žáky prezentace video s ukázkou cíle úlohy 	samostatná práce žáků; vzájemné učení	středně obtížné
5	Pokládání sklenic na podložku	25 minut výklad a příprava 20 minut řešení úlohy	<ul style="list-style-type: none"> Scratch soubor pro žáky prezentace video s ukázkou cíle úlohy 	práce ve dvojici/samostatná práce žáků; vzájemné učení	obtížné až velmi obtížné
6	Hladová myš	25–30 minut výklad a příprava + 15-20 minut řešení úlohy	<ul style="list-style-type: none"> Scratch soubor pro žáky video s ukázkou cíle úlohy 	samostatná práce žáků	lehké

7	Akvárium	30 minut výklad a příprava + 15 minut řešení úlohy	<ul style="list-style-type: none"> • prezentace • video s ukázkou cíle úlohy 	práce ve dvojici/ samostatná práce žáků	obtížné
8	Had a padající jablka	2 vyučovací hodiny (2x 20 minut výklad a příprava a 25 minut řešení úlohy)	<ul style="list-style-type: none"> • pozadí hry • Scratch soubor s vypracovanou první fází pro žáky • video s ukázkou cíle úlohy 	práce ve dvojici/ samostatná práce žáků/skupinová práce	velmi obtížné

Časová dotace:

Vyučovací hodina je rozdělena na tři části. První je úvodní, ve které je zahrnuta i administrativa a další činnosti (docházka, třídní kniha, hledání pomůcek, ...). Předpokládaná časová dotace je 10–20 minut. V této části učitel představí zadání a případně vysvětlí nové bloky či programovací principy. V praxi je třeba počítat s tím, že pokud žáci dělají zápis výkladu do sešitu, v některých třídách děti píšou pomaleji, a proto je zapotřebí zkrátit zápis, nebo rovnou připravit část zápisu na vlepění do sešitu. Ve druhé části žáci samostatně, nebo ve skupinách, bádají a řeší úlohy. Na závěr hodiny učitel provede s žáky rozbor úlohy. Rozbor úlohy zabere přibližně 5 minut.

Doporučení, na co klást důraz v jednotlivých fázích výuky:

- Úvod:
 - Učitel kontroluje, zda žáci mají otevřené správné zadání a rozumí mu. Hledání a otvírání souborů může zpočátku zabrat cca pět minut a žáci potřebují pomoc učitele. Později žáci zvládají otevřít zadání sami nebo si mohou dopomoci navzájem a učitel může využít čas pro administrativu.
 - Učitel žákům předvede výsledek práce, aby žáci viděli, čeho mají dosáhnout.
 - Před zahájením samotné činnosti žáků s nimi učitel vede debatu o podstatě problému a pokládá návodné otázky.
 - Dále před samotnou prací vždy učitel s žáky přečte předpřipravený scénář v zadání úloh. S žáky společně rozebere, co které bloky dělají.
- Práce na úloze:

- Při řešení úloh není nejpodstatnější vyřešení problému, ale způsob uvažování žáků při jeho řešení. Žáci si mohou řešení zaznamenávat, aby věděli, jak postupovali. V praxi to však, až na výjimky, nedělají. Píší pomalu a nejsou schopni se současně věnovat řešení úlohy a zápisu.
- Pokud si žáci nevědí rady, učitel pokládá návodné otázky nebo nechá žáky o problému diskutovat.
- V tabulce 7 jsou uvedeny doporučené/očekávané bloky. Učitel by však neměl žáky odrazovat od jiných způsobů řešení.
- Učitel klade důraz na to, že existuje více způsobů, jak úlohy vyřešit. Správnost řešení žáci ověřují opakovaným spouštěním programu. Pokud program splňuje zadání, tak je naprogramovaný správně.
- V případě, že se práce žákům nedaří, učitel by je měl povzbuzovat a nasměrovat, nebo je upozornit, že „něco“ nemají správně.
- Učitel učí žáky, jak pracovat s chybou. Vede žáky k tomu, aby četli kód, spouštěli scénář a hledali chybu. Hledání chyby je důležitá součást žakovské práce.
- Závěr:
 - Učitel s žáky vytvoří modelové řešení úlohy.
 - Žáci navrhnou další řešení a obhajují své postupy.
 - Žáci uloží svou práci. V případě práce offline se zpočátku jedná o časově náročný úkol. Ukládání práce může zabrat žákům více než pět minut. V této situaci je bohužel zapotřebí ubrat časovou dotaci ze samostatné práce žáků. Později však žáci vše zvládají, případně si mohou i navzájem pomoci.

Materiály⁸

Pro žáky:

- Scratch soubor (úlohy č. 1-5, 8): obsahuje předpřipravený scénář a zadání práce,
- prezentace (úlohy č. 1, 3-5, 7): vedou žáky k řešení problému, jsou v nich uvedeny řešení dílčích částí úloha lze je využít pro chybějící žáky jako podklad pro domácí práci nebo učitel může promítat prezentaci (nebo její část) během výkladu,

⁸ Všechny materiály jsou součástí elektronické přílohy a dostupné na stránce <https://sites.google.com/view/ucimesescratch/>.

- pracovní list (úloha č. 2),
- videa s výsledkem práce (úlohy č. 3-8),
- další materiály: pozadí pro úkol č. 8.

Pro učitele:

- Scratch soubor s ukázkou možného řešení,
- prezentace: u některých úloh (č. 1, 3-5, 7) je prezentace s postupem práce a návodem.

3.2.5 Metody a formy výuky

Při metodě vzájemného učení žáků je třeba klást důraz na to, aby žáci spolužákům nediktovali řešení. Z počátku, než si tento způsob žáci osvojí a jsou si schopni navzájem pokládat návodné otázky, je musí učitel vést. Tato metoda je výhodná i pro žáky, kteří pomáhají spolužákům, protože si musí upořádat své myšlenky a formulovat návodné otázky, které vedou k použití inforatického myšlení k vyřešení úlohy.

Náročnost úloh

Deklarovaná náročnost vychází z míry úspěšnosti řešení a potřebné dopomoci, kterou žáci potřebovali při řešení úlohy. Jejich náročnost postupně stoupá, ale občas jsou zařazeny jednodušší úlohy, aby většina žáků měla šanci zažít úspěch nejen na začátku. Náročnost byla definována následovně:

- velmi lehké: všichni žáci zvládli rozbor i řešení samostatně,
- lehké: většina žáků zvládla rozbor úlohy samostatně, malá část žáků potřebovala dovysvětlit zadání (např. návodné otázky),
- středně obtížné: před řešením úlohy bylo zapotřebí provést její rozbor (společně, ve dvojici, ...),
- obtížné: bylo zapotřebí provést společný rozbor úlohy s celou třídou, klást žákům návodné otázky a kontrolovat proces řešení úlohy u většiny žáků,
- velmi obtížné: i po společném rozboru úlohy část žáků potřebovala vedení při jejím řešení.

Možné potíže a jejich řešení:

- Úlohy jsou příliš složité:
 - Pro jednotlivce:
 - učitel zkontroluje porozumění zadání,
 - žáci mohou vypracovat zjednodušenou variantu úlohy,
 - žáci nemusí vyřešit celou úlohu; zkusí vyřešit alespoň dílčí cíle,
 - vzájemné učení žáků – žáci, kteří mají hotovo, pomáhají pod vedením učitele ostatním.
 - Pro větší část žáků:
 - práce ve dvojích nebo malých skupinách,
 - zadat pouze dílčí cíle.
- Úlohy jsou příliš jednoduché:
 - využít rozšířenou variantu zadání úloh,
 - lze zkrátit časovou dotaci.
- Nedostatek času:
 - Hledat příčinu:
 - žáci nestíhají, protože jsou pro ně úlohy příliš složité → zadat pouze dílčí cíle nebo zjednodušenou variantu,
 - žáci nestíhají, protože příliš pomalu tvoří zápis a rozbor úloh → připravit alespoň část zápisu na vlepění do sešitu,
 - žáci nestíhají, protože se věnují jiným činnostem (např. hrají si, povídají si) → použít běžné kázeňské metody.
 - V případě neočekávané situace, která značně zkrátí čas na práci, je možné zařadit skupinovou výuku (maximálně 3 žáci ve skupině).
- Žáci neukládají svoji práci:
 - zejména ze začátku je důležité žákům zdůrazňovat, že si práci mají ukládat,
 - žáci musí mít dostatečný časový prostor pro uložení práce,
 - kontrolovat žáky, kteří mají pocit, že se jim práce nedaří; někdy mívají tendenci práci záměrně neukládat.
- Žáci nemají zájem o práci:

- V rámci šetření byl nejvíce zapříčiněn subjektivním pocitem obtížnosti úlohy.
 - Zkontrolovat porozumění zadání a přizpůsobit obtížnost zadání úrovni dovedností žáků,
 - pracovat s poznámkami v sešitě,
 - pochválit i za dílčí úspěch,
 - pracovat ve dvojici.
- Motivovat možností vytvořit si vlastní hru.
- Zdůrazňovat, co se naučili v porovnání s minulou hodinou a co všechno už umí.
- Personalizovat zadání. (Jaký sendvič chceš? Kterou příchuť máš nejraději? Které zvíře by mohlo zanechávat odpadky? ...)

3.3 Výzkumný soubor

Výzkumné šetření probíhalo na jedné základní škole v Praze, která má první i druhý stupeň. Školu navštěvuje přibližně 700 žáků. Historicky se jedná o školu s rozšířenou výukou informatiky. Na škole probíhá výuka „nové informatiky“ již od školního roku 2021/2022, škola má 2 plně vybavené počítačové učebny. Ve škole jsou v 5. ročníku tři třídy a na informatiku je každá třída dělena na dvě skupiny.

Úlohy byly koncipovány jako praktická část výuky v devíti po sobě jdoucích vyučovacích hodinách pro tematický celek „Algoritmizace a programování“. Úlohy byly otestovány na šesti skupinách žáků – ve dvou cyklech po třech skupinách. V prvním cyklu byl použit prvotní návrh úloh, v druhém byla použita předpokládaná finální podoba úloh a metodických materiálů a docházelo pouze k drobným úpravám.

Každou skupinu tvořilo přibližně 15 žáků. V prvním cyklu se skupiny od sebe významně lišily. V první skupině bylo několik žáků s odlišným mateřským jazykem, v druhé byli žáci se SPU a sníženou motivací k výuce a v poslední skupině byli několik žáků se zvýšeným zájmem o výuku. V druhém cyklu byly skupiny více vyrovnané jak výkonnostně, tak motivačně. Nehomogenost skupin prvního cyklu se ukázala být výhodou, protože poskytla širokou zpětnou vazbu.

Skupina 1

V této skupině převládal kladný vztah k výuce. Jeden žák měl upravené očekávané výstupy a asistenta pedagoga. Žák zpravidla pracoval na zjednodušených úlohách a asistent žákovi pomáhal se zadáním. Ve třídě byli nově také žáci s odlišným mateřským jazykem. Těmto žákům byla nabídnuta možnost přepnout Scratch do jejich mateřského jazyka. I přes jazykovou bariéru žáci, kteří neměli velkou absenci, zvládli práci na úlohách bez větších problémů.

Skupina 2

Jednalo se o dovednostně velmi nevyváženou skupinu. Ve skupině byli žáci s velmi rozdílným přístupem k výuce. U žáků s SVP byla i přidružená porucha pozornosti a někteří měli sníženou motivaci ke školní práci, proto bylo ze začátku potřeba žáky více motivovat.

Skupina 3

Jednalo se o skupinu, ve které měli žáci lehce nadprůměrné studijní výsledky. Většina žáků nosila pomůcky, dávala pozor během výkladu a snažila se pracovat dle zadání. V této skupině bylo i několik nadanějších žáků, kteří pracovali rychleji. Tito žáci pomáhali pod vedením pedagoga svým spolužákům. Velmi rychle si osvojili, že pomáhání není diktování odpovědí, ale pokládání návodných otázek nebo poukazování na chybu v kódu.

Skupina 4

Tato skupina byla výkonnostně vyrovnaná a žáci měli mírně nadprůměrné výsledky, žáci pracovali stejným tempem.

Skupina 5

Tato skupina byla výkonnostně vyrovnaná. Část žáků měla velmi pomalé tempo psaní a bylo zapotřebí část zápisu přinést vytištěnou na vlepění do sešitu, protože žákům jinak nezbyval čas na samotnou práci.

Skupina 6

V této skupině převažovali žáci se „slabším“ prospěchem a žáci se speciálními vzdělávacími potřebami. Někteří žáci měli větší absenci a nevěnovali odpovídající čas domácí přípravě, což jim komplikovalo práci ve škole. Jednalo se však o aktivní skupinu, která se snažila zadané úkoly splnit.

3.4 Soubor úloh

3.4.1 Úloha č. 1: Vaření s robotem

Odkaz na zadání: <https://scratch.mit.edu/projects/668165572/>

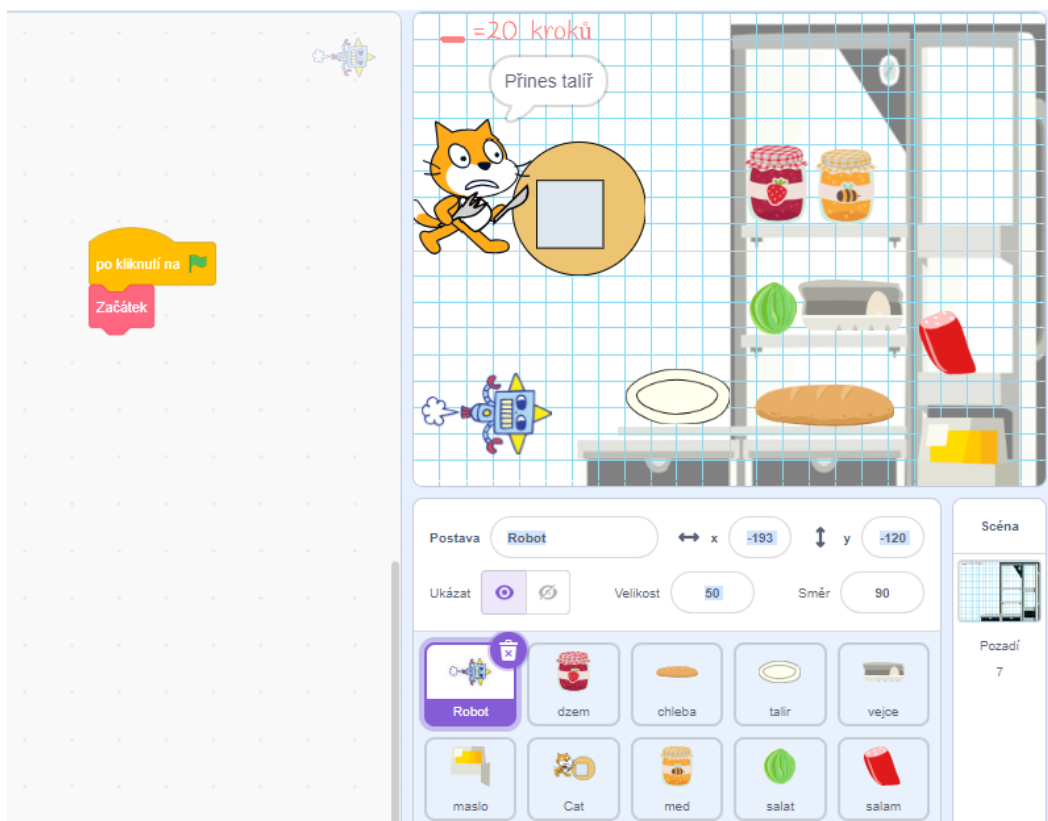
Odkaz na řešení: <https://scratch.mit.edu/projects/981326437/>

Výukové cíle

- Žák vysvětlí, proč je důležité pořadí kroků v algoritmu.
- Žák vysvětlí, k čemu slouží tlačítko „zelené vlajky“ v aplikaci Scratch.
- Žák vytvoří algoritmus pro jednoduchý postup v jazyce Scratch.

Zadání

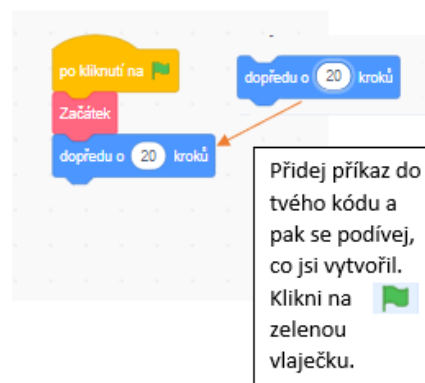
Pomoz robotovi připravit sendvič pro kočičku Scratch. Sendvič samozřejmě podávej na talíři. Všechno jídlo na přípravu sendviče najdeš v ledničce. Ze surovin můžeš vytvořit různé sendviče, tak udělej takový, který by chutnal tobě.



Obrázek 3: Úloha č. 1 – Scratch soubor se zadáním

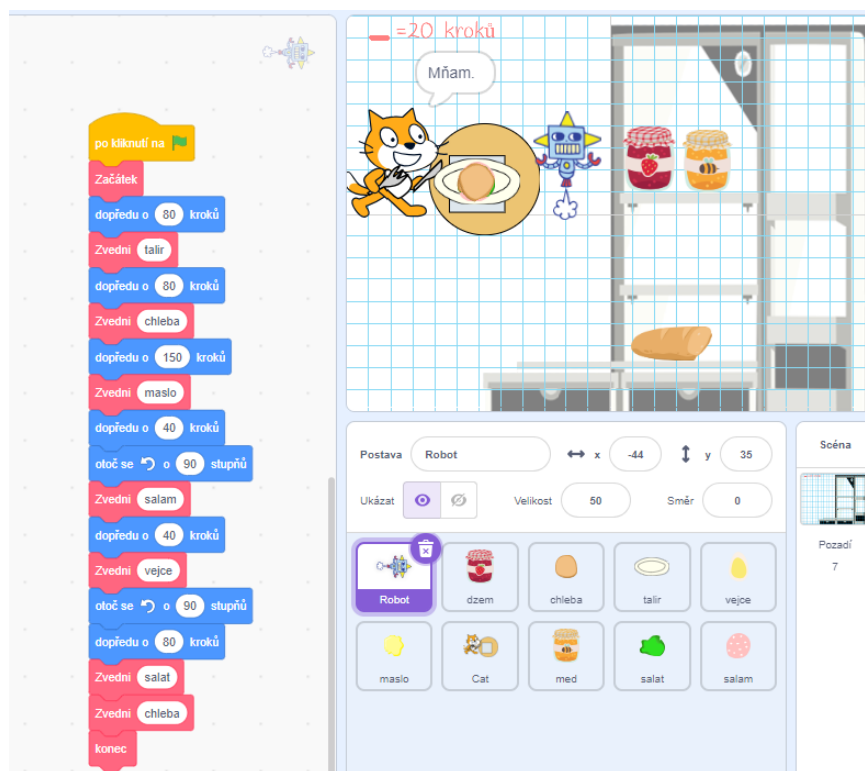
Pokyny pro žáky

Pozor, robot není člověk. Nerozumí lidské řeči, ale má vlastní příkazy ve svém programovacím jazyce. Robotovi musíš říci, kam přesně má pro jídlo do ledničky dojít, tedy o kolik kroků se musí posunout dopředu. Využij příkaz „posunout dopředu o _“. Až robot dojde k jídlu, tak ho může zvednout pomocí bloku „zvedni _“. Do „okýnka“ v bloku „zvedni _“ napiš bez háček a čárek, co má zvednout. Pokud si nejsi jistý/á, tak se podívej na název postavy „jídla“. O kolik kroků půjde robot dopředu? Každý čtvereček je dvacet kroků. Co když je jídlo nad robotem? Otoč robota o 90 stupňů.



Nezapomeň, že robota programuješ. Tvoříš pro něj postup, algoritmus, co má dělat. Každý příkaz musíš přenést do scénáře a připojit ho do svého kódu. Velmi důležité je i pořadí činností. Robot sám neví, zda má namazat máslo na talíř nebo na chleba.

Ukázka možného způsobu řešení



Obrázek 4: Úlohy č. 1- ukázka možného řešení

Metodický rozbor úlohy

Žáci vytváří algoritmus pro výrobu sendviče ve Scratch. Cílem úlohy je, aby si žáci osvojili tvorbu kódu ve Scratch a pochopili princip automatizace činnosti. Robot dělá věci za člověka, ale člověk mu musí zadat správný algoritmus. Každý algoritmus musí mít začátek (počáteční událost) a jasně dané pořadí kroků. Úloha je připravena tak, že se po kliknutí na zelenou vlajku vrátí všechny postavy na výchozí pozici. Pokud žáci pouze klikají na příkazy v nabídce nebo stále dokola upravují kousek kódu ve scénáři, tak po návratu do počáteční pozice program nefunguje.

První aktivita je záměrně jednoduchá a s tématem pro žáky snadno pochopitelným. Zároveň, dle autorčiných zkušeností, většinu dětí téma „jídlo“ zajímá a lze počítat s tím, že aktivita na toto téma pro ně bude atraktivnější. Tato aktivita má žáky také motivovat a dodat pocit sebejistoty při práci v aplikaci Scratch.

Tato aktivita z pohledu žáků napodobuje cvičení z Bobříka informatiky, code.org a dalších online zdrojů pro rozvoj algoritmického myšlení. Pro žáky, kteří se již s podobným typem úloh setkali, slouží jako seznámení s prostředím Scratch. Ostatní žáky připravuje na programovací úlohy v Bobříku informatiky, ve kterých žáci zpravidla musí s panáčkem někam dojít a něco udělat.

Možné varianty úlohy podle náročnosti

- Základní:
 - vytvoření kódu pro výrobu sendviče.
- Rozšířená:
 - přidání více surovin,
 - úprava kostýmu robota,
 - použití bloku „skoč na“.
- Zjednodušená:
 - místo „dopředu o _ kroků“ lze využít vlastní blok „Přesuň o _ čtverečků“.

Vlastní bloky a doporučené příkazy

- Vlastní bloky:
 - „začátek“: slouží k návratu všech prvků do původní pozice,

- „konec“: slouží jako příkaz pro ukončení algoritmu, robot odejde ze scény,
- „zvedni“: slouží k servírování jídla,
- „přesuň o _ čtverečků“: slouží ke zjednodušení posouvání robota pro žáky, kterým činní potíže dopočítat kroky; tento blok není na začátku žákům představen; je ukázán pouze žákům, kteří mají problém s „dopředu o _ kroků“.
- Doporučené příkazy:
 - „dopředu o _ kroků“: žáci se seznámí s příkazem, který budou používat velmi často; zjistí jeho funkci, úskalí (posouvá pouze dopředu) a díky čtvercové síti na pozadí získají představu o délce jednotky „krok“,
 - „otoč doleva o _ stupňů“: žáci zjistí, jak lze otáčet postavy v Scratch; uvědomí si, že otáčení doleva/doprava/nahoru/dolů je o 90° – „postava chodí do čtverce“.
- Další příkazy, které mohou žáci využít:
 - „skoč na x: _ y: _“ nebo „klouzej na x: _ y: _“: Pokud pedagog již dříve s žáky probral souřadnice, tak žáci snadno zjistí pozici suroviny a usnadní si počítání. Tato aktivita je zaměřená na jednotku „krok“, se kterou se bude pracovat v dalších aktivitách, a otáčení postav. Proto by měl vyučující žáky vést k „dopředu o _ kroků“.
 - „nastav směr na _“: Je pro žáky jednodušší, ale v následujícím úkolu budou žáci pracovat s „otoč o _ stupňů“. Pedagog zatím tento blok nepředstavuje, aby se žákům tyto příkazy nepletly.

Doporučený průběh hodiny

Úvod a zadání: (10 minut)

- Učitel ukáže žákům hotový program.
- Učitel s žáky udělá slovní rozbor.
- Učitel žákům předvede, jak se pracuje ve Scratch.

Samostatná práce: (15 minut)

- Žáci naprogramují přípravu sendviče.

Závěr: (5 minut)

- Společný rozbor řešení úlohy.

Možné komplikace

- Žáci netvoří kód, pouze používají bloky jako „tlačítka“.
 - Kliknout na zelenou vlaječku.
- Potraviny se nepokládají na chleba.
 - Pravděpodobně žáci špatně opsali název postavy, kterou chtějí položit na chleba.

Reflexe úlohy č. 1

Skupina 1

Všichni žáci úlohu zvládli. Pouze ojedinele vyučující v průběhu hodiny musel žáky upozorňovat na nutnost tvorby kódu, protože žáci používali bloky ve Scratch jako tlačítka. Žáci měli problém s ukládáním a úlohu odevzdala necelá polovina žáků.

Skupina 2

Úloha samotná žákům nedělala problém, ale někteří ji nezvládli z časových důvodů. Žáci byli z organizačních důvodů spojeni s druhu polovinou třídy, a proto pracovali na úloze ve dvojicích. Bohužel se práce ve dvojicích ukázala jako nevhodná, protože v této skupině byli žáci s poruchou pozornosti a nižší motivací k výuce. Tito žáci, když pracovali ve dvojici, měli tendenci se věnovat jiné činnosti než výuce, nesoustředili se na práci a potřebovali od postavy kočky podrobnější pokyny. Úloha byla proto upravena.

Změna úlohy:

- Byly dodány podrobnější instrukce, co mají žáci dělat, když položí jednu surovinu na chleba („Svačina! Pokud mi nechceš ještě něco přidat na chleba, tak mi na něj můžeš položit další chleba.“).

Skupina 3

Žáci s úlohou neměli problém. Bloky jako tlačítka používali výjimečně, ale po kliknutí na zelenou vlaječku pochopili zadání.

Skupina 4 a skupina 6

Úloha byla pro žáky jednoduchá a většinou je bavila, ale i v těchto skupinách někteří žáci používali bloky jako tlačítka. Žáci sice již programovali ve čtvrtém ročníku, část žáků však učivo zapoměla.

Skupina 5

Z organizačních důvodů byli žáci spojeni s druhou půlkou třídy a pracovali ve dvojicích. Úkol žáci zvládli za 10 minut. Úlohu bohužel nebylo možné rozšiřovat, protože se žáci předtím zdrželi tvorbou zápisu do sešitu.

Závěr

Úloha byla pro žáky jednoduchá a většinou i zábavná. V šetření měli i někteří žáci, kteří se s programováním setkali, tendenci netvořit kód, ale používat bloky jako tlačítka. Tato úloha napomáhá rozvoji algoritmizace a seznamuje žáky s automatizací a je důležitá pro pochopení základních principů práce ve Scratch a řešení úloh krokováním.

3.4.2 Úloha č. 2: Kreslení schodů s robotem

Odkaz na zadání: <https://scratch.mit.edu/projects/756146581/>

Odkaz na řešení: <https://scratch.mit.edu/projects/921669175/>

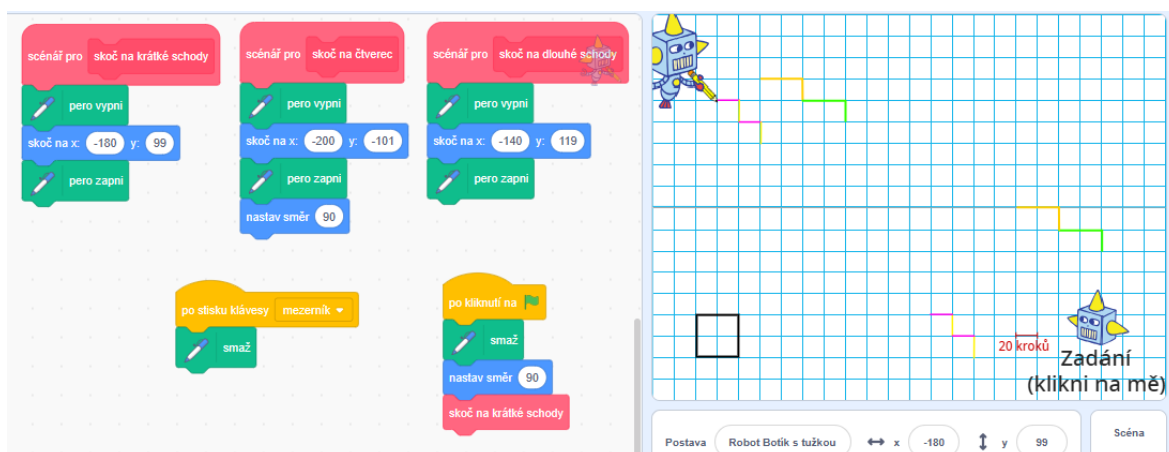
Výukové cíle

- Žák vytvoří algoritmus pro kreslení schodů ve Scratch.
- Žák vysvětlí, k čemu slouží konečný cyklus v programování.
- Žák použije konečný cyklus pro zkrácení kódu ve Scratch.
- Žák předvede význam pořadí kroků pro kreslení dvoubarevných schodů.

Zadání

Pomoz robotovi Botíkovi nakreslit schody a nauč se kreslit ve Scratch. Napřed si nakresli schody do pracovního listu a pak naprogramuj program tak, aby i Botík uměl nakreslit schody.

Po otevření úlohy klikni na hlavu robota Botíka, který ti řekne přesné zadání.



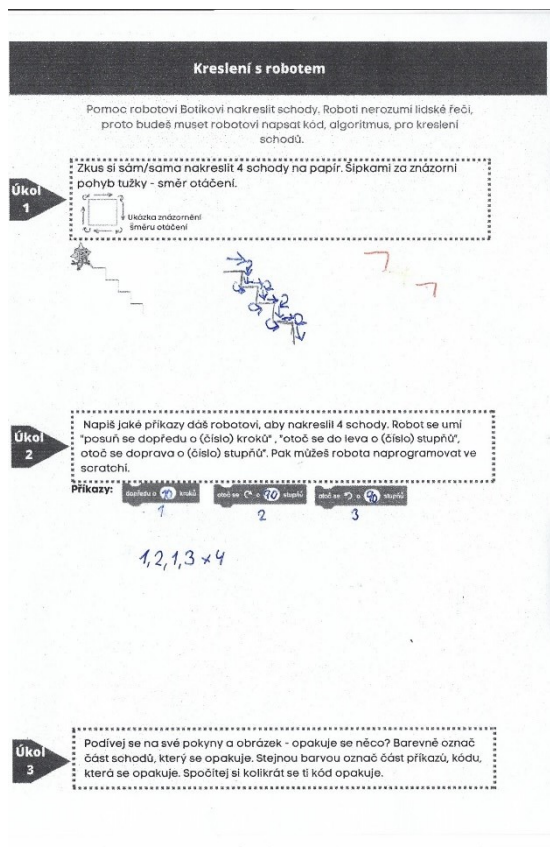
Obrázek 5: Úloha č. 2 – Scratch soubor se zadáním

Pokyny pro žáky

Napřed si do pracovního listu nakresli schody. Do obrázku si zaznamenej směr pohybu tužky. Podle náčrtku naprogramuj robota Botíka. Když otáčíš tužkou, tak použij „otoč se o _ stupňů“. Pozor na správný směr, pro pohyb dopředu použij „dopředu o _ kroků“. Pro změnu barvy použij „nastav barvu pera na _“. Za pomoci bloku „opakuj“ vytvoř co nejkratší kód pro kreslení schodů.

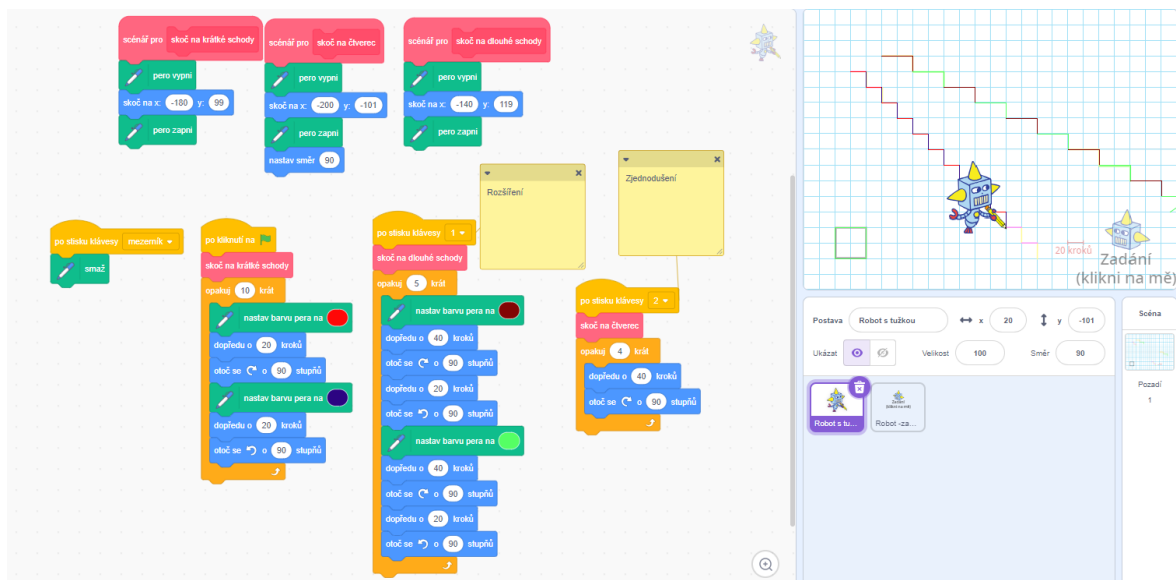
Pro programátory je důležité, aby byl kód co nejkratší a nejpřehlednější. Když je kód krátký a přehledný, lépe se opravuje a upravuje. Také máš menší pravděpodobnost, že uděláš chybu, když se přepíšeš.

Hlavně nemaž žádnou předpřipravenou část kódu! Pak by ti přestalo zadání fungovat.



Obrázek 6: Úloha č. 2 – ukázka žákovské práce: pracovní list

Ukázka možného řešení



Obrázek 7: Úloha č. 2 – ukázka možného řešení

Metodický rozbor úlohy

Žáci tvoří kód pro kreslení schodů ve Scratch. Cílem úlohy je, aby našli opakující se část kódu a využili konečný cyklus. Žáci mají na pozadí předkreslené schody, které mají za úkol dokreslit. Opět je na pozadí čtvercová síť s vyznačenou délkou 20 kroků na stranu čtverce. Žáci programují postavu tužky, kterou drží robot. Robot byl k původní postavě tužky přidán, aby žákům ukazoval směr, kterým tužka kreslí. Při práci žáci využívají pracovní list „Kreslení schodů“, ve kterém znázorňují pohyb tužky a hledají opakující se vzor.

Možné varianty úlohy podle náročnosti

- **Základní:**
 - schody stejné délky a šířky: žáci tvoří kód pro čtvercové schody a nemusí řešit pořadí bloků v kódu; pro uvědomění si pořadí je schod dvojbarevný – horní a boční část chodů mají rozdílnou barvu.
- **Rozšířená:**
 - schody různé délky a šířky: žáci kreslí schody obdélníkového tvaru (délka 40 kroků, výška 20), proto musí skládat bloky ve správném pořadí; předkreslené schody mají rozdílnou barvu lichého a sudého schodu.
- **Zjednodušená:**
 - Čtverec: slouží pro žáky, pro které jsou schody příliš složité.

Doporučený průběh hodiny

Úvod a zadání: (5 minut)

- Na začátku učitel zopakuje s žáky základy práce ve Scratch (Co jsme dělali minulou hodinu? Jak jsme programovali robota? Jak mažeme bloky ve Scratch? ...). Během opakování distribuuje žákům pracovní listy – „Kreslení s robotem“.
- Vyučující vysvětlí žákům zadání práce.
- Učitel žákům předvede „pero“ a nechá je chvíli objevovat kreslení s perem (tloušťku, barvu, vypnout/zapnout, ...). Následně žákům vysvětlí konečný cyklus ve Scratch.
- Před samotnou prací učitel žákům vysvětlí připravený kód v úloze.

Samostatná práce: (20 minut)

- Než se žáci pustí do práce ve Scratch, vypracují pracovní list. Žáci zatím nezapínají Scratch- (cca 5–10 minut)

- Teprve po splnění prvního úkolu v pracovním listu si žáci otevřou úlohu ve Scratch. Během samostatné práce učitel pozoruje žáky a případně jim pomáhá prostřednictvím návodných otázek. Pokud vyučující vidí, že úloha je pro žáka neúměrně složitá, tak úlohu modifikuje na kresbu čtverce. (10-15 minut)

Závěr: (5 minut)

- Společný rozbor řešení úlohy.

Vlastní bloky a doporučené příkazy

- Vlastní bloky:
 - „Skoč na začátek krátkých schodů“, „Skoč na začátek dlouhých schodů“, „Skoč na čtverec“: slouží pro přesunu postavy robota.
- Doporučené příkazy:
 - „dopředu o _ kroků“,
 - „otoč doleva/doprava o 90 stupňů“ ,
 - „nastav barvu pera na _“: žák si má uvědomit pořadí kroků.
- Další příkazy, které mohou žáci využít:
 - „nastav směr na _“: žákům je nutné vysvětlit rozdíl mezi „nastav směr“ a „otoč se“, jinak nebude kód žákům fungovat.
 - „změň barvu o _“: blok cyklicky mění barvu, ta pak ale neznázorňuje jednotlivé kroky; v šetření ho většinou používali žáci, kteří nerozuměli zadání a nahradili jím blok „nastav barvu na _“, který nedokázali umístit do správné části kódu.

Reflexe úlohy č. 2

Skupina 1:

Žáci nejvíce času strávili přemýšlením, jak nakreslit schody. Autorka proto vyzvala žáky, aby si schody nakreslili do sešitu a ke schodům si vyznačili šipkami, jak se pohybovali – vpřed nebo otáčením. Většině žáků to pomohlo, ale z časových důvodů nestihli schody naprogramovat. Žáky také mátl blok pro smazání, který autorka připravila jako „tlačítko“ a nebyl tedy součástí žádného kódu. Po prvním úkolu žáci očekávali, že bloky nepoužívají jako „tlačítka“.

Změny úlohy:

- Blok „smaž“ je spuštěn klávesou mezerník.
- Zařazena výuka „Událostí“ – po stisknutí klávesy „1“ nakreslí schody.

Skupina 2:

V této skupině se žáci seznamovali i s událostmi. Někteří žáci této skupiny v prvním úkolu nepracovali podle zadání, proto měli s úlohou č. 2 velké problémy. Žáci se neorientovali v uživatelském rozhraní a bylo zapotřebí zopakovat úvod do Scratch. Také měli tendence netvořit kód, ale používat bloky jako tlačítka. Pokud žáci měli problém s nakreslením schodů, tak byli vyzváni, aby si zkusili nakreslit chody do sešitu. Zde nastal problém, protože část žáků neměla sešit ani papír.

Všichni žáci zvládli vytvořit čtverec, ale dva žáci nebyli schopni využít cyklus pro opakování příkazů. Pouze 4 žáci dokázali vytvořit schody a použít příkaz „opakuj“.

Pro tuto skupinu byla práce na zadání velmi obtížná. Dokonce ani polovina žáků nezvládla uložit svoji práci.

Změny úlohy:

- Byla přidána zjednodušená varianta pro vytvoření čtverce.
- Přidán předpřipravený vlastní blok pro přesun na začátek schodů, aby se žáci mohli věnovat pouze hledání vzorů.
- Blok „opakuj se“ autorka žákům předpřipravila rovnou do scénáře.
- Mazání mezerníkem autorka žákům připravila rovnou do scénáře.

Skupina 3:

Tato skupina byla velmi rychlá a při řešení úlohy převážně úspěšná. Žáci byli rovnou vyzváni, aby kreslili do sešitu. Pouze dva žáci potřebovali pomoc. Jednomu dopomohl spolužák dovysvětlením zadání. Druhý žák s SPU zvládl pouze nakreslit čtverec. Jednalo se o žáka s poruchou pozornosti, který však po chvíli badání začal řešit i další bloky (např. otiskni se).

Největší výzva pro žáky bylo nalézt opakující se vzor.

Dva žáci byli rychlejší než zbytek třídy. Autorka je nechala experimentovat s tloušťkou pera a změnou barvy pera pro rozšíření úlohy.

Změny úlohy:

- Vytvořen pracovní list, který nahradí kreslení do sešitu.

Skupina 4

Žáci z organizačních důvodů pracovali ve dvojicích, protože byli spojeni s druhou polovinou třídy. Každá dvojice dostala vlastní pracovní list, přesto bylo zapotřebí úlohu rozebrat na tabuli. Při kreslení schodů měli žáci potíže s uvědoměním si pořadí bloků v algoritmu.

Skupina 6

Malá část žáků z organizačních důvodů pracovala ve dvojicích, protože byli spojeni s druhou polovinou třídy. Každá dvojice dostala vlastní pracovní list. Tato skupina měla méně času na práci, protože tvořili velmi pomalu zápis do sešitu. Byl jim zadán pouze první úkol v pracovním listu a pak rovnou programovali ve Scratch. Žáci úlohu zvládli dokončit v daném časovém úseku.

Skupina 5

Skupina 5 vypracovávala tuto úlohu jako poslední. Tato skupina pracovala již samostatně a potvrdilo se, že úloha je v dané podobě pro žáky dostatečně pochopitelná a zvladatelná.

Závěr

Původní zadání bylo pro žáky složité. Výuka událostí se v časové dotaci ukázala být velmi obtížnou a byla přesunuta do dalších úloh. Tato úloha zůstává zaměřena pouze na cykly.

Po společném rozboru kresby čtverce žáci zvládali bez problému provést rozbor kresby schodů. Pracovní list snížil časovou náročnost úlohy. Většina žáků však z pracovního listu využila pouze první úkol a pak zbytek objevovala ve Scratch. V pracovním listu byly ponechány i zbývající úkoly, protože mohou sloužit žákům, kteří mají větší potřebu vizualizace.

3.4.3 Úloha 3: Pokládání sklenic

Odkaz na zadání: <https://scratch.mit.edu/projects/829951776/>

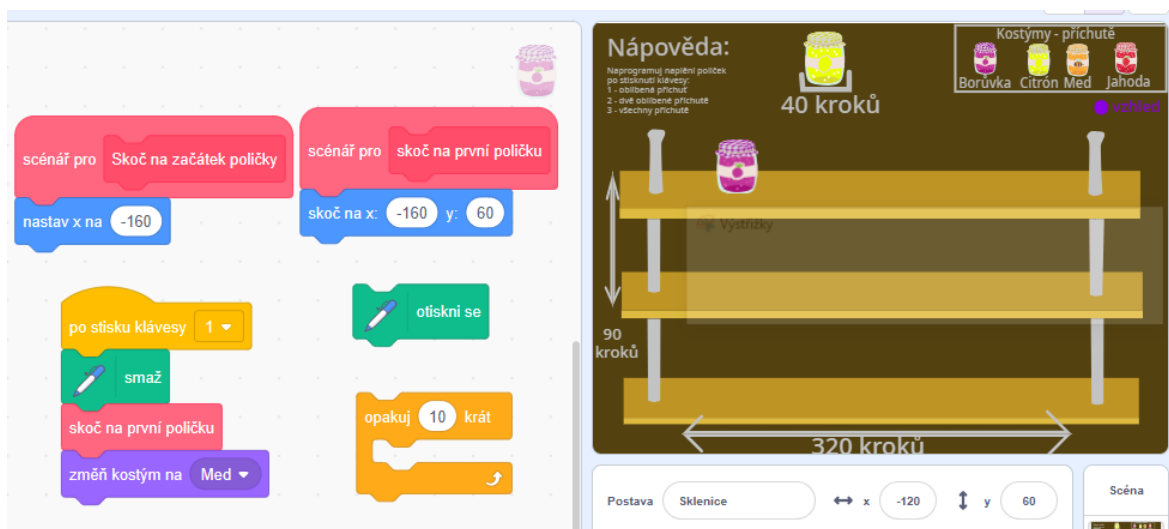
Odkaz na řešení: <https://scratch.mit.edu/projects/931539343>

Výukové cíle

- Žák na příkladu vysvětlí, jak určí začátek algoritmu v programovacím jazyce Scratch.
- Žák nalezne v kódu opakující se vzor.
- Žák použije cyklus s konečným počtem opakování pro zkrácení kódu ve Scratch.
- Žák vytvoří algoritmus pro zadanou úlohu v jazyce Scratch.

Zadání

V budoucnosti nedoplňují zboží do regálů lidé, ale roboti. Vytvoř kód, kterým robotovi pomůžeš zaplnit celou skříň sklenicemi s marmeládou. Pro položení sklenice použij příkaz „otiskni se“, který funguje jako razítko. Kód vytvoř tak, aby se spustil stisknutím klávesy. Po stisknutí klávesy „1“ se zaplní celá skříň sklenicemi s tvou oblíbenou příchutí, po stisknutí klávesy „2“ sklenicemi s dvěma oblíbenými příchutěmi a po stisknutí klávesy „3“ sklenicemi s všemi příchutěmi. Všechny potřebné údaje najdeš v pozadí scény.



Obrázek 8: Úloha č. 3 – Scratch soubor se zadáním

Pokyny pro žáky

Každý kód musí mít začátek. Ve Scratch tento začátek tvoří „Události“. Jakou událostí chceš spustit kód pro pokládání sklenic? Vyber správnou událost.

Pokládání sklenice je pro člověka velmi jednoduchý úkol. Počítač však potřebuje naprogramovat postup krok po kroku. Než budeš programovat, zamysli se, z jakých kroků se skládá pokládání sklenice.

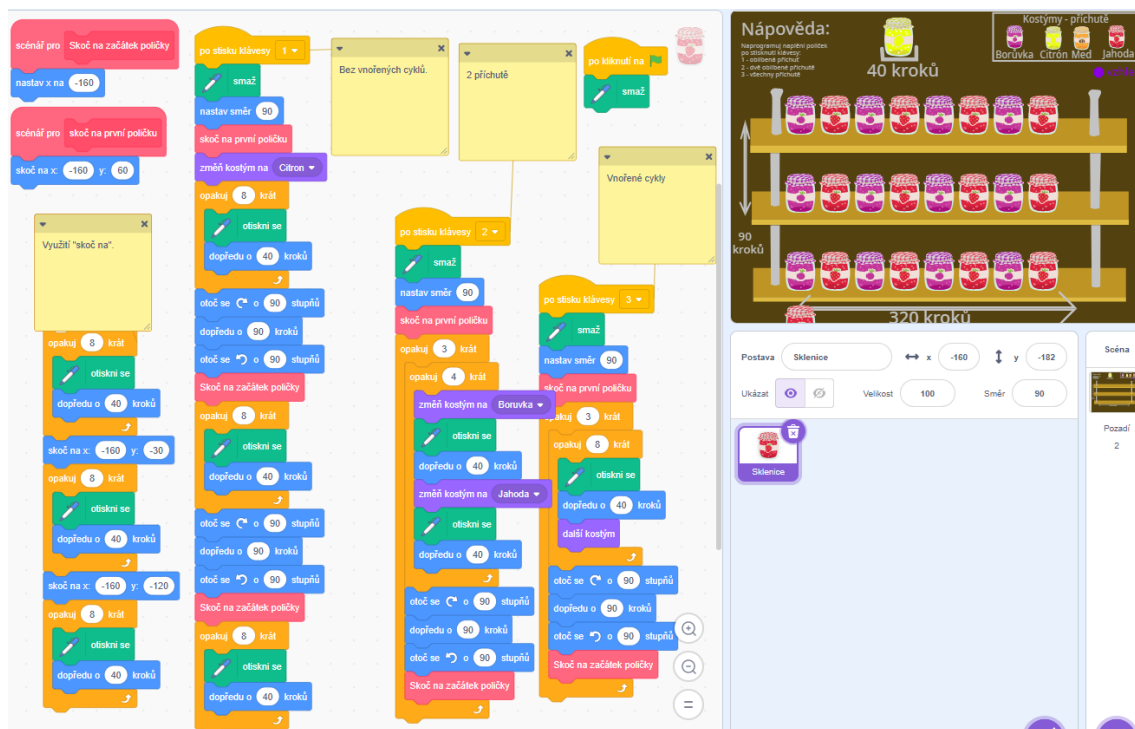
Učitel může položit následující návodné otázky:

- Na jaké činnosti/kroky můžeš rozložit naplnění celé poličky? Co musíš udělat, abys naplnil celou poličku?
- V jakém pořadí budeš činnosti/kroky dělat?
- Jak zjistíš, jak velká musí být mezera mezi sklenicemi?
- Jak zjistíš, kolik sklenic se vejde na poličku?
- Opakuje se nějaká činnost? Jaká a kolikrát?



Pro položení sklenice použij blok „otiskni se“ a pro posunutí použij blok „dopředu o _“. Jak posunout sklenici na další poličku? Pokud potřebuješ sklenici posouvat jiným směrem, tak použij příkaz „otoč se o _“. V případě, že žáci potřebují pomoc, učitel může na tabuli znázornit pohyb sklenice na další poličku.

Ukázka možného způsobu řešení



Obrázek 9: Úloha č. 3 – ukázka možného řešení

Metodický rozbor úlohy

Námět byl vybrán tak, aby reprezentoval pro žáky snadno představitelný problém, jehož řešení vyžaduje konkrétní data (velikost sklenice, délku poličky, název příchutě, ...). Všechny potřebné informace mají žáci v pozadí scény.

Tato úloha slouží k procvičení algoritmizace, zejména pořadí kroků (napřed sklenici položíme, až pak ji přesuneme na další pozici) a hledání vzorů (kód pro položení sklenice a posunutí). V této úloze žáci pracují i s dekompozicí problému (položení jedné sklenice → zaplnění poličky → zaplnění skříně).

Možné varianty úlohy podle náročnosti

- Základní:
 - naplnění skříně sklenicemi s jednou příchutí marmelády.
- Rozšířená:
 - naplnění skříně sklenicemi s dvěma příchutěmi marmelády tak, aby se příchutě pravidelně střídaly,

- naplnění skříně sklenicemi se všemi příchutěmi marmelády,
- zkrácení kódu pomocí vnořených cyklů,
- použití bloků „skoč na pozici“,
- posouvání sklenice pomocí souřadnic x a y. (Po stisknutí mezerníku se změní pozadí na pozadí s nápovědou pro použití osy x a y).
- Zjednodušená:
 - naplnění alespoň jedné poličky sklenicemi s jednou příchutí marmelády.

Vlastní bloky a doporučené příkazy

- Vlastní bloky
 - „skoč na první poličku“: přesune sklenici na začátek první poličky,
 - „skoč na začátek poličky“: přesune sklenici na začátek police.
- Doporučené příkazy:
 - „dopředu o 40 kroků“,
 - „opakuj _ krát“,
 - „otoč o 90 stupňů doleva“ a „otoč o 90 stupňů doprava“,
 - „změň kostým na _“,
 - „další kostým“,
 - „smaž“,
 - „otiskni se“.
- Další příkazy, které mohou žáci využít:
 - „skoč na x:_ y:_“: využít pro posun sklenice na další poličku.

Doporučený průběh hodiny

Úvod a zadání: (10 minut)

- Na začátku učitel zopakuje s žáky základy práce ve Scratch. Zaměřuje se na práci s cykly.
- Vyučující vysvětlí žákům zadání práce.
- Učitel žákům předvede bloky „otiskni se“ a „změň kostým na _“.
- Učitel společně s žáky vytvoří kód, který připraví scénu na pokládání sklenic. Samotný kód pro pokládání sklenic pak žáci tvoří sami.

Samostatná práce: (15 minut)

- Žáci vytvoří kód pro pokládání sklenic na první policičku. (cca 5-10 minut)
- Žáci přemýšlí, jak přesunout sklenici na další policičku a zaplnit celou skříň, následně vytvoří kód. (cca 5–10 minut)

Závěr: (5 minut)

- Na závěr učitel společně se žáky sestaví na tabuli/dataprojektoru možné řešení úlohy.

Možné komplikace

- Žáci používají blok „opakuj 10krát“ bez rozmyšlení. Nemodifikují počet opakování nebo do cyklu vkládají náhodné bloky.
 - Opět zopakovat cykly.
 - Rozebrat krok po kroku příkazy v cyklu. Vést žáky k zjištění, které činnosti potřebují opakovat a které potřebují pouze jednou („smazat“).
 - Nechat žáky vytvořit kód bez cyklů a v něm hledat opakující se vzor.
 - Upozorňovat žáky na informace v pozadí scény.
- Žáci nepoužívají cykly.
 - Viz výše.
- Žáci vkládají do cyklu blok „smazat“.
 - Viz výše.
- Žáci nedokáží přesunout sklenici na další policičku.
 - Znovu rozebrat nákres na tabuli. Případně je možné využít souřadnice na dalším pozadí. (Stisknutím klávesy mezerník se změní pozadí.)

Reflexe úlohy č. 3

Skupina 1

V této skupině byla testována tato úloha po úloze č. 5.

Žáci neměli při řešení větší problémy. Vzhledem k zařazení úlohy po úloze č. 5 někteří žáci měli tendenci používat blok „opakuj dokud nenastane \diamond “. Žáci hledali i vzory v návaznosti na předchozí úlohy (kreslení schodů \rightarrow otáčení sklenice). Všichni žáci, kromě jednoho, zvládli vytvořit řešení pro „jednu příchuť“. Někteří žáci, když experimentovali s otáčením, nebyli schopni sklenici vrátit do původní polohy.

Úprava úlohy:

- Do kódu byl přidán blok „nastav směr na 90 stupňů“, aby se žákům sklenice po různých experimentech vždy vrátila do původní polohy.

Skupina 2

V této skupině byla testovaná tato úloha po úloze č. 4.

Během samostatné práce vznikly dvě skupiny. První skupina žáků pracovala samostatně a nepotřebovala žádnou pomoc, pouze občasné upozornění na možnost využití vnořených cyklů a existence bloku „další kostým“. Žáci opět měli tendenci řešit úlohu pomocí „opakuj dokud nenastane <dotýkáš se barvy _>“. Druhá skupina potřebovala větší vedení a pokládání návodných otázek. Tito žáci tím získali potřebné sebevědomí při práci s aplikací Scratch, začali skládat kód (předtím pouze používali bloky jako tlačítka) a měli radost, že konečně dokončili úlohu.

Úprava úlohy:

- Při zadání úlohy více zdůraznit, aby žáci neupravovali vlastní bloky. Vlastní bloky však zůstávají ukázané, aby žáci mohli alespoň pasivně chápat jejich funkci a žáci s větším zájmem o programování mohli pochopit i samotný předpřipravený program.

Skupina 3

V této skupině žáci neměli s prací na úloze větší problém. Dva žáci byli výrazně rychlejší než zbytek třídy. Tito žáci pod dohledem vyučujícího pomáhali ostatním žákům tím, že jim kladli návodné otázky, řešení spolužákům nesměli nadiktovat. Tento postup se osvědčil.

Skupina 4, 5 a 6

Většina žáků zvládla naplnit celou skříň sklenicemi stejné příchutě, všichni žáci zvládli zaplnit alespoň jednu poličku za použití cyklu. Žáci rozuměli cyklům a byli schopni používat číselné informace z pozadí (velikost sklenice, ...) a zpracovat je do svého kódu. Několik žáků mělo problém s posouváním sklenice, počet kroků odhadovali metodou pokus omyl.

Závěr

Tato úloha je propojením mezi úlohou č. 2 a úlohou č. 4, vznikla, protože úloha č. 4 byla pro žáky původně příliš složitá. Prohlubuje ovládání cyklů a uvědomění si důležitosti pořadí

příkazů v algoritmu. Finální podoba úlohy je pro žáky přiměřeně složitá a rozvíjí vybrané dovednosti.

3.4.4 Úloha č. 4: Pokládání sklenic do poliček rozdílné délky

Odkaz na zadání: <https://scratch.mit.edu/projects/821868138/>

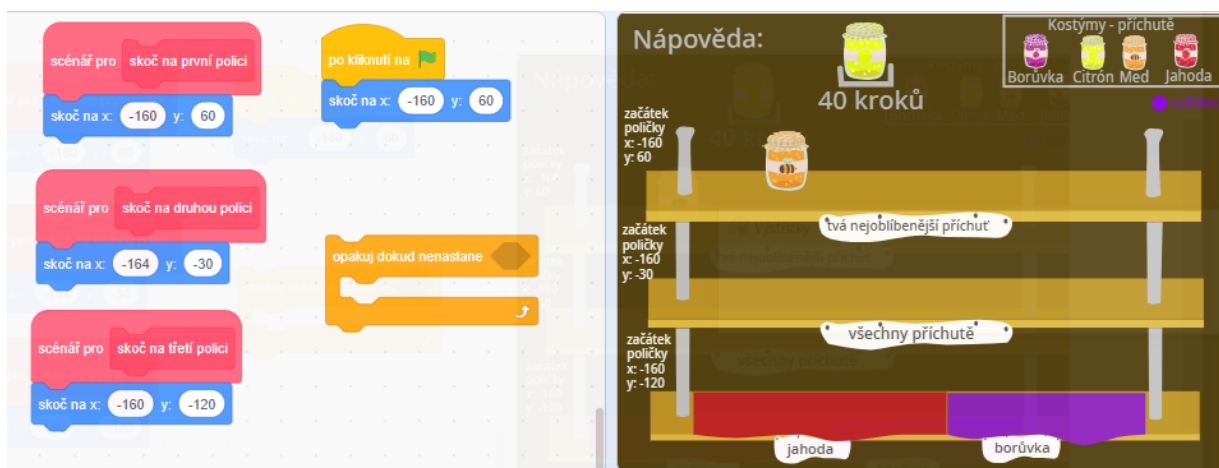
Odkaz na řešení: <https://scratch.mit.edu/projects/821311284/>

Výukové cíle

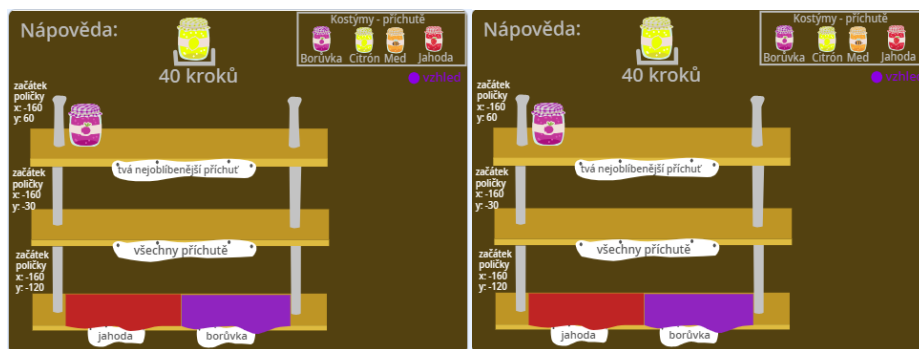
- Žák rozpozná opakující se části kódu.
- Žák určí konec poličky pomocí bloku „dotýkáš se barvy“.
- Žák použije cyklus s podmínkou.
- Žák vytvoří v jazyce Scratch algoritmus pro zadanou úlohu.

Zadání

Minule jsi naprogramoval kód pro naplnění jedné skříně. Všechny skříně však nejsou stejně dlouhé. Zkus vytvořit kód, který naplní sklenicemi s marmeládou jakoukoliv skříň. Jak robot/počítač pozná konec poličky? Při tvorbě kódu použij blok „opakuj dokud nenastane \diamond“.



Obrázek 10: Úlohy č. 4 – Scratch soubor se zadáním



Obrázek 11: Úloha č. 4 – ukázka různých délek skříní

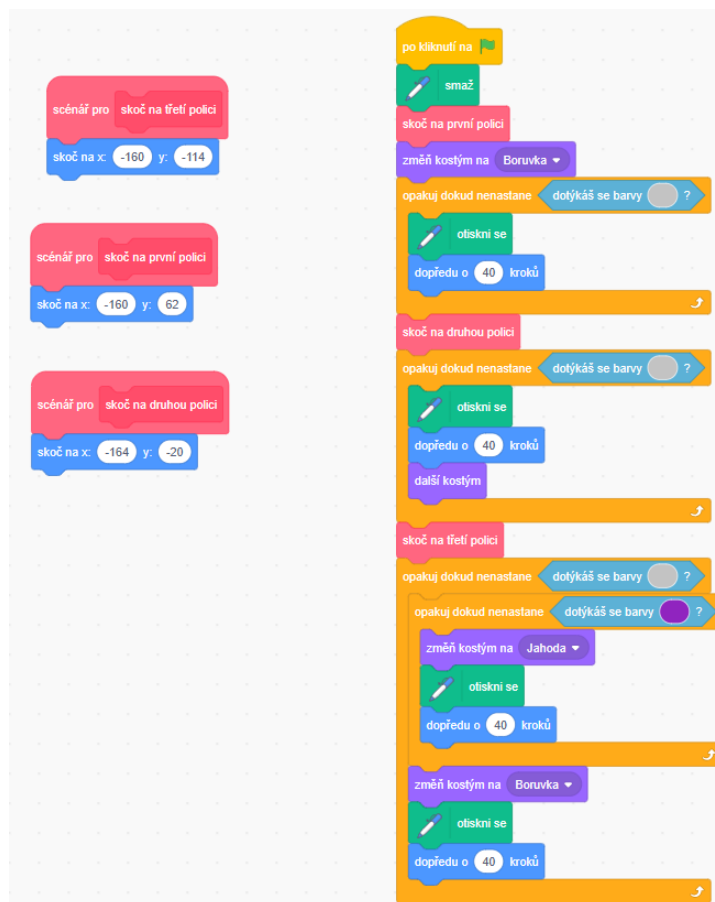
Pokyny pro žáky

V této úloze vytvoříš univerzální kód, který dokáže zaplnit sklenicemi s marmeládou skříně jakékoliv délky. Napřed si zkus vzpomenout, co už umíš naprogramovat. Jak se dnešní kód bude lišit od kódu, který jsi vytvořil minule? Rozděľ kód na části, které už znáš a které musíš vymyslet.

Učitel může použít následující návodné otázky:

- Jak se dnešní kód bude lišit od kódu, který jsi vytvořil minule? (Viz návodné otázky z úlohy č. 3)
 - Rozděľ kód na části, které už znáš a které musíš vymyslet.
- Opakují se nějaké činnosti?
- Jak poznáš, že už se ti na poličku nevejde další sklenice? Jak by to mohl poznat počítač/robot?

Ukázka možného způsobu řešení



Obrázek 12: Úloha č. 4 – ukázka možného řešení

Metodický rozbor úlohy

Úloha navazuje na úlohu č. 3, ve které žáci pokládají sklenice s marmeládou do skříně. V této úloze žáci zobecňují řešení z minulé úlohy, vytváří kód pro poličky různé délky. Z hlediska programování se žáci setkávají s podmínkami. Z hlediska rozvoje IM žáci navazují na hledání vzorů, včetně hledání vzorů v již známých postupech řešení, dále rozvíjí dovednost dekompozice, dělí úlohu na části, které již umí vyřešit, a části, jejichž řešení musí vymyslet. Využijí abstrakci pro rozeznání podstatných informací pro ukončení cyklu pokládání sklenic. Žáci si musí položit otázky: Jak poznám, že skončila polička? Jak to pozná robot nebo počítač?

Chronologicky tato úloha vznikla před úlohou č. 3. Samotná první část této úlohy, pokládání sklenic, zabrala velké části žáků všechen čas pro tvorbu řešení, proto je pokládání sklenic věnovaná celá úloha č. 3.

Možné varianty úlohy podle náročnosti

- Základní:
 - zaplnění dvou poliček sklenicemi s marmeládou.
- Rozšířená:
 - zaplnění třetí poličky sklenicemi s marmeládou.
- Zjednodušená:
 - zaplnění pouze první poličky,
 - použití vlastních bloků místo „skoč na x: _ a y: _“.

Vlastní bloky a doporučené příkazy

- Vlastní bloky
 - „skoč na začátek první poličky“, „skoč na začátek druhé poličky“, „skoč na začátek třetí poličky“: přesune sklenici na začátek příslušné poličky.
- Doporučené příkazy:
 - „opakuj dokud nenastane <>“,
 - „dopředu o _“,
 - „skoč na x: _ y: _“,
 - „otiskni se“,
 - „změň kostým na _“,
 - „další kostým“.
- Další příkazy, které mohou žáci využít:
 - v rámci výzkumného šetření žáci nepoužívali jiné příkazy.

Doporučený průběh hodiny

Úvod a zadání: (10 minut)

- Na začátku učitel zopakuje s žáky základy práce ve Scratch. Zaměřuje se na práci s cykly.
- Vyučující vysvětlí žákům zadání práce.
- Učitel žákům předvede blok „opakuj dokud nenastane <>“.
- Učitel společně s žáky rozebere úlohu a zopakuje, co žáci dělali minulou hodinu.

Samostatná práce: (15 minut)

- Žáci samostatně programují.

Závěr: (5 minut)

- Společný rozbor řešení úlohy.

Možné komplikace

- Žáci mají problém s naprogramováním pokládání sklenic.
 - Viz úloha č. 3.
- Žáci nepoužívají nástroj „kapátko“ pro výběr barvy v podmínce.
 - Znovu ukázat nástroj „kapátko“.
- Žáci nerozumí podmínce v cyklu.
 - Převést na reálný příklad. (Jak poznáš, že je polička plná? Jak poznáš, že se ti na poličku nic nevejde? Jak by to poznal robot? Co mu brání v cestě?)
- Sklenice se nepřesouvá na začátek poličky.
 - Žáci občas mají problém se zápornými čísly v souřadnicích – neopisují znaménko „mínus“.

Reflexe úlohy č. 4

Skupina 1

Úloha byla zařazena před úlohou č. 3. Úloha byla pro žáky velmi složitá. Žákům dělalo problém v časové dotaci vymyslet jen samotný kód pro pokládání sklenic. Řešení úlohy žáci dodělali pouze s velkou dopomocí vyučujícího.

Skupina 2

Úloha byla zařazena před úlohou č. 3. Úloha byla pro žáky subjektivně ještě složitější než pro žáky první skupiny. Zejména to bylo zjevné u žáků, kteří nevypracovali první úlohu. Řešení úlohy č. 4 bylo tvořeno společným rozbohem a diskuzí.

Subjektivní obtížnost byla pro žáky demotivující, proto bylo nutné vytvořit úlohu, která žákům dodá sebevědomí a zopakuje základy práce ve Scratch.

Změna úlohy:

- Vytvoření úlohy č. 3, která je zaměřena pouze na pokládání sklenic. Zároveň zahrnuje události, které měly být před zjednodušením úlohy č. 2 její součástí.

- Odstraněn předpřipravený kód, aby žáci vytvářeli celé řešení samostatně.

Skupina 3

Úloha byla zařazena po úloze č. 3. Pro tuto skupinu byla úloha přiměřeně složitá. Opět byli dva žáci výrazně rychlejší než zbytek třídy. Tito žáci pomáhali spolužákům. Velmi zdařile pokládali návodné otázky a nikomu nediktovali řešení.

Změna úlohy:

- Do pozadí byla přidána nápověda se souřadnicemi začátku políčky.

Skupina 4

Bez větší dopomoci zvládli vypracovat úlohu všichni žáci s výjimkou jednoho, který do cyklu vložil i příkaz „smaž“. Úloha byla pro žáky subjektivně jednoduchá. Proto se autorka rozhodla do úlohy zařadit práci se souřadnicemi, která dělala žákům v první cyklu výzkumného šetření problém v úloze č. 8.

Změna úlohy

- Přidán začátek kódu, ve kterém je blok „skoč na začátek první políčky“ nahrazen blokem „skoč na x:_ y:_“.

Skupina 5 a 6

Úloha byla pro žáky přiměřeně složitá, používali cykly s porozuměním. Žáci také dokázali rozdělit úlohu na část, kterou již umí naprogramovat, a část, kterou musí vymyslet. Největší problém dělaly žákům souřadnice se záporným číslem. Často si neuvědomovali, že do souřadnice musí opsat i znaménko mínus. Po upozornění však neměli s použitím souřadnic žádné větší problémy.

Závěr

Zadání úlohy bylo zapotřebí několikrát upravit. Úloha se zaměřuje na použití podmínek v cyklech. Z hlediska IM se nejvíce zaměřuje na dekompozici (rozdělení úlohy na část, kterou žáci umí, a část, kterou žáci teprve musí vytvořit) a abstrakci (co je podstatné pro zobecnění úlohy z minulé hodiny). Původně byla úloha zařazena jako třetí v pořadí, ukázalo se však, že pro žáky 5. ročníku byla příliš složitá. Po zařazení nové úlohy č. 3 byla tato úloha přiměřeně složitá a žáci se mohli věnovat rozvoji IM. V druhém cyklu výzkumného šetření

byla v rámci úlohy rozvíjena práce se souřadnicemi, která se ukázala jako důležitá v úloze č. 8.

3.4.5 Úloha č. 5: Pokládání sklenic na podložku

Odkaz na zadání: <https://scratch.mit.edu/projects/821686378/>

Odkaz na řešení: <https://scratch.mit.edu/projects/821668124/>

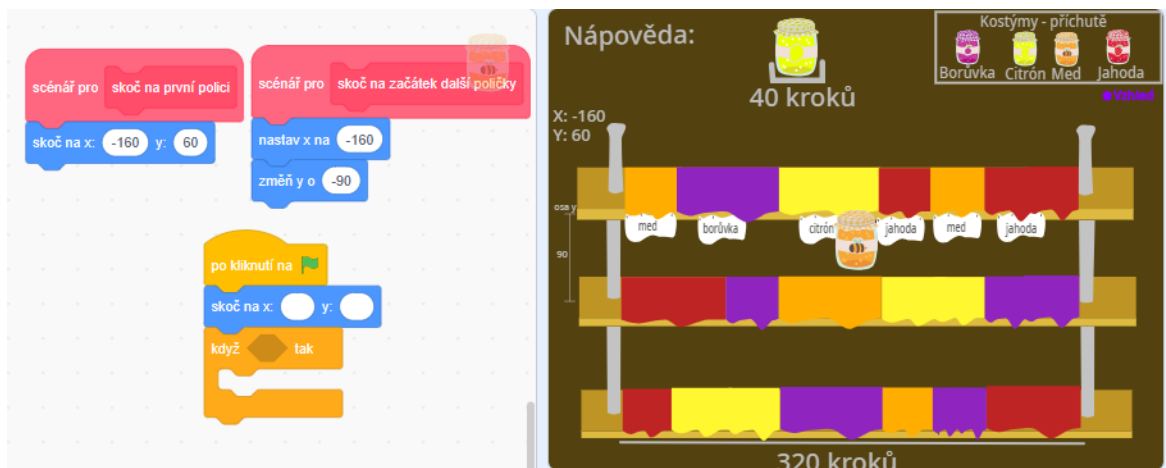
Odkaz na zjednodušenou variantou: <https://scratch.mit.edu/projects/972999779/>

Výukové cíle

- Žák rozpozná opakující se část kódu a využije cyklus s konečným počtem opakování pro zkrácení kódu.
- Žák naprogramuje změnu kostýmu podle barvy podložky za využití příkazu „když“ a podmínky „dotýkáš se barvy _“.
- Žák vytvoří v jazyce Scratch algoritmus pro zadanou úlohu.

Zadání

Už dokážeš vytvořit kód pro pokládání sklenic do libovolně dlouhé poličky. Stále však musíš „ručně“ vybírat příchut'. Zkus vytvořit kód, který podle barvy podložky automaticky vybere sklenici se správnou příchutí a sklenici položí na podložku. Pro rozpoznání příchutě využij bloky „když \diamond tak“ a „dotýkáš se barvy _“.



Obrázek 13: Úloha č. 5 – Scratch soubor se zadáním

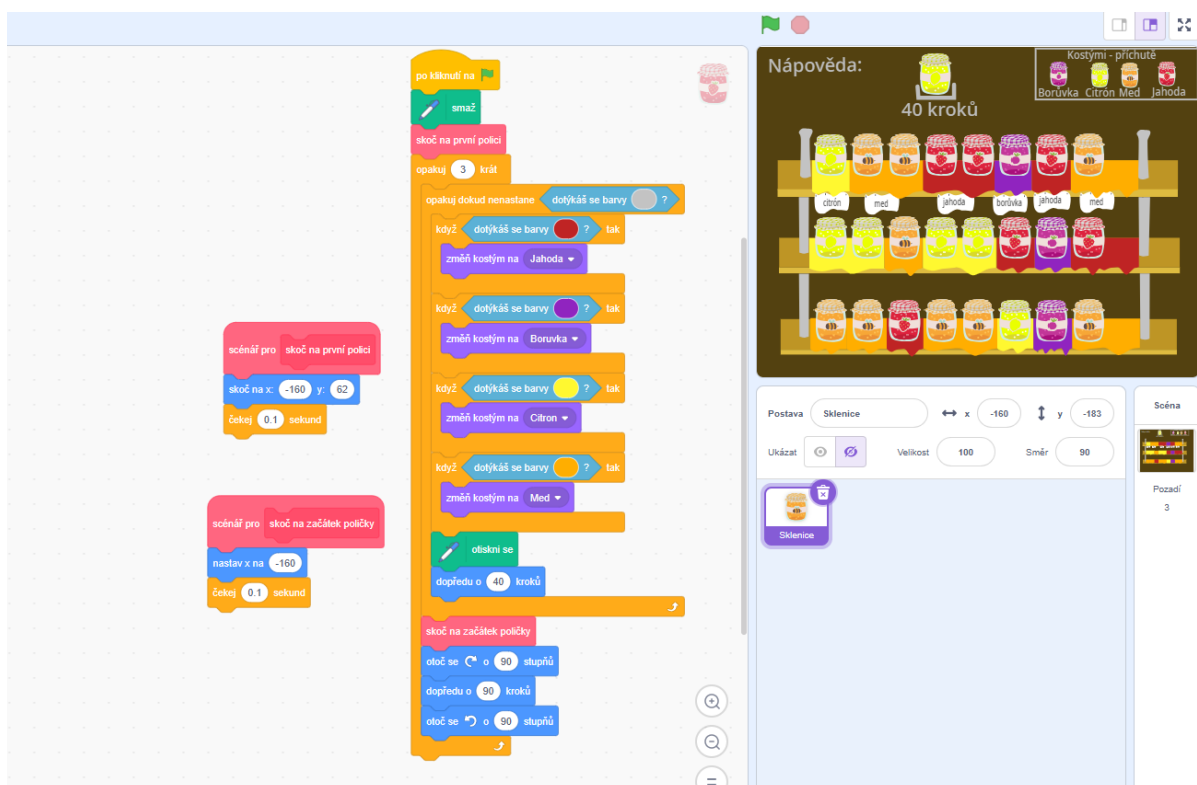
Pokyny pro žáky

V této úloze vytvoříš kód, který zvolí správnou příchut' marmelády podle barvy podložky. Napřed si zkus vzpomenout, co už umíš naprogramovat. Jak se dnešní kód bude lišit od kódu, který jsi vytvořil minule? Rozděl kód na části, které znáš a které musíš vymyslet.

Učitel může použít následující návodné otázky:

- Jak se dnešní kód bude lišit od kódu, který jsi vytvořil minule? (Viz návodné otázky z úlohy č. 3)
 - Rozděl kód na části, které znáš a které musíš vymyslet.
- Jak poznáš, jakou zvolit příchut'?

Ukázka možného způsobu řešení



Obrázek 14: Úloha č. 5 – ukázka možného řešení

Metodický rozbor úlohy

Žáci se seznamují s příkazem „když“. Stejně jako u této úlohy je její dekompozice na již známé problémy, k tomu by měl směřovat i rozbor úlohy. Žáci provádí i dekompozici řešení

úlohy, rozlišení 2 samostatných částí kódu: nastavení kostýmu (příchutě) a pokládání sklenice.

Možné varianty úlohy podle náročnosti

- **Základní:**
 - zaplnění všech políček sklenicemi s marmeládou.
- **Rozšířená:**
 - využití podmínek v cyklu; po stisknutí klávesy „mezerník“ se změní pozadí na skříň o jiné délce.
- **Zjednodušená:**
 - zaplnění pouze jedné poličky,
 - pro pohyb sklenice využít blok „skoč na x:_ y:_“,
 - varianta úlohy zaměřená na debugging, ve které žáci opravují kód (viz závěr úlohy č. 5).

Vlastní bloky a doporučené příkazy

- **Vlastní bloky:**
 - „skoč na začátek první poličky“, „skoč na začátek další poličky“: pro pohyb sklenice na další poličku.
- **Doporučené příkazy:**
 - viz úloha 3,
 - „když <> tak“,
 - „dotýkáš se barvy _“.
- **Další příkazy, které mohou žáci využít:**
 - „opakuj dokud nenastane <>“.

Doporučený průběh hodiny

Úvod a zadání: (5 minut)

- Učitel žákům předvede blok „když <> tak“.
- Učitel vysvětlí žákům zadání práce.
- Učitel společně s žáky rozebere úlohu a zopakuje, co žáci dělali minulou hodinu.

Samostatná práce: (20 minut)

- Žáci samostatně pracují na zadání úlohy.

Závěr: (5 minut)

- Společný rozbor řešení úlohy.

Možné komplikace

- Žáci mají problém s naprogramováním pokládání sklenic.
 - Viz úloha č. 3.
- Žáci nepoužívají nástroj „kapátko“ pro výběr barvy v podmínce.
 - Znovu ukázat nástroj „kapátko“.
- Sklenice se nepřesouvá na začátek poličky.
 - Viz úloha č. 4.

Reflexe úlohy č. 5

Skupina 1

Úloha byla zařazena před úlohou č. 3. Pro žáky byla subjektivně velmi složitá. Bylo zapotřebí věnovat více času rozboru úlohy a pokládat více návodných otázek. Samostatně úlohu nezvládl dokončit žádný žák, což jen podpořilo nutnost vytvoření jednodušší úlohy č. 3.

Skupina 2

Tato skupina již měla zařazenou úlohu č. 3. Žáci čerpali ze znalostí z úloh č. 3 a č. 4, proto s pochopením úlohy neměli větší problémy. Komplikace žákům dělala kombinace poliček s rozdílnou délkou („opakuj dokud nenastane \diamond “) a využití „když \diamond tak“ . S případnou dopomocí učitele všichni zvládli vytvořit kód dle zadání.

Změna úlohy:

- Byla odstraněna změna pozadí po kliknutí na zelenou vlaječku, aby žáci nemuseli vytvářet kód s „opakuj dokud nenastane \diamond “. Tento úkol byl přesunut do rozšířeného zadání. Pozadí se mění po stisknutí klávesy mezerník.

Skupina 3

Žáci pracovali dle zadání. Úloha byla pro žáky subjektivně obtížnější, ale většina žáků zvládla úlohu vypracovat alespoň v zjednodušeném zadání. V této skupině nastala

neočekávaná situace. Cca 8 minut před koncem hodiny musela třída z provozních důvodů opustit počítačovou učebnu a přestěhovat se do učebny bez počítačů. Z těchto důvodů část žáků nezvládla uložit svoji práci.

Skupina 4

Pro žáky byla úloha subjektivně obtížná. Žákům největší problém dělalo propojení cyklů a bloku „když \diamond tak“.

Změna úlohy:

- Vytvoření začátku kódu a připravení bloku „když \diamond tak“.

Skupina 5 a 6

Rozdělení kódu na „rozpoznání příchutě“ a „pokládání sklenice“ bylo v propojení s cykly pro žáky velice obtížné.

Změna úlohy:

- Vytvořena jednodušší varianta zaměřená na debugging (viz závěr úlohy č. 4).

Závěr

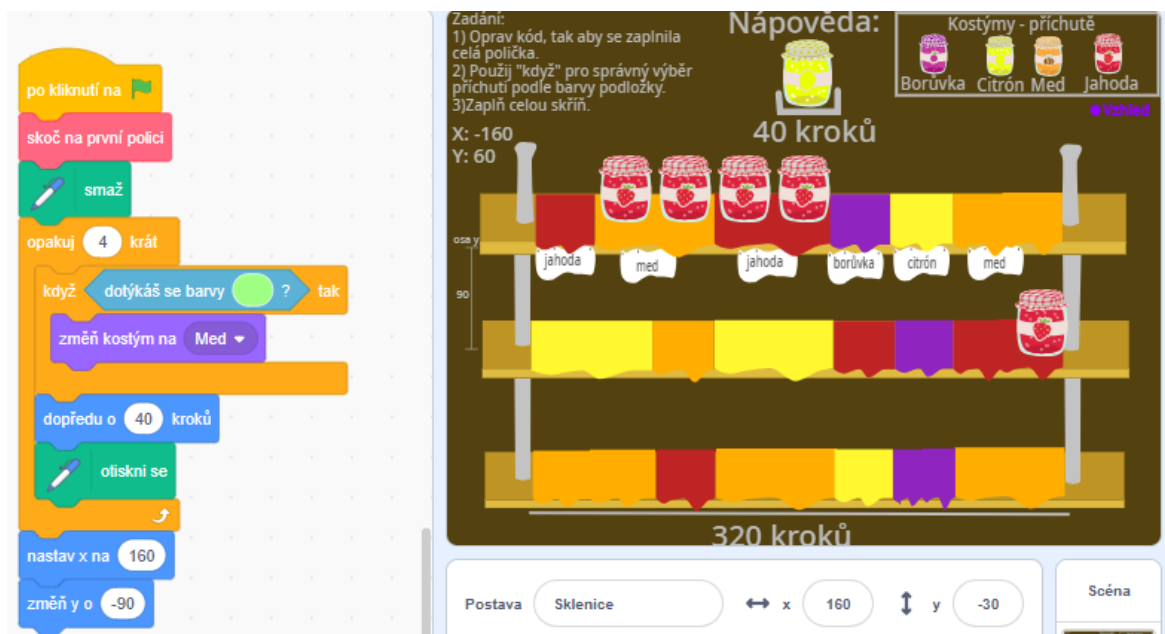
Překvapivě byla úloha pro žáky ve druhém cyklu výzkumného šetření náročnější než v prvním cyklu. Žáci pracovali s předchozími znalostmi, ale dekompozice problému na „rozpoznání příchutě“ a „pokládání sklenice“ ve spojení s opakováním vzorů byla pro žáky nadměrně obtížná. Pro žáky bylo jednodušší vytvořit kód pro „rozpoznávání příchutě a položení sklenice“. Žáci však neměli žádné problémy s používáním souřadnic.



Obrázek 15: Úloha č. 5 – jednodušší řešení

Zjednodušená varianta

Na základě výzkumného šetření byla vytvořena i jednodušší verze zaměřená na práci s chybou. Žáci dostanou předpřipravený kód pro pokládání sklenic, který musí opravit a dodělat dle vzoru. V této variantě je úloha méně zaměřená na dekompozici a více zaměřená na hledání vzorů a algoritmizaci.



Obrázek 16: Úloha č. 5 – Scratch soubor se zjednodušeným zadáním

3.4.6 Úloha č. 6: Hladová myš

Odkaz na řešení: <https://scratch.mit.edu/projects/825953165/>

Výukové cíl

- Žák vytvoří rozbor úlohy do sešitu.
- Žák vytvoří pozadí a postavy ve Scratch.
- Žák vytvoří kód pro pohyb postavy za kurzorem.
- Žák použije nekonečný cyklus pro opakování kódu.
- Žák vytvoří kód pro „zanechávání odpadků“ po dotyku hráčem ovládané postavy s jiným objektem.
- Žák vytvoří algoritmus pro zadanou úlohu v jazyce Scratch.

Zadání

Vytvoř hru, ve které myš jí meloun. Po snědení melounu myš zanechá nesnědený zbytek. Cílem hry je zaplnit celou plochu nesnědenými zbytky – odpadky. Myš se pohybuje za kurzorem – bílou šipkou. Meloun se po snědení objeví znovu na jiném místě.

Pokyny pro žáky

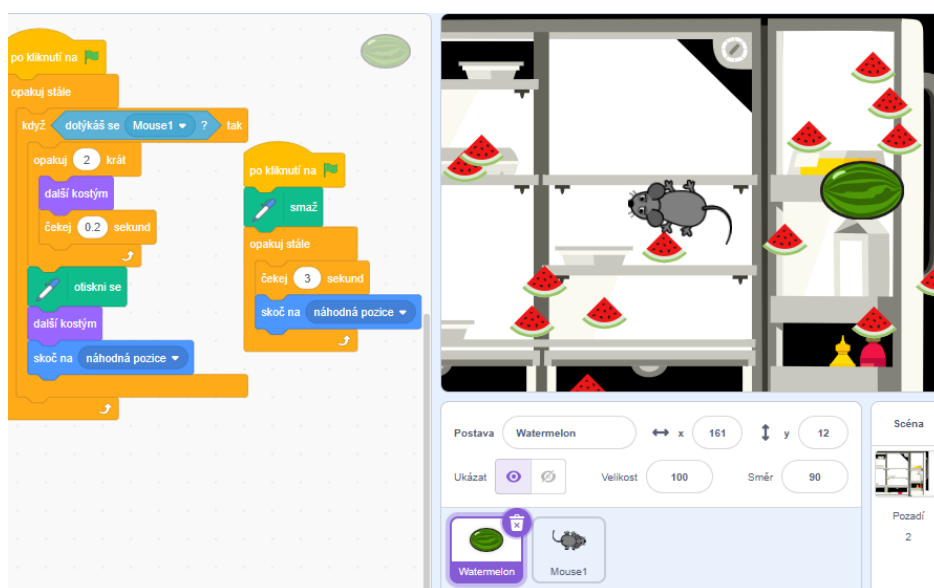
Vytvoř hru pro jednoho hráče. Postava myši jí meloun. Pokaždé, když sní meloun, zanechá odpadek a nový meloun se objeví na jiném místě. Cílem hry je zanechat co nejvíce odpadků. Hráč ovládá postavu myši, která se pohybuje za kurzorem.

Rozděl hru na části „pozadí“ a „postavy“. Jaké budeš potřebovat postavy a pozadí? Jaké činnosti budeš programovat?

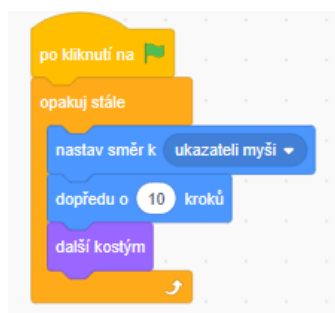
Návodné otázky:

- Ke které postavě naprogramuješ tuto činnost? Která postava vykonává tuto činnost?
- Které bloky budeš k této činnosti potřebovat?
- Kterou počáteční událost použiješ? Jak chceš spustit tento kód?
- Kolikrát se bude tato činnost opakovat?
- Jak dlouho se bude pohybovat myš? (Po celou dobu chodu hry, stále, ...)
- Nепrogramovali jsme již něco podobného? Které bloky jsme k tomu využívali?

Ukázka možného způsobu řešení



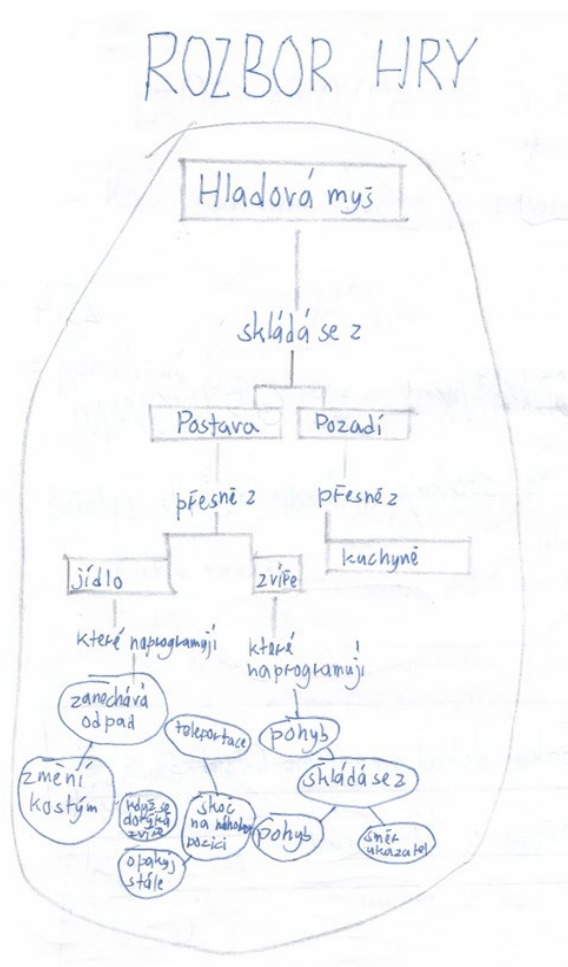
Obrázek 17: Úloha č. 6 – kód postavy melounu



Obrázek 18: Úloha č. 6 – kód postavy myši

Metodický rozbor úlohy

Tato úloha je hlavně zaměřená na dekompozici. Nejdůležitější částí je rozbor zaměřený na rozdělení úlohy na jednotlivé části. Žáci si z rozboru vytvoří zápis (slovní, pojmová mapa, ...). Učitel žákům pomáhá s rozbohem dle potřeb dané skupiny. Z hlediska programování se jedná o první projekt, který žáci vytvoří bez předpřipraveného souboru.



Obrázek 19: Úloha č. 6 – ukázka žákovského rozboru

Možné varianty úlohy podle náročnosti

- **Základní:**
 - postava myši se pohybuje za kurzorem; při dotyku myši a melounu meloun změni kostým, otiskne se a skočí na náhodnou pozici.
- **Rozšířená:**
 - přidat animaci postavy myši a melounu,
 - i když se myš melounu nedotkne, tak meloun sám po chvíli skočí na jiné místo,
 - žáci mohou pracovat s jinými postavami.
- **Zjednodušená:**
 - tato úloha nepracuje se zjednodušenou variantou.

Vlastní bloky a doporučené příkazy

- **Vlastní bloky:**
 - nejsou.
- **Doporučené příkazy:**
 - „nastav směr na _ stupňů“,
 - „dopředu o _ kroků“,
 - „když <> tak a dotýkáš se postavy _“,
 - „čekej _ sekund“,
 - „otiskni se“,
 - „další kostým“,
 - „skoč na náhodnou pozici“.
- **Další příkazy, které mohou žáci využít:**
 - „čekej dokud nenastane <>“: lze tímto blokem nahradit „když <> tak“,
 - „dotýkáš se barvy _“: učitel by měl vést žáky k využití „dotýkáš se postavy _“, aby se žáci naučili i další blok, který mohou využít v dalších projektech.

Doporučený průběh hodiny

Úvod a zadání: (5-10 minut)

- Učitel ukáže žákům hotový program.

- Učitel společně s žáky rozebere úlohu, zaměří se na její dekompozici. S žáky rozdělí hru na postavy a pozadí. U každé postavy rozebere, které činnosti pro ni budou programovat.

Samostatná práce: (15-20 minut)

- Žáci vytvoří zápis z rozboru.
- Žáci programují hru.

Závěr: (5 minut)

- Společný rozbor řešení.

Možné komplikace

- Žák programuje kód k nesprávné postavě.
 - Viz návodné otázky a rozbor úlohy v sešitě.
- Žáci porovnávají řešení mezi sebou a předělávají svůj kód podle řešení spolužáků.
 - Zdůraznit, že úloha může mít více správných řešení.

Reflexe úlohy č. 6

Skupina 2

Žáci naprogramovali pohyb postavy myši, nebo jiného zvířete, a zanechávání odpadků. Žákům bylo zapotřebí více vysvětlit „nastavení směru“ a „opakuj stále“. Žáky aktivita velmi zajímala, vytvářeli vlastní postavy a kostýmy. Pouze dva žáci stihli vytvořit animaci pohybu postavy změnou kostýmů.

Změna úlohy:

- Změna kostýmu při pohybu postavy myši byla zařazena do rozšířené varianty.

Skupina 3

Žáci s úlohou neměli větší problémy a úloha žáky zaujala. Někteří žáci, když spatřili u „šikovného“ spolužáka jiné řešení, zpanikařili a začali od něj opisovat. Bylo třeba zdůraznit, že úloha může mít více správných řešení.

Změna úlohy:

- Zdůraznit více možných správných řešení. (Jak poznáte, že máte úlohu správně naprogramovanou? Postavy dělají to, co vy chcete.)

Skupina 1

Žáci z provozních důvodů vypracovávali tuto úlohu jako poslední v 1. cyklu výzkumného šetření. I tuto skupinu úloha bavila, někteří žáci však nevytvořili rozbor do sešitu. Žáci měli tendenci používat blok „dotýkáš se barvy _“ místo bloku „dotýkáš se postavy _“.

Změna úlohy:

- Zdůraznit „dotýkáš se postavy _“.
- Více dohlížet na zápis rozboru úlohy.

Skupina 4, 5, 6

Ve druhém cyklu s touto úlohou žáci neměli žádné větší potíže. Drobné nedostatky měli někteří žáci při práci s pořadím kroků (postava melounu nejdřív skočila na náhodnou pozici a až pak se otiskla). Úloha byla přiměřeně složitá, žákům se líbila a pracovali s chutí.

Závěr

Pro autorku bylo překvapující, že žáci stále měli potíže s pořadím kroků v kódu. U této úlohy, kterou šlo vyřešit více velmi podobnými způsoby, bylo nutné zdůrazňovat ověření správnosti řešení spuštěním aplikace, nikoliv srovnáváním kódu se spolužáky. Jednalo se o pro žáky nejzábavnější úlohu z tohoto souboru, zvláště pokud měli žáci možnost si vybrat vlastní postavy.

3.4.7 Úloha č. 7: Akvárium

Odkaz na řešení: <https://scratch.mit.edu/projects/839783060/>

Výukové cíle

- Žák vytvoří rozbor úlohy.
- Žák vytvoří pozadí a postavy ve Scratch.
- Žák vytvoří kód pro „plavání“ postavy ryby za použití vlastního bloku.
- Žák použije blok „klonuj sebe“ pro vytvoření dalších ryb.

- Žák vytvoří algoritmus pro zadanou úlohu v jazyce Scratch.

Zadání

Vytvoříš vlastní animaci rybiček, které plavou v akváriu. Nebude to jen tak ledajaké akvárium, bude kouzelné. Na celou animaci potřebuješ pouze jednu postavu! Využiješ k tomu blok „klonuj sebe“ ze skupiny ovládání. Kód pro plavání ryby vytvoř jako vlastní blok, který pojmenuješ „Plavání“. Nestačí ale pouze blok pojmenovat, musíš rybu „naučit“ plavat, naprogramovat jí kód pro plavání.

Pokyny pro žáky

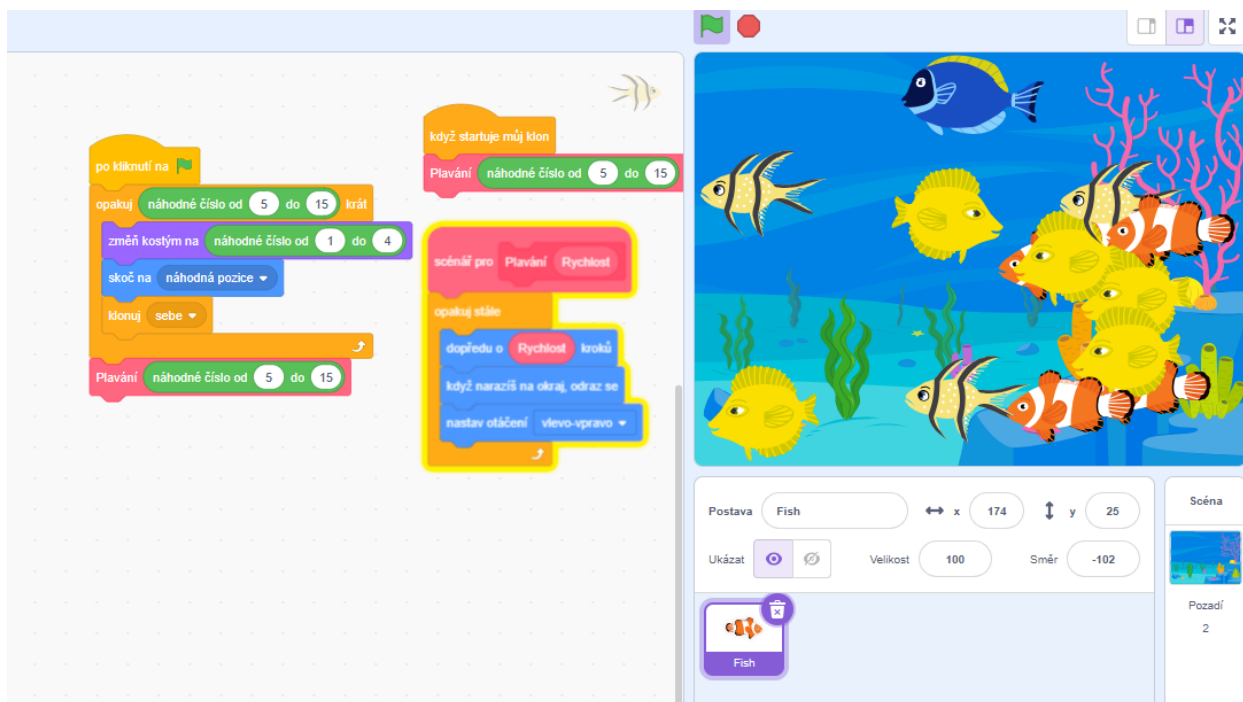
Vytvoř animaci 10 ryb plavajících v akváriu. Pro celou animaci použij pouze jednu postavu. Plavání ryby naprogramuj jako růžový „vlastní blok“ „Plavání“. „Vlastní bloky“ jsou části kódu, které si pojmenuješ. Ve svém kódu si pak tuto část kódu voláš jejím názvem a neprogramuješ ji znovu.

Opět si udělej rozbor animace. Z čeho se animace skládá? Jaké činnosti musíme rybě naprogramovat?

Návodné otázky:

- viz úloha č. 6
- Z jakých činností se skládá plavání ryby? Které bloky k tomu budeš potřebovat?
- Je klonování součástí plavání? Rozmnožuje se ti ryba, když si jen tak plave v akváriu?

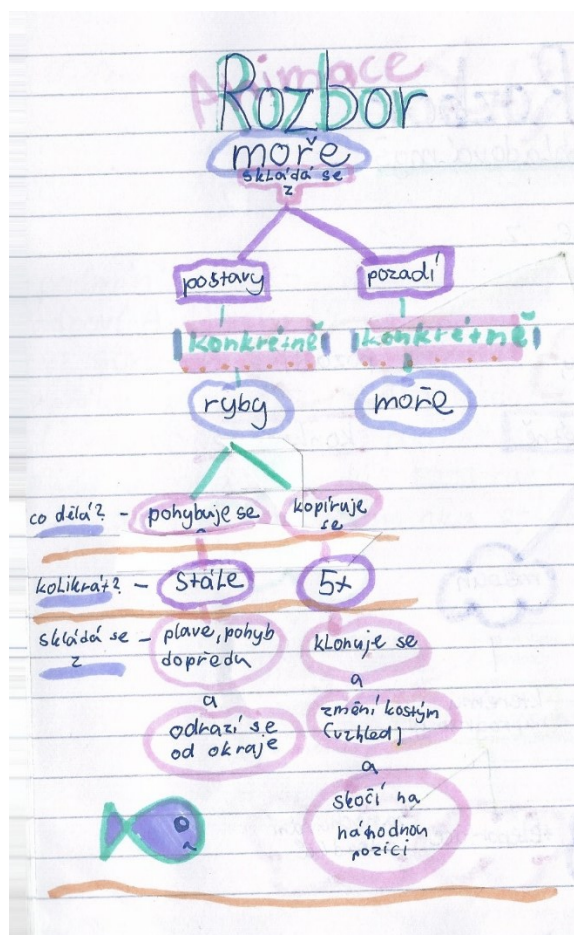
Ukázka možného způsobu řešení



Obrázek 20: Úloha č. 7 – ukázka možného řešení

Metodický rozbor úlohy

Tato úloha je zaměřena hlavně na dekompozici a abstrakci. S dekompozicí navazuje na úlohu č. 6. Žáci opět udělají podrobný rozbor úlohy. V rámci abstrakce žáci klonují a vytvoří kód pro plavání, určí, které činnosti jsou pro plavání podstatné.



Obrázek 21: Úloha č. 7 – ukázka žakovského rozboru

Možné varianty úlohy podle náročnosti

- Základní:
 - žák vytvoří animaci s deseti rybami,
 - jednotlivé klonované ryby mají nastaven vzhled pomocí bloku „další kostým“.
- Rozšířená:
 - náhodná čísla pro počet ryb, rychlost plavání a číslo kostýmu,
 - vlastní blok pro tvorbu ryby („Vytvoř rybu“), který obsahuje klonování a změnu kostýmu.
- Zjednodušená:
 - ryby nemění kostým,
 - ryby se neklonují. (velké zjednodušení)

Vlastní bloky a doporučené příkazy

- Vlastní bloky:
 - „Plavání“ – žáci tvoří sami
- Doporučené příkazy
 - „dopředu o _ kroků“
 - „když narazíš na okraj, odraz se“
 - „klonuj sebe“
 - „když startuje můj klon“
 - „skoč na náhodnou pozici“
- Další příkazy, které mohou žáci využít:
 - „klouzej _ sekund na náhodnou pozici“ – pohyb ryby
 - „když <> tak“ a „otoč se o 180 stupňů“ – pro odražení od okraje

Doporučený průběh hodiny

Úvod a zadání: (10 minut)

- Učitel ukáže žákům hotový program.
- Učitel společně s žáky rozebere úlohu.
- Učitel ukáže tvorbu vlastního bloku a vysvětlí, k čemu slouží.
- Učitel demonstruje klonování a představí blok „Když startuje můj klon“ („Událost“ pro naklonovanou rybu. Sem naprogramuj, co bude dělat tvoje ryba.)

Samostatná práce: (15 minut)

- Žáci vytvoří zápis z rozboru.
- Žáci pracují na úloze.

Závěr: (5 minut)

- Společný rozbor řešení úlohy.

Možné komplikace

- Žák nerozumí „vlastním blokům“.
 - Přiblížit žákům „vlastní bloky“ jako pojmenování části kódu, kterou pak tento blok zastupuje.
- Žák nerozlišuje část kódu pro plavání a část kódu pro tvorbu ryby.

- Přiblížit na reálných příkladech.

Reflexe úlohy č. 7

Skupina 2

Žáci z provozních důvodů pracovali ve dvojicích. Žáci měli problém s tvorbou vlastních bloků a s oddělením bloků pro kódy pro plavání a klonování.

Změna úlohy:

- Klást větší důraz na rozbor.
- Důkladněji vysvětlit vlastní bloky.

Skupina 3

Žáci pracovali samostatně, s úlohou neměli větší problémy a připadala jim zajímavá.

Skupina 1

Žáci pracovali samostatně, ale dvěma žákům dělalo problém určit, které bloky jsou součástí plavání. Pro tuto skupinu byla časová dotace nedostatečná. Z celé skupiny včas dokončil a odevzdal práci pouze jeden žák.

Skupiny 4, 5

Žáci neměli s úlohou větší problémy a pracovali samostatně. Úloha jim připadala zajímavá, ale nikdo nestihl vypracovat rozšířenou verzi, i když se o to několik žáků pokusilo.

Skupina 6

V této skupině odpadly tři hodiny po sobě. Třítýdenní pauza zapříčinila, že někteří žáci měli větší problémy i se samotným programováním a potřebovali pomoc vyučujícího. Nebyli schopni samostatně použít blok „když startuje můj klon“. Práci odevzdali všichni žáci, ale správně ji vyřešili pouze 2.

Závěr

Tato úloha žáky zaujala a pracovali s velkým nasazením, ale ukázalo se, že časová dotace nebyla vždy dostatečná. Přestože žáci neměli se samotným řešením úlohy větší problémy, někteří potřebovali více času pro její zdárné dokončení. Nejobtížnější pro žáky bylo

pochopení funkce vlastních bloků. Stěžejní pro vypracování úlohy byl její kvalitní rozbor, ve kterém konkretizovali, co je „plavání“ u postavy ryby.

3.4.8 Úloha č. 8: Had a padající jablka

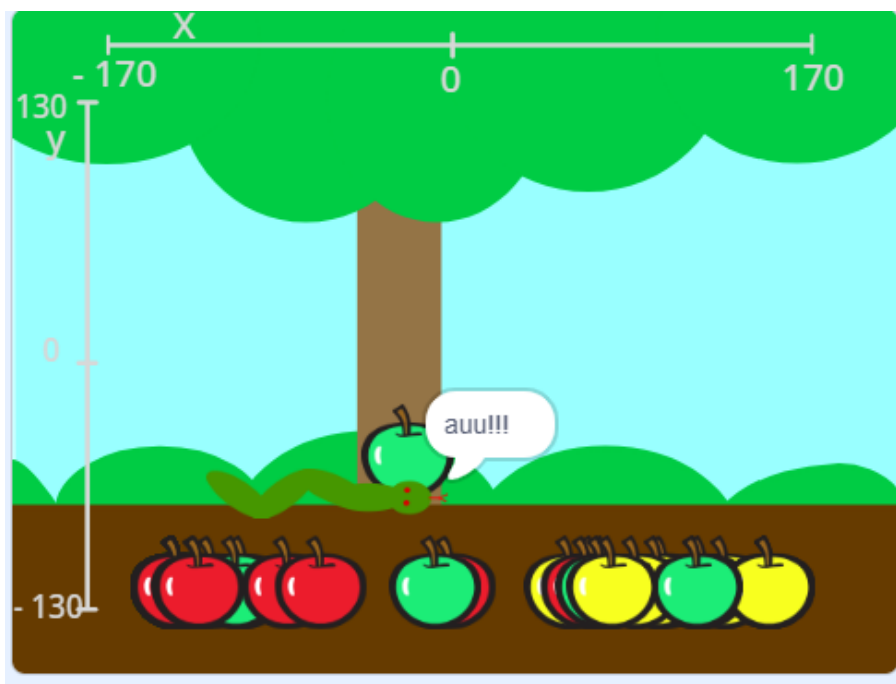
Odkaz na řešení: <https://scratch.mit.edu/projects/847661630/>

Výukové cíle

- Žák vytvoří rozbor úlohy do sešitu.
- Žák nakreslí vlastní postavy ve Scratch.
- Žák nahraje do Scratch pozadí z počítače.
- Žák použije nekonečný cyklus pro opakování kódu.
- Žák vytvoří pohyb postav za pomoci souřadnic.
- Žák vytvoří kód pro „pád“ postavy jablka za použití vlastního bloku.
- Žák vytvoří algoritmus pro zadanou úlohu v jazyce Scratch.

Zadání

Vytvoříš úplně celou hru, včetně návrhu vlastních postav. Ve hře „Had a padající jablka“ se postava hada vyhýbá padajícím jablkům. Když jablko spadne na hada, tak had řekne „au.“



Obrázek 22: Úloha č. 8 – had a padající jablka

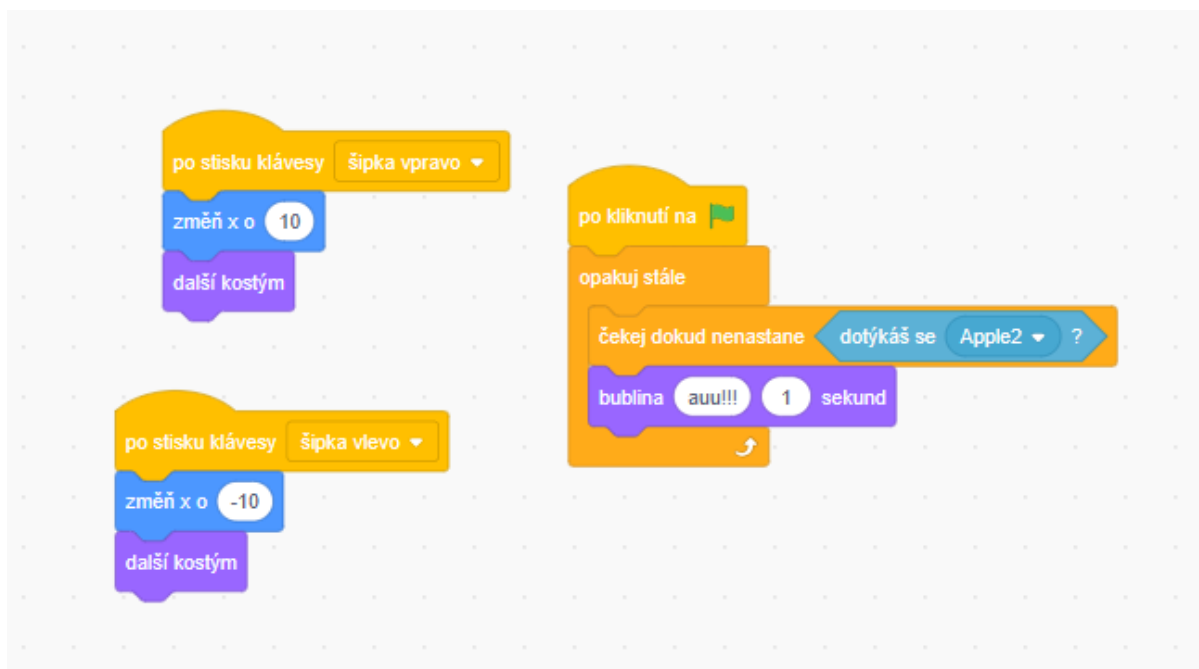
Pokyny pro žáky

Naprogramuj hru pro jedno hráče, ve které se had vyhýbá jablkům. Nakresli a naprogramuj vlastní postavy hada a jablka. Podívej se na ukázkou hry. Z jakých postav a pozadí se skládá hra? Jaké činnosti jednotlivé postavy vykonávají?

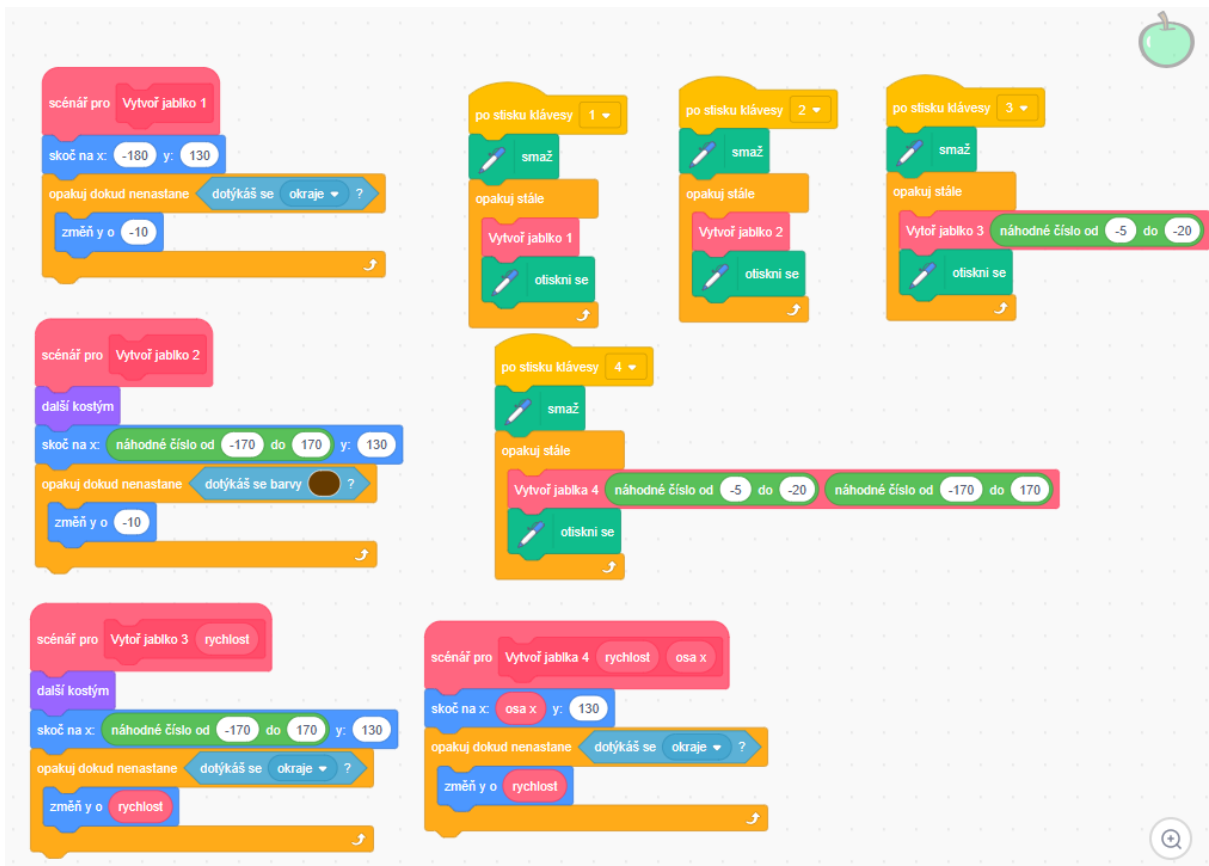
Návodné otázky

- Ke které postavě naprogramuješ tuto činnost? Která postava vykonává tuto činnost?
- Které bloky budeš k této činnosti potřebovat?
- Kterou počáteční událost použiješ? Jak chceš spustit tento kód?
- Jak vypadá hra na začátku?
- Kolikrát se bude tato činnost opakovat?
- Naprogramovali jsme již něco podobného? Které bloky jsme k tomu využívali?
- Z čeho se skládá padání jablka?
- Jak pohybujeme postavy „dozadu“ a „dolů“? (Nakreslit na tabuli osu souřadnic)
- Jak vytvoříme animaci při pohybu postavy hada?
- Jak postava hada pozná, že se dotýká jablka?

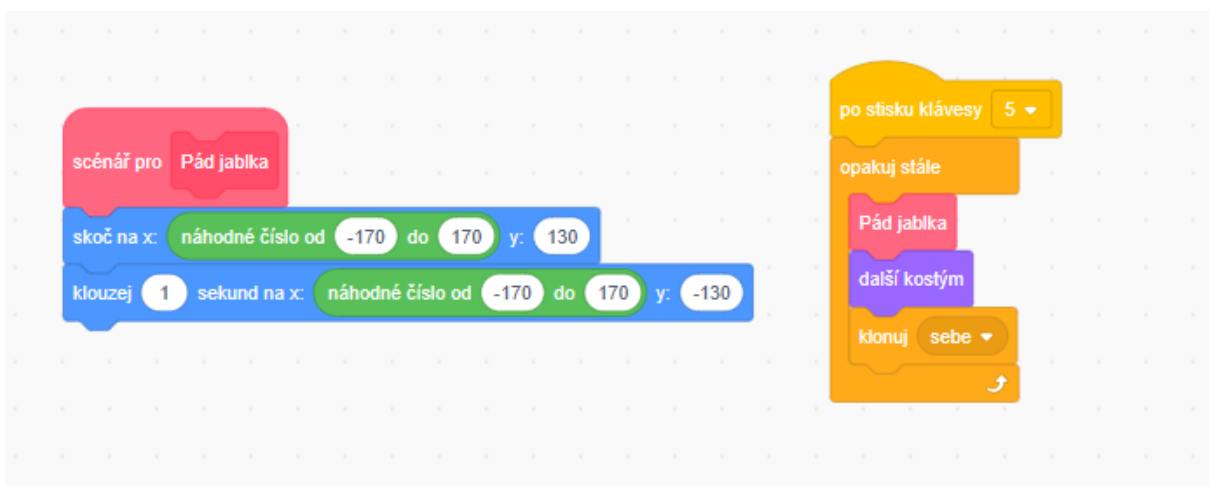
Ukázka možného způsobu řešení



Obrázek 23: Úloha č. 8 – kód pro postavu hada



Obrázek 24: Úloha č. 8 – postava jablka: ukázky různých možných řešení s „opakuji dokud nenastane“



Obrázek 25: Úloha č. 8 – postava jablka: ukázka řešení pouze za pomoci souřadnic

Metodický rozbor úlohy

Tato úloha je závěrečný projekt, ve kterém žáci uplatní vše, co se již naučili v tomto souboru úloh. Žáci vytvoří celý projekt samostatně, včetně návrhu a tvorby vlastní postavy. Zadání umožňuje mnoho různých správných řešení a tím klade větší nároky na učitele, který se musí žákům více věnovat individuálně.

Tato úloha je ve srovnání s ostatními časově náročná, počítá s časovou dotací dvou vyučovacích hodin. Závěrečná úloha je komplexní a vyžaduje pochopení předchozích úloh. Z řešení žáků bude možné velmi dobře možné vyčíst, s čím měli problémy (se souřadnicemi, pořadím kroků, cykly, ...). Učitel musí počítat s tím, že ne všichni žáci dokončí úlohu v plném rozsahu, někteří ji možná nedokončí vůbec. Důležité však není dokončení úlohy, ale způsob, jakým žáci o úloze uvažují.

Možné varianty úlohy podle náročnosti

- **Základní:**
 - hra pro jednoho hráče, ve které se had vyhýbá padajícím jablkům, po doteku s jablkem řekne „au“,
 - jablko padá vždy ze stejného místa,
 - pro zvýšení zájmu dětí je možná nahradit postavy „had“ a „jablko“ vlastními postavami dle jejich volby.
- **Rozšířená:**
 - použití náhodných čísel pro pozicování postavy jablka,
 - vytvoření a naprogramování animace pohybu hada.
- **Zjednodušená:**
 - pro úsporu času žák dostane nakreslené postavy,
 - had nereaguje na pád jablka,
 - žáci pro pád jablka vytvoří vlastní blok,
 - jablko padá pouze jednou,
 - práce ve dvojici,
 - přidat hodinovou dotaci.

Vlastní bloky a doporučené příkazy

- Vlastní bloky:
 - „pád jablka“: žák vytvoří sám.
- Doporučené příkazy:
 - „změň x o $_$, změň y o $_$ “,
 - „opakuj dokud nenastane $\langle \rangle$ “,
 - „dotýkáš se postavy/okraje“,
 - „skoč na x: $_$ y: $_$ “,
 - „otiskni se“,
 - „když $\langle \rangle$ tak“,
 - „bublina 1 sekunda“.
- Další příkazy, které mohou žáci využít:
 - „klouzej na x: $_$ y: $_$ “: nahradí kód pro pád jablka,
 - „klonuj sebe“: nahradí „otiskni se“.

Doporučený průběh hodiny

Tato úloha je rozdělena na dvě vyučovací hodiny.

1. hodina:

Úvod a zadání: (10 minut)

- Učitel s žáky provede rozbor ukázky hry.
- Učitel žákům předvede nahrávání pozadí z počítače (pozadí: „Had a padající jablka - pozadí.png“).
- Učitel žákům předvede tvorbu vlastní postavy.

Samostatná práce: (25 minut)

- Žáci vytvoří zápis z rozboru.
- Žáci vytvoří vlastní postavu „had“.
- Žáci naprogramují pohyb hada.

Závěr: (5 minut)

- Společný rozbor řešení úlohy.

2. hodina:

Úvod a zadání: (5 minut)

- Učitel zopakuje rozbor hry.
- Učitel předvede nahrávání uloženého projektu. (Žákům, kteří chyběli, poskytnete soubor z minulé hodiny.)

Samostatná práce: (25 minut)

- Žáci vytvoří a naprogramují postavu jablka.
- Žáci programují reakci hada na padající jablko.

Závěr: (10 minut)

- Společný rozbor řešení úlohy.

Možné komplikace

- Žáci neví, co mají dělat (chyběli, zapomněli látku, ...).
 - Odkázat je na příslušnou kapitolu v sešitě/prezentaci.
 - Pokud jim nestačí prezentace, individuálně jim pomoc.
- Žáci mají problémy se zápornými hodnotami v souřadnicích.
 - Upozornit na znaménko minus.
- Žáci neví, jak mají naprogramovat, že jablko je na stromě.
 - Znovu vysvětlit souřadnice.
- Žáci nepoužijí „opakuj stále“, jablko spadne jen jednou.
 - Viz „návodné otázky“ (Kolikrát se bude tato činnost opakovat?)
- Žáci chyběli na předchozí hodině.
 - Mít pro ně předpřipravený vzorový výstup předchozí hodiny.

Reflexe úlohy č. 8

Skupina 1

Pro žáky byla subjektivně úloha složitá. Největší problém měli se souřadnicemi, jak se zápornými čísly, tak s rozlišením os x a y . Pouze dva žáci zvládli úlohu v zjednodušené verzi (jablko padá pouze jednou), zbytek třídy nezvládl úlohu ani v zjednodušené verzi. Žáky nadchla možnost kreslení vlastních postav, celková náročnost celé úlohy žáky spíše odradila.

Změna úlohy:

- Klást větší důraz na rozbor úlohy.
- Více rozebrat souřadnice.
- Zařadit animaci postavy hada do rozšířené varianty.

Skupina 2

Pro žáky byla úloha obtížná, tvorba vlastních postav však žáky motivovala k jejich programování. I v této skupině byl největší problém pochopení souřadnic.

Když žáci potřebovali pomoc, tak je učitel odkazoval na příslušnou část v sešitě/prezentaci. Žáci na úloze pracovali, ale potřebovali by větší hodinou dotaci. Polovina třídy zvládla dodělat alespoň zjednodušenou verzi (jablko padá pouze jednou nebo bez reakce hada na jablko). Většina žáků pochopila princip vlastních bloků.

Skupina 3

Podrobnější rozbor hry žákům pomohl s porozuměním „pádu jablka“, ale opět byl největší problém se souřadnicemi. Jeden žák například používal blok „změň osu x o $_$ “ pro pád jablka. V této skupině zvládli dva žáci celé základní zadání a jeden žák rozšířené zadání.

Změna úlohy:

- Zařadit práci se souřadnicemi do předchozích úloh.

Skupina 4

Zařazení práce se souřadnicemi do úloh č. 4 a č. 5 pomohlo žákům s jejich pochopením. V této skupině žáci chtěli použít pro ukončení cyklu pro pád jablka dotyk s barvou země. S touto variantou autorka nepočítala a barva hlíny byla stejná jako barva kmene. Během hodiny autorka žákům přebarvila kmen na jinou barvu, aby kód fungoval za všech okolností. Úlohu zvládla dokončit polovina třídy, čtvrtina žáků vytvořila pouze zjednodušenou variantu (jablko padá pouze jednou, had nereaguje na jablko...). Zbylá čtvrtina třídy nezvládla úlohu ani v zjednodušeném zadání. Bylo to zapříčiněno zejména neznalostmi z předchozích hodin a nedokončenými poznámkami v sešitě.

Skupina 5

I v této skupině žáci používali k ukončení cyklu pro pád jablka dotyk s barvou země. Opět během hodiny autorka žákům přebarvila kmen na jinou barvu. Všichni žáci zvládli alespoň zjednodušenou verzi.

Změna úlohy:

- Změna barvy kmenu v pozadí scény.

Skupina 6

Této skupině odpadlo mnoho hodin informatiky, úlohu č. 8 nebylo možné z časových důvodů realizovat.

Závěr

V prvním cyklu výzkumného šetření byla úloha pro žáky velmi složitá. Žáci sice téma „osy souřadnic“ probírali v hodinách matematiky, ale byla pro ně stále novým konceptem. Někteří žáci zaměňovali osu x za osu y nebo měli problém pracovat se zápornými čísly. Z těchto důvodů se autorka rozhodla zařadit práci se souřadnicemi do úloh č. 4 a č. 5. Ve druhém cyklu výzkumného šetření již žáci výrazně méně chybovali.

Tato úloha měla mnoho různých správných řešení, žáci dokonce vymysleli i jeden způsob řešení, se kterým autorka nepočítala. Pro ukončení cyklu pro pád jablka použili dotyk s barvou „hlíny“, která byla v původním zadání stejná jako barva kmene stromu. Autorka musela operativně žákům během hodiny přebarvovat „kmen“. Na základě tohoto zjištění vytvořila nové pozadí, které již počítá s tímto řešením.

Žáci byli velmi motivovaní k práci na této úloze, i přes její obtížnost, protože si mohli vytvořit a naprogramovat vlastní postavy.

3.5 Shrnutí poznatků akčního výzkumu

Autorka navrhla ucelený soubor osmi úloh, který metodou akčního výzkumu ověřila v pedagogické praxi. Úlohy jsou zaměřeny na rozvoj IM. Obsah úloh je vytvořen tak, aby pokryl znalosti jazyku Scratch, jak jsou uvedeny v modelovém ŠVP. Ověření proběhlo ve dvou cyklech celkově na 6 skupinách žáků. Ukázky vybraných žákovských řešení jsou přiloženy ve formě přílohy (příloha č. 4 – 31).

První dvě úlohy byly zaměřené na seznámení žáků s principem algoritmizace a s tvorbou kódu v jazyku Scratch, proto byl v metodických pokynech věnován větší prostor i rozboru doporučených bloků. Úlohy č. 3 – 5 jsou zaměřeny na práci s informacemi, které lze využít pro rozvoj hledání vzorů, abstrakce a dekompozice. Úlohy č. 6 – 8 kladou důraz na samostatnou práci, děti tvoří hry a animace. Toto téma umožňuje výrazně prohloubit využití dekompozice a abstrakce. Úloha č. 8 je závěrečný projekt, který propojuje všechny dosavadní poznatky jak z hlediska IM, tak z hlediska Scratch. Dává učiteli velkou zpětnou vazbu, co se žáci naučili, jak přemýšlí a s čím potřebují větší dopomoc.

Žáci nejlépe zvládali hledání vzorů, ve kterém většinou nechybovali. Pro autorku nečekaným zjištěním bylo, že největší problém žákům dělalo pořadí kroků v kódu. Správná dekompozice byla stěžejním krokem k vyřešení úloh. Kvalitní rozbor úlohy výrazně zkrátil čas nutný k dokončení práce a vedl k přemýšlení nad úlohou. Bez rozboru žáci často zkoušeli uhodnout řešení metodou „pokus omyl“.

Úlohy prošly mnoha úpravami. První verze souboru úloh se ukázala být pro děti příliš náročnou. Zadání bylo pro děti obtížné a komplexní, část zadání byla přesunuta do rozšířených variant. Autorka vytvořila i zjednodušené varianty zadání, jednu úlohu bylo dokonce nutné rozdělit na dvě. Pro některé úlohy bylo dostatečné přidat podrobnější instrukce nebo vhodný předpřipravený kód. Celkově byl pro snížení náročnosti úloh velmi důležitý jejich důkladný rozbor.

Na základě poznatků autorky s ověřováním jednotlivých úloh byly upravovány i metodické pokyny. Jedná se primárně o „doporučený průběh hodiny“, „metodický rozbor úlohy“, „pokyny pro žáky“, „možné komplikace a jejich řešení“.

Ve výzkumném šetření se ukázalo, že je velmi důležité, jakým způsobem je učivo reprezentováno. Žáci pracovali s velkým nasazením i na objektivně obtížných úlohách, pokud jim téma připadalo zajímavé. Největší úspěch měly úlohy č. 1, č. 6 a č. 8, ve kterých vytvářejí vlastní objekty podle svého výběru.

4 Závěr

Cílem této diplomové práce bylo analyzovat informatické myšlení z hlediska základních konceptů a konkrétní cílové skupiny, žáků 5. ročníku, a na základě získaných poznatků sestavit soubor na sebe navazujících aktivit a úloh se zvyšující se obtížností v programovacím prostředí Scratch, které respektují a rozvíjí základní koncepty informatického myšlení.

Hlavní cíl by realizován prostřednictvím čtyř dílčích cílů. První dílčí cíl „Analyzovat pojem ‚informatické myšlení‘“ byl naplněn analýzou informačních zdrojů v kapitole 2.2, která rozebírá jednotlivá pojetí nejen samotného pojmu, ale i jednotlivé koncepty IM. Pro komparaci byly vybrány definice IM od Jaennette Wingové, MŠMT, iMyšlení, ISTE a CSTA a Královské společnosti. Pro komparaci vymezení konceptů IM bylo vybráno pojetí od Evropské komise a PRADA. Na základě analýzy bylo pro potřeby této práce informatické myšlení následně definováno jako *„způsob myšlení, který se zaměřuje na popis problému, jeho analýzu a hledání efektivních řešení“*. (Co je to informatické myšlení 2018) V rámci této práce se autorka přiklonila k následujícím konceptům IM: algoritmizace, hledání sekvencí a vzorů, dekompozice a abstrakce.

Druhým dílčím cílem bylo „Zmapovat předpoklady pro rozvoj informatického myšlení žáků 5. ročníku ZŠ“, který byl naplněn kapitolou 2.3. Tato kapitola se zabývá revizí RVP, analyzuje dvanáct výukových materiálů pro vzdělávací oblast Informatika pro 5. ročník jak z hlediska obsahové a formální stránky, tak i finanční náročnosti. Na základě zvolených kritérií byly vybrány jako nejvhodnější tištěné učební materiály od nakladatelství Fraus a digitální/online materiály vzniklé v rámci projektu PRIM. V neposlední řadě se kapitola zabývala výběrem vhodných nástrojů pro tematický celek „Algoritmizace a programování“. Byla provedena analýza nástrojů pro rozvoj IM dané cílové skupiny, ve které byla zkoumána jejich využitelnost pro rozvoj IM, přiměřená složitost pro cílovou skupinu a dostupnost. Na základě zvolených kritérií bylo vybráno programovací prostředí Scratch.

Třetí dílčí cíl „Sestavit ucelený soubor úloh pro žáky 5. ročníku, který rozvíjí informatické myšlení se zaměřením na jeho koncepty“ a čtvrtý dílčí cíl „Prostřednictvím výzkumného šetření v praxi ověřit soubor úloh na několika skupinách žáků 5. ročníku“ byly realizovány

praktickou částí. Úlohy byly ověřovány metodou akčního výzkumu, a proto byly oba cíle naplňovány paralelně, nikoliv postupně.

V praktické části byl navržen soubor osmi na sebe navazujících úloh s postupně gradující obtížností, který byl ověřen metodou akčního výzkumu, jenž probíhal ve dvou cyklech po třech skupinách žáků. Na základě akčního výzkumu byl soubor upraven tak, aby lépe reflektoval potřeby žáků. Autorka obzvláště pečlivě zvažovala časté chyby žáků a možnosti jejich řešení. Na základě těchto poznatků byly vytvořeny další materiály (např. prezentace, pracovní list). Součástí této práce jsou i metodické pokyny pro učitele, seznam vhodných bloků a předpřipravené soubory se zadáním.

Úlohy byly vyhodnoceny ze dvou hledisek, infromatického myšlení a pedagogického. Koncepty hledání vzorů a sekvencí i abstrakce žákům nepůsobily větší obtíže a byly intuitivně pochopeny. Z hlediska IM byly pro žáky náročnější koncepty algoritmizace, zejména správné pořadí kroků, a dekompozice. Žáci 5. ročníku měli velké potíže se samostatným rozborem úloh. Jednotlivé úlohy bylo s žáky nutné nejprve společně prodiskutovat, pak již mohli pracovat samostatně.

Z hlediska pedagogického se ukázalo, že pro motivaci žáků k práci je mnohem důležitější subjektivně zajímavé téma zadání než objektivní obtížnost úlohy. V rámci výzkumného šetření se osvědčila metoda vzájemného učení žáků a také možnost zadání úlohy ve více variantách (základní, rozšířená, zjednodušená).

Na webových stránkách <https://sites.google.com/view/ucimesescratch/> byl zveřejněn výstup práce. Přáním autorky je, aby tato práce byla inspirativní pro učitele prvního stupně a mohli její výstup, nebo alespoň jeho část, zařadit do výuky, případně se jím inspirovat. Sama autorka soubor nadále využívá ve své vlastní výuce a již proběhl třetí cyklus, který nepotřeboval další úpravy.

Seznam použitých informačních zdrojů

- About us*. Online. CSTA. Dostupné z: <https://csteachers.org/about-csta/>. [cit. 2022-10-11].
- AGH, Peter. *Hledá se Puffy: hybridní pracovní sešit pro 5. ročník základní školy : práce s daty a datová výchova : první dataktivka*. Ilustroval Sabina AGHOVÁ. Plzeň: Fraus, 2022. ISBN 978-80-7489-785-6.
- AGH, Peter. *Informatika 2: hybridní pracovní učebnice pro 5. ročník základní školy : Uffi a Uffi*. 2. vydání. Ilustroval Sabina AGHOVÁ. Škola s nadhledem. Plzeň: Fraus, 2023. ISBN 978-80-7489-885-3.
- ANDERSON, Nicole D. A Call for Computational Thinking in Undergraduate Psychology. Online. *Psychology Learning & Teaching*. 2016, roč. 15, č. 3, s. 226-234. ISSN 1475-7257. Dostupné z: <https://doi.org/10.1177/1475725716659252>. [cit. 2023-11-07].
- Autoři projektu*. Online. Umíme to. Dostupné z: <https://www.umimeto.org/text-project-authors>. [cit. 2024-05-21].
- BEBBINGTON, Shaun. *What is computer programming?* Online. Year of Code. Dostupné z: <https://yearofcodes.tumblr.com/what-is-programming>. [cit. 2024-03-09].
- BERKI, Jan a DRÁBKOVÁ, Jindra. *Základy informatiky pro 1. stupeň základní školy*. Online. Technická univerzita v Liberci, 2020. ISBN 978-80-7494-520-5. Dostupné z: https://imysleni.cz/images/vzdelavaci_materialy/Inf/NS-Zaklady-informatiky.pdf. [cit. 2024-05-20].
- BLAHO, Andrej; KALAŠ, Ivan a MORAVČÍK, Milan. *Informatika s Emilem 3: programování pro 1. stupeň základní školy : pracovní sešit*. Žďár nad Sázavou: H-edu, [2020]. ISBN 978-80-907769-0-6.
- BOCCONI, Stefania; CHIOCCARIELLO, Augusto; DETTORI, Giuliana; FERRARI, Anusca a ENGELHARDT, Katja. *Developing Computational Thinking in Compulsory Education - Implications for policy and practice*. Online. Lucembursko: Publications Office of the European Union, 2016. ISBN 978-92-79-64442-9. Dostupné z: <https://doi.org/10.2791/792158>. [cit. 2023-11-06].

BRDIČKA, Bořivoj. *Informatické myšlení jako výukový cíl*. Online. Spomocník. 2014-04-16. Dostupné z: <https://spomocnik.rvp.cz/clanek/18689/INFORMATICKE-MYSLENI-JAKO-VYUKOVY-CIL.html>. [cit. 2023-10-07].

Co je informatické myšlení? Online. IMyšlení. 2018. Dostupné z: <https://www.imysleni.cz/informaticke-mysleni/co-je-informaticke-mysleni>. [cit. 2023-10-11].

Comparison Between Different Block-based Programming Platforms. Online. Duck Learning. 2021, 2021-04-20. Dostupné z: <https://ducklearning.com/blogs/parent-and-educator-resources/comparison-between-different-block-base-programming-languages>. [cit. 2024-05-19].

Computational Thinking Competencies. Online. ISTE. 2023. Dostupné z: <https://iste.org/standards/computational-thinking-competencies>. [cit. 2023-10-11].

Computational thinking in classroom. Online. ISTE. 2023. Dostupné z: <https://iste.org/areas-of-focus/computational-thinking-in-the-classroom>. [cit. 2023-10-11].

Computational Thinking. Online. Code.org. Dostupné z: <https://code.org/curriculum/course3/1/Teacher>. [cit. 2024-06-21]

DENNING, Peter J. Remaining trouble spots with computational thinking. Online. *Communications of the ACM*. 2017, roč. 60, č. 6, s. 33-39. ISSN 0001-0782. Dostupné z: <https://doi.org/10.1145/2998438>. [cit. 2023-10-07].

DONG, Yihuan; CATETE, Veronica; JOCIUS, Robin; LYTLE, Nicholas; BARNES, Tiffany et al. PRADA. Online. In: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. New York, NY, USA: ACM, 2019, s. 906-912. ISBN 9781450358903. Dostupné z: <https://doi.org/10.1145/3287324.3287431>. [cit. 2023-11-07].

Exploring Computational Thinking. Online. Google Research. 2010. Dostupné z: <https://research.google/blog/exploring-computational-thinking/>. [cit. 2023-11-06].

- FILIPI, Zbyněk; MAINZ, Denis a FADRHONC, Jan. *Práce s daty pro 5. až 7. ročník základní školy*. Online. Západočeská univerzita, 2020. ISBN 978-80-261-0917-4. Dostupné z: <https://pracesdaty.zcu.cz/>. [cit. 2024-05-20].
- GROVER, Shuchi a PEA, Roy. Computational Thinking in K–12. Online. *Educational Researcher*. 2013, roč. 42, č. 1, s. 38-43. ISSN 0013-189X. Dostupné z: <https://doi.org/10.3102/0013189X12463051>. [cit. 202-01-02].
- GUZDIAL, Mark. *A Definition of Computational Thinking from Jeannette Wing*. Online. Computing Ed Research – Guzdial's Take. 2011. Dostupné z: <https://computinged.wordpress.com/2011/03/22/a-definition-of-computational-thinking-from-jeanette-wing/>. [cit. 2023-10-07].
- GUZDIAL, Mark. Education Paving the way for computational thinking. [online] *Communications of the ACM*, č. 51, s 25-27. DOI: [10.1145/1378704.1378713](https://doi.org/10.1145/1378704.1378713) Dostupné z: https://www.researchgate.net/publication/234812396_Education_Paving_the_way_for_computational_thinking [cit. 2023-01-05].
- HRNČIŘÍKOVÁ, Marcela; BALLA, David; ENEVOVÁ, Pavla a HYNEK, Miroslav. *Informatika v pohodě 5: pracovní učebnice pro 5. ročník ZŠ v souladu s RVP*. Praha: Taktik, 2023. ISBN 978-80-7563-566-2.
- Informatika a digitální kompetence s Emilem a robotika s Emou*. Online. H-Edu. Dostupné z: <https://www.h-edu.cz/informatika>. [cit. 2024-05-21].
- Jak rozvíjet infromatické myšlení*. Online. Eduskop. 2010. Dostupné z: <https://eduskop.cz/courses/course-v1:UWB+PRIM-02+2020/about>. [cit. 2023-11-06].
- JANDOUREK, Jan. *Sociologický slovník*. Praha: Portál, 2001. ISBN 80-7178-535-0.
- JANÍK, Tomáš. *Akční výzkum pro učitele: Příručka pro teorii a praxi*. Online. Brno: Pedagogická fakulta Masarykovy univerzity Katedra pedagogiky, 2003. Dostupné z: https://is.muni.cz/el/1441/jaro2006/ZS1BP_ZPM/um/um/TJ_akcni_vyzkum.pdf. [cit. 2024-04-04].
- JANKO, Tomáš. Učebnice: druhy, funkce, práce s učebnicí. Online. In: . Dostupné z: https://is.muni.cz/el/ped/jaro2015/SZ7BK_SD1/Ucebnice_druhy_funkce_prace_s_ucebnicemi.pdf.

KALAŠ, Ivan a MIKOVÁ, Karolína. *Základy programování ve Scratch pro 5. ročník základní školy*. Online. Jihočeská univerzita v Českých Budějovicích, Pedagogická fakulta, 2020. ISBN 978-80-7394-782-8. Dostupné z: <https://imysleni.cz/ucebnice/zaklady-programovani-ve-scratchi-pro-5-rocnik-zakladni-skoly>. [cit. 2024-05-20].

KALAŠ, Ivan; BLAHO, Andrej a MORAVČÍK, Milan. *Informatika s Emilem 4: programování pro 4. a 5. ročník základní školy : pracovní sešit*. Žďár nad Sázavou: H-edu, [2020]. ISBN 978-80-907769-1-3.

KRÁLOVSKÁ SPOLEČNOST. *Shut down or restart?: The way forward for computing in UK schools*. Online. 2012. Dostupné z: <https://royalsociety.org/~media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf>. [cit. 2023-10-11].

Kurzy Eduskop, *Jak rozvíjet informatické myšlení*. 2019 YouTube video. Dostupné z: <https://code.org/curriculum/course3/1/Teacher>. [cit. cit. 2023-11-07]

LESSNER, Dan; LÁNA, Martin; PODRÁZKA TOMKOVÁ, Michala a HAUT, Jiří. *Základy informatiky pro střední školu*. Online. Jihočeská univerzita v Českých budějovicích, Pedagogická fakulta, 2020. ISBN 978-80-7394-785-9. Dostupné z: <https://popelka.ms.mff.cuni.cz/~lessner/301/titulka/index.php>. [cit. 2023-10-07].

LESSNER, Daniel. ANALYSIS OF TERM MEANING „COMPUTATIONAL THINKING.“ Online. *Journal of Technology and Information*. 2014, roč. 6, č. 1, s. 71-88. ISSN 1803537X. Dostupné z: <https://doi.org/10.5507/jtie.2014.006>. [cit. 2023-10-07].

LODI, Michael a MARTINI, Simone. Computational Thinking, Between Papert and Wing. Online. *Science & Education*. 2021, č. 30, s. 883–908. Dostupné z: <https://doi.org/10.1007/s11191-021-00202-5>. [cit. 2023-06-20].

MANNILA, Linda; SETTLE, Amber; DAGIENE, Valentina; DEMO, Barbara; GRGURINA, Natasa et al. *Computational Thinking in K-9 Education*. Online. 2014. Dostupné z: <https://doi.org/10.1145/2713609.2713610>. [cit. 2024-05-16].

MBlock. Online. Dostupné z: <https://mblock.cc/>. [cit. 2024-03-21].

Microsoft MakeCode. Online. Microsoft. Dostupné z: <https://www.microsoft.com/en-us/makecode>. [cit. 2024-03-21].

Modelové školní vzdělávací programy pro základní vzdělávání. Online. Informatické myšlení. 1. 10. 2022. Dostupné z: <https://www.imysleni.cz/svp/svp-zv>. [cit. 2024-02-24].

Modelový ŠVP: NEBOJÁCNĚ VPŘED. Online. In: . 2022. Dostupné z: https://www.imysleni.cz/images/SVP/SVP2_nebojacne_vpřed_v20230323.pdf. [cit. 2024-02-24].

Modelový ŠVP: OPATRNĚ VPŘED. Online. In: . 2022. Dostupné z: https://www.imysleni.cz/images/SVP/SVP1_opatrne-vpred_v20221001.pdf. [cit. 2024-02-24].

MÖNIG, Jens a HARVEY, Brian. *SNAP! Reference Manual*. Online. 2020. Dostupné z: <https://snap.berkeley.edu/>. [cit. 2024-03-21]

MORBACHEROVÁ, Jana. *Informatika 5: učebnice pro 5. ročník*. Brno: Nová škola - Duha, [2023]. ISBN 978-80-88285-96-0.

MŠMT. *Strategie digitálního vzdělání do roku 2020*. Online. 2014. Dostupné z: <https://msmt.gov.cz/uploads/DigiStrategie.pdf>. [cit. 2023-10-11].

NAVRÁTIL, Pavel. *Informatika pro 5. ročník ZŠ*. Computer Media, 2024. ISBN 978-80-7402-469-9.

NAVRÁTIL, Pavel. *Informatika pro 5. ročník ZŠ: Metodika pro učitele*. Computer Media, 2024. ISBN 978-80-7402-469-6.

NEŠPOR, Zdeněk R. a BURIÁNEK, Jiří. *Analýza obsahová*. Online. Sociologická encyklopedie. 2017b, 10. 11. 2018. Dostupné z: https://encyklopedie.soc.cas.cz/w/Anal%C3%BDza_obsahov%C3%A1 [cit. 2024-04-04].

NEŠPOR, Zdeněk R. a PETRUSEK, Miloslav. *Analýza obsahová (MSgS)*. Online. Sociologická encyklopedie. 2017c, 10. 11. 2018. Dostupné z: https://encyklopedie.soc.cas.cz/w/Metoda_srovn%C3%A1vac%C3%AD. [cit. 2024-04-04].

NEŠPOR, Zdeněk R.; LINHART, Jiří a VODÁKOVÁ, Alena. *Metoda srovnávací*. Online. Sociologická encyklopedie. 2017a, 11. 12. 2017. Dostupné z: https://encyklopedie.soc.cas.cz/w/Metoda_srovn%C3%A1vac%C3%AD. [cit. 2024-04-04].

O Scratchi. Online. Scratch. Dostupné z: <https://scratch.mit.edu/about>. [cit. 2023-03-21].

Opatření č. j.: MSMT-40117/2020-4 Opatření ministra školství, mládeže a tělovýchovy, kterým se mění Rámcový vzdělávací program pro základní vzdělávání. Dostupné z https://msmt.gov.cz/file/54861_1_1/ [cit. 2024-05-20].

Operational Definition of Computational Thinking. Online. *ISTE*. 2011. Dostupné z: https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf. [cit. 2023-10-11].

PAPERT, Seymour. From the math wars to the new new math [Keynote Lecture] *17th ICMI Study Conference: Mathematics Education and Digital Technologies.*, 2006, Vietnam, *Přepis dostupný z t* <http://dailypapert.com/wp-content/uploads/2012/05/Seymour-Vietnam-Talk-2006.pdf>, <https://drive.google.com/file/d/0BzmXPBo2CiPTNjJMOGIc2dvVDA>

Preview. Online. MICROSOFT. Learn Microsoft. Dostupné z: <https://learn.microsoft.com/en-us/training/modules/computational-thinking-importance-education/preview>. [cit. 2023-11-07].

Prof. Dr. Alexander Repenning. Online. University of Colorado. Dostupné z: <https://home.cs.colorado.edu/~ralex/>. [cit. 2024-04-20].

Programování pro děti: naučte se programovat při tvorbě skvělých her. 1. vyd. Brno: Computer Press, 2013. 159 s. ISBN 978-80-251-3809-0.

Programování s Emilem 1. díl - pracovní sešit. Online. H-Edu. Dostupné z: <https://www.h-edu.cz/eshop/4c6ee2c5-8fe0-4d45-ab29-bbf97457de37>. [cit. 2024-05-21].

PROCHÁZKA, Josef; LAPEŠ, Jakub a TOCHÁČEK, Daniel. *Základy informatiky pro 1. stupeň základní školy*. Online. Západočeská univerzita, 2020. ISBN 978-80-7603-086-

2. Dostupné z: <https://imysleni.cz/ucebnice/edukacni-robotika-s-lego-wedo-2-0-pro-1-stupen-zakladni-skoly>. [cit. 2024-05-20].

PRŮCHA, Jan; WALTEROVÁ, Eliška a MAREŠ, Jiří. *Pedagogický slovník*. 4., aktualiz. vyd. [i.e. Vyd. 5.]. Praha: Portál, 2008. ISBN 978-80-7367-416-8.

Rámcové vzdělávací programy. Online. Národní ústav pro vzdělávání. Dostupné z: <https://archiv-nuv.npi.cz/t/rvp.html>. [cit. 2024-06-21].

Rámcový vzdělávací program pro základní vzdělávání [online]. Praha: MŠMT, 2023 [cit. 2024-01-02]. Dostupné z: https://www.edu.cz/wp-content/uploads/2023/07/RVP_ZV_2023_cista_verze.pdf

REPENNING, Alexander. Moving Beyond Syntax: Lessons from 20 Years of Blocks Programming in AgentSheets. Online. *Journal of Visual Languages and Sentient Systems*. 2017, roč. 3, č. 1, s. 68-91. ISSN 25753916. Dostupné z: <https://doi.org/10.18293/VLSS2017-010>. [cit. 2024-04-20].

Scratch. Online. Dostupné z: <https://scratch.mit.edu/>. [cit. 2023-02-17].

SELBY, Cynthia a WOOLLARD, John. Computational thinking: the developing definition. Online. 2013. Dostupné z: https://eprints.soton.ac.uk/356481/1/Selby_Woollard_bg_soton_eprints.pdf. [cit. 2023-10-07].

SHUTE, Valerie J.; SUN, Chen a ASBELL-CLARKE, Jodi. Demystifying computational thinking. Online. *Educational Research Review*. 2017, roč. 22, s. 142-158. ISSN 1747938X. Dostupné z: <https://doi.org/10.1016/j.edurev.2017.09.003>. [cit. 2023-10-07].

Snap! Build Your Own Blocks. Online. Dostupné z: <https://snap.berkeley.edu/>. [cit. 2024-03-21].

STEWART, William a BAEK, Kwanoo. Analyzing computational thinking studies in Scratch programming: A review of elementary education literature. Online. *International Journal of Computer Science Education in Schools*. 2023, roč. 6, č. 1, s. 35-58. ISSN 2513-8359. Dostupné z: <https://doi.org/10.21585/ijcses.v6i1.156>. [cit. 2023-10-07].

SUSANTI, Reni Dwi a TAUFİK, Marhan. Analysis of Student Computational Thinking in Solving Social Statistics Problems. Online. *SJME (Supremum Journal of Mathematics Education)*. 2021, roč. 5, č. 1. ISSN 2548-8163. Dostupné z: <https://doi.org/10.35706/sjme.v5i1.4376>. [cit. 2023-11-07].

TIOBE Index for March 2024. Online. Tiobe. Dostupné z: <https://www.tiobe.com/tiobe-index/>. [cit. 2024-03-14].

TRNKA, Josef. Konstrukční výzkum (design-based research) v přírodovědných didaktikách. *Scientia in educatione*. 2011, č. 2, s. 3–14. ISSN 1804-7106. Dostupné z: <https://ojs.cuni.cz/scied/article/view/11/12> [cit. 2024-04-04].

Umíme informatiku. Online. Dostupné z: <https://www.umimeinformatiku.cz/>. [cit. 2024-05-21].

VITOVSKÝ, Antonín. *Moderní slovník softwaru: výkladový anglicko-český a česko-anglický*. Praha: AV Software, 2006. ISBN 80-901428-8-5.

What is computational thinking? Online. University of York. Dostupné z: <https://online.york.ac.uk/what-is-computational-thinking/>. [cit. 2023-11-07].

What is Logo? Online. Logo Foundation. 2011. Dostupné z: <https://web.archive.org/web/20110721121752/http://el.media.mit.edu/logo-foundation/logo/index.html>. [cit. 2024-06-19].

WING, Jeannette M. Computational Thinking. Online. *Communications of the ACM*. Č. 49, 2006 s. 33-35. Dostupné z: <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>. [cit. 2022-10-16].

WING, Jeannette M. Computational Thinking: What and Why?, 2010 Online. Dostupné z: <https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>. [cit. 2022-10-16].

YADAV, Aman a CAELI, Elisa. *Unplugged Approaches to Computational Thinking: a Historical Perspective*. Online. 2019. Dostupné z: <https://doi.org/10.1007/s11528-019-00410-5>. [cit. 2023-01-05].

YADAV, Aman; ZHOU, Ninger; MAYFIELD, Chris; HAMBRUSCH, Susanne a KORB, John T. Introducing computational thinking in education courses. Online. In:

Proceedings of the 42nd ACM technical symposium on Computer science education.
New York, NY, USA: ACM, 2011, s. 465-470. ISBN 9781450305006. Dostupné z:
<https://doi.org/10.1145/1953163.1953297>. [cit. 2023-11-06].

Seznam příloh

Příloha 1: Porovnání „klasického výzkumu“ a „akčního výzkumu“ (Janík 2003 s. 8) ..	120
Příloha 2: Porovnání „klasického výzkumu“ a „akčního výzkumu“ (Janík 2003 s. 8)...	121
Příloha 3: Porovnání cílů výzkumu Příloha (Trna 2011, s. 6).....	122
Příloha 4: Pracovní list „Kreslení s robotem“	123
Příloha 5: Ukázka správného řešení úlohy č. 1	124
Příloha 6: Ukázka správného řešení úlohy č. 2 včetně rozšířeného zadání.....	125
Příloha 7: Ukázka řešení úlohy č. 2: řešení metodou „pokus-omyl“	126
Příloha 8: Ukázka řešení úlohy č. 2 – čtverec bez cyklů.....	126
Příloha 9: Ukázka správného řešení úlohy č. 3: použití vnořených cyklů	127
Příloha 10: Ukázka řešení úlohy č. 3: řešení metodou "pokus-omyl"	128
Příloha 11: Ukázka řešení úlohy č. 3: využití záporných čísel.....	129
Příloha 12: Ukázka řešení úlohy č. 3: špatné pořadí kroků.....	130
Příloha 13: Ukázka správného řešení úlohy č. 4 včetně rozšířeného zadání a využití „skoč na x: _ y: _“	131
Příloha 14: Ukázka řešení úlohy č. 4: špatné pořadí kroků.....	132
Příloha 15: Ukázka řešení úlohy č. 4: špatné pořadí kroků "smaž"	132
Příloha 16: Ukázka správného řešení úlohy č. 5	133
Příloha 17: Ukázka řešení úlohy č. 5: problém s použitím cyklů a příkazu „když <> tak“	133
Příloha 18: Ukázka správného řešení úlohy č. 5 včetně rozšířeného zadání.....	134
Příloha 19: Ukázka řešení úlohy č. 5: jednodušší verze řešení základní varianty.....	135
Příloha 20: Ukázka řešení úlohy č. 6: jiné postavy, chybí "smaž"	136
Příloha 21: Ukázka řešení úlohy č. 6: "smaž" u postavy "myši".....	136
Příloha 22: Ukázka řešení úlohy č. 6: pohyb postavy "myš"	136
Příloha 23: Ukázka řešení úlohy č. 6: vlastní postavy.....	137
Příloha 24: Ukázka řešení úlohy č. 6: špatné pořadí kroků.....	137
Příloha 25: Ukázka správného řešení úlohy č. 7: naprogramování odrážení od okraje	138
Příloha 26: Ukázka řešení úlohy č. 7: využití náhodných čísel.....	138
Příloha 27: Ukázka řešení úlohy č. 7 bez "když startuje můj klon"	139

Příloha 28: Ukázka správného řešení úlohy č. 8	139
Příloha 29: Ukázka správného řešení úlohy č. 8 s „dotýkáš se barvy _“	140
Příloha 30: Ukázka řešení úlohy č. 8 bez otisknutí	140
Příloha 31: Ukázka řešení úlohy č. 8 se záměnou bloků „změň y o _“ a „změň x o _“	141

Seznam elektronických přílohy

Příloha 1 01 Vaření s robotem.sb3
Příloha 2 01 řešení.sb3
Příloha 3 01 Vaření s robotem.pdf
Příloha 4 02 Kreslení s robotem.sb3
Příloha 5 02 řešení.sb3
Příloha 6 02 Pracovní list.pdf
Příloha 7 03 Pokládání sklenic.sb3
Příloha 8 03 řešení.sb3
Příloha 9 03 Pokládání sklenic.pdf
Příloha 10 03.mp4
Příloha 11 04 Pokladní sklenic do poliček různé délky.sb3
Příloha 12 04 řešení.sb3
Příloha 13 04 Pokladní sklenic do různých poliček.pdf
Příloha 14 04.mp4
Příloha 15 05 Pokladní sklenic do různých poliček.sb3
Příloha 16 05 řešení.sb3
Příloha 17 05 Pokladní sklenic na podložku.pdf
Příloha 18 05.mp4
Příloha 19 06 řešení.sb3
Příloha 20 06.mp4
Příloha 21 07 řešení.sb3
Příloha 22 07 Akvárium.pdf
Příloha 23 06.mp4
Příloha 24 08 Had a padající jablka.sb3
Příloha 25 08 řešení.sb3
Příloha 26 08 Had a padající jablka - pozadí.png
Příloha 27 08.mp4

Seznam obrázků

Obrázek 1: Průběh akčního výzkumu (Janík 2003, s. 9).....	10
Obrázek 2: Uživatelské rozhraní Scratch – obrázková příloha pro žáky	40
Obrázek 3: Úloha č. 1 – Scratch soubor se zadáním	55
Obrázek 4: Úlohy č. 1- ukázka možného řešení.....	56
Obrázek 5: Úloha č. 2 – Scratch soubor se zadáním	61
Obrázek 6: Úloha č. 2 – ukázka žákovské práce: pracovní list	62
Obrázek 7: Úloha č. 2 – ukázka možného řešení	62
Obrázek 8: Úloha č. 3 – Scratch soubor se zadáním	67
Obrázek 9: Úloha č. 3 – ukázka možného řešení	69
Obrázek 10: Úlohy č. 4 – Scratch soubor se zadáním	73
Obrázek 11: Úloha č. 4 – ukázka různých délek skříní	74
Obrázek 12: Úloha č. 4 – ukázka možného řešení	75
Obrázek 13: Úloha č. 5 – Scratch soubor se zadáním	79
Obrázek 14: Úloha č. 5 – ukázka možného řešení	80
Obrázek 15: Úloha č. 5 – jednodušší řešení	84
Obrázek 16: Úloha č. 5 – Scratch soubor se zjednodušeným zadáním	85
Obrázek 17: Úloha č. 6 – kód postavy melounu.....	86
Obrázek 18: Úloha č. 6 – kód postavy myši.....	87
Obrázek 19: Úloha č. 6 – ukázka žákovského rozboru	87
Obrázek 20: Úloha č. 7 – ukázka možného řešení	92
Obrázek 21: Úloha č. 7 – ukázka žákovského rozboru	93
Obrázek 22: Úloha č. 8 – had a padající jablka	96
Obrázek 23: Úloha č. 8 – kód pro postavu hada.....	97
Obrázek 24: Úloha č. 8 – postava jablka: ukázky různých možných řešení s „opakuj dokud nenastane <>“	98
Obrázek 25: Úloha č. 8 – postava jablka: ukázka řešení pouze za pomoci souřadnic	98

Seznam tabulek

Tabulka 1: Porovnání konceptů IM podle C. Shelby a J. Woollarda (C. Selby a J. Woollard 2013, s. 5)	19
Tabulka 2 Porovnání konceptů IM (Bocconi a Chiocciariello et al. 2016 s. 17)	20
Tabulka 3: Rozbor výukových materiálů z formálního hlediska	30
Tabulka 4: Rozbor výukových materiálů z obsahového hlediska – RVP	31
Tabulka 5: Rozbor výukových materiálů z obsahového hlediska – Rozvoj IM.....	32
Tabulka 6: Rozbor výukových materiálů z ekonomického hlediska.....	33
Tabulka 7: Rozbor úloh z pohledu IM.....	43
Tabulka 8: Využití bloků v jednotlivých úlohách	46
Tabulka 9: Metodický rozbor úloh	47

Příloha 1: Porovnání „klasického výzkumu“ a „akčního výzkumu“ (Janík 2003 s. 8)

„KLASICKÝ VÝZKUM“	„AKČNÍ VÝZKUM“
<p style="text-align: center;">↓</p> <ul style="list-style-type: none"> • základní výzkum • získávání poznatků • výzkumník se chce něco dozvědět • učitelé, žáci jsou „zkoumání“ 	<p style="text-align: center;">↓</p> <ul style="list-style-type: none"> • změna vlastního vyučování • získávání poznatků • učitel se chce něco dozvědět • učitel zkoumá
CÍLE	
VÝZKUMNÉ OTÁZKY	
<ul style="list-style-type: none"> • odvozeny ze výzkumných záměrů... • odvozeny na základě odborné literatury 	<ul style="list-style-type: none"> • vyplynou z potřeb učitele • mohou se v průběhu výzkumu změnit
DESIGN (PLÁN) VÝZKUMU	
<ul style="list-style-type: none"> • stanoven před začátkem výzkumu 	<ul style="list-style-type: none"> • vyvíjí se, může být v průběhu měněn
„KLASICKÝ VÝZKUM“	„AKČNÍ VÝZKUM“

VÝZKUMNÝ VZOREK	
<ul style="list-style-type: none">• reprezentativní• po výběru zůstává stejný	<ul style="list-style-type: none">• nereprezentativní• může být kdykoliv změněn
SBĚR DAT	
<ul style="list-style-type: none">• stanoveno před začátkem výzkumu v plánu výzkumu	<ul style="list-style-type: none">• probíhá nedogmaticky• metody sběru dat mohou být kdykoliv změněny
METODY VYHODNOCOVÁNÍ DAT	
<ul style="list-style-type: none">• statistické metody• inferenční statistika• teorie pravděpodobnosti	<ul style="list-style-type: none">• analýza dat• deskriptivní statistika
JAZYK	
<ul style="list-style-type: none">• jazyk vědy	<ul style="list-style-type: none">• jazyk učitelů
VÝSLEDKY	
<ul style="list-style-type: none">• často jsou k dispozici teprve až po několika letech• jsou zobecněné a objektivní• výzkumníci získají poznatky	<ul style="list-style-type: none">• jsou brzy k dispozici• jsou platné „tehde“ a tady“<ul style="list-style-type: none">• jsou subjektivní• učitelé získají poznatky
DOPAD	
<ul style="list-style-type: none">• je možné zpětně působit na jednání učitelů jako profesní skupiny<ul style="list-style-type: none">• vnější zkušenosti• snaha ovlivňovat jednání učitele zvnějšku	<ul style="list-style-type: none">• s jistou pravděpodobností lze bezprostředně zpětně působit na jednání konkrétního učitele<ul style="list-style-type: none">• vlastní zkušenosti• jednání je autonomní

Příloha 3: Porovnání cílů výzkumu Příloha (Trna 2011, s. 6)

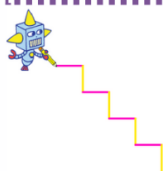
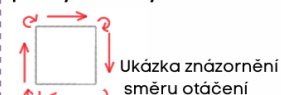
Teoretické (theoretical) cíle	Zaměření na vysvětlení jevů prostřednictvím logické analýzy a syntézy teorií, principů a výsledků jiných typů výzkumů, především empirických studií; vyžaduje vysokou úroveň syntézy a zobecnění; předpokládá dlouhodobou vědeckou práci.
Empirické (empirical) cíle	Zaměření na zjišťování stavu fungování vzdělávacího systému (výkonnost, hodnocení, sociální interakce, vzdělávací design aj.), stanovení účinků faktorů vzdělávání nebo aspektů technologických inovací v kontrolovaných podmínkách s využitím kvantitativních výzkumných metod; v současnosti je populární a mohou se mu věnovat začínající výzkumníci.
Interpretační (interpretivist) cíle	Zaměření na popis a výklad prvků a jevů ve vzdělání (výkonnost, hodnocení, sociální interakce, inovace aj.) s využitím kvalitativních výzkumných metod; narůstá popularita, třebaže je občas kritizováno (statistická neprůkaznost apod.).
Postmoderní (postmodern) cíle	Zaměření na kritické zkoumání stavu a podmínek současného vzdělávacího systému, s cílem odhalit skryté problémy z hlediska alternativních názorů (multikulturnost, gender aj.); výzkum je vzácný, protože se mu věnuje málo výzkumníků i v důsledku malé poptávky po těchto výzkumech.
Vývojové (development) cíle	Zaměření na dvojitý cíl: rozvinout tvůrčí přístupy k řešení problematiky zefektivnění vzdělávání a současně budovat konkrétní postupy a nástroje, které mohou vést k tomuto rozvoji; rozvoj tohoto výzkumu, který je také označován jako konstrukční experiment či formativní výzkum, má aktuální podporu i díky svému propojení s praxí.
Akční (action) cíle	Zaměření na konkrétní projekt (produkt, technologický prvek), který má zefektivnit vzdělávání; je blízký vývojovému výzkumu, je zde ale kladen malý důraz na zobecnění pro rozvoj teorie, což bývá akčnímu výzkumu vytýkáno, a ten je pak považován jen za hodnotící fázi jiného výzkumu; hlavním cílem je řešení konkrétního problému, na určitém místě a v krátkém časovém úseku; je považován za legitimní formu výzkumu za předpokladu, že zpráva je publikována a může být využita i v jiném výzkumu.

Kreslení s robotem

Pomoc robotovi Botíkovi nakreslit schody. Roboti nerozumí lidské řeči, proto budeš muset robotovi napsat kód, algoritmus, pro kreslení schodů.

Úkol 1

Zkus si sám/sama nakreslit 4 schody na papír. Šípkami za znázorni pohyb tužky - směr otáčení.



Úkol 2

Napiš jaké příkazy dáš robotovi, aby nakreslil 4 schody. Robot se umí "posuň se dopředu o (číslo) kroků", "otoč se doleva o (číslo) stupňů", "otoč se doprava o (číslo) stupňů". Pak můžeš robota naprogramovat ve Scratch.

Příkazy:

dopředu o kroků

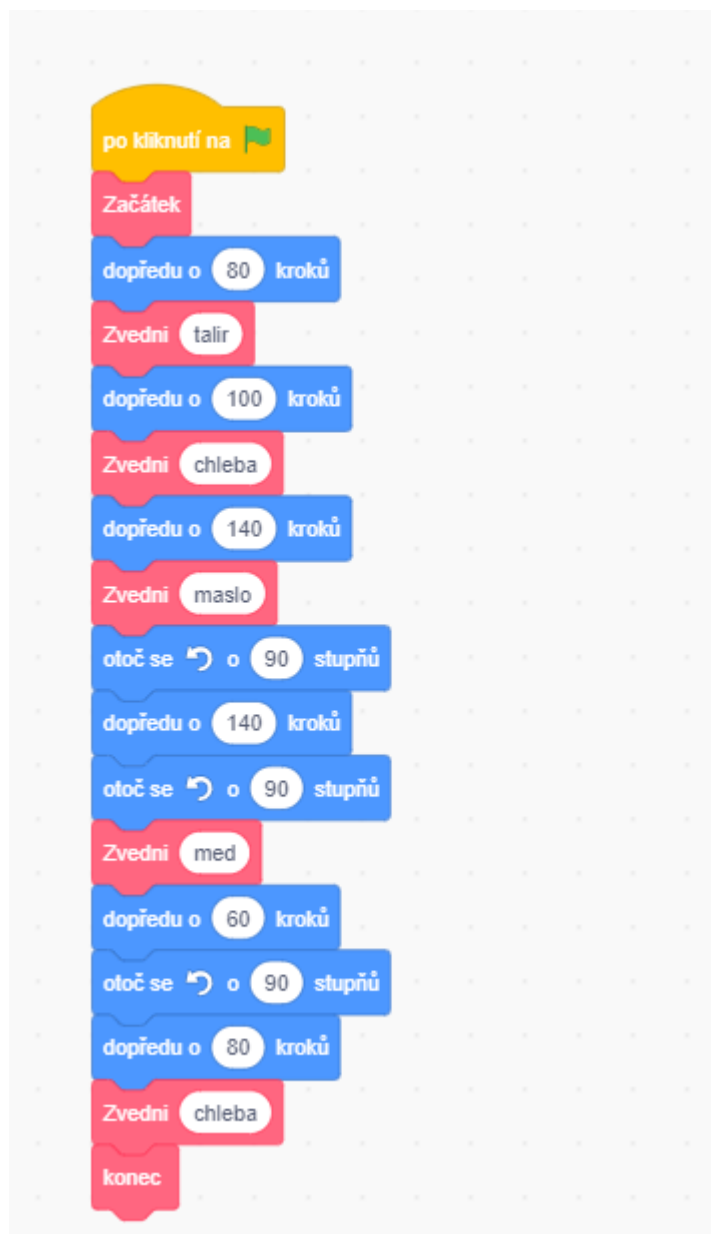
otoč se o stupňů

otoč se o stupňů

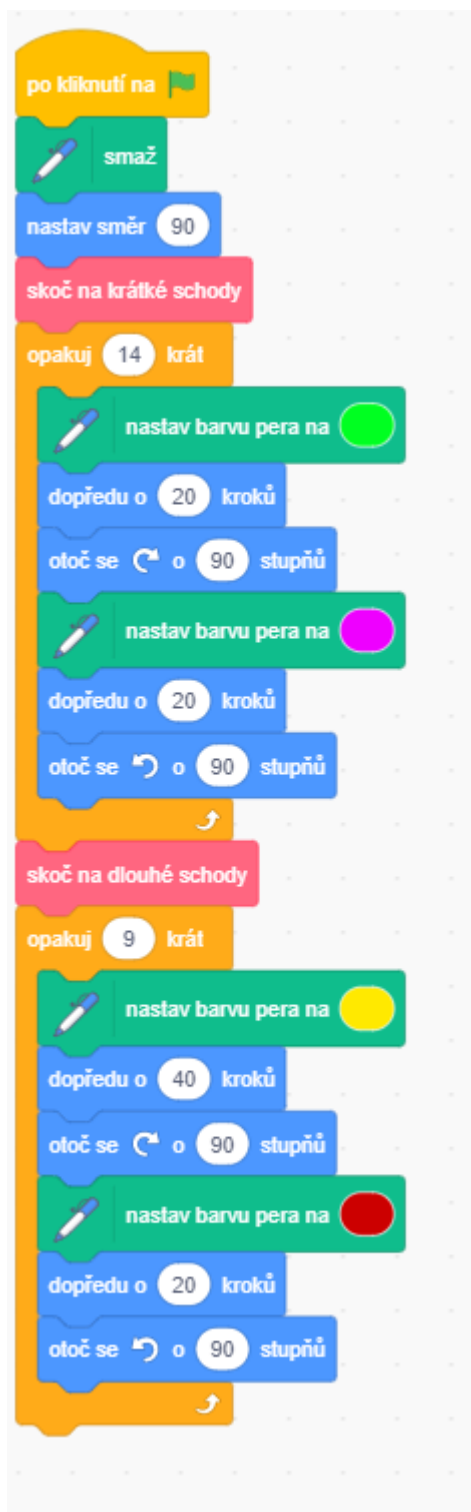
Úkol 3

Podívej se na své pokyny a obrázek - opakuje se něco? Barevně označ část schodů, který se opakuje. Stejnou barvou označ část příkazů, kódu, která se opakuje. Spočítej si kolikrát se ti kód opakuje.

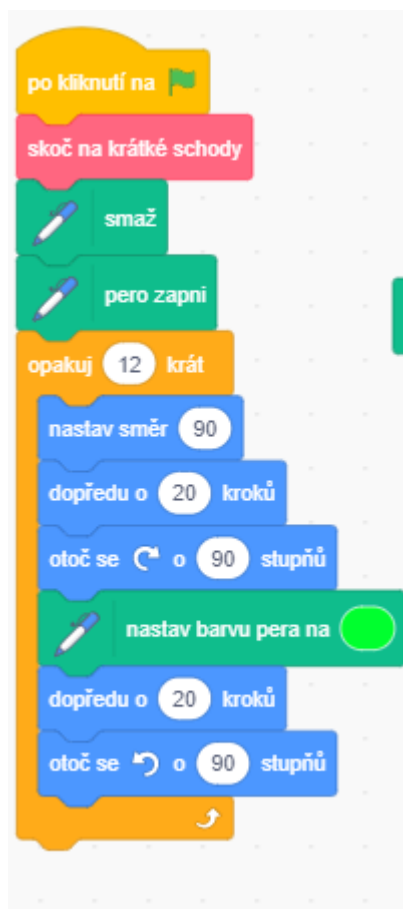
Příloha 5: Ukázka správného řešení úlohy č. 1



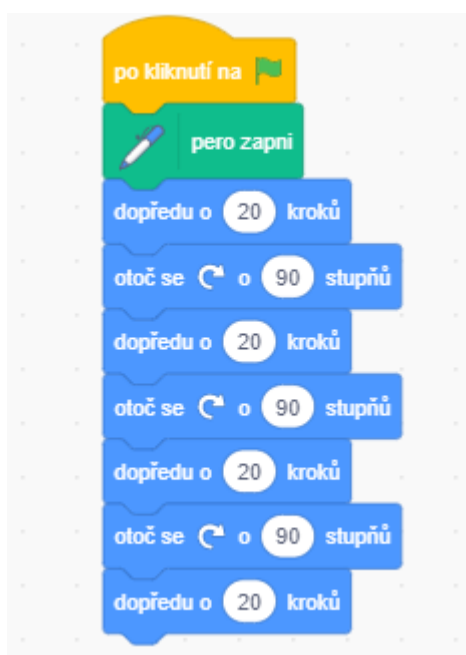
Příloha 6: Ukázka správného řešení úlohy č. 2 včetně rozšířeného zadání



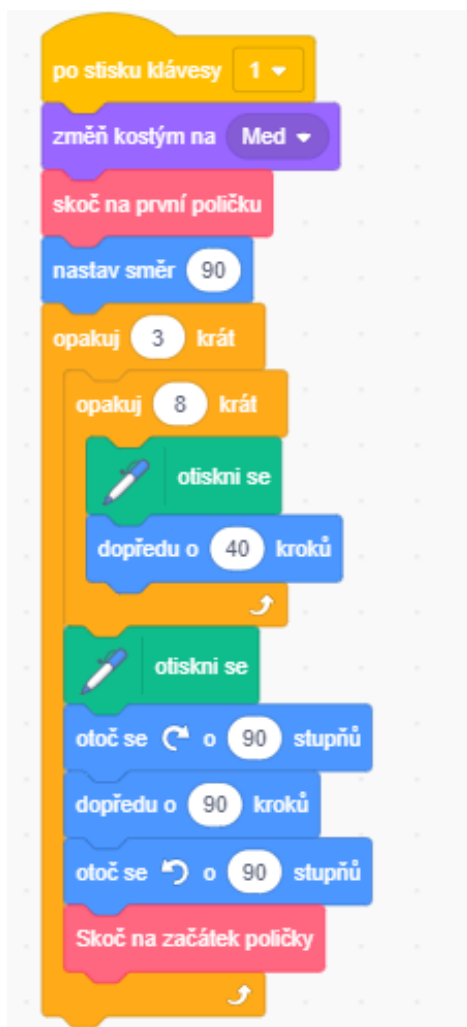
Příloha 7: Ukázka řešení úlohy č. 2: řešení metodou „pokus-omyl“



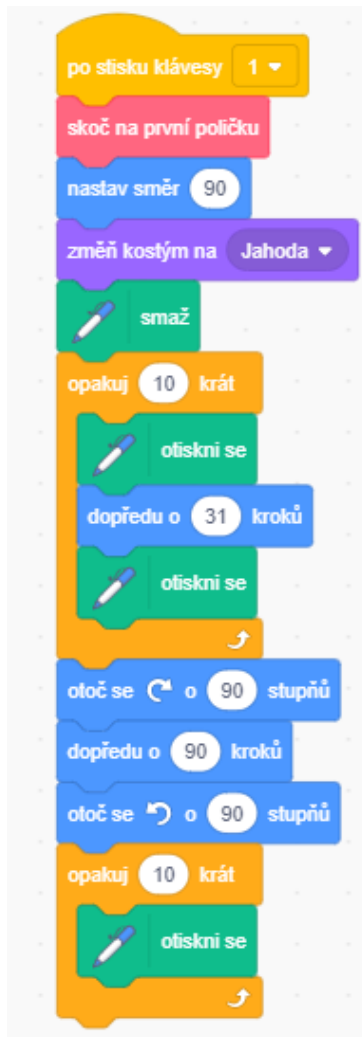
Příloha 8: Ukázka řešení úlohy č. 2 – čtverec bez cyklů



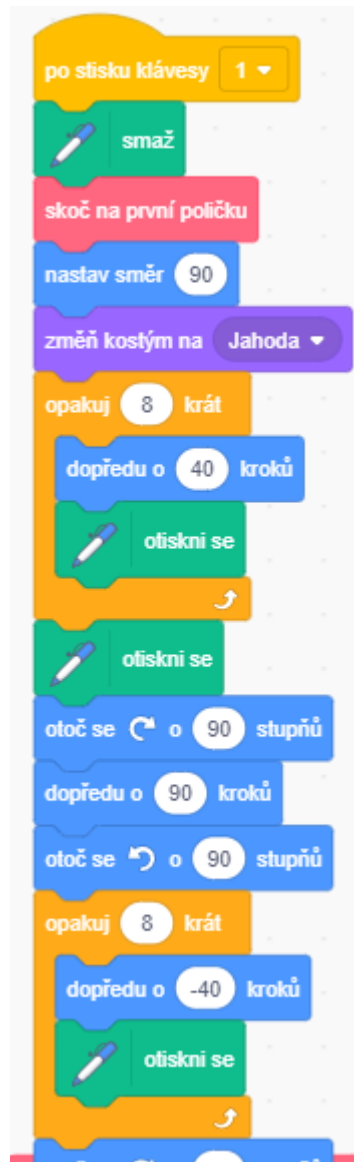
Příloha 9: Ukázka správného řešení úlohy č. 3: použití vnořených cyklů



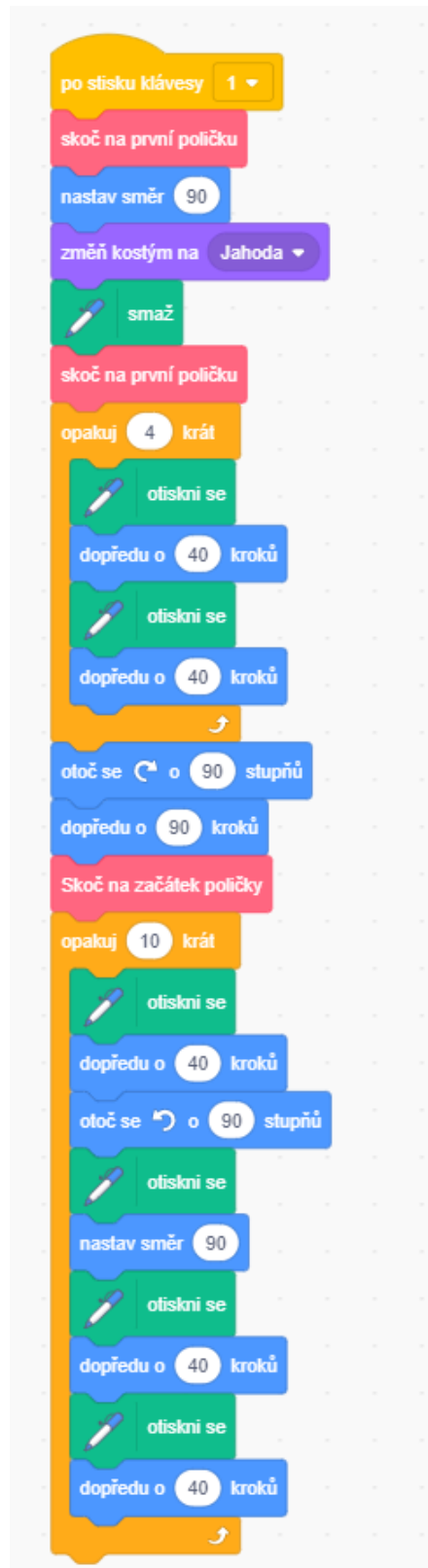
Příloha 10: Ukázka řešení úlohy č. 3: řešení metodou "pokus-omyl"



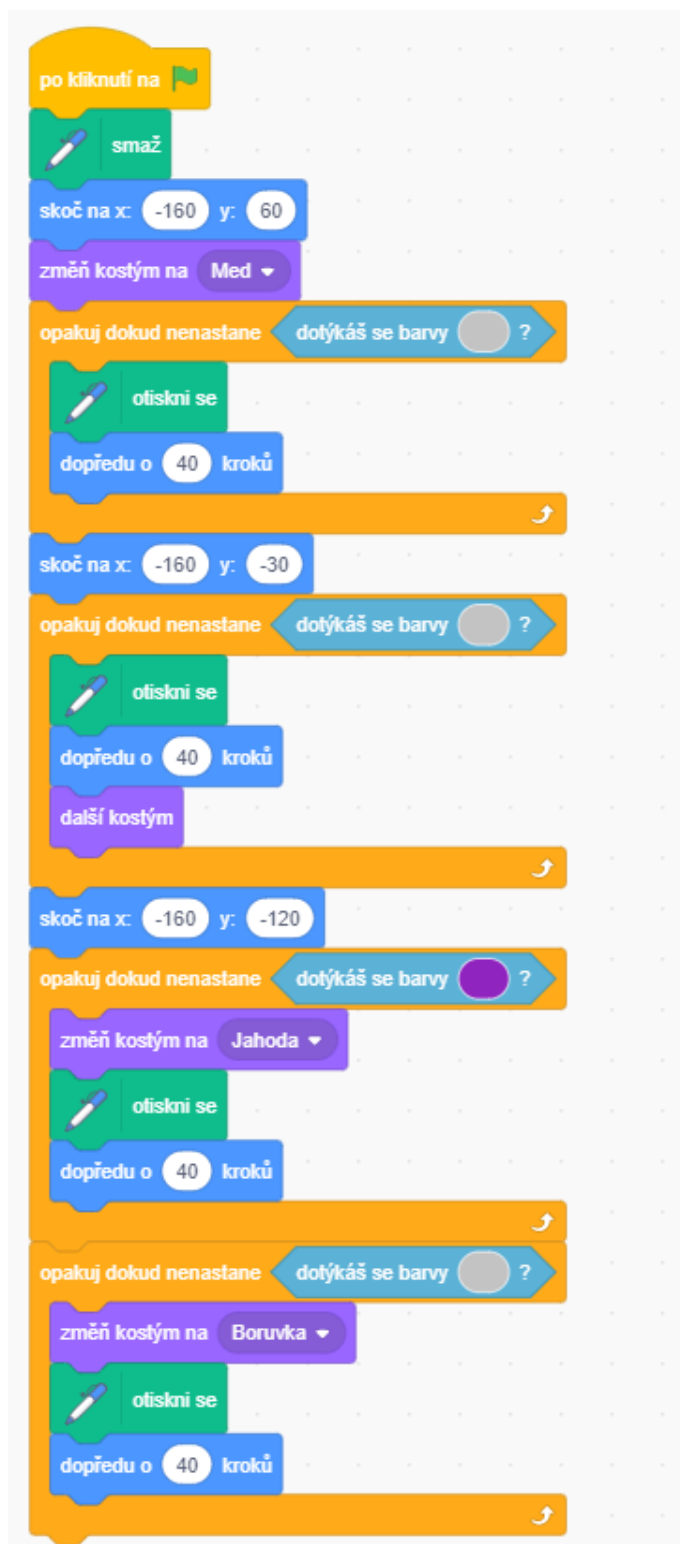
Příloha 11: Ukázka řešení úlohy č. 3: využití záporných čísel



Příloha 12: Ukázka řešení úlohy č. 3: špatné pořadí kroků



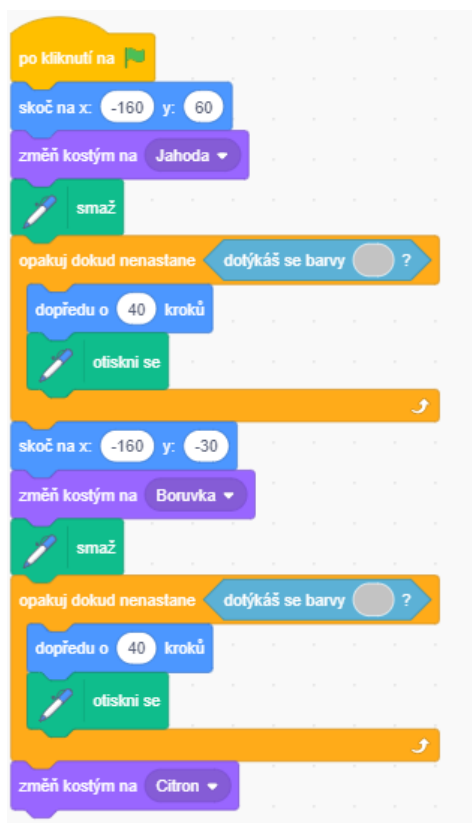
Příloha 13: Ukázka správného řešení úlohy č. 4 včetně rozšířeného zadání a využití „skoč na x:_y:“



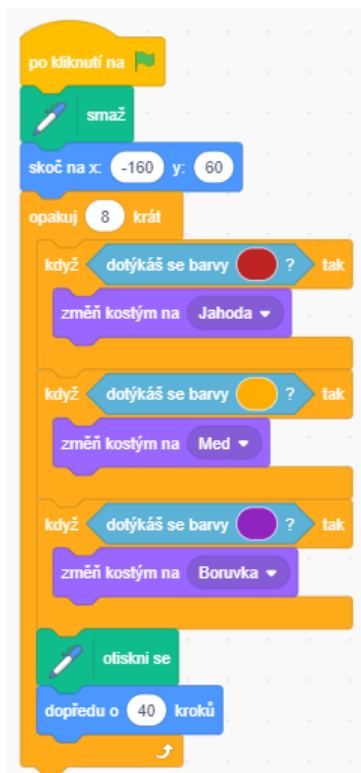
Příloha 14: Ukázka řešení úlohy č. 4: špatné pořadí kroků



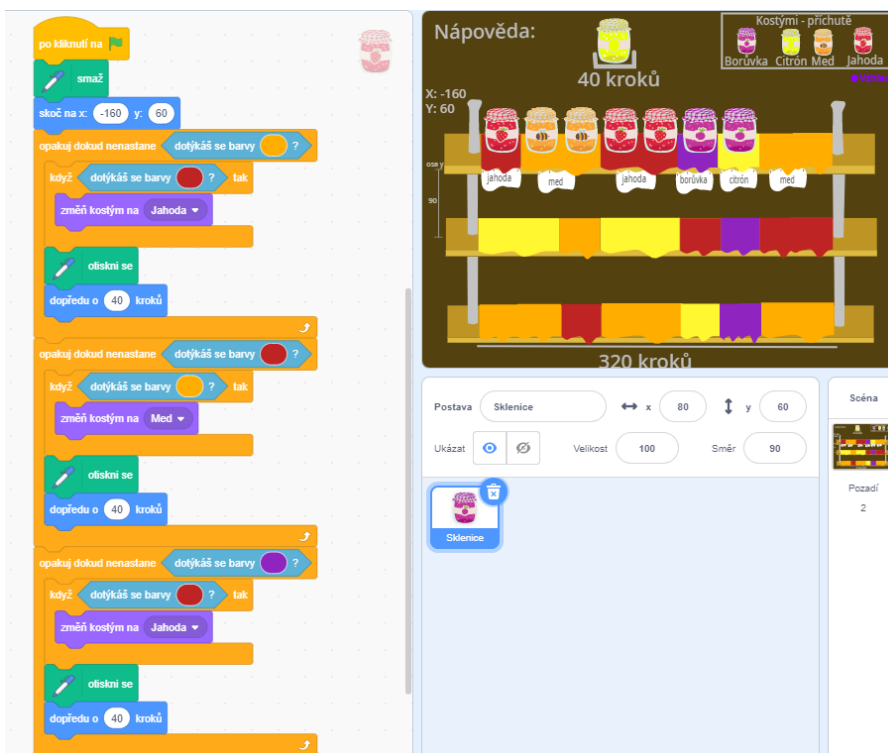
Příloha 15: Ukázka řešení úlohy č. 4: špatné pořadí kroků "smaž"



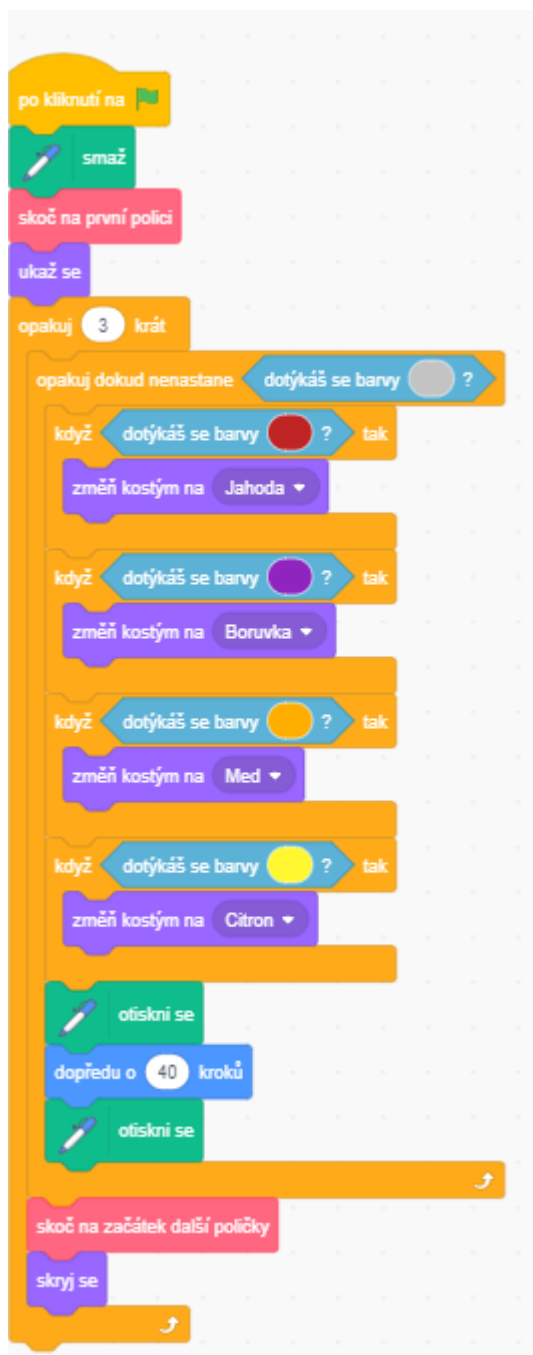
Příloha 16: Ukázka správného řešení úlohy č. 5



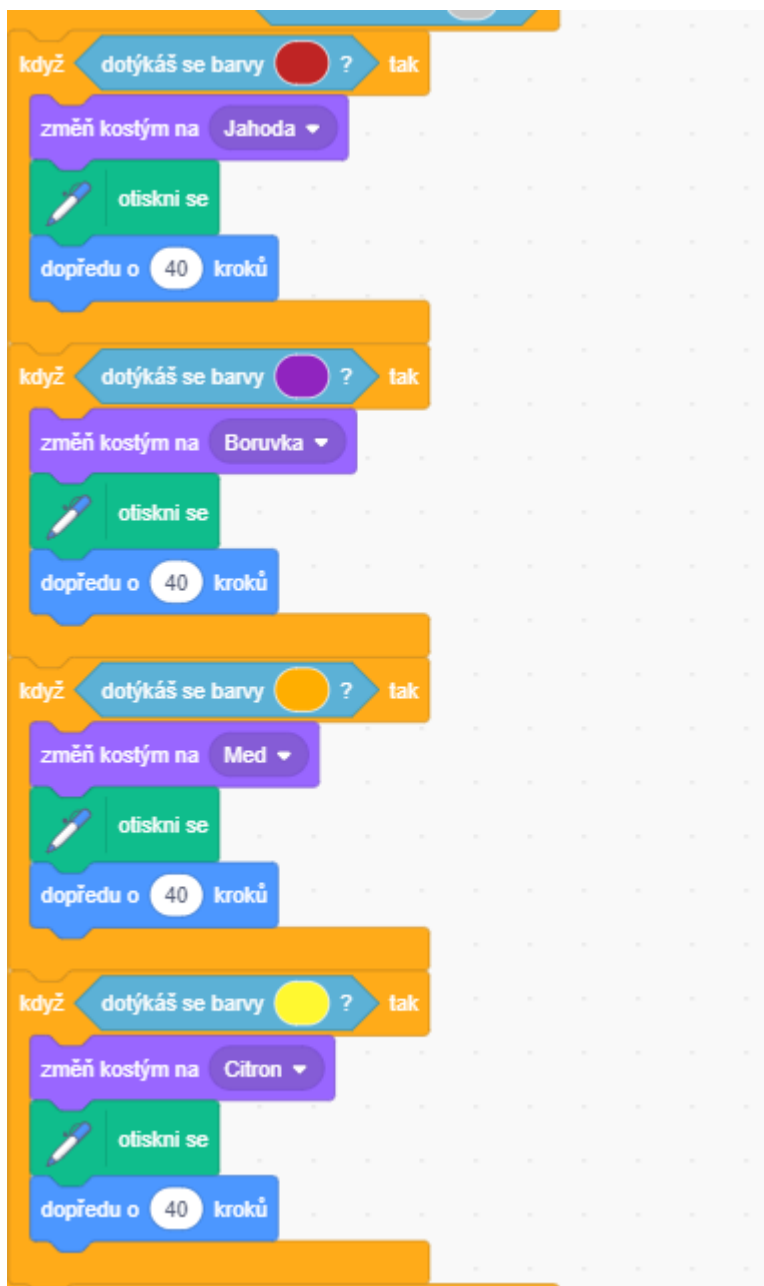
Příloha 17: Ukázka řešení úlohy č. 5: problém s použitím cyklů a příkazu „když <> tak“



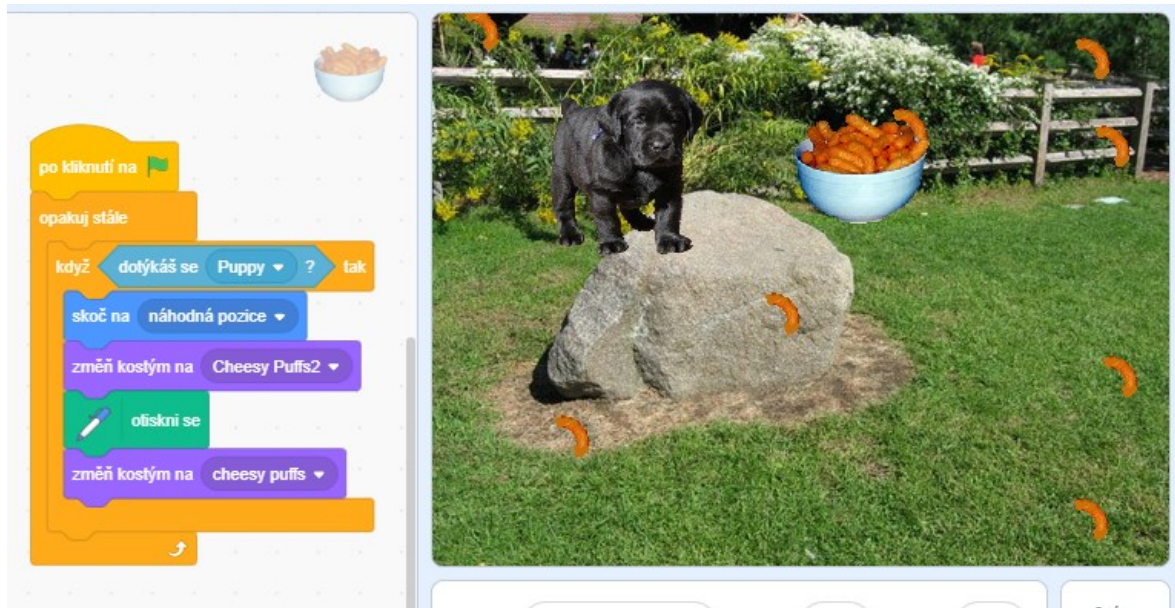
Příloha 18: Ukázka správného řešení úlohy č. 5 včetně rozšířeného zadání



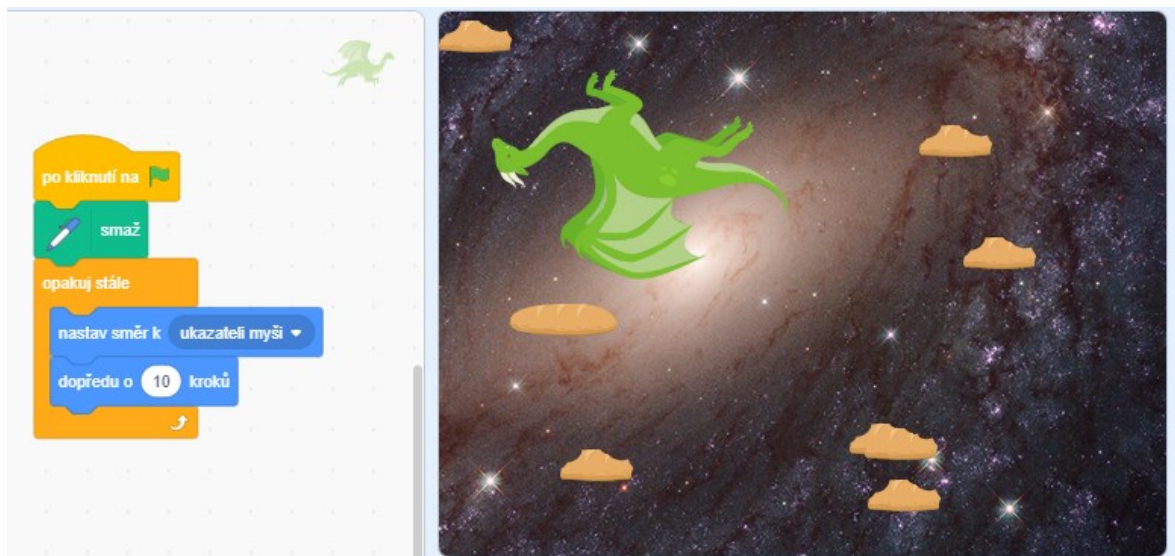
Příloha 19: Ukázka řešení úlohy č. 5: jednodušší verze řešení základní varianty



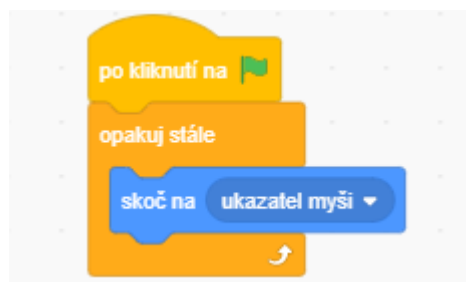
Příloha 20: Ukázka řešení úlohy č. 6: jiné postavy, chybí "smaž"



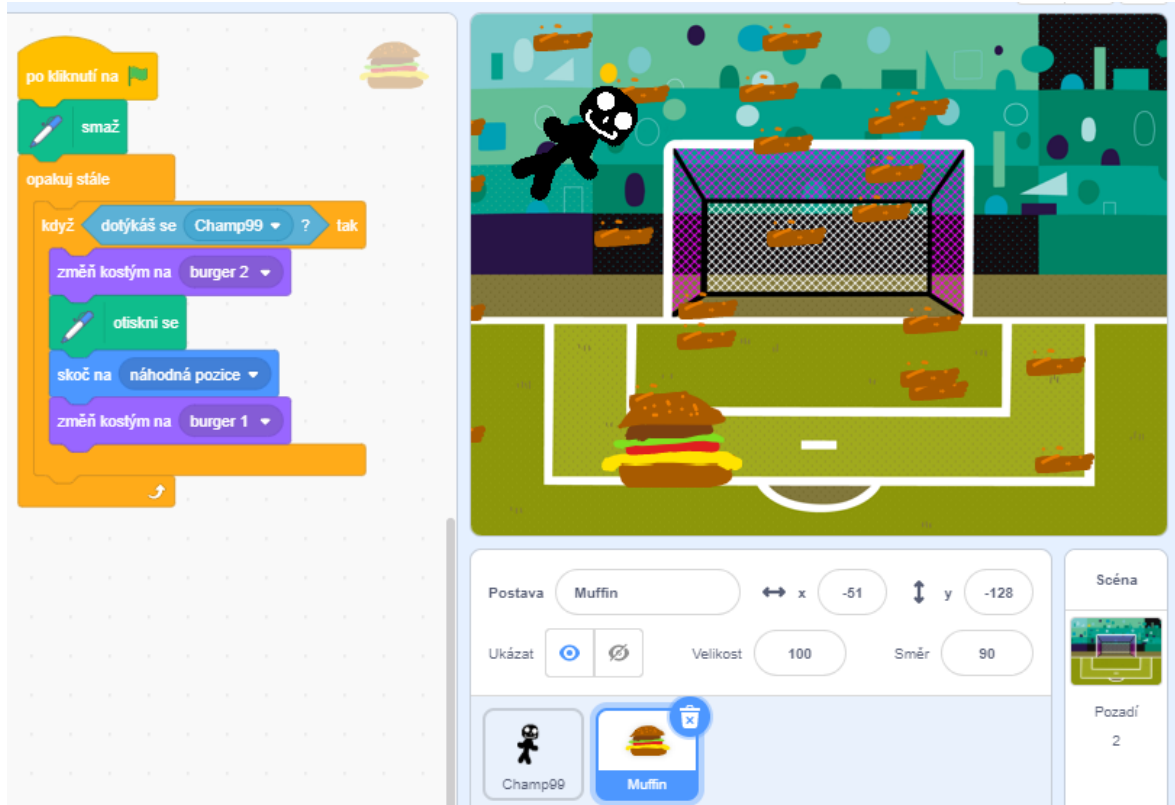
Příloha 21: Ukázka řešení úlohy č. 6: "smaž" u postavy "myši"



Příloha 22: Ukázka řešení úlohy č. 6: pohyb postavy "myš"



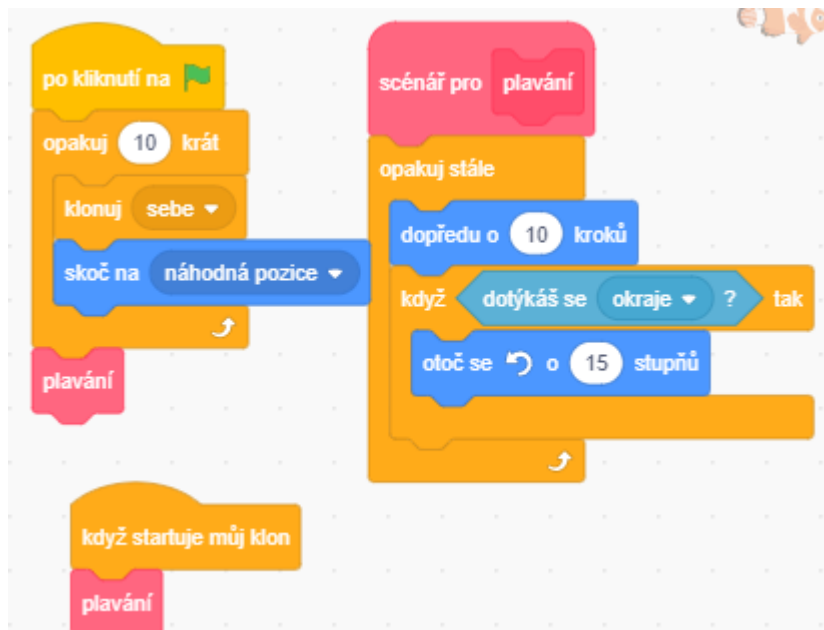
Příloha 23: Ukázka řešení úlohy č. 6: vlastní postavy



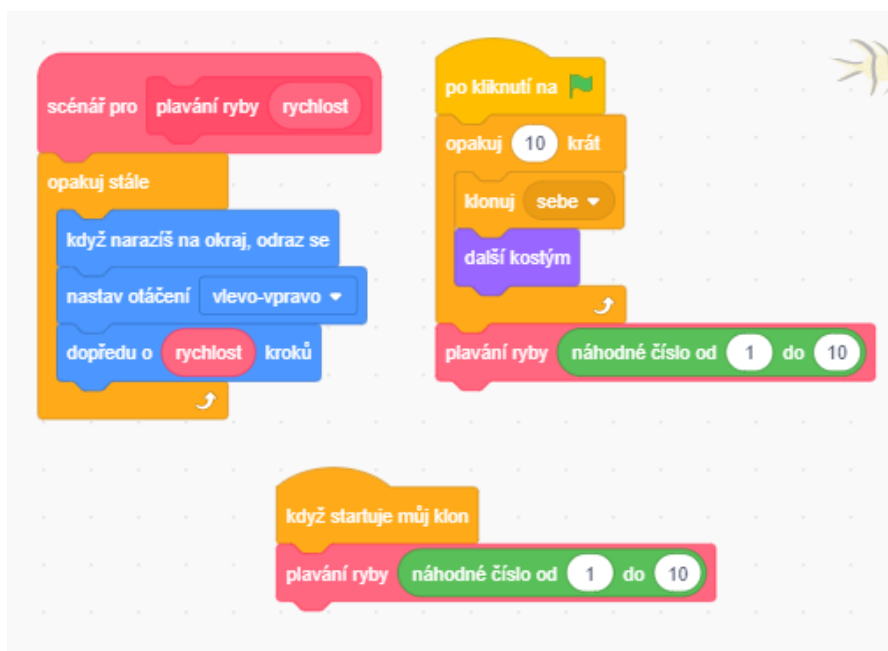
Příloha 24: Ukázka řešení úlohy č. 6: špatné pořadí kroků



Příloha 25: Ukázka správného řešení úlohy č. 7: naprogramování odrážení od okraje



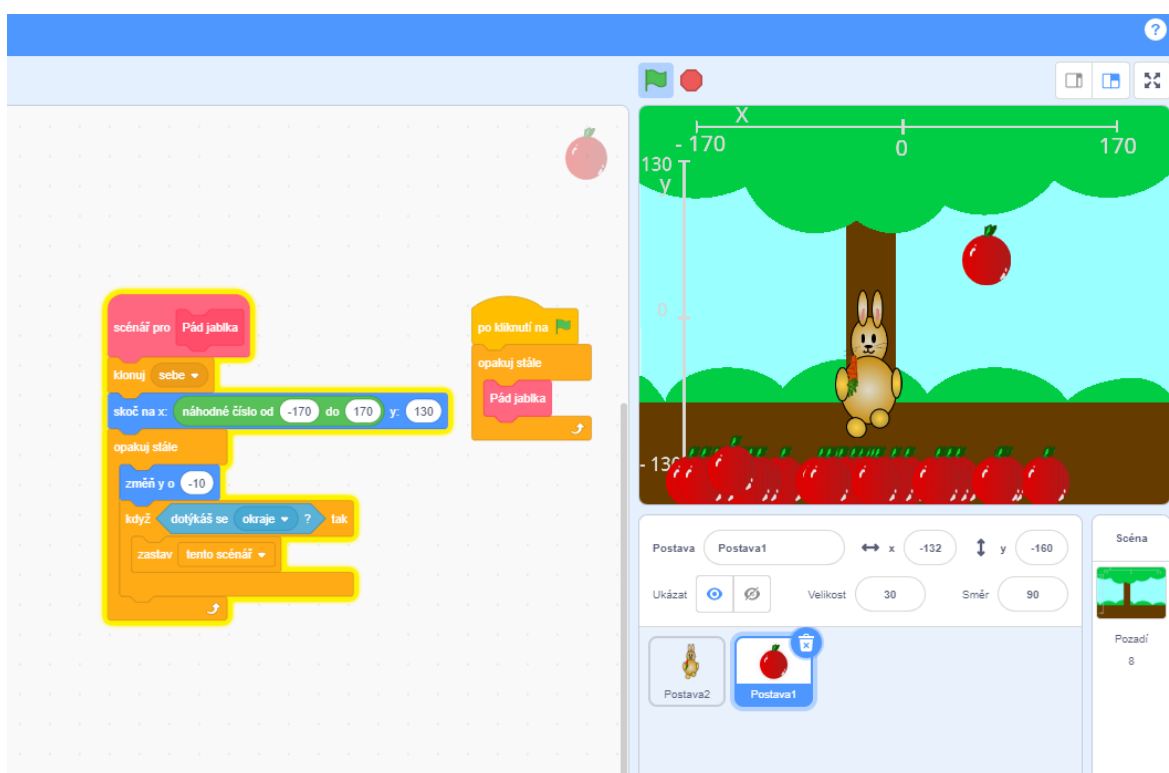
Příloha 26: Ukázka řešení úlohy č. 7: využití náhodných čísel



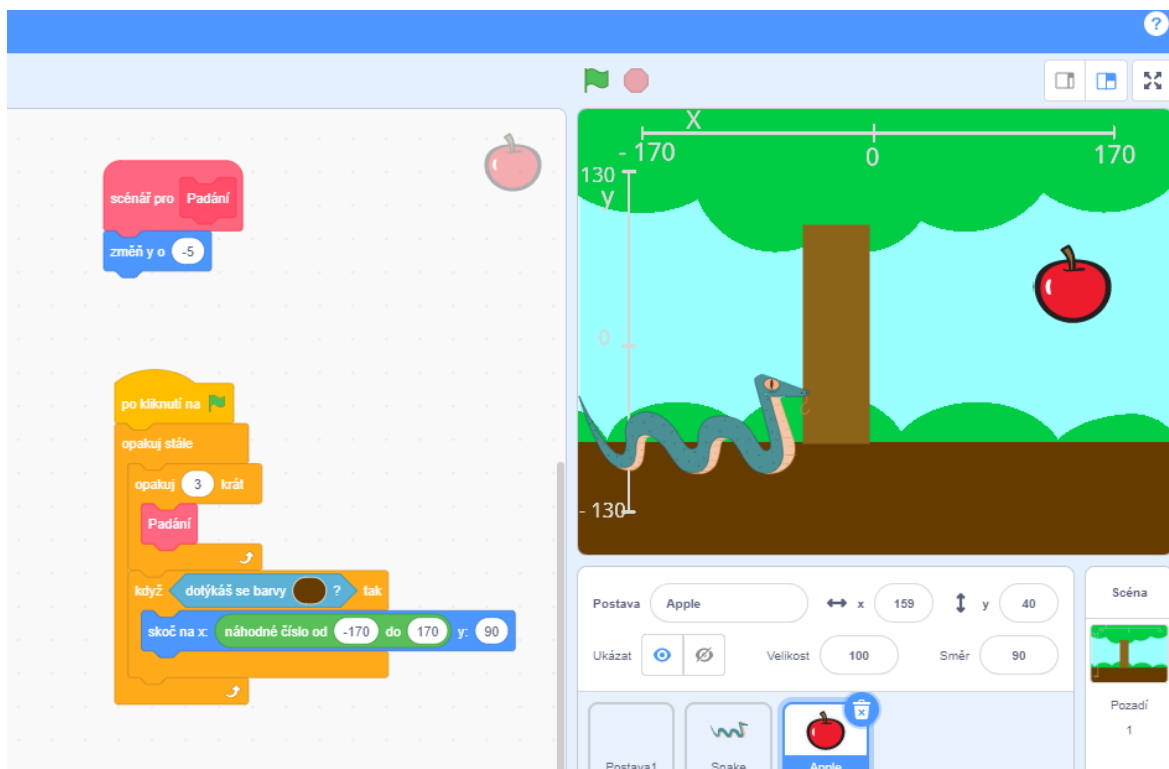
Příloha 27: Ukázka řešení úlohy č. 7 bez "když startuje můj klon"



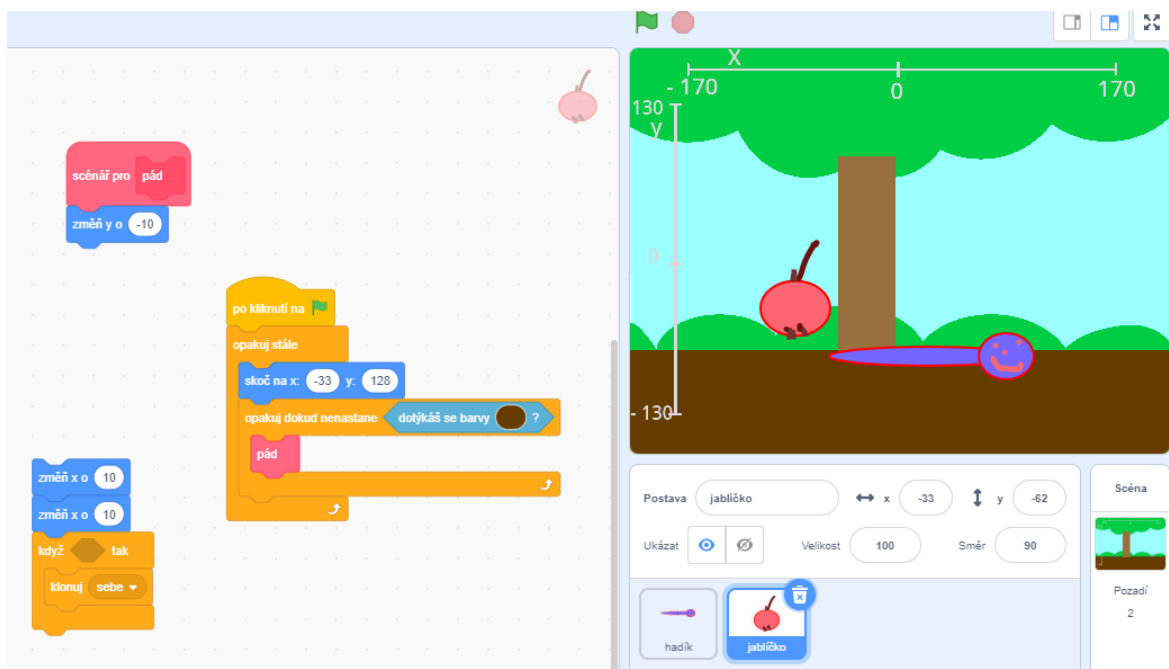
Příloha 28: Ukázka správného řešení úlohy č. 8



Příloha 29: Ukázka správného řešení úlohy č. 8 s „dotýkáš se barvy _“



Příloha 30: Ukázka řešení úlohy č. 8 bez otisknutí



Příloha 31: Ukázka řešení úlohy č. 8 se záměnou bloků „změň y o _“ a „změň x o _“

