



Pedagogická  
fakulta  
Faculty  
of Education

Jihočeská univerzita  
v Českých Budějovicích  
University of South Bohemia  
in České Budějovice

JIHOČESKÁ UNIVERZITA V ČESKÝCH  
BUDĚJOVICÍCH

Pedagogická fakulta

Katedra informatiky

**Výuka programování ve Scratch zaměřená na  
vytváření pojmů**

**Concept-oriented education if programming  
in Scratch**

Diplomová práce

**Vypracovala:** Bc. Kristýna Holečková

**Vedoucí práce:** doc. PaedDr. Jiří Vaníček, Ph.D.

České Budějovice 2017

**ZADÁNÍ DIPLOMOVÉ PRÁCE**  
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Kristýna HOLEČKOVÁ**  
Osobní číslo: **P15302**  
Studijní program: **N7503 Učitelství pro základní školy**  
Studijní obory: **Učitelství technické výchovy pro 2. stupeň základních škol**  
**Učitelství informatiky pro 2. stupeň základních škol**  
Název tématu: **Výuka programování ve Scratch zaměřená na vytváření**  
**pojmu**  
Zadávající katedra: **Katedra informatiky**

**Z á s a d y p r o v y p r a c o v á n í :**

Značná část současných výukových materiálů pro výuku programování pro začátečníky předkládá žákům jako výukové jednotky tzv. projekty, tedy sled aktivit různého typu, jejichž postupným prováděním žák vytvoří nějaký produkt, např. hru nebo animaci. Aktivita jsou pak navzájem doplňkové, každá vyžaduje jiné kompetence a jiný problém k vyřešení a jejich realizací nedochází ke zpevnění žákových znalostí. Cíle této práce je poskytnout žákovi sady krátkých úloh, tzv. etud, týkajících se vždy téhož učiva, téhož pojmu, nahlíženého v různých situacích a postupně zesložitujících se. Žák pak může rychleji vytvořit univerzální mentální model vyučovaného pojmu a lépe a více do hloubky mu tak porozumět.

Studentka vytvoří sadu úloh pro výuku základů programování pro základní školu a ověří funkčnost prostředí ve výuce. Po vyzkoušení úloh ve škole bude provedeno vyhodnocení a jejich případné následné upravení. Jednou z částí práce bude vyhodnocení úloh, které většinou dělají největší problémy žákům. Dále studentka vytvoří manuál, kde bude detailně popsáno pro učitele, na jaké principu je úloha založena, rozvíjené kompetence u žáků a návrh správného řešení. Součástí tohoto manuálu bude i zdůraznění konkrétních částí v úloze, na které by se učitelé měli nejvíce zaměřit.

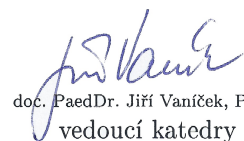
Rozsah grafických prací: **CD ROM**  
Rozsah pracovní zprávy: **60**  
Forma zpracování diplomové práce: **tištěná**  
Seznam odborné literatury: **viz příloha**

Vedoucí diplomové práce: **doc. PaedDr. Jiří Vaníček, Ph.D.**  
Katedra informatiky

Datum zadání diplomové práce: **22. března 2016**  
Termín odevzdání diplomové práce: **30. dubna 2017**



Mgr. Michal Vančura, Ph.D.  
děkan



doc. PaedDr. Jiří Vaníček, Ph.D.  
vedoucí katedry

V Českých Budějovicích dne 22. března 2016

## Prohlášení

Prohlašuji, že svoji diplomovou práci jsem vypracovala samostatně, pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 25. 4. 2017

Bc. Kristýna Holečková

## Abstrakt

Cílem diplomové práce je vytvořit sadu krátkých úloh pro výuku programování na základních školách v prostředí Scratch, které pomohou žákovi pochopit a porozumět do hloubky danému vyučovacímu pojmu z oblasti algoritmizace a programování. Sady obsahují úlohy vztahující se k jednomu pojmu, které ale bude vždy podáno jiným způsobem. Nejprve jsem vybrala některé klíčové pojmy z výuky programování pro 2. stupeň ZŠ a k těmto pojům si stanovit kompetence, které žáci potřebují na to, aby tomu pojmu porozuměli. Následně k těmto kompetencím jsem hledala nebo vytvářela úlohy, ve kterých se tyto kompetence budou rozvíjet.

Diplomová práce je rozdělena na praktickou a teoretickou část. V teoretické části jsou uvedena teoretická východiska z oblasti tvorby pojmů a konstruktivistického přístupu k výuce a následně jsou popsány veškeré úlohy, včetně rozvíjených kompetencí u žáka. Dále teoretická část diplomové práce obsahuje manuál, ve kterém je detailně popsáno pro učitele, na jakém principu jsou úlohy založeny a návrh správného řešení. Součástí tohoto manuálu je i zdůraznění metodických poznámek v úloze, na které by se učitelé měli nejvíce zaměřit. Praktická část práce obsahuje právě sady úloh, které byly vyzkoušeny přímo ve výuce na základní škole a byly průběžně vyhodnocovány a případně upraveny.

## Klíčová slova

Programování, algoritmizace, výuka, vytváření pojmů, základní škola, Scratch

## Abstract

The focus of this diploma thesis is to create a set of short tasks for teaching programming at primary schools focused on Scratch programming environment. Those tasks should help students to deeply understand of particular term, functions and syntax of commands. Also this set will contain tasks focused on particular term but with different approach of learning the term. First

I chose some of the key concepts of teaching programming for 2nd grade primary school. For each set and task I had to set which term will the task be focused on and how the task will accomplish students key learning preferences which is important for learning students. Those task has been developed based on this learning preferences.

Diploma thesis is divided into two parts, practical and theoretical part. Theoretical part contains the theoretical starting points from the field of conceptual design and constructivist approach to teaching and consequently contains description of all task including students developing competenses. This part also contain manual for teachers containing pcinipal, core and correct solution of the tasks all described in detail with highlights of important parts for teacher to carefully focuse on. The second and practical part of the diploma theses contains sets of tasks which have been directly tested in teaching on primary school and have been edited during this testing period of time.

## **Keywords**

Programming, algorithm, education, concept-oriented education, primary school, Scratch

## Poděkování

Děkuji tímto panu doc. PaedDr. Jiřímu Vaníčkovi, Ph.D. za odborné vedení mé diplomové práce, za velmi cenné rady a připomínky, které mi v průběhu vypracovávání této práce dával a za velice dobrou komunikaci.

Dále bych chtěla poděkovat přátelům za oporu a především rodině, která mě podporovala po více stránkách při studiu na vysoké škole.

Děkuji také všem, kteří se jakkoliv podíleli na zkvalitnění této diplomové práce.

Těž děkuji katedře informatiky a matematiky za možnost zapojení se do projektu GAJU 121/2016/S, díky kterému jsem měla možnost rozvíjet tuto práci v zahraničí a zúčastnit se na konferenci v Novém Městě na Moravě.

# Obsah

<b>1 Úvod</b>	<b>10</b>
1.1 Východiska práce . . . . .	11
1.2 Cíle práce . . . . .	11
1.3 Metoda práce . . . . .	11
<b>2 Současná situace výuky programování na základních školách</b>	<b>12</b>
2.1 Inovace ICT kurikula 2017 . . . . .	13
2.2 Programování na základních školách ve Slovinsku . . . . .	14
<b>3 Programovací prostředí Scratch</b>	<b>14</b>
3.1 Výhody . . . . .	15
3.2 Nevýhody . . . . .	16
3.3 Další dětská programovací prostředí . . . . .	16
<b>4 Výuka programování zaměřená na porozumění pojmům</b>	<b>18</b>
4.1 Konstruktivistický přístup ve výuce . . . . .	18
4.2 Zaměření na expoziční a fixační metodu v rámci výchovně-vzdělávacího procesu . . . . .	18
4.3 Ohlasy učitelů . . . . .	20
<b>5 Vytvořené sady úloh</b>	<b>20</b>
5.1 Sada úloh na naučení a procvičení skládání pojmů . . . . .	21
5.1.1 01 Skládání příkazů - Kočička . . . . .	23
5.1.2 02 Skládání příkazů - Chození . . . . .	25
5.1.3 03 Skládání příkazů - Pozdrav . . . . .	27
5.1.4 04 Skládání příkazů - Fáze měsíce . . . . .	28
5.1.5 05 Skládání příkazů - Růst rostliny . . . . .	31
5.1.6 06 Skládání příkazů - Říkanka . . . . .	32
5.1.7 07 Skládání příkazů - Tanec . . . . .	33
5.2 Opakování . . . . .	35
5.2.1 01 Opakování - Chození rovně . . . . .	37
5.2.2 02 Opakování - Chození rovně a do strany . . . . .	39
5.2.3 03 Opakování - Fáze měsíce . . . . .	41
5.2.4 04 Opakování - Květina . . . . .	42



5.2.5	05 Opakování - Kreslení čtverce . . . . .	44
5.2.6	06 Opakování - Kreslení kružnice . . . . .	46
5.2.7	07 Opakování - Kreslení více čtverců . . . . .	48
5.2.8	08 Opakování Tabulka - Kreslení více čtverců . . . . .	50
5.3	Podmínky . . . . .	52
5.3.1	01 Podmínka - Šváb . . . . .	53
5.3.2	02 Podmínka - Myška . . . . .	55
5.3.3	03 Podmínka - Motýl a žabka . . . . .	56
5.3.4	04 Podmínka - Měnící se kočka . . . . .	57
5.3.5	05 Podmínka - Vítek jde domů . . . . .	58
5.3.6	06 Podmínka - Žralok hledá rybku . . . . .	60
5.4	Manuál pro učitele . . . . .	62
<b>6</b>	<b>Ověření úloh a způsobu této výuky na základní škole</b>	<b>63</b>
6.1	Ověřování úloh formou pozorování ve výuce . . . . .	63
6.1.1	Výsledky pozorování ve výuce . . . . .	63
6.1.2	Nejčastější opravy úloh . . . . .	64
6.2	Ověřování úloh formou otevřeného dotazníku . . . . .	64
6.2.1	Kladené otázky v dotazníkovém šetření: . . . . .	64
6.2.2	Zhodnocení zájmu dětí o programování na základě do- tazníkového šetření a pozorování ve výuce . . . . .	65
6.3	Shrnutí . . . . .	67
<b>7</b>	<b>Publikování práce</b>	<b>68</b>
<b>8</b>	<b>Závěr</b>	<b>69</b>

# 1 Úvod

Chceme-li, aby se děti naučily programovat a zároveň pochopily různé programovací pojmy a příkazy, je třeba připravit pro učitele informatiky úlohy zaměřené na tuto problematiku. V této práci představím svůj návrh na výuku programování, která bude zaměřena právě na vytváření pojmů za účelem lepšího pochopení programování z více hledisek a tím i jeho porozumění více do hloubky.

Téma, programování ve Scratch zaměřené na vytváření pojmů, jsem si vybrala z důvodu podpoření a zkvalitnění výuky programování na ZŠ. Nejen podle mého názoru výuka programování do základních škol bezpochyby patří, ovšem dodnes je toto téma na základních školách spíše výjimkou. Přitom právě programování rozvíjí u žáků logické, algoritmické a analytické myšlení, kreativnost, informatické ale i matematické dovednosti, a navíc je zde možnost propojení i s dalšími znalostmi z jiných předmětů.

Tato práce se liší od ostatních tím, že se žáci učí pomocí sad krátkých úloh rozeznávat funkci jednotlivých příkazů, vytvářejí si tím představu, co jaký pojem znamená ať už se jedná o pojem příkaz, podmínka, cyklus atd. Díky tomuto přístupu pochopí princip programování více do hloubky.

V rámci této práce bych chtěla všem učitelům ukázat, že programování se dá učit i jednoduchou a zábavnou formou v prostředí, které je kreativní a uzpůsobené dětem pro výuku základů programování. Úlohy byly navíc ověřeny přímo ve výuce na ZŠ a díky zpětné vazbě formou pozorování při výuce a následně otevřeným dotazníkem mohu říct, že většina žáků byla za tyto hodiny programování ráda, programování je bavilo, a dokonce by mu chtěli věnovat i více času ve výuce informatiky.

Ve své práci navrhuji vlastní postup při výuce informatiky a nový směr, jakým by se měla výuka programování nést. K této práci jsou připraveny sady úloh připravené v on-line dětském programovacím prostředí Scratch, které jsou ověřeny z výuky na ZŠ a zároveň je k těmto sadám připravený manuál pro učitele, kde jsou veškeré úlohy představeny, zahrnuje i postup práce a řešení úloh, různá doporučení při práci s úlohami, ale i postřehy z hodin, tedy upozornění na nejčastější problémy a chyby žáků při řešení úloh.

## 1.1 Východiska práce

Donedávna bylo programování na základních školách jen výjimečné. Dnes jsou již pro učitele a žáky dostupné výukové materiály pro výuku programování pro začátečníky. Tyto materiály jsou ale povětšinou jen projekty (např. hry) s různým typem aktivit, kterými žák vytvoří nějaký výsledný produkt. Ovšem tyto aktivity jsou zaměřeny na více pojmů, vyžadují celou řadu dovedností z různých oblastí, tudíž žáci nemají dané téma ucelené a tím mu nedokáží porozumět do hloubky.

## 1.2 Cíle práce

Hlavním cílem této práce bylo vytvořit sady krátkých úloh v prostředí Scratch, přičemž každá sada se bude zaměřovat pouze na jeden daný pojem nebo téma. Úlohy jsou rozdílně pojaté, aby žák pochopil pojem z více hledisek a tím mu i porozuměl více do hloubky. Hotové úlohy jsou vyzkoušeny přímo ve výuce na základní škole a následně případné nedostatky v úlohách jsou doladěny.

Dalším cílem bylo vytvořit jakýsi manuál pro učitele, kde jsou jednotlivé úlohy detailně popsány, na jakém principu jsou založeny a budou zdůrazněny konkrétní části kódu v úloze, na které by se učitelé měli nejvíce zaměřit. Dále v manuálu budou uvedené i rozvíjené kompetence u žáka a samozřejmě i návrh způsobu správného řešení úloh.

## 1.3 Metoda práce

Rozhodla jsem se, že vytvořím výuku programování žáků na ZŠ tak, aby žáci pochopili pojmy v programování více do hloubky.

Nejprve jsem tedy po nastudování potřebné literatury postupně vytvářela sady úloh, které se budou věnovat jen na jeden hlavní pojem, například jen podmínkám. Musela jsem si tedy říct dané pojmy a k těmto pojmům jsem si stanovila kompetence, které žáci potřebují na to, aby tomu pojmu porozuměli. Následně k těmto kompetencím jsem hledala nebo vytvářela úlohy, ve kterých se tyto kompetence budou rozvíjet. Slovem pojem se myslí název a funkce nějakého příkazu. V této práci konkrétně jsou tři sady a každá z nich učí žáky jinému základnímu pojmu. Jedna učí skládání příkazů, druhá opakování neboli

cykly a třetí podmínky. Do těchto tří oblastí se řadí zase další pojmy. Například u opakování se používá parametr nebo u podmínky konkrétní její výraz atd. Zároveň jsou zde úlohy s různými metodami učení. Některé úlohy žáci musí vytvořit téměř celé sami, někdy zase musí opravit kód tak, aby program fungoval správně, jinde zase musí například říct nebo nakreslit, co program vlastně dělá. Pokud se žáci naučí těmto pojmům porozumět, budou vědět, jak fungují, jak se používají v různých typech situací, poté budou schopni vytvářet náročnější programy a tím i lépe porozumí principu programování více do hloubky.

Vytvořené úlohy jsem poté zkoušela přímo ve výuce na základní škole, kde jsem v průběhu výuky aplikovala pozorovací metodu na ověřování úloh, zda jsou vytvořené tak, aby jim žáci rozuměli, dokázali je samostatně zpracovávat, zda je vůbec takovéto úlohy baví, a hlavně jestli těmito úlohami si žáci opravdu vytvářejí pojmy, které na základě svých představ a poznatků následně dokážou aktivně používat i v řešení dalších úloh. Na základě pozorování žáků při hodině jsem následně nějaké úlohy upravovala, aby byly co nejoptimálnější při další výuce a žáci jim co nejlépe porozuměli.

Na konci působení na základní škole jsem ještě provedla ověřování formou dotazníkového šetření, které zjišťovalo, zda žáky tyto hodiny programování bavily, jaké úlohy je zaujaly nejvíce a které naopak nejméně a proč.

Na základě vytvoření těchto sad úloh jsem ještě vytvořila dokument pro učitele tzv. „*Manuál pro učitele*“ s popisem úloh, včetně návrhu správného řešení dané úlohy a případnými doporučeními pro vedení výuky a upozorněními na nejčastější chyby žáků při plnění úloh.

## 2 Současná situace výuky programování na základních školách

Zařazení programování do výuky již na ZŠ je velmi důležitým krokem k rozvíjení kompetencí jako abstraktní a logické myšlení, tvořivost, preciznost a řešení problémů. Bohužel se ale ani v dnešní době programování příliš nezařazuje do výuky na základních školách. Učitelé dávají přednost spíše klasickým uživatelským tématům jako grafika, práce s kancelářskými programy, práce s internetem apod. [1][15]

Kniha *Oborové didaktiky: vývoj – stav – perspektivy* obsahuje kapitolu rozebírající také aktuální stav výuky informatiky na českých školách a rozebírá i možné důvody, proč učitelé učí raději právě spíše klasické uživatelské programy, kde žáci jsou spíše tzv. konzumenty a uživateli, nikoliv tvůrci a autory. Může to být způsobeno jak vzděláním učitelů a jejich kompetencí k předmětu IT, tak například nedostatečnou hodinovou dotací.[2]

Programování přitom lze spojit i s jinými předměty a o to může být zajímavějším i zábavnějším. Matematika je u programování zcela běžná, ale také lze vymýšlet úlohy zaměřené na fyziku, přírodopis, zeměpis a další předměty.

Když jsem se poohlížela po školách, kde bych vyzkoušela úlohy v prostředí Scratch v rámci diplomové práce, udělala jsem malý průzkum, v kolika a jakých základních školách učitelé zařadili programování do výuky na 2. stupni ZŠ. Ptala jsem se na osmi náhodně vybraných školách, z nichž pouze dvě mají programování v rámci výuky informatiky. Na těchto školách jsem se zeptala učitelů informatiky, zda znají dětské programovací prostředí Scratch. Všichni odpověděli, že toto prostředí znají, spíše o něm jen slyšeli, ale nikdy v něm samostatně nepracovali. Na základě výsledků tohoto průzkumu mohu konstatovat, že je třeba pomoci současným i budoucím učitelům informatiky a ulehčit jim práci s učením se v prostředí Scratch tím, že jim dám předlohu různých úloh, které by mohli zařadit do výuky na ZŠ. Dále jsou tyto úlohy uzpůsobené žákům, aby pochopili a porozuměli programovacím pojmům více do hloubky.

## 2.1 Inovace ICT kurikula 2017

Na konferenci Počítač ve škole 2017 byla přednáška NUV<sup>1</sup>, kde byli učitelé informováni o inovaci ICT<sup>2</sup> kurikula, která se týká hlavně předmětu Informatika a informační technologie. Národní ústav pro vzdělávání zde představil nový návrh pro zaměření výuky informatiky na ZŠ a SŠ. Hlavním cílem těchto změn je začít učit v hodinách informatiky informatická témata jako algoritmizace a programování, informační systémy, informační technologie, pojem informace, reprezentace a modely. Mezi průřezové kompetence NUV řadí pojmy digitální gramotnosti, tedy Člověk, společnost a digitální technologie; Tvorba

---

<sup>1</sup>NUV - Národní ústav pro vzdělávání

<sup>2</sup>ICT - Information and Communication Technologies v překladu *Informační a komunikační technologie*

digitálního obsahu; Informace, sdílení a komunikace v digitálním světě; Řešení problémů v digitálním prostředí. Uživatelské ovládání počítače by se zařadilo do ostatních výukových oborů – např. Excel – matematika; textový editor – český jazyk; grafické editory – výtvarná výchova; prezentace – další předměty jako zeměpis, přírodopis atd. Znamená to tedy, že by bylo více času věnovat se v ICT programování a algoritmizaci, k čemuž se tato práce výborně hodí.[3]

## 2.2 Programování na základních školách ve Slovinsku

Díky možnosti výměnného pobytu formou mobility freemover, který proběhl v dubnu 2016, jsem měla možnost prozkoumat tuto problematiku i ve Slovinsku, konkrétně v rámci studia na Slovinské univerzitě v Lublani na fakultě informatiky, kde se též zabývají různými programy na výuku informatiky žáků na základních školách. Ve většině základních školách ve Slovinsku je programování běžnou součástí výuky informatiky. Dokonce v některých školách se Scratch učí už na prvním stupni ZŠ.[4]

## 3 Programovací prostředí Scratch

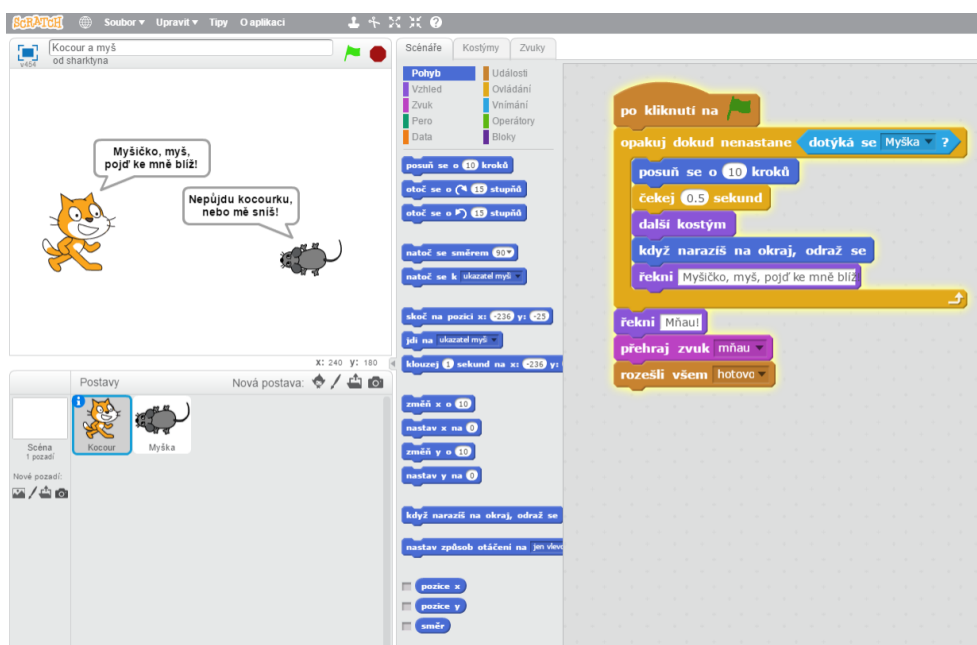
Programovací prostředí Scratch je výborným online nástrojem pro naučení se základů programování. Jedná se o velmi graficky povedené a kreativní prostředí, ve kterém se dobře orientuje i pracuje jak dětem, tak i dospělým. Velkou výhodou Scratche je, že veškeré úlohy, které uživatel vytvoří, jsou uloženy v rámci jeho účtu online a tím není třeba úlohy ukládat do počítače (pokud uživatel vyloženě nechce). Další velkou výhodou je sdílení úloh mezi uživateli a tím i možnost prohlížení jiných her a úloh v tomto prostředí.[5][18]

Prostředí Scratch je vhodná programovací aplikace jak na výuku a procvičování informatického, tak i matematického myšlení. V článku *Želví geometrie pro 21. století aneb Scratch ve výuce algoritmizace* je popsáno a vysvětleno na mnoha příkladech, jak je možné zkombinovat výuku právě informatiky s matematikou zábavnou formou. [6]

Programování v prostředí Scratch se provádí pomocí grafických příkazů (ikon), které se skládají na sebe a tím se vytváří program. Ke skládání těchto příkazů je určena pravá polovina prostředí. Pokud uživatel v hlavní nabídce

příkazů nenajde vhodný příkaz, má k dispozici tzv. bloky, díky nimž si pomocí několika příkazů vytvoří samostatný vlastní příkaz.

Toto prostředí se dělí na 3 části. První část, která se nachází nalevo, je obrazovka pro okamžitou zpětnou vazbu, kde se uživateli zobrazují naprogramované objekty, které si uživatel může zvolit z připravené knihovny Scratche, vložit si vlastní objekty stažené z internetu nebo vzaté z počítače a nebo si je může sám nakreslit. Uprostřed prostředí je hlavní nabídka příkazů, které jsou rozčleněny do jakých si skupin (šuplíků). Samotné příkazy vybrané z hlavní nabídky příkazů, případně vytvořené pomocí bloků, se skládají napravo do „nekonečného prostoru“.



Obrázek 1: Prostředí Scratch

### 3.1 Výhody

- Podpora kreativní práce
- Hezké grafické prostředí
- Online prostředí
- Registrace zdarma

- Komunikace s dalšími uživateli (komentáře ke sdíleným úlohám, zprávy)
- Sdílení projektů
- Prostředí v češtině (možnost změnit jazyk)
- Propojení s Lego roboty a senzorickou deskou Picoboard
- Vytváření studií (sdílená „složka“ s několika úlohami)

## 3.2 Nevýhody

- Flash aplikace – některé OS tyto aplikace nepodporují
- Není k dispozici mobilní aplikace
- Není možnost konvertovat do běžného uživatelského (textového) kódu pro pokročilejší programátory

## 3.3 Další dětská programovací prostředí

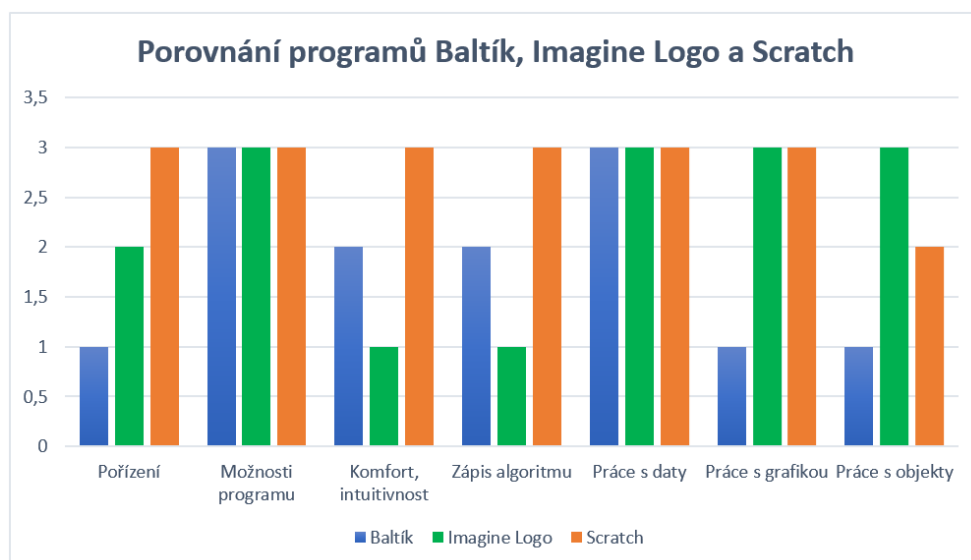
Jedním z dalších dětských programovacích prostředí je *Baltík*. Jedná se o výukový multimediální programovací a kreslicí nástroj pro děti a mládež, který rozvíjí logické myšlení a tvořivost. Pomocí *Baltíka* se děti mohou naučit i základní ovládání počítače, základní práci s textovým editorem a multimédií nebo práci na internetu a posílání elektronické pošty. V *Baltíkovi* programuje pomocí ikon, nikoliv pomocí textového kódu. Nevýhodou tohoto prostředí je, že není online a jedná se o placený program, který má zdarma pouze zkušební verzi, ve které ovšem nelze ukládat skripty ani scény.[7]

Dalším zajímavým programovacím prostředím je *Imagine Logo*, zde se jedná o textově orientovaný jazyk (například jako *Python* nebo *JavaScript*). Podporuje paralelní programování a má též propracovanou ideu obrázkových tvarů želv. Má některé nové prvky, které jsou typické pro programy pod *Windows*, např. překrývající se grafické plochy (jako listy papíru), tlačítka i s obrázky, posuvné lišty, textová pole, lišty tlačítek apod. Nechybějí ani multimédia, Internet a též vzájemná spolupráce *Imagine-programů* v síti. *Imagine Logo* je stejně jako *Baltík* placeným programem, který má také zdarma zkušební verzi, ve které nejdu ukládat vytvořené úlohy a programy.[8]



A nakonec bych ráda ještě zmínila programovací prostředí *EasyLogo*. Jedná se o velmi jednoduchý program (jednodušší než *Baltík*, *Imagine Logo* nebo *Scratch*), který zvládnou ovládat i děti na 1. stupni ZŠ. Prostředí *EasyLogo* má výstupní prostor s viditelnou mřížkou (editor) a poté postranní panel, který obsahuje řadu karet s ikonami příkazů, které uživatel může táhnout do editoru a tím skládat program. Kromě jednoduchosti je také výhodou tohoto prostředí fakt, že tento program je volně ke stažení.[9]

Inspirovala jsem se v diplomové práci *Výuka základů programování v prostředí Scratch* autora Jana Krejsy, který porovnával programovací prostředí *Baltík*, *Imagine Logo* a *Scratch* a bodoval jednotlivá prostředí. Na základě jeho porovnávání programů jsem vytvořila podobný graf, který porovnává možnosti jednotlivých programovacích prostředí a bodovala jsem je ve škále 0 - 4 (0 = nejhorší; 3 = nejlepší). Konkrétně nás oba zajímaly možnosti programu, pořízení (placený/neplacený program), komfort a intuitivnost, zápis algoritmu, práce s daty, grafikou a objekty. Z obrázku 2 znázorňující graf, který je podobně použit přímo v Krejsově práci, je zřejmé, že z těchto tří programů má nejlepší hodnocení právě Scratch. [1]



Obrázek 2: Porovnání programů Baltík, Imagine Logo a Scratch

## 4 Výuka programování zaměřená na porozumění pojmům

Aby žáci správně chápali pojem, vztahy mezi ním a dalšími pojmy, je zapotřebí, aby žák měl správně vytvořený tzv. *mentální model*. Tento model si člověk vytváří na základě daného pojmu podle představy, co takový pojem představuje, jak funguje a jak se chová. Jednotlivá stádia, kterými tvorba pojmu u jedince prochází, jsou popsána v knize profesora Milana Hejného *Teória vyučovania matematiky 2*. [10][11]

### 4.1 Konstruktivistický přístup ve výuce

Typ výuky je rozlišován podle aktivity žáka v hodině na transmisivní nebo konstruktivistický. Transmisivní přístup je v podstatě klasický frontální výklad zprostředkovávající poznatky v již nějaké hotové podobě. Naopak konstruktivistický přístup je o aktivní tvorbě poznatků (vytvoření znalosti) samotným žákem. Znamená to tedy, že na základě předchozích zkušeností a znalostí se pracuje s novými učícími se situacemi v aktivním podílu učícího se žáka. Učení je chápáno jako aktivní, adaptivní a zkušenostní proces. Žák se snaží porovnávat podobnosti a rozdíly ve svém vlastním kognitivním schématu v nové situaci, ve které se ocitne, a přitom toto schéma může měnit na základě předchozích zkušeností. [11]

### 4.2 Zaměření na expoziční a fixační metodu v rámci výchovně-vzdělávacího procesu

Výchovně vzdělávací proces se dělí podle jednotlivých metod na motivační, expoziční, fixační, diagnostický a aplikační. Ve vyučovacím procesu je motivace žáků jedním z nejpodstatnějších faktorů. Úkolem učitele je žáky k učení motivovat, vzbudit v nich zájem, vytvářet vhodné podmínky pro výuku. Bez správné motivace samozřejmě nebude docházet ani ke zvyšování efektivity učební činnosti žáků. „Čím vyšší je motivace žáků, tím lepší jsou výsledky jejich učení, čím lepší jsou učební výsledky, tím více jsou žáci k učení motivováni.“ [12] Během expozice dochází u žáků k vytváření pojmů, představ, formálních vědomostí a základů pro budování návyků. [13] K vytváření pojmů dochází prostřed-

nictvím myšlenkových operací (analýza, syntéza, dedukce, indukce, srovnávání, analogie atd.). Podstatou úspěšného osvojování poznatků je přesné vnímání a soustavné pozorování. Na základě vjemů a představ (výsledků vnímání a pozorování) je možno budovat pojmový aparát a hlouběji rozvíjet myšlení žáků. Bohužel také ve výuce někdy dochází ke vzniku tzv. formálních vědomostí, což znamená, že žák si sice nový poznatek pamětně osvojí, ale jeho smyslu nerozumí a není schopen jej aplikovat při řešení úloh. Poté následuje fixace, která se ve vyučovacím procesu realizuje opakováním a cvičením. Cílem fixace je upevnování již osvojených dovedností a vědomostí. Dále se provádí diagnostikování znalostí žáků, prověřování vědomostí at' už zkoušením, testováním a následně hodnocením. V poslední fázi, tedy aplikační, by žáci měli být schopni aplikovat osvojené vědomosti a dovednosti v praxi, tedy v našem případě vytvořit zcela samostatně nějaký delší a náročnější program, který v sobě bude mít více probraných pojmů. [14]

Často se u výuky programování stává, že cílem je vytvořit nějakou hru, nikoliv pochopení a porozumění pojmům v programování více do hloubky. Ve vytváření takových úloh se používá několik pojmů najednou v jednom programu a žáci tak nemají možnost pochopit a porozumět pojmům více do hloubky, když se každému zvláště' nevěnují delší dobu a nezkouší je v kratších úlohách, kde pochopí jejich podstatu a možnost práce s nimi.[15]

Tato práce se od ostatních liší právě tím, že jsem se ve výuce programování zaměřila hlavně na expoziční a fixační metodu vyučovacího procesu. Motivační metodu aplikuji v průběhu celé výuky různými způsoby. Například soutěžemi mezi žáky, ukázkou hotové úlohy, kterou žáci mají vytvořit, dobrou známkou za správné vyhotovení úlohy apod. Žáci pracují převážně samostatně bez učitelova navádění, experimentují a snaží se řešit krátké úlohy, které na sebe navazují. Tím si žáci vytváří nějaké představy a pojmy a učí se s nimi dále pracovat a rozvíjet tím i dosavadní vědomosti a dovednosti, ale zároveň i tím, že úlohy jsou vytvořeny tak, aby se každá řešila jiným způsobem. Upevnování, tedy fixace, osvojených dovedností (prací s pojmy a představami) dochází v opakování a řešení nových úloh, kde se procvičuje stále jeden a ten samý pojem, ale vždy jiným způsobem. Znalosti žáků následně ověřuji pozorováním jejich práce při výuce a testováním (písemným i slovním). Na základě dobrých výsledků při pozorování a testování by mohla následovat úloha, kde by žáci zcela

samostatně aplikovali své vědomosti.

Stejný výchovně-vzdělávací proces probíhá při vyučování matematiky. Žáci se seznamují s novými pojmy jako například matematickými operacemi součet, rozdíl, součin a podíl. Nejprve si osvojují jeden pojem a to součet. Ten procvičují na několika příkladech a začínají od nejlehčího po ty těžší. Následuje ověřování žakových znalostí a aplikování samotného pojmu do složitějšího příkladu. Následuje pojem podíl a tak dále.

### 4.3 Ohlasy učitelů

Učitelé, se kterými jsem mohla řešit problematiku ohledně programování na základní škole a zároveň konkrétně výuku v prostředí Scratch zaměřenou na vytváření pojmů, byli velmi nakloněni pro takový způsob výuky, kdy žáci budou pracovat na sadě úloh zaměřených vždy pro jeden pojem. Též byli potěšeni, že budou moci použít již vytvořené úlohy, tudíž jim tím alespoň částečně usnadním práci a budou vědět, jakým směrem dál mají se žáky v programování pokračovat, jak vytvářet další úlohy atp. Též jsem se setkala s učiteli, kteří se Scratchem nezačali, protože s ním nikdy nepracovali, a ještě nenašli čas nebo odvalu na prozkoumání tohoto prostředí. Po seznámení s cíli diplomové práce na toto téma byli tito učitelé velice rádi, že nějaké takovéto úlohy budou k dispozici a budou mít také inspiraci, co ve Scratchi je možné vytvořit.

## 5 Vytvořené sady úloh

V této kapitole bych chtěla představit úlohy vytvořené v rámci této diplomové práce. Jak už jsem psala výše, vytvořila jsem sady úloh, kterými si žáci osvojují pojmy v programování, učí se s těmito pojmy pracovat různými způsoby a metodami a tím se programování učí chápat více do hloubky.

Vytvořila jsem celkem tři sady, z nichž každá má různý počet úloh. Úlohy jsou poskládány tak, jak by měly jít za systematicky za sebou. Každá sada je zhotovena od nejjednodušších úloh po ty nejtěžší. Ve druhé sadě (opakování) se předpokládá, že žáci již umí pracovat s příkazy a správně je poskládat. Ve třetí sadě (podmínky) se zase již předpokládá, že žáci mají zvládnuté i cykly. Zároveň jsem se snažila, aby úlohy byly metodicky rozdílné, tedy že každá úloha

je jinak postavená než ta předchozí a podobně. V některých úlohách žáci mají správně přiřazovat již připravené příkazy, v jiných si žáci musí příkazy sami najít a správně je použít. Také jsou zde úlohy, kde žáci mají před sebou kód programu a musí říct nebo nakreslit, co ten daný program má udělat. Zároveň se úlohy mohou řešit jak ve dvojicích, tak samostatně, to už záleží na samotném učiteli.

Dostala jsem možnost vyzkoušet mnou vytvořené úlohy na základní škole L. Kuby v Českých Budějovicích. Měla jsem k dispozici dvě šesté a jednu sedmou třídu. Vyučování probíhalo souvisle 7 týdnů, kdy hodinu informatiky měli žáci jednou týdně 45 minut. Za tuto dobu jsem s nimi vyzkoušela úlohy na skládání příkazů, opakování a podmínky. Pár vhodných úloh pro tento styl výuky jsem převzala a upravila od jiných autorů (jsou citováni v komentářích daných úloh), ale většinu úloh jsem musela vytvořit sama. Inspirovala jsem se v několika knižních publikacích, jako například v *Imagine Logo: učebnice programování pro děti*, ale i v konkrétních webových aplikacích jako *Hodina kódu*. [8][17]

Své poznatky z pozorování práce žáků při řešení úloh, jako například nejčastější chyby, případná doporučení, co je třeba s žáky předem zopakovat a podobně, uvádím u každé úlohy zvlášť a zároveň jsou uvedeny i v manuálu pro učitele k těmto úlohám (o manuálu více v kapitole 5.4 na straně 62).

## 5.1 Sada úloh na naučení a procvičení skládání pojmů

V této sadě úloh se mají žáci naučit správně napojovat a poskládat příkazy tak, aby program pracoval podle zadání a instrukcí. Zároveň se postupně v úlohách žáci učí hledat chyby, kterých se při tvorbě programů mohli dopustit a zároveň je i správně opravit.

Rozvíjí tím jak základní klíčové kompetence, tak i kompetence nebo schopnosti jako preciznost, logické, algoritmické a abstraktní myšlení, kreativitu, řešení problémů a generalizaci, neboli žáci vyřeší jeden konkrétní problém a toto řešení může použít i v dalších úlohách.

Název úlohy + odkaz na Scratch	Bližší informace k úloze	Typ úlohy – poskládání již použitých příkazů, použití příkazů z nabídky nebo oprava kódu
1) Kočička	Podrobnosti na straně 23	Použití připravených příkazů
2) Chození	Podrobnosti na straně 25	Použití připravených příkazů – zhodnocení, zda se využijí všechny připravené příkazy
3) Pozdrav	Podrobnosti na straně 27	Použití připravených příkazů
4) Fáze měsíce	Podrobnosti na straně 28	Použití připravených příkazů
5) Růst rostliny	Podrobnosti na straně 31	Použití připravených příkazů
6) Říkanka	Podrobnosti na straně 32	Oprava již poskládané říkanky
7) Tanec	Podrobnosti na straně 33	Nalezení potřebných příkazů v nabídce a správně je použít

Tabulka 1: Tabulka s informacemi o úlohách v sadě na skládání pojmů

### 5.1.1 01 Skládání příkazů - Kočička

Úloha Kočička je zaměřená na skládání příkazů. Programovaný objekt, tedy kočička, se má podle zadání posunout o 10 kroků, změnit kostým na Hnědou kočičku a následně zamňoukat. Tato úloha je zařazená mezi první úlohy, které si žáci ve Scratchi vyzkouší. Učí se v ní skládat a přetahovat příkazy tak, aby kočička vykonala postupně vše podle zadání.

#### Doporučení

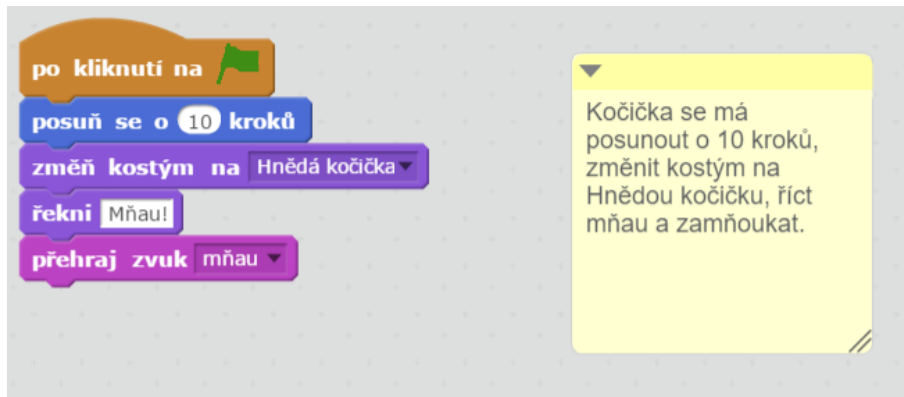
Žákům by se nejprve mělo představit samotné prostředí, vysvětlit jim, co jsou příkazy, kam příkazy přetahovat, aby skládaly program a kde se jim bude zobrazovat to, co naprogramovali. První úloha je vhodná na představení samotného prostředí, ale i na ukázkou práce s příkazy. Žáci by poté měli být schopni sami dokončit kód tak, aby odpovídal zadání.

#### Počáteční stav úlohy



Obrázek 3: Počáteční stav úlohy - 01 Skládání příkazů - Kočička

### Možné správné řešení



Obrázek 4: Možné správné řešení - 01 Skládání příkazů - Kočka

### Doporučení pro učitele

Tuto úlohu mají žáci hotovou poměrně rychle. Pokud to již mají poskládané správně a Kočka dělá vše, co má dělat podle zadání, mohou učitelé zkusit dát jim jiné zadání tak, aby si žáci zkusili přeskládat již poskládané příkazy.



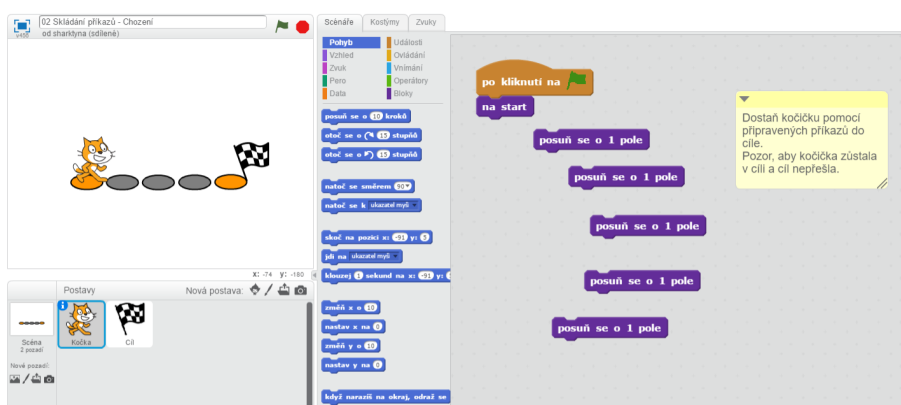
### 5.1.2 02 Skládání příkazů - Chození

V této úloze žáci musí správně spočítat, kolikrát má vlastně kočička popojít, aby došla přesně na políčko cíle. Je zde malý chytáček, že je připravený jeden příkaz „posuň se o 1 pole“ navíc.

#### Úprava úlohy

V původní neupravené úloze byl příkaz „zpět na start“. Z původního příkazů byli žáci dost zmatení a často ho dávali na konec úlohy, tudíž kočička došla do cíle a následně skočila na start, kde program skončil. Z důvodu nepřesného znění příkazů a matení žáků jsem příkaz upravila na „na start“.

#### Počáteční stav úlohy



Obrázek 5: Počáteční stav úlohy - 02 Skládání příkazů - Chození

## Možné správné řešení



Obrázek 6: Možné správné řešení - 02 Skládání příkazů - Chození

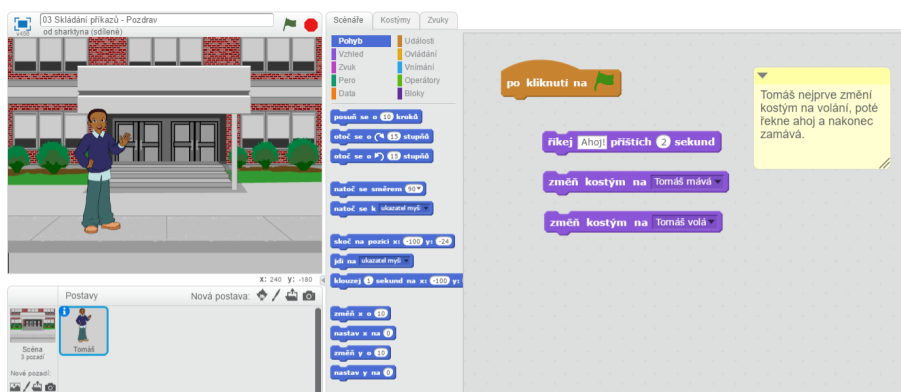
## Nejčastější chyby žáků při řešení této úlohy

Žáci často napojí všechny připravené příkazy a tudíž jim kočička přejde start a zastaví se až za oranžovým políčkem, ovšem zadání zní, aby se kočička zastavila v cíli, nikoliv za cílem. Další nejčastější chybou je, že žáci dávají příkaz „na start“ nakonec, tudíž kočičku na start musí dosadit ručně, poté zapnou program, kočička jim dojde do cíle a vrátí se zpět na start.

### 5.1.3 03 Skládání příkazů - Pozdrav

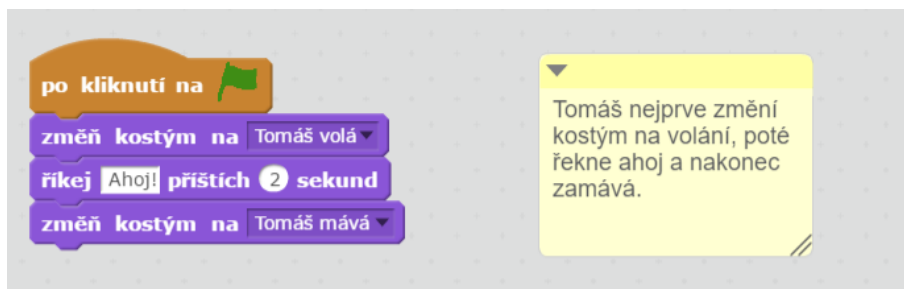
V této úloze mají žáci za úkol přiřadit příkazy postavě jménem Tomáš přiřadit příkazy tak, aby Tomáš změnil kostým tak, aby volal a následně aby říkal Ahoj a následně opět má změnit kostým, aby mával.

#### Počáteční stav úlohy



Obrázek 7: Počáteční stav úlohy - 03 Skládání příkazů - Pozdrav

#### Možné správné řešení

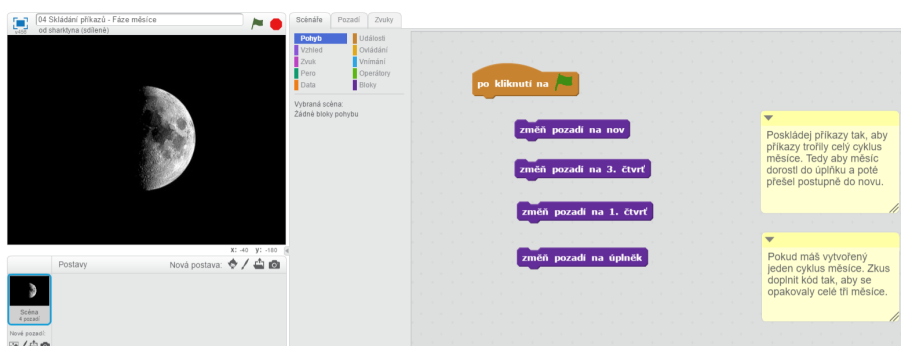


Obrázek 8: Možné správné řešení - 03 Skládání příkazů - Pozdrav

### 5.1.4 04 Skládání příkazů - Fáze měsíce

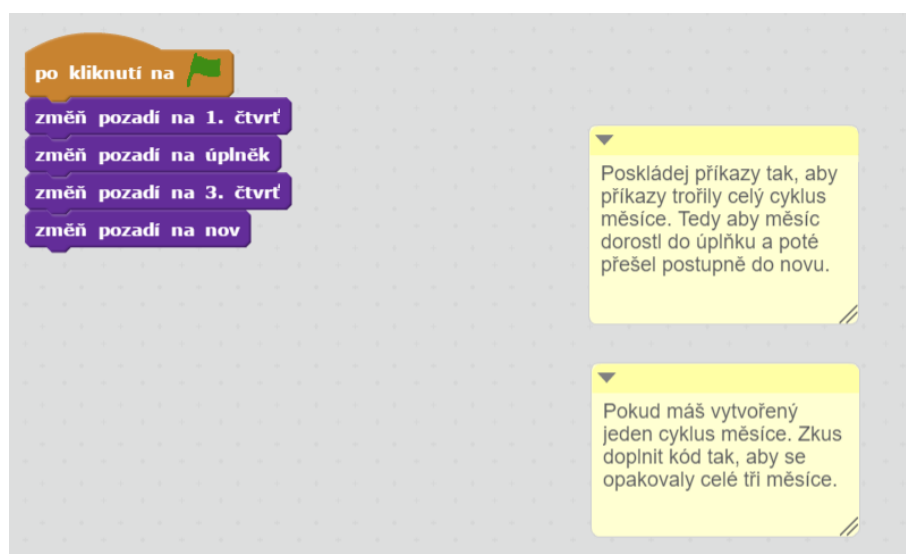
V této úloze by žáci měli pracovat se znalostmi ze zeměpisu, konkrétně z tématu vesmír, který se probírá nejčastěji hned na začátku tohoto předmětu. Tudíž by fáze měsíce měly být pro žáky pouhým opakováním, přesto ale tyto neznalosti byly nejčastějším problémem při řešení úlohy. Žáci mají přiřadit a poskládat příkazy tak, aby se opakoval celý cyklus měsíce. Tedy například z novu se dostat do úplňku a poté opět do novu.

#### Počáteční stav úlohy



Obrázek 9: Počáteční stav úlohy - 04 Skládání příkazů - Fáze měsíce

## Možné správné řešení



Obrázek 10: Možné správné řešení - 04 Skládání příkazů - Fáze měsíce

## Nejčastější chyby žáků při řešení této úlohy

Žáci často příkazy poskládají v pořadí 1.čtvrt', 3. čtvrt', úplněk, nov. Je to často tím, že neznají pojem čtvrt' a neví, jakým způsobem je měsíc na zemi osvětlený. Proto doporučuji ověřit, zda žáci znají princip „dorůstání“ a „couvání“ měsíce – měsíc nejprve musí „dorůst“ do úplňku a teprve potom začne „couvat“.

## Doplňková úloha - výsledek

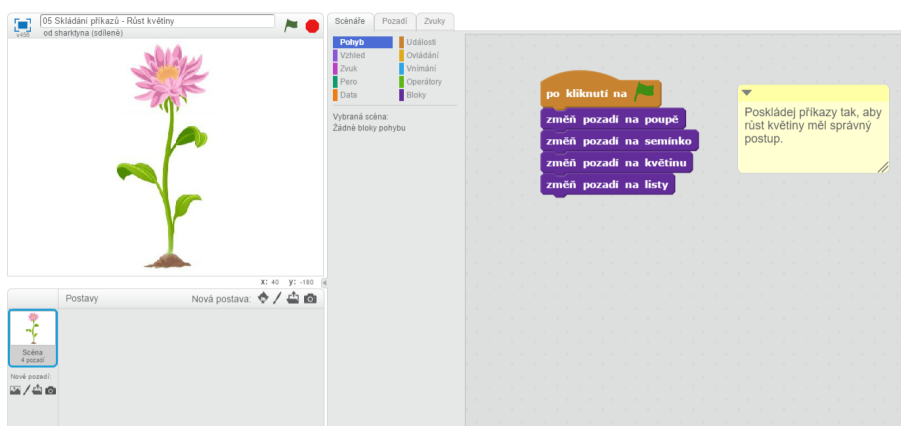
The image shows a sequence of programming blocks on a grey grid background. The first block is orange and says "po kliknutí na" with a green flag icon. It is followed by a vertical stack of 12 purple blocks, each with the text "změň pozadí na" followed by a phase: "1. čtvrt", "úplněk", "3. čtvrt", "nov", "1. čtvrt", "úplněk", "3. čtvrt", "nov", "1. čtvrt", "úplněk", "3. čtvrt", and "nov". To the right of the blocks are two yellow text boxes with a downward arrow icon. The top box contains the text: "Poskládej příkazy tak, aby příkazy trojily celý cyklus měsíce. Tedy aby měsíc dorostl do úplňku a poté přešel postupně do novu." The bottom box contains the text: "Pokud máš vytvořený jeden cyklus měsíce. Zkus doplnit kód tak, aby se opakovaly celé tři měsíce."

Obrázek 11: Doplňková úloha - 04 Skládání příkazů - Fáze měsíce

### 5.1.5 05 Skládání příkazů - Růst rostliny

V této úloze již žáci mají spojené příkazy hned na počátku. Musí tedy umět příkazy rozpojit a následně správně přiřazovat a skládat. Jejich úkolem je poskládat příkazy tak, aby růst květiny postupoval ve správném pořadí. V této úloze žáci častou zkouší několik možností a spíše tipují, jak by to mohlo být správně. Tedy zkouší i vícekrát měnění pořadí příkazů.

#### Počáteční stav úlohy



Obrázek 12: Počáteční stav úlohy - 05 Skládání příkazů - Růst květiny

#### Možné správné řešení

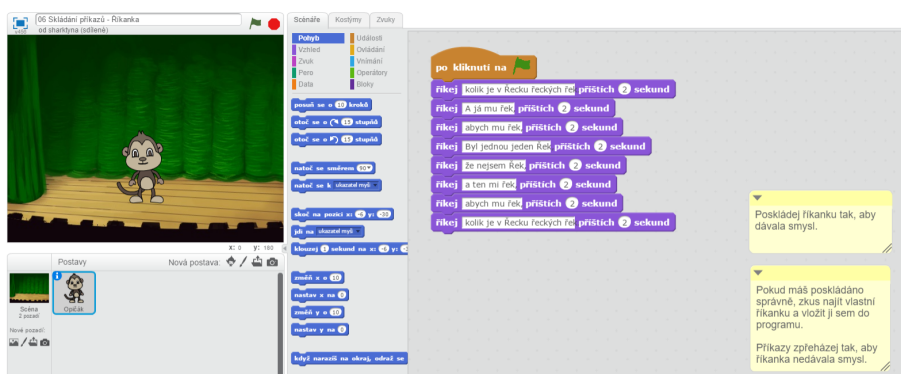


Obrázek 13: Možné správné řešení - 05 Skládání příkazů - Růst květiny

### 5.1.6 06 Skládání příkazů - Říkanka

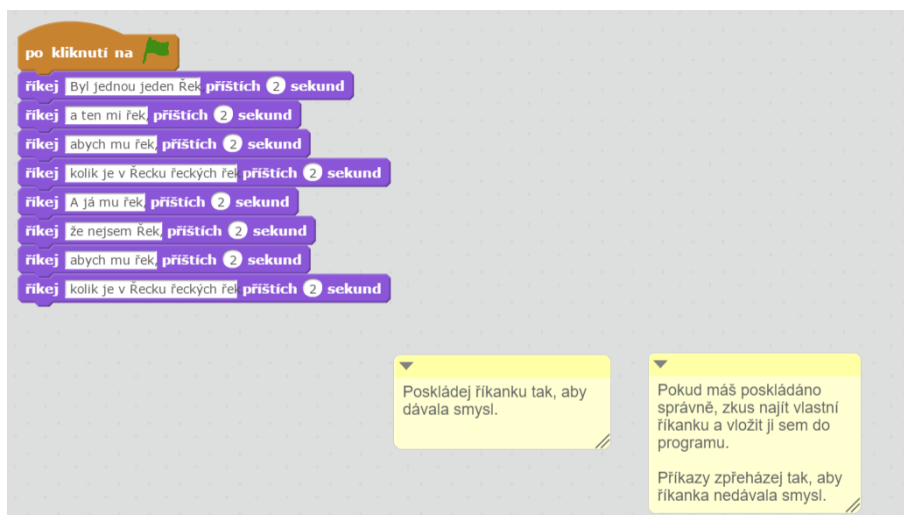
V této úloze žáci opět musí příkazy nejprve rozpojit a teprve následně skládat tak, jak jde říkanka za sebou. V další části po správném poskládání se žáci učí upravovat příkazy. V tomto případě text, který má Opičák říkat (více v doporučení).

#### Počáteční stav úlohy



Obrázek 14: Počáteční stav úlohy - 06 Skládání příkazů - Říkanka

#### Možné správné řešení



Obrázek 15: Možné správné řešení - 06 Skládání příkazů - Říkanka



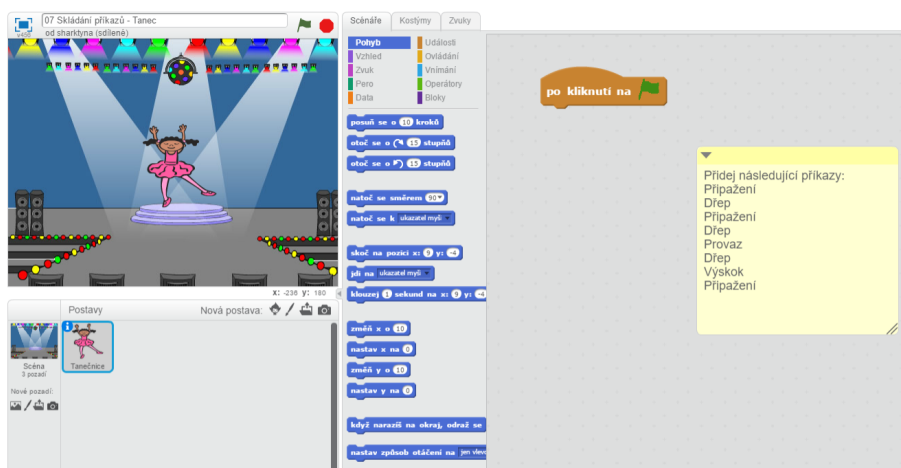
## Doporučení pro učitele

Někteří žáci tuto říkanku neznají, tudíž po nějaké době jim učitelé mohou doporučit, aby si ji zkusili najít na internetu a poskládat ji podle předlohy. Po poskládání této říkanky mají žáci připravit promíchané příkazy vlastní říkanky. Pokud je žádná nenapadá, může to být i nějaká známá písnička např. *Prší, prší* nebo *Skákal pes přes oves*. Když mají vymyšlenou a proházenou říkanku, doporučuji učitelům, aby žáky nechali mezi sebou skládat říkanky a aby tomu věnovali třeba zbytek hodiny, aby si žáci zkusili co nejvíce říkanek spolužáků. Žáky to baví, protože mají v podstatě volnost v hodině, baví je mezi sebou soutěžit a hlavně je to povzbudí do další hodiny informatiky.

### 5.1.7 07 Skládání příkazů - Tanec

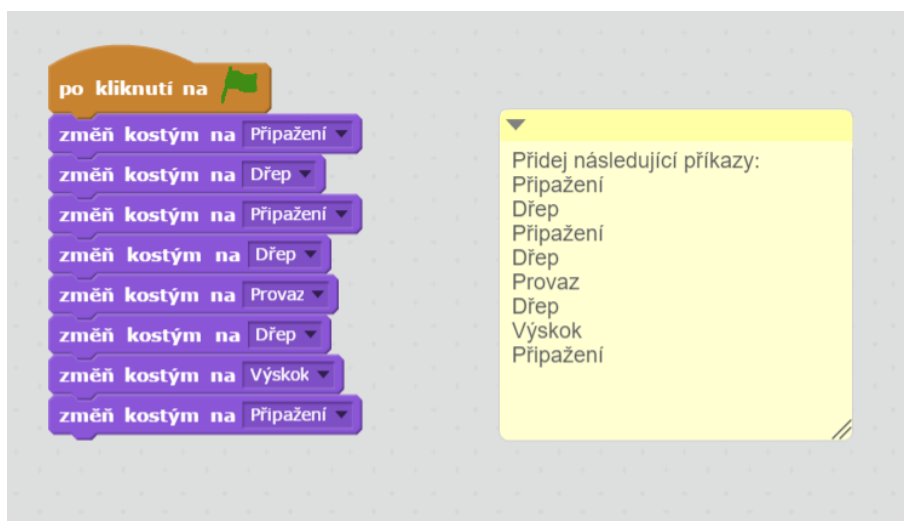
V této úloze se žáci musí začít orientovat již v nabídce příkazů a sami si musí příkazy vyhledat, přetáhnout do pravé pracovní plochy a následně skládat na sebe. Dále se v této úloze učí konkretizovat znění příkazu. V tomto případě musí určit, na co se má změnit kostým u postavy Tanečnice. Úkolem žáků je přiřadit příkazy Tanečnici tak, aby zatancovala sestavu, která je znázorněná v komentáři.

### Počáteční stav úlohy



Obrázek 16: Počáteční stav úlohy - 07 Skládání příkazů - Tanec

## Možné správné řešení



Obrázek 17: Možné správné řešení - 07 Skládání příkazů - Tanec

## Nejčastější problém při řešení této úlohy

Žáci mají problém vyhledat příkazy, které by odpovídaly zadání. Učitelé jim mohou třeba napovědět, například aby Tanečnice mohla provést následující příkazy, musí změnit Vzhled, kde je několik možností, co má vlastně na sobě změnit.

## 5.2 Opakování

Před plněním první úlohy této sady by již žáci měli umět poskládat příkazy, automaticky program vyzkoušet a případně začít hledat chyby. Také by se měli alespoň částečně orientovat v nabídce příkazů.

V této sadě úloh se mají žáci naučit poskládat příkazy do cyklu a mimo cyklus tak, aby program pracoval podle zadání a instrukcí. Zároveň by měli umět co nejvíce zjednodušit a zkrátit program o příkazy tak, aby byl co nejoptimálnější. Také by měli umět správně určit, kolikrát se cyklus musí opakovat, aby program vykonal to, co uživatel chce.

I touto sadou úloh žáci rozvíjí jak základní klíčové kompetence, tak i kompetence nebo schopnosti jako preciznost, logické, algoritmické a abstraktní myšlení, kreativitu, řešení problémů a generalizaci.

Název úlohy + odkaz na Scratch	Bližší informace k úloze	Typ úlohy – poskládání již použitých příkazů, použití příkazů z nabídky nebo oprava kódu
1) <b>Chození rovně</b>	Podrobnosti na straně 37	Použití připravených příkazů – zhodnocení, zda se využijí všechny připravené příkazy, když bude použitý cyklus s určitým počtem opakování
2) <b>Chození rovně a do strany</b>	Podrobnosti na straně 39	Použití připravených příkazů a úprava počtu opakování v cyklu – zhodnocení, zda se využijí všechny připravené příkazy, když bude použitý cyklus s určitým počtem opakování
3) <b>Fáze měsíce</b>	Podrobnosti na straně 41	Použití připravených příkazů a úprava počtu opakování v cyklu

<b>4) Květina</b>	Podrobnosti na straně 42	Oprava programu
<b>5) Kreslení čtverce</b>	Podrobnosti na straně 44	Použití připravených příkazů a úprava počtu opakování v cyklu a úprava stupňů v otáčení
<b>6) Kreslení kružnice</b>	Podrobnosti na straně 46	Použití připravených příkazů a úprava počtu opakování v cyklu
<b>7) Kreslení více čtverců</b>	Podrobnosti na straně 48	Vyhodnocení, co program dělá. Zjišťování kreslení na papír.; následné přesunutí přímo do programu a s použitím připravených příkazů a zároveň doplněním kódů z nabídky; úprava počtu opakování v cyklu
<b>7) Tabulka - Kreslení více čtverců</b>	Podrobnosti na straně 50	Použití připravených příkazů a zároveň doplnění kódů z nabídky; úprava počtu opakování v cyklu

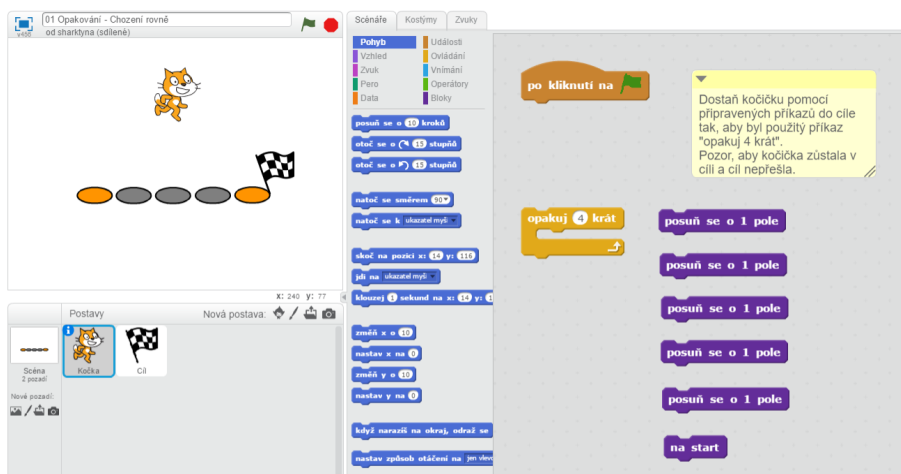
Tabulka 2: Tabulka s informacemi o úlohách v sadě na opakování

### 5.2.1 01 Opakování - Chození rovně

Tato úloha je stejná jako v první sadě Skládání příkazů, ovšem zde už žáci musí dostat kočku do cíle pomocí cyklu, který má nastavené opakování na počet 4.

U této úlohy se žákům musí říct, že příkazy se mohou dávat dovnitř cyklu. Následně jim vysvětlit, jak cyklus s takovými příkazy pracuje a co znamená to číslo 4 (opakuj 4 krát).

#### Počáteční stav úlohy



Obrázek 18: Počáteční stav úlohy - 01 Opakování - Chození rovně

## Možné správné řešení



Obrázek 19: Možné správné řešení - 01 Opakování - Chození rovně

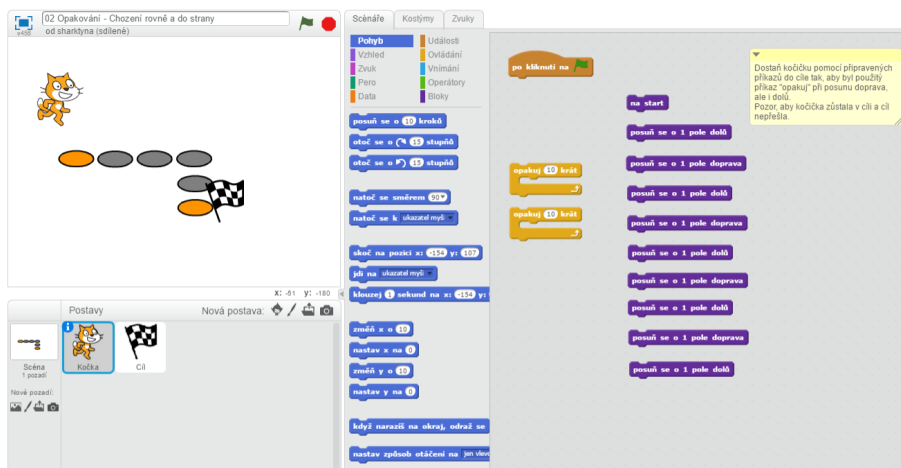
## Nejčastější chyby žáků při řešení této úlohy

Žáci často do cyklu vloží všechny příkazy, tudíž kočička přejde všechna políčka a nakonec zmizí z obrazovky (poté stačí dvakrát kliknout na příkaz „na start“ a kočička se ukáže zpět na startu).

### 5.2.2 02 Opakování - Chození rovně a do strany

Tato úloha je podobná jako první v této kapitole, jen se kočička neposouvá jen rovně (doprava), ale i dolů. Opět ale žáci musí použít co nejméně kódů a použít oba cykly, tak, aby kočička došla do cíle, ale ne dál.

#### Počáteční stav úlohy



Obrázek 20: Počáteční stav úlohy - 02 Opakování - Chození rovně a do strany

## Možné správné řešení



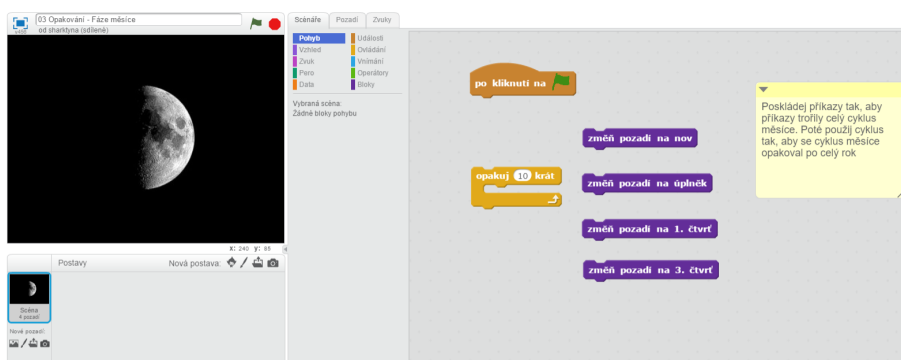
Obrázek 21: Možné správné řešení - 02 Opakování - Chození rovně a do strany



### 5.2.3 03 Opakování - Fáze měsíce

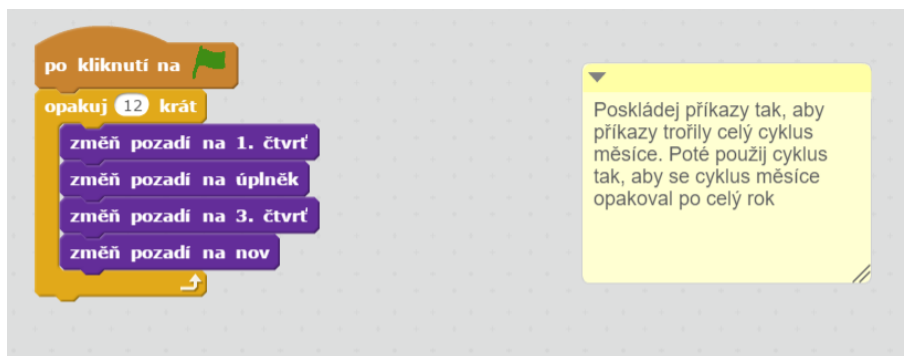
Tato úloha se také opakuje z předešlé sady. Ovšem zde si již žáci musí uvědomit, kolik měsíců má rok a kolikrát se tedy jeden celý cyklus měsíce ročně opakuje. Žáci tedy mají nejdříve poskládat příkazy tak, aby se měsíc měnil po dobu jednoho měsíce a následně použít opakování po dobu jednoho roku.

#### Počáteční stav úlohy



Obrázek 22: Počáteční stav úlohy - 03 Opakování - Fáze měsíce

#### Možné správné řešení



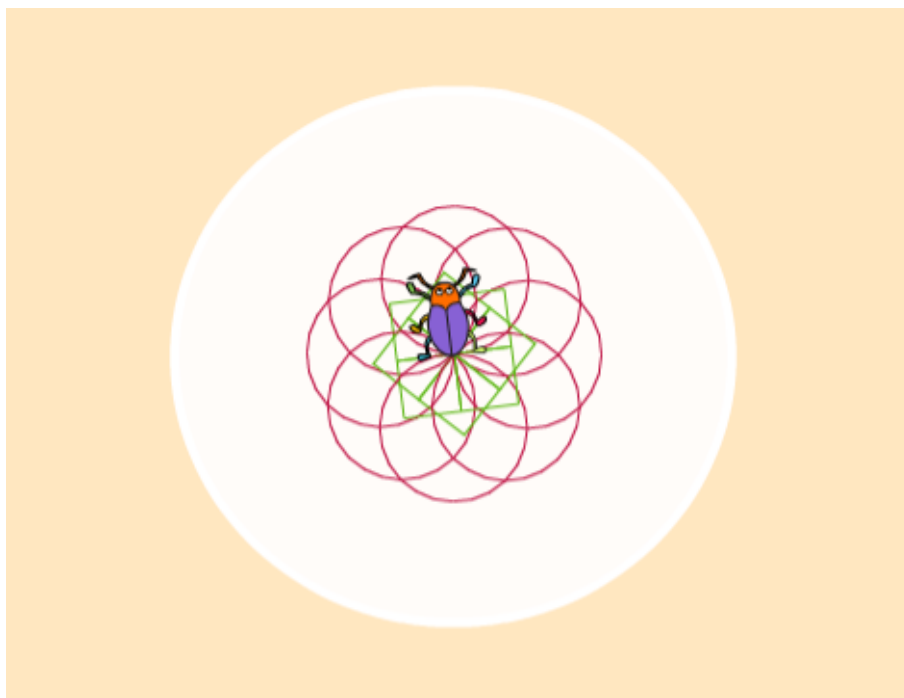
Obrázek 23: Možné správné řešení - 03 Opakování - Fáze měsíce

### 5.2.4 04 Opakování - Květina

V této úloze brouk za sebou maluje kružnice a čtverce ve tvaru podle toho, jak se příkazy nastaví - v tomto případě do tvaru květiny.

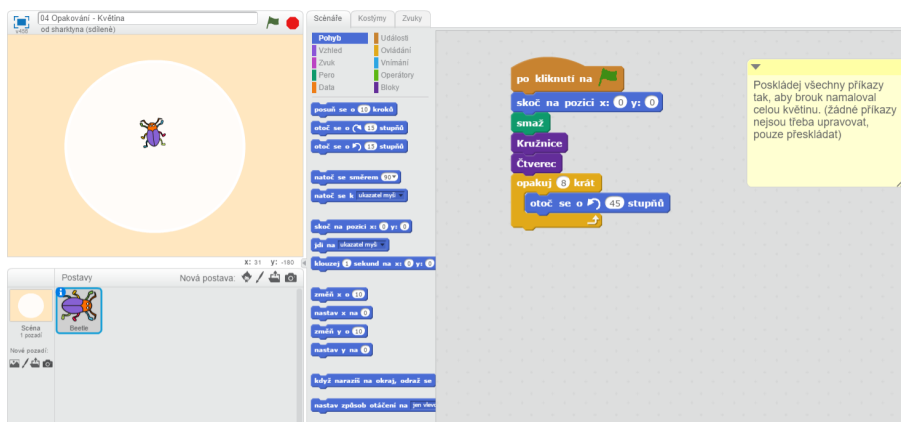
#### Doporučení pro učitele

Učitelé mohou zobrazit žákům náhled výsledku, co program dělá (neukazovat kód, pouze na stránce projektu program spustit) a žáci at' se pokusí sami ten kód poskládat podle ukázky.



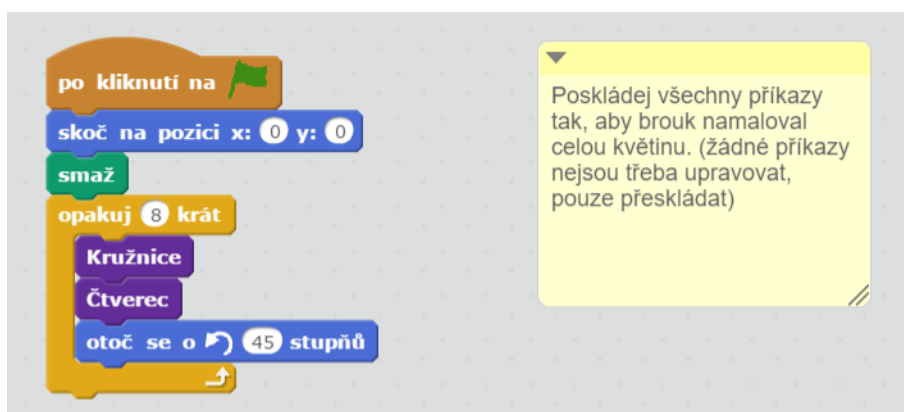
Obrázek 24: Náhled úlohy Květina

## Počáteční stav úlohy



Obrázek 25: Počáteční stav úlohy - 04 Opakování - Květina

## Možné správné řešení



Obrázek 26: Možné správné řešení - 04 Opakování - Květina

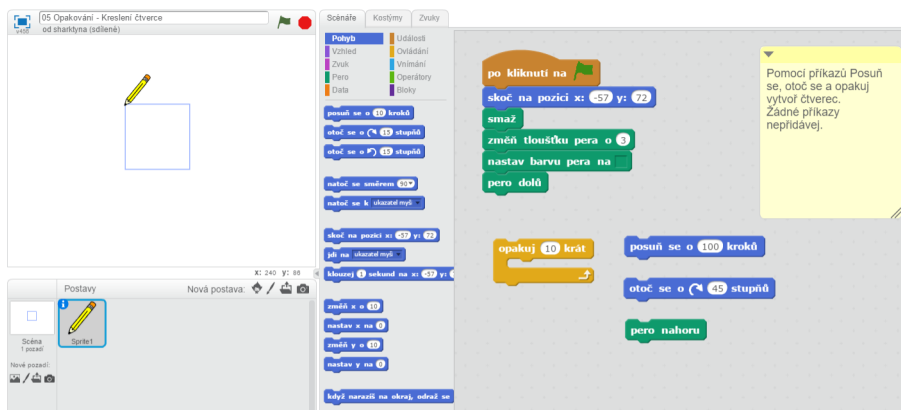
### 5.2.5 05 Opakování - Kreslení čtverce

V této úloze je použita postava tužka, která díky příkazům z nabídky *Pero* za sebou nechává čáru, pokud se pero nastaví „dolu“. Pokud je tužka směrem „nahoru“, žádnou čáru za sebou nenechává. Může se tak nakreslit jakýkoliv tvar. V tomto případě mají žáci za úkol nakreslit čtverec.

#### Doporučení pro učitele

Učitelé by se měli ujistit, že žáci ví, kolik stupňů svírají dvě strany čtverce a kolik vlastně má čtverec stran. Dále je třeba žákům vysvětlit rozdíl mezi „perem nahoru“ a „perem dolu“. Kdy tužka bude kreslit a kdy naopak kreslit nemá. Může se jim to ukázat na tabuli s křídou – když je křída nahoru, kreslit s ní nelze, protože se nedotýká tabule, ale jakmile ji dáme dolů, přiložíme ji na tabuli, kreslit začne.

#### Počáteční stav úlohy



Obrázek 27: Počáteční stav úlohy - 05 Opakování - Kreslení čtverce

## Možné správné řešení



Obrázek 28: Možné správné řešení - 05 Opakování - Kreslení čtverce

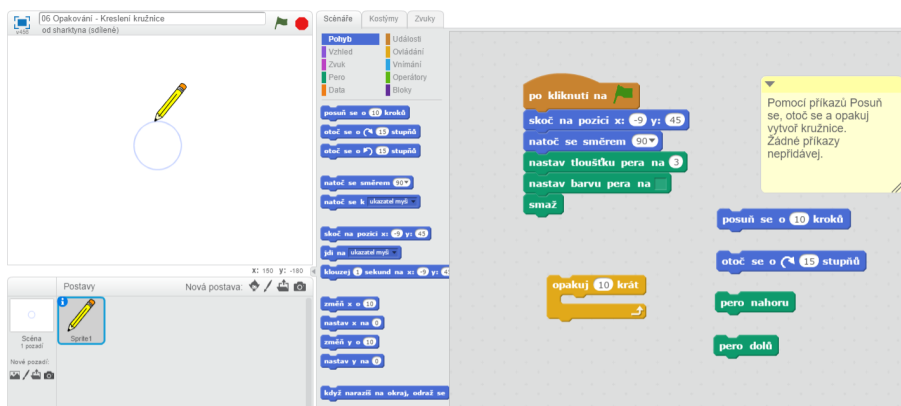
### 5.2.6 06 Opakování - Kreslení kružnice

Tato úloha je podobná té předchozí se čtvercem. Pomocí opakování a příkazů posuň se a otoč se, mají žáci vytvořit kružnici.

#### Doporučení pro učitele

Pro jednoduchost at' žáci nemění počet kroků ani stupňů. Žáci by měli vypočítat, kolikrát se tužka musí posunout a otočit se o  $15^\circ$ , aby se nakreslila celá kružnice – tedy celkem se otočit o  $360^\circ$ .

#### Počáteční stav úlohy



Obrázek 29: Počáteční stav úlohy - 06 Opakování - Kreslení kružnice

## Možné správné řešení



Obrázek 30: Možné správné řešení - 05 Opakování - Kreslení kružnice

### 5.2.7 07 Opakování - Kreslení více čtverců

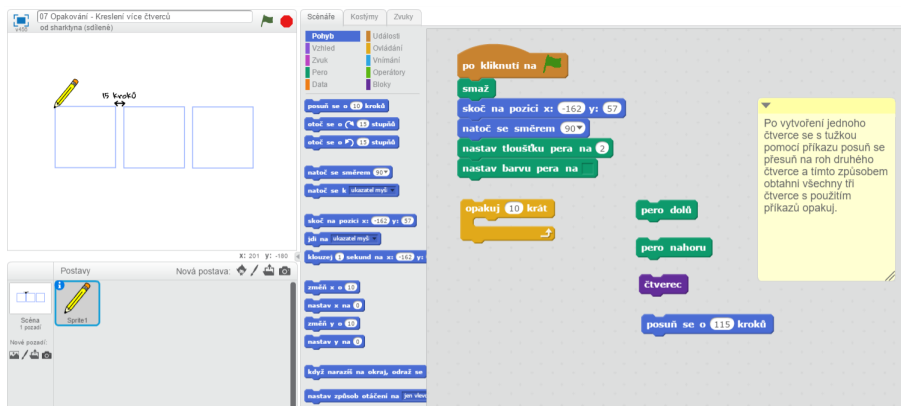
Tato úloha se dá považovat za rozšířenou k úloze čtverec na straně 5.2.5. Žáci již ví, jak se vytvoří čtverec, proto je zbytečné, aby znovu psali tento kód. V této úloze mají vytvořit celou řadu čtverců, ovšem čtverce mají mít mezi sebou mezeru 15 kroků. Musí si tedy uvědomit přesné pozice tužky, kde tužka skončí po jednom tahu atd. Poté přijdou na to o kolik kroků se vlastně musí posunout, aby byla mezera mezi čtverci opravdu 15 kroků.

#### Doporučení pro učitele

Učitele mohou žákům nejprve zobrazit již hotový kód (bez obrázku se čtverci) a třeba zajistit, aby zkusili nakreslit na papír, co program provádí. Mohou je inspirovat tím, že komu se to povede nakreslit správně, dostane jedničku.

Poté ať si program (ten neposkládaný) otevřou ve Scratchi a zkusí ho správně poskládat.

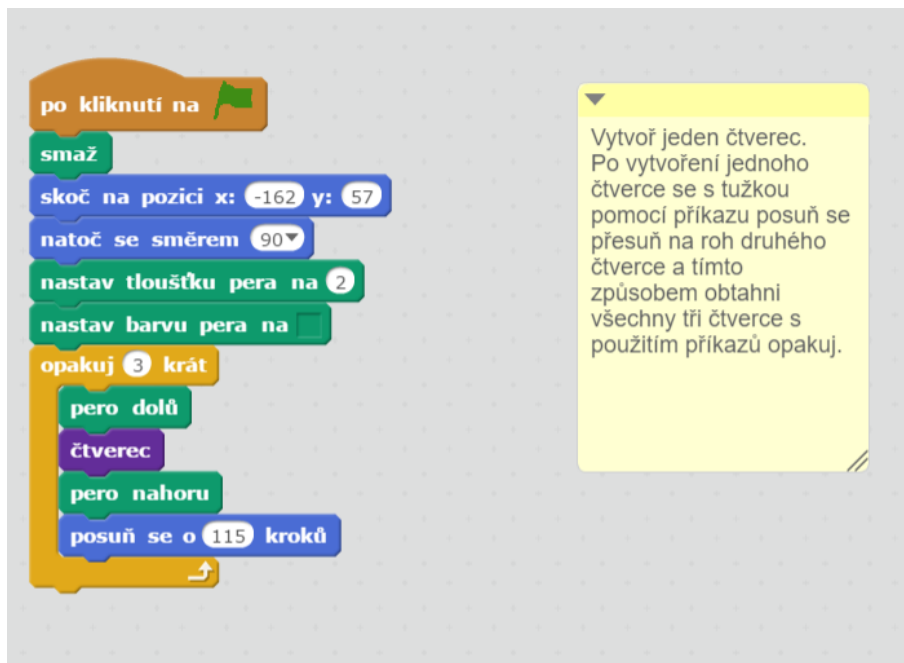
#### Počáteční stav úlohy



Obrázek 31: Počáteční stav úlohy - 07 Opakování - Kreslení více čtverců



## Možné správné řešení



Obrázek 32: Možné správné řešení - 07 Opakování - Kreslení více čtverců

## Nejčastější chyby žáků při řešení této úlohy

Největší problém dělalo žákům přijít na to, o kolik se má tužka posunout, aby další čtverec v řadě byl odsazený o 15 kroků. Zapomínají připočítat ještě stěnu čtverce, o kterou se musí poposunout navíc.

### 5.2.8 08 Opakování Tabulka - Kreslení více čtverců

Tato úloha je podobná předchozí úloze. Žáci již ví, jak se vytvoří čtverec, ale již i řadu čtverců. V této úloze mají vytvořit tabulku ze dvou řad čtverců opět s mezerami 15 kroků mezi sebou. Zde využívají nový příkaz *Skoč na pozici*. Tímto příkazem si žáci mohou uvědomit, co jsou pozice  $x$  a  $y$  a k čemu jsou dobré.

#### Doporučení pro učitele

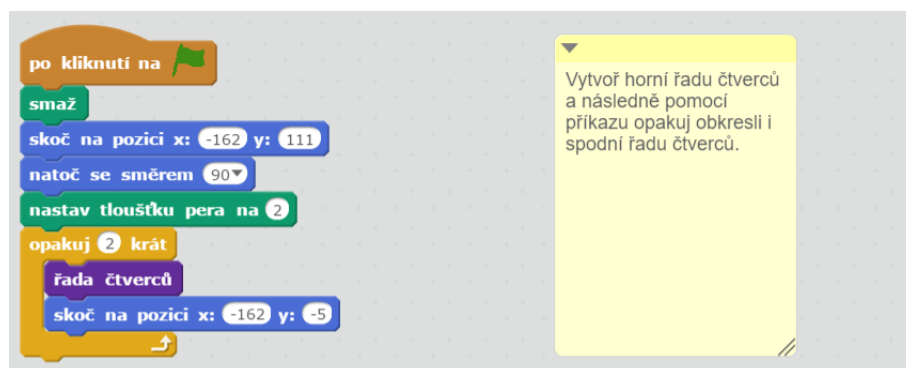
Ti rychlejší si mohou v této úloze pohrát s tloušťkou čar a barvou. Učitelé jim mohou dát zadání, aby každý čtverec měl vždy jinou barvu a tloušťku (vlastnosti čtverce se mohou změnit ve nejspodnější části programu – v části skládání příkazů v bloku čtverec).

#### Počáteční stav úlohy



Obrázek 33: Počáteční stav úlohy - 08 Opakování Tabulka - Kreslení více čtverců

## Možné správné řešení



Obrázek 34: Možné správné řešení - 08 Opakování Tabulka - Kreslení více čtverců

### 5.3 Podmínky

Před plněním první úlohy této sady by již žáci měli umět pracovat s cykly, správně vybrat cyklus s daným počtem opakování nebo s nekonečné opakování. Dále umět rozoznat přiřadit příkazy buď do cyklu nebo mimo cyklus a zároveň používat vnořené cykly. Také by se žáci již měli snažit o to nejúspornější a nejoptimálnější zápis kódu programu.

V této sadě úloh se mají žáci naučit poskládat příkazy do větve podmínky nebo do původní větve programu a vybrat správný výraz podmínky tak, aby program pracoval podle zadání a instrukcí. Zároveň by žáci měli umět kombinovat cykly s podmínkami.

Také touto sadou úloh žáci rozvíjí jak základní klíčové kompetence, tak i kompetence nebo schopnosti jako preciznost, logické, algoritmické a abstraktní myšlení, kreativitu, řešení problémů a generalizaci.

Název úlohy + odkaz na Scratch	Bližší informace k úloze	Typ úlohy – poskládání již použitých příkazů, použití příkazů z nabídky nebo oprava kódu
1) Šváb	Podrobnosti na straně 53	Správné vložení podmínky včetně další větve programu do těla kódu – tedy do těla opakování
2) Myška	Podrobnosti na straně 55	Použití připravených příkazů
3) Motýl a žabka	Podrobnosti na straně 56	Dohledání vhodného příkazu pro skrytí motýla a vložení příkazu do vhodné větve programu
4) Měnící se kočka	Podrobnosti na straně 57	Použití připravených příkazů
5) Vítek jde domů	Podrobnosti na straně 58	Použití připravených příkazů
6) Žralok hledá rybku	Podrobnosti na straně 60	Dohledání vhodného příkazu pro skrytí rybky a vložení příkazu do vhodné větve programu

Tabulka 3: Tabulka s informacemi o úlohách v sadě na podmínky

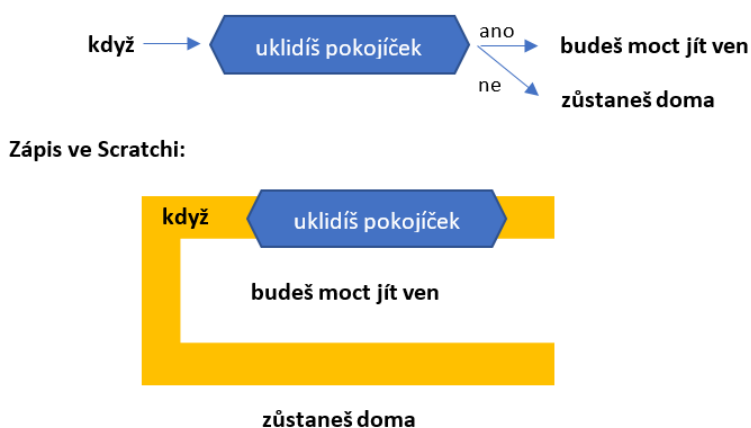
### 5.3.1 01 Podmínka - Šváb

V této úloze se při každém dotknutí kurzorem myši zvýší počet švábů. Žáci mají za úkol přiřadit správně podmínku a celou vedlejší větev správně do těla kódu.

#### Doporučení pro učitele

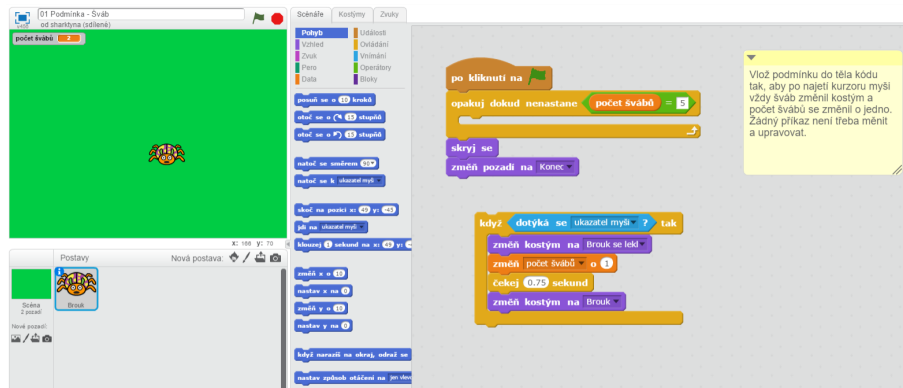
Učitelé mohou zkusit se žáky zopakovat, co znamená v češtině podmínka a at' každý zkusí nějakou podmínku říct. Na nějakém příkladu jim mohou zkusit vysvětlit, co je výraz v těle podmínky a co je další větev programu. Nejlépe nakreslit na tabuli.

**Příklad:** *Když si uklidíš pokojíček, budeš moct jít ven za kamarády.*



Obrázek 35: Nákres podmínky žákům na tabuli

## Počáteční stav úlohy



Obrázek 36: Počáteční stav úlohy - 01 Podmínky - Šváb

## Možné správné řešení

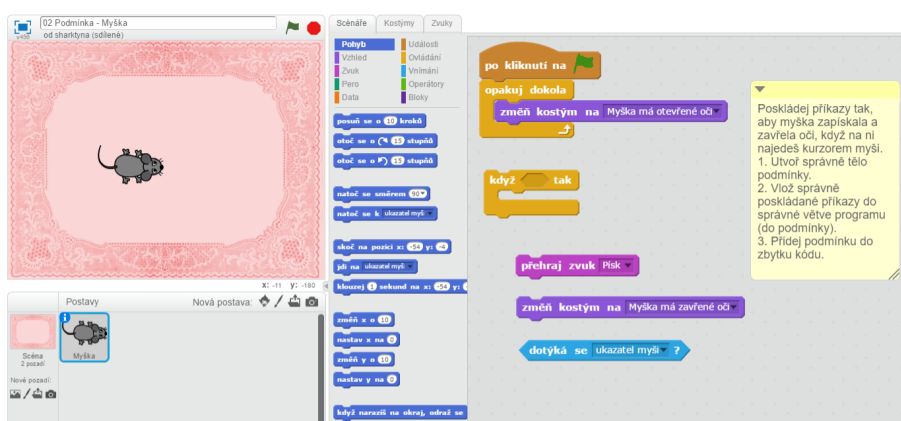


Obrázek 37: Možné správné řešení - 01 Podmínky - Šváb

### 5.3.2 02 Podmínka - Myška

Další kreativní úlohou, je úloha na *obrázku č. 38*, která je zaměřená na správné použití podmínky v opakování. Zde žáci mají za úkol sestavit podmínku tak, aby myška zavřela oči a pískla, když na ni uživatel najede kurzorem. Pomocí této úlohy žáci zjistí, jak se sestavuje podmínka, kdy se používá a jak vlastně funguje.

#### Počáteční stav úlohy



Obrázek 38: Počáteční stav úlohy - 02 Podmínky - Myška

#### Možné správné řešení

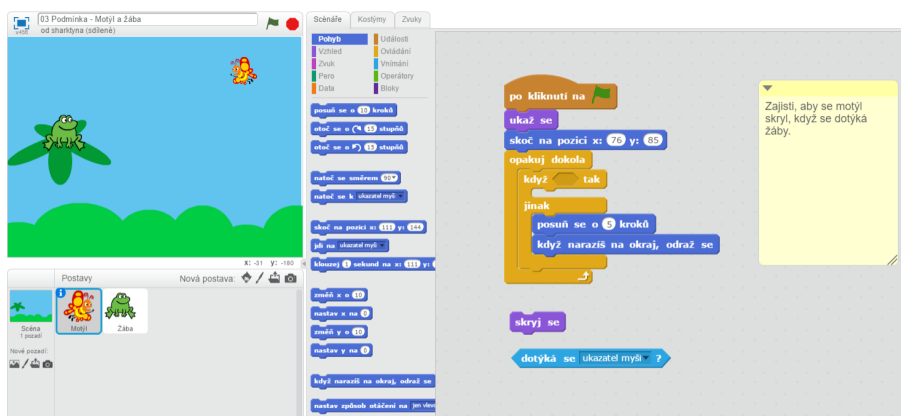


Obrázek 39: Možné správné řešení - 02 Podmínky - Myška

### 5.3.3 03 Podmínka - Motýl a žabka

V této úloze mají žáci za úkol zařadit výraz podmínky a přiřadit do kódu připravený příkaz tak, aby motýl zmizel, když se dotkne žabky. Žáci by se tímto měli procvičit práci s výrazem.

#### Počáteční stav úlohy



Obrázek 40: Počáteční stav úlohy - 03 Podmínky - Motýl a žabka

#### Možné správné řešení



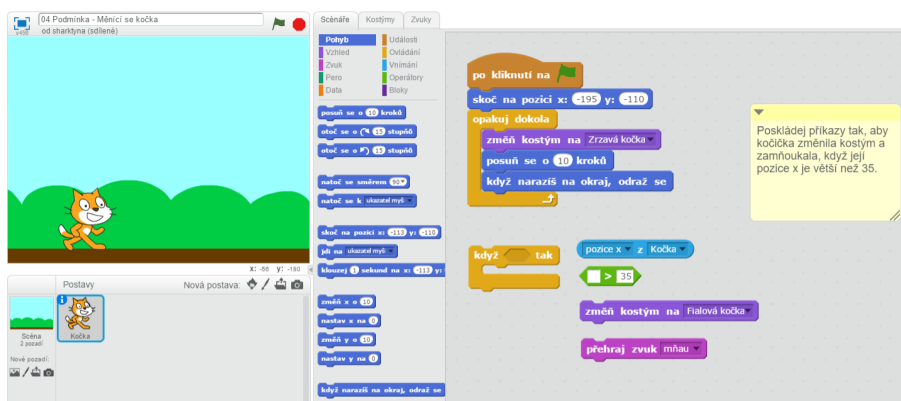
Obrázek 41: Možné správné řešení - 03 Podmínky - Motýl a žabka



### 5.3.4 04 Podmínka - Měnící se kočka

V této úloze se kočička pohybuje ze strany do strany a při pozici  $x$ , která bude větší než 35, se změní kočičky kostým na hnědou kočičku a zamňouká. Žáci mají sestavit výraz podmínky a vložit příkazy do vedlejší větve podmínky a následně celý kód zařadit do programu.

#### Počáteční stav úlohy



Obrázek 42: Počáteční stav úlohy - 04 Podmínky - Měnící se kočka

#### Možné správné řešení

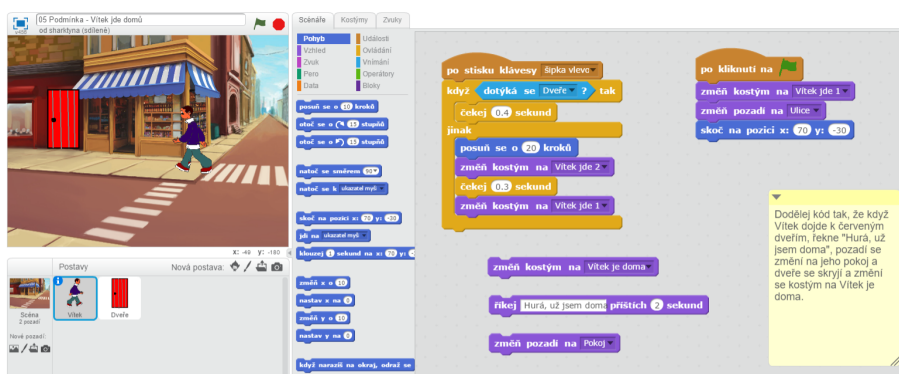


Obrázek 43: Možné správné řešení - 04 Podmínky - Měnící se kočka

### 5.3.5 05 Podmínka - Vítek jde domů

V této úloze mají žáci dodělat kód tak, aby když postava Vítek dojde k červeným dveřím, řekl „Hurá, už jsem doma“, změnil se pozadí na jeho pokoj, Vítek změnil kostým, že je Vítek doma a dveře se skryjí. Žáci zde musí pracovat jak s postavou Vítek, tak i se Dveřmi a seznamuje se s dalším typem podmínky „*když/jinak*“.

#### Počáteční stav úlohy



Obrázek 44: Počáteční stav úlohy - 05 Podmínky - Vítek jde domů

## Možné správné řešení



Obrázek 45: Možné správné řešení - 05 Podmínky - Vítek jde domů

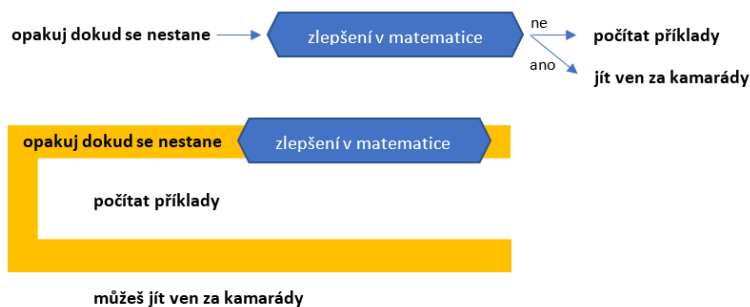
### 5.3.6 06 Podmínka - Žralok hledá rybku

Žáci v této úloze mají najít příkaz, který zajistí, že rybka zmizí, když se dotkne žraloka. Je to v podstatě podobná úloha jako Motýl a žabka (na straně 41), žáci by zde měli využít poznatků právě z předchozí úlohy a zároveň poznat podmínku v cyklu a její význam.

#### Doporučení pro učitele

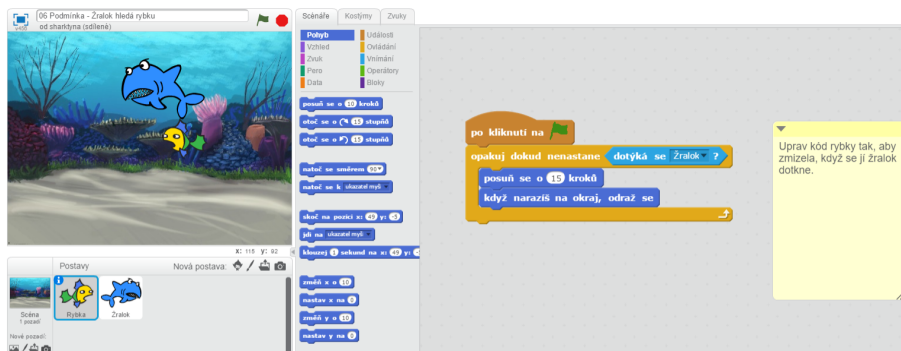
Žákům je třeba vysvětlit, co znamená příkaz „*opakuj dokud se nestane*“. Vysvětlit, jaký je rozdíl mezi tímto příkazem a klasickým cyklem nebo podmínkou a jaké příkazy se dávají do další větve programu.

**Příklad:** *Opakuj počítání příkladů, dokud se nezlepšíš v matematice. Poté můžeš jít ven za kamarády.*



Obrázek 46: Nákres opakování s podmínkou žákům na tabuli

## Počáteční stav úlohy



Obrázek 47: Počáteční stav úlohy - 06 Podmínky - Žralok hledá rybku

## Možné správné řešení



Obrázek 48: Možné správné řešení - 06 Podmínky - Žralok hledá rybku

## 5.4 Manuál pro učitele

Všechny tyto úlohy jsou představené i v manuálu pro učitele, který společně s článkem k tomuto tématu publikuji na několika portálech (více v kapitole Publikování práce).

Výukové prostředí formou studia, tedy složky s připravenými projekty, najdete přímo v prostředí Scratch <sup>3</sup>. Ve studiu jsou úlohy pojmenované tak, aby bylo jasné pořadí úloh, jak jdou za sebou, následně do jaké sady patří a následně už je samotný název dané úlohy. Takto jsou úlohy pojmenované i v manuálu, kde je navíc rovnou přiřazený odkaz přímo na tu danou úlohu v prostředí Scratch. Tudíž učitel si může prohlédnout úlohu v manuálu, nějaká doporučení případně upozornění na nejčastější chyby a problémy žáků, a poté rovnou přejít do samotného prostředí, kde si tu úlohu může projít a vyzkoušet jí.

V dokumentu jsou popsány úlohy v počátečním stavu, ale také náhled možného správného řešení, který by mohl učiteli pomoci tehdy, pokud sám správné řešení nenajde. Jak jsem prala v předchozím odstavci, jsou zde také různá doporučení pro vedení výuky a popsané nejčastější chyby žáků při řešení dané úlohy.

---

<sup>3</sup>Odkaz do studia výukového prostředí Scratch <https://scratch.mit.edu/studios/1768046/>

## 6 Ověření úloh a způsobu této výuky na základní škole

Jak už je napsáno výše, s každým pojmem se žáci seznamují pomocí celé sady úloh. Vždy by se mělo začínat nejjednoduššími úlohami s jen pár příkazy, a především takovými příkazy, které žák už zná, případně jsou velmi snadné na vysvětlení a pochopení. V rámci tohoto způsobu výuky učitel vysvětluje nejnütnější věci (například představuje žákům nový pojem) a žáci jinak pracují aktivně a buď samostatně nebo ve dvojicích.

Mezi tradičně častější typ úloh, kdy žáci sami vytvářejí kód, jsou zde zařazeny i úlohy, kde žáci musí přijít na chybu v kódu, případně odhadnout, vyslovit hypotézu, nebo nakreslit, co daný kód dělá.

Díky rozmanitosti a kreativnímu prostředí, kterým Scratch bezpochyby je, se fantazii ve tvorbě úloh meze nekladou. Proto by neměl být problém pro zkušeného učitele vymýšlet další a další úlohy, které by žáky mohly bavit a naučilo se to, co jsme si dali jako cíl pro danou hodinu nebo několik hodin.

### 6.1 Ověřování úloh formou pozorování ve výuce

V průběhu výuky programování jsem ověřovala pozorovací metodou ohlasy žáků na vytvořené úlohy a zároveň, zda jsou právně vytvořené, tedy jestli žáci rozumí zadání, mnou vytvořeným příkazům nebo jestli jdou za sebou ve správném pořadí podle obtížnosti. Dále jsem také zjišťovala, zda u žáků pomocí těchto úloh dochází k prohlubování znalostí programovacích pojmů a představ a zda žáky vůbec tyto úlohy baví a jak na ně reagují.

#### 6.1.1 Výsledky pozorování ve výuce

Na základně vyhodnocení pozorování jsem zjistila, že žáky většina úloh opravdu baví. Často si žáci po splnění úlohy ještě následně hrály s dalšími příkazy a úlohu si upravovali podle sebe (například dávali kočičce ještě příkazy aby něco řekla nebo zamňoukala, nebo peru nastavovali jinou barvu, tloušťku i směr). Pokud jsem viděla, že žáci mají problém s nějakou úlohou, že přesně nerozumí mnou vytvořenému příkazu nebo úlohu je pro ně ještě příliš těžké vyřešit, napsala jsem si na papír poznámku, o co konkrétně se jedná.

Po každé vyučovací hodině jsem měla chvíli čas na to, abych případně na opravu časově nenáročné nedostatky doladila, abych hned opravenou úlohu mohla vyzkoušet na další hodině a ověřit tím, zda je již úloha v pořádku. Pokud úloha potřebovala větší opravu, napsala jsem si k tomu po výuce poznámku a opravila jsem ji až doma a vyzkoušela jsem ji další den na polední třídě, se kterou jsem sady úloh zkoušela.

### 6.1.2 Nejčastější opravy úloh

Aby úlohy byly nakonec nejoptimálněji poskládané za sebou a aby se s nimi žákům pracovalo co nejlépe a nejsnáze a zároveň se jim všechny spojitosti dávaly dohromady, musela jsem u některých úloh pozměnit pořadí, a to nejčastěji u podmínek, kde se často pracuje s dalšími neznámými pojmy, ale třeba již také s opakováním. Dále jsem musela pozměnit názvy postav, aby měly český název a zároveň například u kostýmů změnit názvy, aby žáci věděli, jaký kostým mají vlastně přiřadit nebo změnit (například 5.1.3 na straně 27). Také jsem například u úlohy 5.1.2 na straně 25 změnil příkaz z bloku „*na start*“. Původní příkaz byl „*zpět na start*“. To ovšem žáky mátl a dávali tento příkaz na konec místo hned na začátek programu.

## 6.2 Ověřování úloh formou otevřeného dotazníku

Na konci působení na této škole jsem realizovala dotazník formou otevřených otázek. Požádala jsem žáky, aby mi zcela upřímně napsali, jestli je Scratch bavil, případně která úloha je zaujala nejvíc a jestli by se ve Scratchi chtěli učit dál nebo se k němu ještě někdy vrátit. Tento dotazník byl anonymní, tudíž jsem nechala zcela na žácích, zda se pod svůj názor podepíší, či nikoliv.

Tímto dotazníkem jsem chtěla zjistit, jestli je programování bavilo, zaujal je typ úloh, které jsem pro ně připravila, a hlavně jestli tímto způsobem výuky žáci pochopí princip funkčnosti pojmů, na které byly úlohy sestaveny.

### 6.2.1 Kladené otázky v dotazníkovém šetření:

1. Bavily tě hodiny programování?
2. Vzpomeneš si na nějaké úlohy, které jsme dělali?



3. Vypiš všechny, na které si vzpomeneš.
4. Jaké úlohy tě nejvíce bavily a proč?
5. Jaké úlohy tě naopak nebavily a proč?
6. Chtěl(a) by ses ke Scratchi někdy vrátit?

### 6.2.2 Zhodnocení zájmu dětí o programování na základě dotazníkového šetření a pozorováním ve výuce

Podle reakcí žáků, jejich zaujetí a také podle jejich výpovědí v dotazníku jsem zjistila, že nejvíc žáky zaujmou úlohy, kde jsou nějaké zvuky, nebo jsou provedeny vtipnou formou. Jedna z výpovědí žáků: *„Pracování ve Scratchi se mi moc líbilo. Moc mi nešlo sestavit čtverce, ale zase mě bavilo vše, co bylo se zvukem a pohybem koček a myši.“*

Ovšem má zkušenost po dokončení těchto úloh byla, že je často těžké udržet pozornost a kázeň žáků, aby se dalo pokračovat na dalších úlohách. Nicméně je pro mě a věřím, že i další učitele důležité, že žáky tyto „aktivní“ úlohy, kde se používá zvuk a pohyb, více motivují a díky tomu je Scratch baví.

Žáci zkoušeli i úlohy s cykly na daný počet opakování. Použila jsem úlohy s obtahováním čtverců (*obrázek č. 31*), které byly tohoto typu celkem tři. Nejprve žáci měli obkreslit jeden čtverec, v další úloze zase tři čtverce vedle sebe a poslední úloha byla se třemi čtverci ve dvou řadách, což už pro některé byla těžší úloha. Tyto úlohy sloužily převážně k procvičení a pochopení používání cyklů.

Z pozorování žáků při výuce, ale i poté z jejich odpovědí v dotazníku jsem zjistila, že některé žáky bavily úlohy, které byly spíše o pohybu a zvuku, ale naopak zase byli i žáci, kteří se předvedli hlavně v úlohách se čtverci a bavily je více. Dokonce i v těchto úlohách žáci často experimentovali a měnili barvu a tloušťku čar, poté i směr čar atd. Jedna z výpovědí žáků:

*„Nejvíc mě bavilo to obtahování čtverců. Měl jsem to vždy hotové za chvíli a pak jsem ty čáry různě upravoval.“*

Některé úlohy bavily žáky více a některé zase méně. Jak se dalo očekávat, některé žáky Scratch nezaujal, nebavil je a nechtěl se v něm už dále učit. Jedna z výpovědí žáků:

*„Moje nejoblíbenější úloha byla s myškou. Myška měla pípat, když jsem na ni najela počítačovou myší. Měla jsem tu úlohu i první vypracovanou. Upřímně moc mě Scratch nebavil, ale všechny úlohy jsem vypracovala.“*

Ačkoliv jsem se snažila všem žákům věnovat tak, aby všichni pochopili práci se Scratchem a co v něm vlastně mají dělat, dostala jsem i zpětnou vazbu, že Scratch žáka nebavil i třeba z důvodu, že nepochopil nějaké úlohy. Proto je při výuce programování opravdu třeba zřetelně vysvětlit práci ve Scratchi a následně správně a srozumitelně zadávat úlohy. Nezájem žáka mohl být způsobený i tím, že tempo, kterým se programováním vyučovalo, bylo na něj příliš rychlé a žák nestihl vstřebat všechny informace tak, aby to pochopil, a tudíž se přestal snažit. Také jsem ve třídě zaznamenala žáky, kteří každou chvíli vyrušovali, nedělali to, co měli a snažili se být za každou cenu vtipní a ve finále úlohy hotové neměli, protože nedávali pozor. U nich byla samozřejmě odpověď, že je Scratch nebavil. Jedna z výpovědí žáků:

*„Líbila se mi úloha s vrabcem. Dělal mi problém několik úloh, protože jsem je nepochopil. Scratch mě celkově moc nebavil.“*

Naštěstí takových žáků bylo méně. Většina žáků (přesně 75 %) byla s výukou ve Scratchi potěšena, přišel jim zajímavý a byla to pro ně zase příjemná změna ve výuce informatiky. Jedna z výpovědí žáků:

*„Celkově mě Scratch bavil a určitě si nějaké úlohy zkusím doma. Nejvíc se mi líbila úloha, kdy jsme dělali kružnici nebo čtverce. Také mě bavily úlohy, když jsme tam mohli přidávat nějaký zvuk. Není žádná úloha, která by mě vyložené nebavila, protože každá byla něčím zajímavá.“*

Výpovědi žáků se poměrně lišily i v zálibě jednotlivých úloh. Některé žáky bavily převážně úlohy, kde se pracovalo se zvukem nebo pohybem objektů a zase naopak je nebavily úlohy, kde měli obtahovat čtverce (obrázek č. 31). Jedna z výpovědí žáků:

*„Pracování ve Scratchi se mi moc líbilo. Moc mi nešlo sestavit čtverce, ale zase mě bavilo vše, co bylo se zvukem a pohybem koček a myší. Neříkám, že mi to zrovna šlo, ale bavilo mě to. Možná, že se k tomu jednou vrátím.“*

Celkem jsem učila 79 žáků, z nichž 72 jich bylo poslední týden, kdy jsem rozdávala dotazník. Tedy dostala jsem 71 odpovědí, které mi pomohly k ověření, zda děti programování může bavit a zda je oslovily úlohy, které jsem vytvořila. Po prozkoumání a roztrídění všech odpovědí jsem zjistila, že 54 žákům (75 %) se hodiny programování líbily, většina úloh je bavila a rádi by se učili programovat dál. Zbýlých 17 žáků odpovědělo, že je programování neoslovilo například z důvodu obtížnosti.

### 6.3 Shrnutí

Mohu tedy tvrdit, že děti převážně programování baví, pokud se jim podá zábavnou a srozumitelnou formou v kreativním a jednodušším prostředí. Právě i díky dotazníku jsem zjistila, že žáci si Scratch pustí s radostí i doma a berou ho spíše jako hru než povinnost, kterou se musí učit, protože to po nich učitel vyžaduje. Jedna z výpovědí žáků: „*Celkově mě Scratch bavil, nějaké úlohy si zkouším i doma a hraju si v něm.*“

## 7 Publikování práce

K této práci jsem vytvořila článek, který popisuje tuto problematiku a zároveň představuje mou práci. Aby se toto téma dostalo do širšího podvědomí společnosti, tedy především k co nejvíce učitelům, tento článek a manuál pro učitele jsem zveřejnila na několika portálech.

- Prvním portálem je *DUMY*, který využívá velké množství učitelů. Je to vhodný portál na publikování výukových materiálů. Soubory lze stáhnout [zde](#)<sup>4</sup>.
- Druhým portálem je e-sborník konference *Počítač ve škole* (rok 2017). Tato konference je pořádána již několikátý rok a je též mezi učiteli velmi známá a oblíbená. [Zde](#)<sup>5</sup> je možnost stažení článku.
- Třetím portálem je *Metodický portál RVP*. Zde článek s manuálem čeká na schválení, tudíž zatím (v době dopsání diplomové práce) není zařazen mezi články.

Dále článek na toto téma je již zaslán do *Školních novin*, společnosti *AV Media* a časopisu *e-Mole*, což jsou média, která též publikují školní témata. Tyto časopisy jsou dodávány přímo do škol k dispozici učitelům.

---

<sup>4</sup><http://dumy.cz/material/165388-manual-pro-ucitele-v-ramci-vyuky-programovani-ve-scratch-zamerena-na-vytvareni-pojmu>

<sup>5</sup><http://www.pocitacveskole.cz/system/files/soubory/sbornik/2017/Holeckova.pdf>

## 8 Závěr

Na závěr bych tedy chtěla shrnout, že programování za základní školy určitě oprávněně patří a děti dokonce i baví. Prostředí Scratch se je vhodným dětským programovacím prostředím, ve kterém se dětem dobře pracuje a mohou zde rozvíjet nejen svoji kreativnost, ale i infromatické, abstraktní a logické myšlení. Programování děti může bavit i díky tomu, že se může učit jinými způsoby, například ve skupinkách, ve dvojicích, formou projektové výuky apod.

Mnou dotázaní učitelé tuto formu výuky programování hodnotí velice kladně a oceňují, že bude k dispozici jakási průprava ve formě sady úloh a tím budou vědět, jakým stylem začít vyučovat programování, aby ho děti pochopily více do hloubky a zároveň aby je i bavilo.

Touto prací jsem se pokusila vytvořit zcela nový přístup k výuce programování v tom, že žáci si formou aktivního přístupu postupně si vytvářejí pojmy a představy, které následně v průběhu dalších úloh prohlubují, ale zároveň prohlubují své schopnosti a kompetence v rámci těchto úloh. Na základě porozumění a pochopení těmto pojům by měli být žáci schopni sami vytvářet a zpracovávat již náročnější úlohy a programy, kde se již vyskytuje více pojmů najednou.

Vytvořila jsem tedy tři sady úloh na skládání příkazů, opakování a podmínky. Každá tato sada vede žáky vždy k pochopení jednoho hlavního pojmu (např. podmínky) a zároveň se žáci seznamují a pracují i s jinými pojmy (výraz v podmínce). Tyto sady úloh byly vyzkoušeny přímo ve výuce na základní škole a v průběhu nebo následně po vyzkoušení úloh jsem je, pokud bylo třeba, upravila, aby co nejvíce vyhovovaly.

Pro učitele jsem k těmto úlohám vytvořila manuál, který je bude provázet všemi sadami, úlohu po úloze a zároveň je bude upozorňovat na možná úskalí či rady.

Také díky inovaci ICT kurikula bude v hodinách informatiky více prostoru pro výuku programování, algoritmizaci apod. Proto tedy doufám, že touto prací přispějí ke zlepšení situace výuky programování na základních školách a že učitelé se již nebudou tolik bát přejít na toto téma a třeba tento typ úloh dál rozvíjet.

## Literatura

- [1] KREJSA, Jan. Výuka základů programování v prostředí Scratch. České Budějovice, 2013. Dostupné také z: <http://theses.cz/id/b5f11x/>.
- [2] STUHLÍKOVÁ, Iva, Tomáš JANÍK, Zdeněk BENEŠ, et al. Oborové didaktiky: vývoj, stav, perspektivy. Brno: Masarykova univerzita, 2015. Syntézy výzkumu vzdělávání. ISBN 978-80-210-7769-0.
- [3] RŮŽIČKOVÁ, Daniela. Přednáška na konferenci Počítač ve škole 2017. Nové Město na Moravě, 11. 4. 2017.
- [4] DEMŠAR, Janez. Osobní sdělení. Ljubljana, 7. 4. 2016.
- [5] SCRATCH - Imagine, Program, Share [online]. Cambridge, USA: Lifelong Kindergarten Group, 2005 [cit. 2015-11-05]. Dostupné z: <https://scratch.mit.edu/>
- [6] VANÍČEK, J. Želví geometrie pro 21. století aneb Scratch ve výuce algoritmizace. In Hašek, R. (ed.): Sborník příspěvků 7. konference Užití počítačů ve výuce matematiky. České Budějovice: Jihočeská univerzita v Českých Budějovicích, 2015, s. 259-267. ISBN 978-80-7394-549-7. Dostupné z [http://home.pf.jcu.cz/~upvvm/2015/sbornik/Sbornik\\_UPVM\\_2015.pdf](http://home.pf.jcu.cz/~upvvm/2015/sbornik/Sbornik_UPVM_2015.pdf)
- [7] PECINOVSKÝ, Rudolf. Baltík: učebnice programování nejen pro děti. S.l.: SGP Systems, 2000.
- [8] BLAHO, Andrej a Ivan KALAŠ. Imagine Logo: učebnice programování pro děti. Vyd. 1. Brno: Computer Press, 2006, 48 s. Česká škola (Computer Press). ISBN 80-251-1015-x.
- [9] SALANCI, Ľubomír: EasyLogo-discovering basic programming concepts in a constructive manner. In: Constructionist approaches to creative learning, thinking and education: Lessons for the 21st century. Bratislava : FMFI UK, 2010. ISBN 978-80-89186-66-2, ISBN 978-80-89186-65-5. Dostupné z: <http://www.salanci.sk/EasyLogo/Paper.pdf>
- [10] HEJNÝ, Milan. Teória vyučovania matematiky 2. Bratislava: SPN, 1987. ISBN 80-08-00014-7

- 
- [11] VANÍČEK, J. Počítačové kognitivní technologie ve výuce geometrie. Praha: PedF UK, 2009, 212 s. ISBN 978-80-7290-394-8
- [12] CHALOUPKOVÁ, Kateřina. Alternativní výukové metody v hodinách cizích jazyků. Brno, 2010. Dostupné také z: [https://is.muni.cz/th/350874/pedf\\_m/Diplomova\\_prace.pdf](https://is.muni.cz/th/350874/pedf_m/Diplomova_prace.pdf). Bakalářská práce. Masarykova univerzita; Pedagogická fakulta; Katedra pedagogiky. Vedoucí práce Mgr. Lucie Pištěková.
- [13] ZORMANOVÁ, Lucie. Obecná didaktika: pro studium a praxi. Vydání 1. Praha: Grada, 2014, 239 stran. ISBN 978-80-247-4590-9.
- [14] ŠUMBEROVÁ, Klára. Alternativní výukové metody v hodinách cizích jazyků. Brno, 2015. Dostupné také z: [https://is.muni.cz/th/350874/pedf\\_m/Diplomova\\_prace.pdf](https://is.muni.cz/th/350874/pedf_m/Diplomova_prace.pdf). Diplomová práce. Masarykova univerzita; Pedagogická fakulta; Katedra fyziky, chemie a odborného vzdělávání. Vedoucí práce Mgr. Irena Plucková, Ph.D.
- [15] Programování pro děti: naučte se programovat při tvorbě skvělých her. Brno: Computer Press, 2013. ISBN 978-80-2513-809-0.
- [16] BLAHO, A., a L. SALANCI. Informatics in Primary School: Principles and Experience. In: I. KALAŠ a R. T. MITTERMEID, eds. ISSEP 2011. Springer Berlin Heidelberg, 2011, s. 129–142.
- [17] CODE.ORG. Hodina kódu. Frozen. Kurs programování [online]. Code.org, 2014 [cit. 2015-01-23]. Dostupné z: <http://studio.code.org/s/frozen>
- [18] MIKOLAJOVÁ, K. Vytváranie hier v prostredí Scratch – cesta k programovaniu na II. stupni ZŠ. In: Andrejková, G., ed. DidInfo 2011. Banská Bystrica: Univerzita Mateja Bela, 2011, s. 31.
- [19] METZ, Cade. Google Blockly Lets You Hack With No Keyboard. WIRED [online]. 6.12.2012 [cit. 2015-11-05]. Dostupné z: <http://www.wired.com/2012/06/google-blockly/>

- [20] KALAŠ, I. BLAHO, A. ComeniusLOGO: tvorivá informatika. Bratislava : Slovenský ústav technickej normalizácie, 1998. ISBN 80-967999-0-8. 1.časť.
- [21] RESNICK, M. a kol. Scratch: Programming for all. Communications of the ACM [online]. 2009, 52(11) [cit. 2015-01-15]. Dostupné z: <http://web.media.mit.edu/simmres/papers/Scratch-CACM-final.pdf>
- [22] BAGGE, Phil. Code-IT Primary Programming: How to Teach Primary Programming Using Scratch. 1. Great Britain: The University of Buckingham Press, 2015. ISBN 9781908684578.



## Seznam obrázků

1	Prostředí Scratch . . . . .	15
2	Porovnání programů Baltík, Imagine Logo a Scratch . . . . .	17
3	Počáteční stav úlohy - 01 Skládání příkazů - Kočička . . . . .	23
4	Možné správné řešení - 01 Skládání příkazů - Kočička . . . . .	24
5	Počáteční stav úlohy - 02 Skládání příkazů - Chození . . . . .	25
6	Možné správné řešení - 02 Skládání příkazů - Chození . . . . .	26
7	Počáteční stav úlohy - 03 Skládání příkazů - Pozdrav . . . . .	27
8	Možné správné řešení - 03 Skládání příkazů - Pozdrav . . . . .	27
9	Počáteční stav úlohy - 04 Skládání příkazů - Fáze měsíce . . . . .	28
10	Možné správné řešení - 04 Skládání příkazů - Fáze měsíce . . . . .	29
11	Doplňková úloha - 04 Skládání příkazů - Fáze měsíce . . . . .	30
12	Počáteční stav úlohy - 05 Skládání příkazů - Růst květiny . . . . .	31
13	Možné správné řešení - 05 Skládání příkazů - Růst květiny . . . . .	31
14	Počáteční stav úlohy - 06 Skládání příkazů - Říkanka . . . . .	32
15	Možné správné řešení - 06 Skládání příkazů - Říkanka . . . . .	32
16	Počáteční stav úlohy - 07 Skládání příkazů - Tanec . . . . .	33
17	Možné správné řešení - 07 Skládání příkazů - Tanec . . . . .	34
18	Počáteční stav úlohy - 01 Opakování - Chození rovně . . . . .	37
19	Možné správné řešení - 01 Opakování - Chození rovně . . . . .	38
20	Počáteční stav úlohy - 02 Opakování - Chození rovně a do strany . . . . .	39
21	Možné správné řešení - 02 Opakování - Chození rovně a do strany . . . . .	40
22	Počáteční stav úlohy - 03 Opakování - Fáze měsíce . . . . .	41
23	Možné správné řešení - 03 Opakování - Fáze měsíce . . . . .	41
24	Náhled úlohy Květina . . . . .	42
25	Počáteční stav úlohy - 04 Opakování - Květina . . . . .	43
26	Možné správné řešení - 04 Opakování - Květina . . . . .	43
27	Počáteční stav úlohy - 05 Opakování - Kreslení čtverce . . . . .	44
28	Možné správné řešení - 05 Opakování - Kreslení čtverce . . . . .	45
29	Počáteční stav úlohy - 06 Opakování - Kreslení kružnice . . . . .	46
30	Možné správné řešení - 05 Opakování - Kreslení kružnice . . . . .	47
31	Počáteční stav úlohy - 07 Opakování - Kreslení více čtverců . . . . .	48
32	Možné správné řešení - 07 Opakování - Kreslení více čtverců . . . . .	49

---

33	Počáteční stav úlohy - 08 Opakování Tabulka - Kreslení více čtverců . . . . .	50
34	Možné správné řešení - 08 Opakování Tabulka - Kreslení více čtverců . . . . .	51
35	Nákres podmínky žákům na tabuli . . . . .	53
36	Počáteční stav úlohy - 01 Podmínky - Šváb . . . . .	54
37	Možné správné řešení - 01 Podmínky - Šváb . . . . .	54
38	Počáteční stav úlohy - 02 Podmínky - Myška . . . . .	55
39	Možné správné řešení - 02 Podmínky - Myška . . . . .	55
40	Počáteční stav úlohy - 03 Podmínky - Motýl a žabka . . . . .	56
41	Možné správné řešení - 03 Podmínky - Motýl a žabka . . . . .	56
42	Počáteční stav úlohy - 04 Podmínky - Měnící se kočka . . . . .	57
43	Možné správné řešení - 04 Podmínky - Měnící se kočka . . . . .	57
44	Počáteční stav úlohy - 05 Podmínky - Vítek jde domů . . . . .	58
45	Možné správné řešení - 05 Podmínky - Vítek jde domů . . . . .	59
46	Nákres opakování s podmínkou žákům na tabuli . . . . .	60
47	Počáteční stav úlohy - 06 Podmínky - Žralok hledá rybku . . . . .	61
48	Možné správné řešení - 06 Podmínky - Žralok hledá rybku . . . . .	61

## Seznam tabulek

1	Tabulka s informacemi o úlohách v sadě na skládání pojmů . . .	22
2	Tabulka s informacemi o úlohách v sadě na opakování . . . . .	36
3	Tabulka s informacemi o úlohách v sadě na podmínky . . . . .	52

## Přílohy

1. CD - součástí práce je CD, na kterém je přiložený manuál pod názvem *manual\_pro\_ucitele.pdf* a článek k tomuto tématu pod názvem *clanek\_Holeckova.pdf*. Dále je na disku přiložená tato diplomová práce pod názvem *Holeckova\_diplomovaPrace.pdf*.