

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Využití robota Sphero SPRK+ při výuce algoritmizace a
programování



2022

Vedoucí práce: Mgr. Tomáš Kühr,
Ph.D.

Martin Kostka

Studijní obor: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor: Martin Kostka
Název práce: Využití robota Sphero SPRK+ při výuce algoritmizace a programování
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2022
Studijní obor: Aplikovaná informatika, prezenční forma
Vedoucí práce: Mgr. Tomáš Kühn, Ph.D.
Počet stran: 30
Přílohy: 1 CD/DVD
Jazyk práce: český

Bibliographic info

Author: Martin Kostka
Title: Sphero SPRK+ Robots in the Teaching of Algorithms and Programming
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2022
Study field: Applied Computer Science, full-time form
Supervisor: Mgr. Tomáš Kühn, Ph.D.
Page count: 30
Supplements: 1 CD/DVD
Thesis language: Czech

Anotace

Tato práce se zabývá způsoby výuky informatiky z pohledu rámcových a školních vzdělávacích programů pro 2. stupeň základních škol. V teoretické části se dále zabývá principy digitálního a inforatického myšlení zavedených do nových revizí rámcových vzdělávacích programů. V praktické části je pak představen výukový plán společně s robotem Sphero SPRK+, které by mohly být v budoucnu aplikovány do výuky tohoto předmětu se zaměřením na algoritmizaci a programování. K práci je také přiložena sbírka úloh zaměřená na procvičení základních znalostí algoritmizace a programování na platformách Scratch a Sphero Edu.

Synopsis

This thesis examines ways of teaching computer science from the perspective of Framework Educational Programme for the secondary schools. In the theoretical part, thesis deals with the principles of digital and computational thinking introduced in the new revisions of the Framework Educational Programme. In the practical part, the teaching plan is presented together with a robot Sphero SPRK+, which could be used for future teaching of this subject with a focus on algorithmization and programming. This part also includes a collection of exercises aimed to practicing the basic knowledge of algorithmization and programming on the Scratch and Sphero Edu platforms.

Klíčová slova: rámcový vzdělávací program; digitální gramotnost; inforatické myšlení; algoritmus; Sphero SPRK+; STEM; výukový plán

Keywords: Framework Educational Programme; digital literacy; computational thinking; algorithm; Sphero SPRK+; STEM; teaching plan

Děkuji Mgr. Tomáši Kührovi, Ph.D. za vedení práce a cenné rady.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Úvod	7
2	Vzdělávací systém v ČR	7
2.1	Rámcové a školní vzdělávací programy	7
2.2	Rámcový vzdělávací program v informatice	8
3	Digitální gramotnost a infromatické myšlení	9
3.1	Digitální gramotnost	10
3.2	Infromatické myšlení	10
3.3	Začlenění předmětu ICT do výuky	11
4	Algoritmizace	13
4.1	Definice algoritmu	13
4.2	Podmínky a zápisy algoritmu	14
4.3	Typy algoritmů	14
5	Robot Sphero SPRK+	15
5.1	Ovládání a programování	16
5.2	Platforma Scratch	16
5.3	STEM	17
5.4	Využití SPRK+ při výuce	18
5.5	Nedostatky SPRK+	18
5.5.1	Nepřesnost způsobená vnějšími vlivy	19
5.5.2	Vliv setrvačnosti	19
6	Výukový plán	20
6.1	Cíle výukového plánu	20
6.2	Návrh výukového plánu	21
	Závěr	25
	Conclusions	26
A	Sbírka úloh pro Sphero SPRK+	27
B	Sbírka úloh pro Scratch	27
C	Obsah přiloženého CD/DVD	27
	Seznam zkratk	28
	Literatura	29

Seznam obrázků

1	Systém RVP a ŠVP	8
2	Postoj ICT k ostatním předmětům	11
3	Koncept rozvoje digitální gramotnosti a inforatického myšlení	12
4	Robot Sphero SPRK+	15
5	Blokový kód na platformě Sphero Edu	17
6	Princip STEM vzdělávání	18
7	Vykreslení	19

Seznam algoritmů

1	Euklidův algoritmus	13
---	-------------------------------	----

Seznam tabulek

1	Výukový plán.	21
---	-----------------------	----

1 Úvod

Výuka informatických předmětů v posledních letech nabírá změn, i přes to na některých základních školách výuka dodnes spočívá hlavně v práci s textovými a tabulkovými editory. Umět pracovat s těmito editory je dnes nezbytnou dovedností. Není však pouze jedinou, kterou by si žáci a studenti měli ze škol odnést. Informatika je rozsáhlým oborem, pro které jsou současné vzdělávací osnovy nedostačující, a proto již dochází k revizím studijního plánu, které by měly do budoucna změnit způsob výuky informatiky.

Tato práce shrnuje důležité poznatky o systému vytváření nových vzdělávacích plánů, o způsobu nové výuky informatiky včetně jejich výstupů a vysvětlení. V další části je pak představen výukový plán s robotem Sphero SPRK+ používaný ve výuce.

Jedna z velkých změn, která by měla přijít, je zaměření vzdělávání na rozvoj digitálního a informatického myšlení. Studie pod názvem [ICILS](#) z roku 2018, která se zabývala kvalitou digitální gramotnosti žáků napříč několika zeměmi, přišla se závěrem, že pouhá 2 % žáků dokázala vyhledat a vyhodnotit požadované informace v digitálním prostředí. Druhá skupina 18 % žáků naopak nedosáhla ani na nejnižší stupeň digitálních dovedností, i přesto, že většina účastníků má na školách přístup k těmto digitálním technologiím a pravidelně je využívá.[1]

Mají-li se žáci pohybovat v digitálním světě, je potřeba, aby se s rozvojem digitálních a informatických dovedností začalo už na základních školách. Nejprve je však potřeba popsat, jak ve školství fungují rámcové vzdělávací programy a jaký je postoj informatiky vůči nim.

2 Vzdělávací systém v ČR

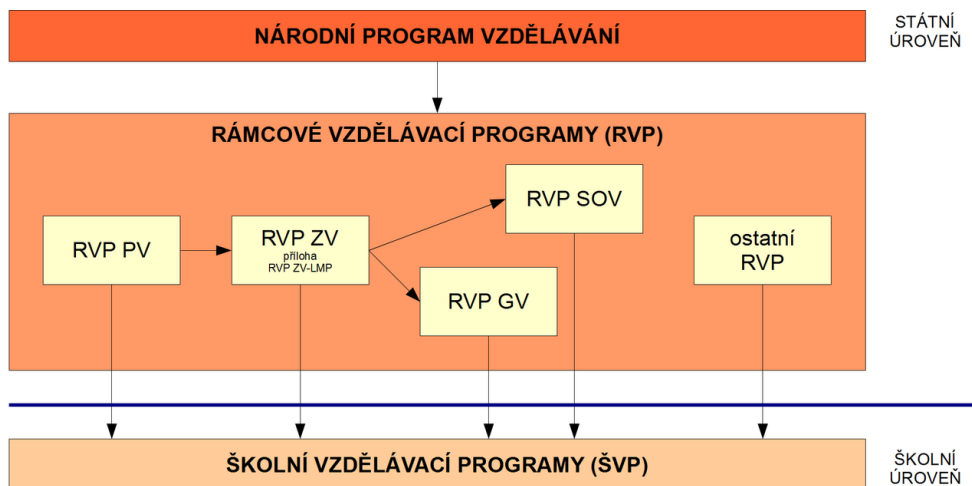
Vzdělávání na českých školách můžeme rozdělit do 3 kategorií: primární, sekundární a terciární. Primárním vzděláváním rozumíme státem povinné vzdělávání na základních školách pro děti ve věku od 6 do 15 let [2], které se dále dělí na první a druhý stupeň. Do sekundárního vzdělávání se řadí gymnázia, střední odborné školy a střední umělecké školy. Terciární vzdělávání je nejvyšší kategorií, do které řadíme vysoké školy nebo vyšší odborné školy. [3]

Žáci se pak na většině škol dostanou poprvé do kontaktu s informatikou na konci prvního stupně ZŠ. [4]

2.1 Rámcové a školní vzdělávací programy

Ministerstvo školství, mládeže a tělovýchovy (MŠMT) je hlavní státní institucí spravující program národního vzdělávání. Jejím účelem je přes zákony definovat zásady a cíle vzdělávání. Tyto zákony pak projednává a schvaluje Poslanecká sněmovna ČR. Národní program vzdělávání vymezuje [Rámcový vzdělávací program \(RVP\)](#) pro jednotlivé obory vyučované na základních a středních školách. RVP pak definují výstupy v jednotlivých stupních vzdělávání. Školy na základě

těchto výstupů vytvářejí **Školní vzdělávací program (ŠVP)**. Jedním z výstupů ŠVP, který se aplikuje ve výuce je pak výukový plán vycházející z těchto RVP. ŠVP jsou pak schvalovány ředitelem dané školy, případně školskou radou. [5]



Obrázek 1: Systém RVP a ŠVP. Převzato z: https://cs.wikipedia.org/w/index.php?title=RC3A1mcovC3BD_vzdC49B1C3A1vacC3AD_program&oldid=20342426

Jednotlivé školy tedy tyto rámcové programy přebírají a upravují podle svých potřeb. Učitelé na školách tak nejsou vázáni pevnými osnovami popisující učivo, které se má v dané hodině probrat, ale mají možnost si výuku upravit podle sebe a podle potřeb celé třídy (případně také podle odborného zaměření dané školy). Školy tímto mohou do určité míry upravovat časovou dotaci věnovanou jednotlivým předmětům v závislosti na tom, zda je daná škola více zaměřena matematicky, umělecky nebo sportovně. [6]

2.2 Rámcový vzdělávací program v informatice

RVP dosud pro informatiku na ZŠ vymezovalo 1 vyučovací hodinu týdně na prvním i druhém stupni. Výstupy v těchto hodinách byly zaměřené převážně na uživatelskou práci s počítačem. Výuka informatiky pak na většině škol začínala na konci prvního stupně základních škol. Nově bude výuka informatiky podle nových RVP rozdělena do 4 sekcí: [7]

1. Data, informace a modelování,
2. Algoritmizace a programování,
3. Informační systémy,
4. Digitální technologie.

Tyto jednotlivé RVP pro základní školy již byly schváleny MŠMT a v platnost mají vstoupit od 1. září 2023¹. Takto velká změna ve výuce informatiky byla naposledy v roce 2004 a je tedy zřejmé, že za tu dobu se dnešní svět v oblasti informatiky a její výuky proměnil. [8]

Na prvním a druhém stupni základního vzdělávání je výuka založena na aktivních činnostech, při kterých se žáci seznamují s novými digitálními technologiemi a inforatickými pojmy. Žáci navrhuji a objevují řešení různých problémů, o kterých diskutují s ostatními. Součástí je také pochopení, jak digitální technologie fungují a jejich bezpečné používání. Tímto poznáváním se učí chápat základní práci s daty, principy kódování a způsoby, jak chránit své soukromí. Očekávané výstupy RVP v sekci Algoritmizace a programování vypadají následovně: [9]

- po přečtení jednotlivých kroků algoritmu nebo programu vysvětlí celý postup; určí problém, který je daným algoritmem řešen
- rozdělí problém na jednotlivě řešitelné části a navrhne a popíše kroky k jejich řešení
- vybere z více možností vhodný algoritmus pro řešený problém a svůj výběr zdůvodní; upraví daný algoritmus pro jiné problémy, navrhne různé algoritmy pro řešení problému
- v blokově orientovaném programovacím jazyce vytvoří přehledný program s ohledem na jeho možné důsledky a svou odpovědnost za ně; program vyzkouší a opraví v něm případné chyby; používá opakování, větvení programu, proměnné
- ověří správnost postupu, najde a opraví v něm případnou chybu

RVP také definuje minimální doporučenou úroveň výstupů určenou například pro školy jiného zaměření nebo pro žáky se speciálními potřebami.

3 Digitální gramotnost a inforatické myšlení

Jak již bylo zmíněno, ve výuce nyní bude kladen důraz na rozvoj digitálního a inforatického myšlení. Pro začátek můžeme říct, že se jedná o komplex dovedností, které jedinec v dnešní době využívá v běžném životě nebo ve společnosti. Ve výuce informatiky bude pro novou akreditaci klíčové také rozšíření časové dotace pro výuku, rozvoj oborových kompetencí v ostatních předmětech, a nakonec využití digitálních technologií ve výuce a vzdělávání.

¹Nejedná se však o mezní termín. V platnost pro oba stupně ZŠ začne povinně platit nejdříve v září 2024.

3.1 Digitální gramotnost

Podle NÚV digitální gramotnost představuje souhrn digitální kompetencí, které jedinec využívá s digitálními technologiemi při práci, učení a ve volném čase. Základní charakteristikou těchto kompetencí je využití digitálních technologií při řešení nejrůznějších problémů v běžném životě: NÚV rozlišuje tyto jednotlivé digitální kompetence jedince do 3 oblastí: [10]

- Člověk, společnost a digitální technologie,
- Tvorba digitálního obsahu,
- Informace, sdílení a komunikace v digitálním světě.

Do první oblasti spadá zejména schopnost jedince se zapojit do společnosti prostřednictvím digitálních technologií. Může se jednat o nejrůznější činnosti jako jsou například využívání digitálního fotoaparátu, připojení mobilního zařízení do Wi-Fi sítě atp. Druhá oblast se zaměřuje na tvorbu (úpravu) vlastního (cizího) digitálního obrazu. Sem můžeme zařadit focení digitálním fotoaparátem nebo vytvoření prezentace pro svůj projekt. Do poslední oblasti pak spadá například využívání sociálních sítí a komunikačních platforem za účelem sdílení informací.

3.2 Informatické myšlení

Další částí, kterou popisují revize rámcových programů je informatické myšlení. Definování informatického myšlení není jednoduché, jak se může na pohled zdát. Nejjednodušší definice pro širokou veřejnost říká, že informatické myšlení je *schopnostní myslet jako informatik při řešení problémů*. [11] U dětí je například vhodné si každé ráno dopředu rozvrhnout činnosti, které je potřeba udělat. Tyto činnosti si dokážeme krok po kroku seřadit podle našich preferencí. Do jisté míry se může zdát, že při tomto plánování člověk informaticky přemýšlí, ale tomu tak není. Jak uvádí Daniel Lessner ve svém článku o analýze pojmu informatického myšlení, tak při takových jednoduchých situacích používáme svůj selský rozum. Definice pojmu informatické myšlení není tedy pevně zavedená a nemůžeme tedy uvažovat pouze jednu. Řada odborníků, zabývajících se touto problematikou má na tyto definice odlišné pohledy podle toho, ke kterému odvětví informatické myšlení přiřazují. Podle portálu imysleni.cz, který stojí za projektem Podpora rozvíjení informatického myšlení, je informatické myšlení způsob myšlení, zaměřující se na popis problému, jeho analýzu a hledání efektivních řešení. Mezi jeho hlavní body výstupů pak řadí: [12]

- posouzení různých řešení daného problému a vybrání toho nejvhodnějšího,
- rozdělení většího problému na několik menších samostatně řešitelných problémů,
- plánování a řízení činností,

- vytváření a popisování postupů (které může použít někdo jiný),
- uspořádání a strukturování souborů a dat, ...

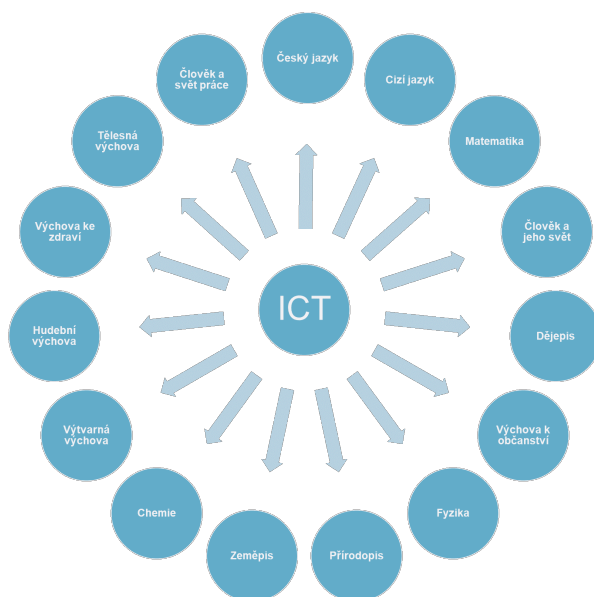
Plánování a řízení činností lze navíc doplnit o schopnost rozplánovat a optimalizovat algoritmus tak, aby se vyhnul opakujícím se vzorům.

Jak již bylo zmíněno ve studii v úvodu výše, samotné využívání digitálních technologií nerozvíjí informatické myšlení a je tedy důležité z pohledu vzdělávání pojmy digitální a informatická gramotnost nezaměňovat.

3.3 Začlenění předmětu ICT do výuky

Jedním z dalších cílů, který sebou nese revize výukového plánu je začlenění digitálních technologií a informatického myšlení do oborových kompetencí ostatních předmětů. Pan Lessner dále zmiňuje, že rozvoj informatické myšlení v neinformatických předmětech nemá za cíl dané předměty omezovat. Naopak by právě měl prohloubit znalosti vyučované v těchto předmětech a lépe chápat souvislosti. Můžeme říct, že to samé platí i pro rozvoj digitální gramotnosti.

NÚV tak ve své revizi počítá s rozdělením rozvoje digitálních a informatických kompetencí tak, aby nezasahovala do osnov ostatních předmětů, ale aby na druhou stranu „obohatila“ samotnou výuku daného předmětu.



Obrázek 2: Postoj ICT k ostatním předmětům. *Převzato z: https://www.edu.cz/wp-content/uploads/2020/08/Navrh_revizi_ICT-1-1.pdf*

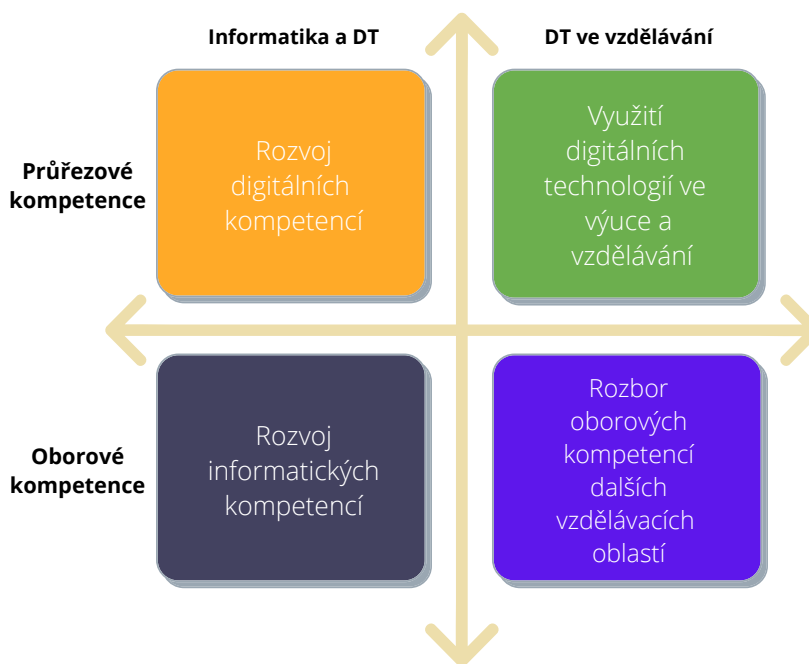
Z obrázku 2 je patrné, jaký postoj by měla mít výuka ICT vůči ostatním předmětům. Doposud byla výuka ICT odtržena od ostatních předmětů a nebyla zde „provázanost“ vzdělávacích plánů s ostatními předměty. Jako příklad této

provázanosti je možné uvést kombinaci Přírodopis – Chemie, kde obsahy těchto dvou přírodovědeckých předmětů jsou sice odlišné, ale zároveň obsah jednoho předmětu rozšiřuje obsah toho druhého a naopak.

Jedna z dalších myšlenek související zmíněná v [10] zní: „*Současně spolu s pronikáním digitálních technologií do nejrůznějších oblastí a činností člověka rostou nároky na související vědomosti, dovednosti, postoje, přitom je problematické je rozvíjet a dále aplikovat odtrženě od kontextu ostatní práce žáků ve škole... Obor informatika má vlastní cíle, jejich význam roste a základ, který může nabídnout ostatním oborům se proměňuje.*“

Rostoucí nároky a rychlý vývoj těchto digitálních technologií pak vnáší neefektivitu do vzdělávání, jelikož žáci musejí řešit konkrétní problémy dvakrát. Nejprve bez digitálních technologií ve výuce neinformatického předmětu a poté s digitálními technologiemi v informatice.

Způsob, jakým tyto nedostatky řešit popisuje NÚV tím, že rozděluje rozvoj digitálních a informatických kompetencí žáka. Na obrázku 3 vidíme rozdělení 4 oblastí podle toho, který předmět se vyučuje, pro tyto potřeby je můžeme rozlišovat na informatické a neinformatické. Z obrázku je zřejmé, že digitální kompetence žáka budou rozvíjeny v kontextu napříč ostatními předměty a informatické kompetence pak budou rozvíjeny v rámci informatického předmětu.



Obrázek 3: Koncept rozvoje digitální gramotnosti a informatického myšlení. Převzato z: https://www.edu.cz/wp-content/uploads/2020/08/Navrh_revizi_ICT-1-1.pdf

Horizontální dělení člení výuku na vzdělávací cíle a na ně navázané výukové aktivity podle toho, co učitel při výuce vnímá jako prvotní. Levá část se zaměřuje

na rozvoj digitálních či inforatických kompetencí v předmětu informatika a v pravé části se poté zaměřuje na daný (neinforatický) předmět a k jeho výuce využívá digitálních technologií. V tomto případě se učitelé řídí výstupy daného předmětu a platí, že učitelé přitom zohledňují ostatní kompetence žáků.

Vertikální dělení pak rozděluje výuku podle průřezových a oborových kompetencí. Do průřezových kompetencí řadíme aktivity v ostatních předmětech, kde žáci pracují s digitálními technologiemi. Tyto aktivity jsou podřízené vzdělávacímu plánu vyučovaného předmětu. Do oborových kompetencí pak řadíme aktivity, které jsou oborově zaměřené podle vzdělávacího obsahu předmětu. Toto dělení je více podmíněné ŠVP a způsoby, jakými učitelé vedou výuku.

4 Algoritmizace

Jak již bylo zmíněno, výstupem této bakalářské práce je výukový plán zaměřený na algoritmizaci a programování. Proto si v této práci pojem algoritmizace trochu představíme.

4.1 Definice algoritmu

Slovo „algoritmus“ bylo odvozeno ze jména arabského matematika Al-Chorezmí. V té době bylo jeho významem provádění aritmetických operací v arabských číslicích. [13]

Stejně jako u inforatického myšlení, slovo „algoritmus“ nemá danou univerzální definici. Jedna z definic říká, že algoritmem rozumíme konečnou posloupnost či sekvenci pravidel, která je aplikována na konečnou množinu dat a umožňuje řešit třídu problémů podobného typu. Místo spojení *sekvenci pravidel*, by bylo vhodnější použít spíše *sekvenci příkazů*. Příkazem pak rozumíme jednu elementární instrukci.

Euklidův algoritmus, vytvořený řeckým matematikem Euklidem je považován za jeden z nejstarších algoritmů.² Tento algoritmus slouží pro nalezení největšího společného dělitele 2 čísel. Existuje řada implementací a modifikací toho algoritmu, například pro 3 a více čísel. Algoritmus pak může vypadat takto:

Algoritmus 1 Euklidův algoritmus

Input: celá čísla m, n

Output: m

```
1: while  $n \neq 0$  do  
2:    $r \leftarrow m \bmod n$   
3:    $m \leftarrow n$   
4:    $n \leftarrow r$   
5: end while  
6: return  $m$ 
```

²Jeho původ sahá do doby 300 let př. n. l.

4.2 Podmínky a zápisy algoritmu

Každý algoritmus musí splňovat následujících pět podmínek: [14]

1. elementárnost – algoritmus se skládá z jednoduchých kroků,
2. resultativnost – každý algoritmus musí mít alespoň jeden výstup,
3. konečnost – algoritmus musí po určitém počtu kroků skončit,
4. obecnost – algoritmus dokáže vyřešit problém pro danou množinu vstupů,
5. determinovanost – v algoritmu musí být každý krok přesně definován.

Nesplňuje-li algoritmus jednu z těchto podmínek, pak algoritmus není správně sestaven a hrozí, že skončí chybou.

Existuje několik způsobů, jak algoritmus vyjádřit. První a nejjednodušší způsob je slovní vyjádření. Typickým příkladem slovního vyjádření může být recept na vaření. Dalším způsobem vyjádření algoritmu je prostřednictvím programovacího jazyka, například jazyka C. Poslední způsob, který zde uvedeme je vyjádření pomocí pseudokódu. Jedná se o neformální zápis s určitými pravidly, které můžeme zařadit mezi vyjádření slovním způsobem a programovacím jazykem. Příkladem takového pseudokódu může být již zmíněný Euklidův algoritmus 1.

4.3 Typy algoritmů

Algoritmy používané v informatice můžeme dále rozdělit do několika typů. Některé algoritmy mohou spadat do více typů. [15]

Rekurzivní algoritmy se ve svém těle opakovaně volají, dokud daný problém nevyřeší (dokud nenastane koncová podmínka). Pokud máme na začátku problém, tak s každým opakovaným voláním tohoto algoritmu dochází ke zjednodušení daného problému. Algoritmus by poté měl po konečném počtu kroků dojít k požadovanému výsledku.

Algoritmy typu rozděl a panuj řeší daný problém tak, že jej opakovaně rozdělují na dvě poloviny. Algoritmus poté tyto dílčí podproblémy řeší, až je jejich řešení snadno vyřešitelné.

Dynamicky programovací algoritmy si uchovávají výsledky dílčích podproblémů, které potom využívá k efektivnějším vyřešení složitějších problémů.

Hladové algoritmy při každém kroku hledají ideální řešení nebo spíše to nejlepší, co se v tuto chvíli nabízí. Jeho nevýhodou je, že nezohledňuje problém (nebo jeho vývoj) do hloubky. Může se tak stát, že ne vždy algoritmus nalezneme optimální řešení.

Algoritmy hrubé síly mají jednoduchou myšlenku, jsou však velmi neefektivní. Myšlenka spočívá v procházení všech možných řešení problémů do doby, než nalezne výsledek. Pokud bychom chtěli zjistit heslo náhodné osoby, o které nemáme žádné informace, je prolomení hrubou silou jediným řešením.

Algoritmy zpětného sledování využívají při řešení problému procházení více možností a zároveň si uchovávají možné kandidáty pro řešení podproblému. Pokud se řešení nepodaří nalézt (nebo je řešení nežádoucí), tak se algoritmus vrátí o krok zpět.

5 Robot Sphero SPRK+

Sphero je americká společnost založená v roce 2010 zaměřující se na vývoj programovatelných robotů převážně pro vzdělávací účely. Sphero SPRK+ je jedním z modelů této společnosti. Roboti jsou určeny pro děti od 5 let a mají za cíl seznámit děti se základy programování.



Obrázek 4: Robot Sphero SPRK+. Převzato z: <https://www.alphr.com/technology/1003918/sphero-sprk-review-a-little-ball-of-educational-fun/>

Sphero SPRK+ je robot kulatého tvaru z plastu o velikosti dlaně. Uvnitř robota se nachází obvodová deska, na které najdeme 2 motory, procesor (CPU), dvě LED (jednu hlavní a vedlejší) a nakonec tzv. IMU jednotkou. IMU je zařízení obsahující několik senzorů, které zaznamenávají zrychlení (akcelerometr) a nebo orientaci v prostoru (gyroskop a magnetometr). Robot je dále vybaven Bluetooth čipem zajišťující komunikaci s vývojovým prostředím. Dále se uvnitř nachází baterie s bezdrátovým dobíjením. Díky absenci portů a celoplošnou ochranou je vhodný i pod vodu.

5.1 Ovládání a programování

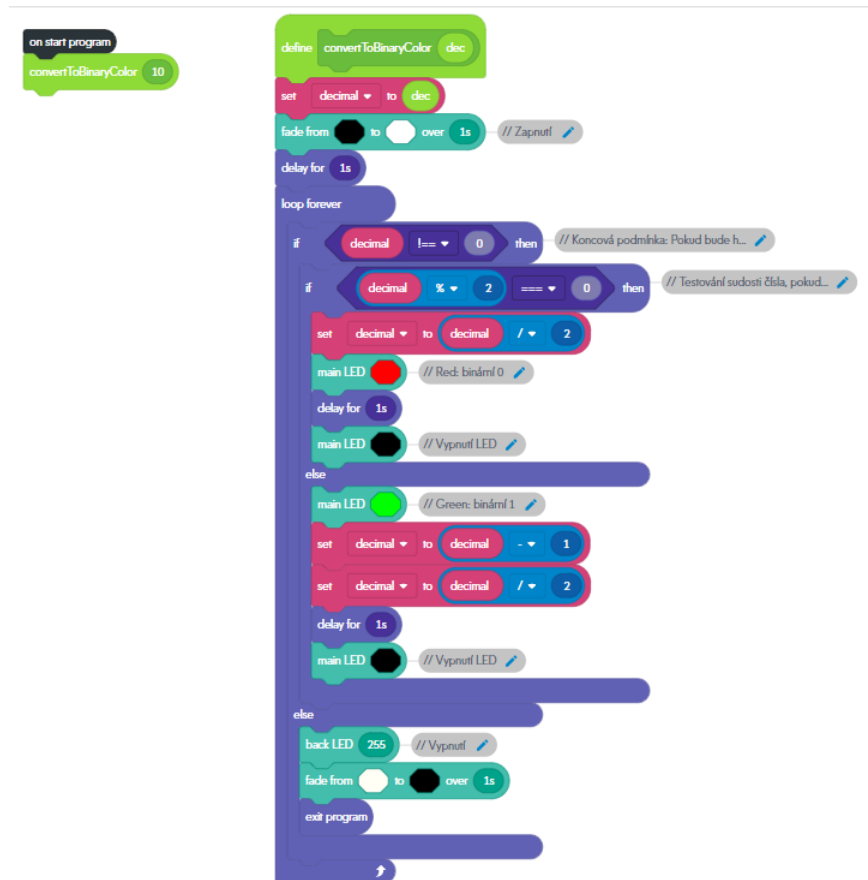
Vývojové prostředí Sphero Edu dostupné pro mobilní a desktopová zařízení slouží k programování robota a nabízí 3 způsoby ovládání. Tím nejjednodušším způsobem je ovládání pouhým prstem na displeji, případně klávesami WASD na klávesnici. Jelikož se jedná o intuitivní přístup, tak jej zde dále rozebírat nebudeme. Dalšími dvěma způsoby jsou psaní v tzv. blokovém kódu a v jazyce Javascript. Nás bude především zajímat blokový kód, protože jeho používání je jeden ze základních výstupů v revizích RVP³.

V blokovém kódu se instrukce programují v ucelených blocích, které uživatel přetahuje po vymezené ploše. Tyto bloky nahrazují běžné příkazy programovacích jazyků, kde je potřeba znát syntax jazyka. Bloky lze do sebe zanořovat a tím pádem je možné vytvářet i složitější programy s cykly a větvenými. Výhodou blokového kódu je také lepší přehlednost a orientace v kódu. Na obrázku 5 níže můžeme vidět blokový kód pro převod čísla z desítkové soustavy do binární, kde hodnoty 1 a 0 reprezentujeme zelenou a červenou barvou. I proto je programování v blokovém jazyce vhodné pro začátečníky, zejména pak pro mladší žáky. Příkladem další takové platformy je například Scratch.

5.2 Platforma Scratch

Scratch je on-line výukové programovací prostředí využívající blokových instrukcí (bloků), které nahrazují běžné příkazy programovacích jazyků. Pro naše účely i účely výuky pracujeme ve Scratch s rozšířením Pero, které umožní pracovat s vektorovou grafikou, jinak také želví grafikou. Výhodou těchto on-line platform je jejich přesnost. Na některých úlohách je pak možné vidět rozdíly mezi méně přesným robotem a želví grafikou. Právě proto je jedna ze sbírek úloh zaměřená právě na programování ve Scratch.

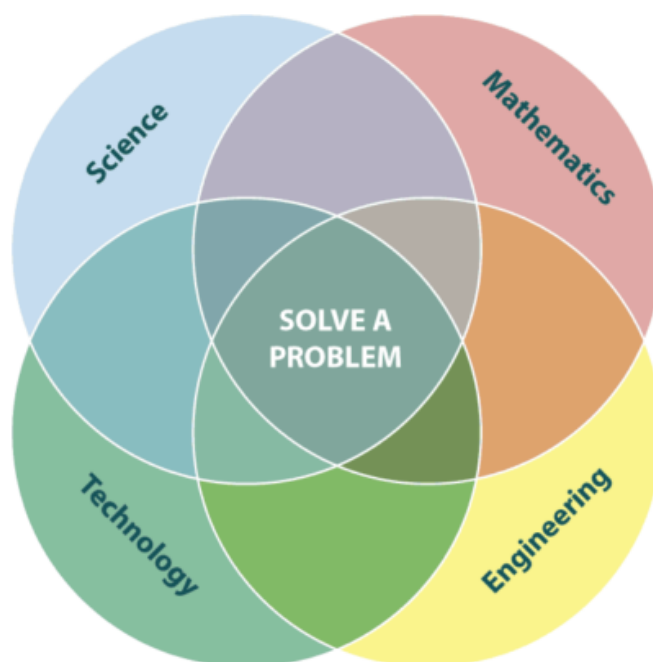
³Ve skutečnosti se žáci poprvé setkají s blokovým kódem už na konci prvního stupně.



Obrázek 5: Blokový kód na platformě Sphero Edu

5.3 STEM

Pojem **STEM** stojící za názvem *Science, Technology, Engineering and Mathematics* je vzdělávací koncept původem ze Spojených států, který původně měl za cíl zvýšit zájem veřejnosti (studentů) o studium těchto oborů. A to především kvůli klesajícímu zájmu o tyto přírodní a technické oblasti nebo také jako snahu poukázat na důležitost těchto oborů. Sdružení těchto oborů pod zkratku STEM pak poukazuje na provázanost jednotlivých oborů mezi sebou. Výuka pomocí této metodiky využívá při výuce vzdělávací pomůcky pro žáky a studenty různých věkových kategorií. [16]



Obrázek 6: Princip STEM vzdělávání. Převzato z: <https://www.jeduedu.cz/stem/>

5.4 Využití SPRK+ při výuce

Součástí bakalářské práce v příloze je sbírka úloh pro Scratch a robota SPRK+, které mohou být využity při výuce.

Úlohy jsou rozděleny na 2 části a jsou navrženy tak, aby si žák vyzkoušel aplikovat své znalosti v praxi. První část úloh spočívá v procvičení algoritmického myšlení na platformě Scratch. Ve Scratch jsou úlohy zaměřené výhradně na práci s vektorovou grafikou, která je součástí balíčku Pero. Ve druhé části jsou pak úlohy navrženy pro programování robota SPRK+ v aplikaci Sphero Edu.

Při výuce programování s roboty je pro žáky pohodlnější využívat právě blokově orientovaný jazyk oproti těm běžným. Neplatí to však pouze pro žáky na základní škole, ale i pro starší studenty. Podle této studie [17], která testovala 3 skupiny studentů programující roboty ve výuce, se ukázalo, že skupina využívající vizuálního programovacího jazyka dosáhla lepších studijních výsledků, narozdíl od studentů programujícího robota v textovém programovacím jazyce. Nutno podotknout, že studenti, kteří se účastnili této studie měli již základy programování. Žáci by se měli při výuce algoritmizace a programování naučit především algoritmicky přemýšlet, což jim programování v blokovém jazyce umožňuje.

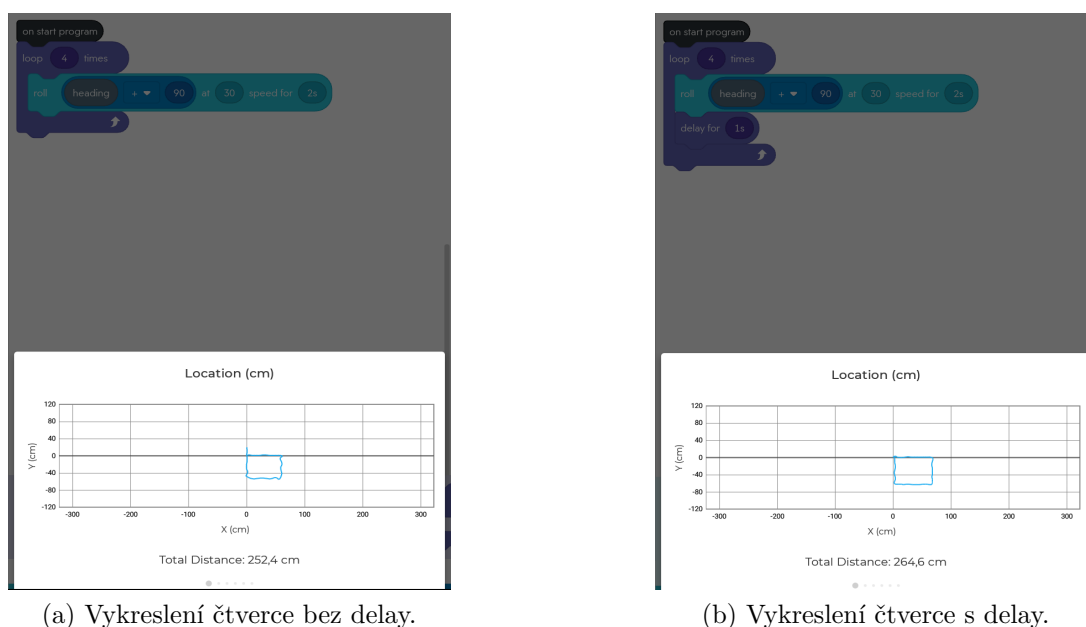
5.5 Nedostatky SPRK+

Na rozdíl od programování ve vektorové grafice se při vytváření úloh přišlo na několik nedostatků spojené s robotem SPRK+.

5.5.1 Nepřesnost způsobená vnějšími vlivy

Vykreslení jednoduchých geometrických obrazců SPRK+ zvládá relativně dobře. Dochází jen k menší odchylce v délce. Občas se ale dají pozorovat nepřesnosti v dráze. Problém nastává, pokud chceme vykreslit složitější obrazec s cykly, například čtverce s rostoucí délkou. Zde se oproti vektorové grafice ve Scratch podstatně liší, protože s každým průchodem v cyklu dochází ke zvětšení chyby. Na tyto chyby může mít vliv mírně nakloněná podlaha anebo také tření robota o rovinu.

U některých úloh lze nepřesnost snížit tak, že každý pohyb robota proložíme instrukcí čekej (anglicky delay). Díky tomu umožníme robotovi plynulé zastavení nebo aspoň dostatečné snížení jeho setrvačnosti. Zde na obrázku níže vidíme vykreslení čtverce s instrukcí delay a bez ní. Délka strany čtverce by měla být 60 cm.



Obrázek 7: Vykreslení

Jak je možné vidět, při použití instrukce delay jsme dostali přesnější vykreslení čtverce. Nicméně ani jedna z možností vhodně nezobrazuje délku dráhy robota (obvod čtverce), jelikož $O = 4 \cdot 60$ cm.

5.5.2 Vliv setrvačnosti

Pomocí závislostí mezi dráhou s , rychlostí v a časem t lze dopočítat délku dráhy, kterou chceme, aby robot urazil. Pokud chceme, aby robot urazil vzdálenost 2 metry v čase $t = 4$ sekundy, pomocí vztahu $s = v \cdot t$ zjistíme, že potřebná rychlost je 50 cm/s (tím, že počítáme v cm je potřeba dráhu převést na cm). Při nižších rychlostech je dráha ve skutečnosti shodná s vypočtenými hodnotami, avšak

při vyšších rychlostech dochází ke značné odchylce s vypočtenými hodnotami. Nehledě na to, že robot je vybaven IMU jednotkou.

I přes tyto nedostatky je na druhou stranu robot vhodný pro fyzikální a výpočetní experimenty. Jako příklad lze uvést experiment, kdy můžeme dopředu vypočítat počet otáček, který robot urazí na dráze o délce 240 cm. Pokud se budeme pohybovat dopředu rychlostí 40 cm/s po dobu 6 sekund, urazíme onu vzdálenost 240 cm (aby platil vztah $s = v \cdot t$). Počet otáček pak vypočítáme jako podíl délky dráhy a obvodu kružnice. Proměnná d představuje průměr robota SPRK+. Průměr robota je 7,3 cm.

$$pocetOtacek = \frac{s}{\pi d} \quad (1)$$

V dalším případě se v experimentu pokusíme ověřit přesnost IMU jednotky a vlivu setrvačnosti. Představme si dráhu o délce 1 metr tak, že část dráhy mezi startem a cílem bude nakloněná rovina. Jinak řečeno, start bude položen ve větší výšce než cíl. Chceme, aby robot urazil vzdálenost právě 1 m, tedy 100 cm. Pokud bychom na startu měli kuličku, kterou postrčením uvedeme do rovnoměrného přímočarého pohybu⁴, tak v momentě, kdy dosáhne této nakloněné roviny, dojde ke zrychlení kuličky a začne se pohybovat rovnoměrně zrychleným pohybem. V důsledku toho dojde k nárůstu zrychlení a kulička poté předem stanovenou dráhu přejede. V případě našeho robota dojdeme k podobnému výsledku, i když by se mohlo zdát, že robot bude se zrychlením počítat (právě díky IMU jednotce). Z měření, ze kterého se vychází testovalo robota na dráze 220 cm. Měření se srovnává také s daty, při kterém byla dráha v jedné rovině. Při naklonění roviny došlo k odchylce přibližně o 60 cm. Při „vodorovné“ rovině došlo k odchylce přibližně o 20 cm. Data zpracována robotem při pohybu jsou dostupná v příloze. Vzhledem k tomu, že se jedná o výukového robota, můžeme tyto nedostatky zanedbávat. Žáci by však s těmito odchylkami měli být seznámeni.

6 Výukový plán

Výukový plán je jeden z klíčových dokumentů vycházející ze ŠVP. Měl by obsahovat veškerá probíraná témata v daném předmětu, dále pak náplň popřípadě výstupy v daném tématu a nakonec časovou dotaci v hodinách. Časová dotace by měla odpovídat náročnosti jednotlivých témat a součet těchto hodin by měl padnout do rozmezí 35 až 45 vyučovacích hodin, který je pro informatiku vyhrazen.

6.1 Cíle výukového plánu

Mezi základní cíle výukového plánu zařadíme tyto hlavní body:

1. Naučit žáky algoritmicky přemýšlet a popsat problém, se kterým se setkají.

⁴Předpokládáme, že do kuličky strčíme takovou silou, aby v rovině urazila právě 100 cm.

2. Umět rozdělit problémy na menší podproblémy, které žák bude schopen samostatně řešit.
3. Řešení nějakého problému převést do podoby algoritmického návrhu – pro naše potřeby uvažujeme pouze blokový kód.
4. Spustit program a případné chyby v něm opravit.

6.2 Návrh výukového plánu

Tabulka 1: Výukový plán.

Téma	Výstupy	Počet h.
Teoretický úvod do algoritmizace a pseudokódu	Žák zná definice algoritmu a dokáže vyjmenovat jeho základní podmínky. Je schopen algoritmicky popsat některou ze svých každodenních činností.	2
Instrukce používané v algoritmech	Probrány jsou základní logické spojky, operátory a některé algoritmické instrukce: přiřazení, inkrementace, ... Žáci jsou seznámeni s větvením programu pomocí podmíněných výrazů a cyklů.	2
Blokový kód	Seznámení se s programovacím prostředím Scratch. Žák se umí orientovat v jednotlivých blocích, které prostředí nabízí a dosud získané znalosti si vyzkouší aplikovat v tomto blokovém prostředí. Do prostředí je přidáno rozšíření Pero, které lze přirovnat k programování v želví grafice.	2
Úlohy na platformě Scratch	Procvičení dosud získaných znalostí na úlohách ve Scratch.	2
Základní datové typy	Zná základní datové typy používané v programování a rozezná mezi nimi rozdíly.	2
Funkce	Žák chápe základní princip funkcí. Naučí se je používat pro zpřehlednění svého kódu.	1
Pokračování na další straně.		

Tabulka 1 – pokračování

Téma	Výstupy	Počet h.
Představení robotické hračky Sphero SPRK+	Seznámení se s robotem Sphero SPRK+, instalace aplikace Sphero Edu a základní orientace v IDE. Je schopen robota spárovat se zařízením.	1
Vytváření vlastního programu v blokovém kódu v Sphero Edu	Žák si vyzkouší sestavit jednoduchou aplikaci sestavenou v blokovém kódu. Robota si vyzkouší naprogramovat pro pohyb v různých směrech, využívá k tomu LED senzory a ostatní funkce, které prostředí nabízí. Ve svém programu případné chyby dokáže opravit.	3
Programování ve Sphero Edu: úloha na výpočet dráhy (úloha 1)	Žák si vyzkouší Sphero robota aplikovat v úloze, ve které se bude snažit předem vypočítat požadovanou vzdálenost. K tomu může použít uživatelsky definované proměnné.	1
Programování ve Sphero Edu: čtverec (úloha 2)	Při úlohách si osvojí používání cyklů při řešení daných problémů.	1
Programování ve Sphero Edu: zigzag (úloha 3)	Při úlohách si osvojí používání cykly při řešení daných problémů.	1
Programování ve Sphero Edu: hodiny (úloha 4)	V této hodině si žáci zopakují práci s úhly a jejich implementaci do programu. Například si vyzkouší, jakým postupem nasměrovat robota tak, aby ukázal na správnou ručičku na hodinách podle zadaného vstupu.	2
Pokračování na další straně.		

Tabulka 1 – pokračování

Téma	Výstupy	Počet h.
Číselné soustavy	Jsou zopakovány základní číselné soustavy – především binární, osmičková a hexadecimální. Žáci již mají o nich základní přehled, jelikož se jedná o jednu z částí probíranou v předmětu Data, informace a modelování. Znají tedy i jejich využití (např. hexadecimální má zastoupení v počítačové grafice u pixelů, ...). Naučí se mezi těmito hlavními soustavami převádět.	2
Programování ve Sphero Edu: binární soustava (úloha 5)	Úloha má za cíl prohloubit znalosti převodu mezi soustavami. Zde se jedná o převod z desítkové do binární, kterou se žák s pomocí učitele pokusí implementovat.	2
Programování ve Sphero Edu: bludiště (úloha 6)	Žáci si ve třídě vytvoří bludiště podle návrhu, přičemž stěny budou reprezentovány páskou. Bludiště si také mohou pokusit postavit podle vlastního návrhu. Jejich cílem bude naprogramovat Sphero robota tak, aby dokázal bludiště bezpečně opustit (aniž by přešel lepicí pásku).	2
Programování ve Sphero Edu: co se vykreslí (úloha 7)	V této úloze je zadán algoritmus, který představuje pohyb robota po zemi. Jejich úkolem je po přečtení pseudokódu přijít na to, jaký tvar Sphero vykreslí. Pokud si nejsou jistí mohou si postupně jednotlivé kroky zaznamenat na papír.	2
Programování ve Sphero Edu: slalom (úloha 8)	Žáci si sestavili vlastní dráhu s překážkami. Cílem hodiny je sestavit program tak, aby Sphero projížděl oblouky mezi jednotlivými překážkami.	2
Programování ve Sphero Edu: síť (úloha 9)	Žáci si opět procvičí aplikování cyklů a rozložení kódů do jednotlivých funkcí.	2
Pokračování na další straně.		

Tabulka 1 – pokračování

Téma	Výstupy	Počet h.
Programování ve Sphero Edu: šance na zelenou (úloha 10)	Ke konci předmětu si žáci zopakují své matematické znalosti (práce se zlomky, procenty) a navrhnu algoritmus pro výpočet šance generování zelené barvy.	2
Shrnutí celého předmětu	Na závěr celého předmětu je žákům shrnuta teorie z oblasti algoritmizace a základů imperativního programování.	1

Závěr

V této práci jsme si představili hlavní změny rámcového vzdělávacího programu pro informatiku. Společně s tím jsme zmínili klíčové výstupy a rozdělení výuky informatiky na 2. stupni základní školy se zaměřením na algoritmizaci a programování. Dále jsme si popsali rozdíly mezi digitální gramotností a informatickým myšlením, a také jak k těmto dovednostem přistupovat v rámci výuky informatických i neinformatických předmětů. K tomu jsme krátce shrnuli základní pojmy z algoritmizace.

Důležitou součástí práce je pak představení robota Sphero SPRK+ a jeho využití při výuce. Spolu s tím byly zmíněny jeho nedostatky, které však mohou vhodně doplnit výuku fyziky a matematiky. Společně propojením těchto oblastí pod zkratku STEM dává vznik novému přístupu k výuce informatiky a dalších technických oborů napříč školami.

Zmínili jsme také hlavní benefity programování robotů v blokovém jazyce, a to jak v prostředí Sphero Edu, tak i v on-line vývojovém prostředí Scratch, kde pracujeme především s obdobou želví grafiky. Poté jsme si na experimentu ukázali nepřesnosti robota, které však můžeme pro účely výuky zanedbávat, ale je vhodné s nimi seznámit žáky. I přes tyto nedostatky robot SPRK+ vyhovuje výstupům rámcových vzdělávacích programů, a proto se hodí robota zahrnout do nové výuky informatiky.

V závěru pro obě tyto vývojové platformy byly vytvořeny sbírky úloh s řešením, které mohou sloužit učitelům jako opora při výuce.

Conclusions

We have presented the main changes to the school educational programs for computer science. We have also mentioned the main outcomes and the partition of learning computer science at the 2nd grade of primary schools with focus on algorithmization and programming. Then we described the differences between digital literacy and computational thinking and how to use these skills in the context of teaching informatic and non-informatic subjects. We summarised the basic concepts of algorithmization.

The important part of this work is the introduction of the Sphero SPRK+ robot and its use in teaching. We presented its main defects, but on the other side they could fill in teaching of physics and mathematics. By linking these areas under the shortcut STEM, it gives a new attitude to teaching computer science and other technical subjects.

We have also mentioned the main benefits of programming robots in a block-based language both in the Sphero Edu IDE and in the Scratch IDE, where we worked primarily with a form of turtle graphics. Then we did an experiment to see the inaccuracies of the robot, which we can neglect for teaching purposes, but it's useful to introduce them to students. Despite these defects the SPRK+ robot goes down well with school educational programs, and therefore it is suitable to include the robot in the new curriculum.

In the conclusion, for both of these platforms have been created collections of exercises together with solutions that can be used as a teacher's tool.

A Sbírka úloh pro Sphero SPRK+

Příloha s názvem *martin_kostka_sbirka_uloh_SPRK+* obsahuje sbírku úloh s řešeními pro programování robota Sphero SPRK+. Veškeré kódy jsou poté dostupné v příloze.

B Sbírka úloh pro Scratch

Příloha s názvem *martin_kostka_sbirka_uloh_Scratch* obsahuje sbírku úloh s řešeními pro programování na platformě Scratch. Veškeré kódy jsou poté dostupné v příloze.

C Obsah přiloženého CD/DVD

Na samotném konci textu práce je uveden stručný popis obsahu přiloženého CD/DVD, tj. jeho závazné adresářové struktury, důležitých souborů apod.

doc/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

src/

Kompletní zdrojové texty úloh pro platformy Scratch a Sphero Edu se všemi potřebnými (příp. převzatými) zdrojovými texty, knihovnami a dalšími soubory potřebnými pro bezproblémové spuštění.

Navíc CD/DVD obsahuje:

data/

Ukázková a testovací data použitá v práci a pro potřeby testování práce při tvorbě posudků a obhajoby práce.

literature/

Vybrané položky bibliografie, příp. jiná užitečná literatura vztahující se k práci.

video/

Vybrané nahrávky s robotem na některých úlohách.

U veškerých cizích převzatých materiálů obsažených na CD/DVD jejich zahrnutí dovolují podmínky pro jejich šíření nebo přiložený souhlas držitele copyrightu. Pro všechny použité (a citované) materiály, u kterých toto není splněno a nejsou tak obsaženy na CD/DVD, je uveden jejich zdroj (např. webová adresa) v bibliografii nebo textu práce nebo v souboru `readme.txt`.

Seznam zkratk

ICILS International Computer and Information Literacy Study

IMU Inertial Measurement Unit

NÚV Národní ústav vzdělávání

RVP Rámcový vzdělávací program

STEM Science & Technology & Engineering & Math

ŠVP Školní vzdělávací program

Literatura

- [1] Neumajer, Ondřej. *Mezinárodní výzkum digitálních dovedností žáků ICILS 2018*. 2020. Dostupný z: <https://spomocnik.rvp.cz/clanek/22347/%20MEZINARODNI-VYZKUM-DIGITALNICH-DOVEDNOSTI-ZAKU-ICILS-2018.html>.
- [2] MŠMT. *Povinná školní docházka*. Dostupný z: <https://www.msmt.cz/vzdelavani/zakladni-vzdelavani/plneni-povinne-skolni-dochazky-v-zahranici-od-1-9-2017>.
- [3] MŠMT. *Rámcový vzdělávací program*. Dostupný z: https://cs.wikipedia.org/wiki/Vzd%C4%5C%9B1%5C%3%5C%A1v%5C%3%5C%A1n%5C%C3%5C%AD_v_%5C%4%5C%8Cesku.
- [4] Slížek, David. *Změny ve výuce informatiky*. Dostupný z: <https://www.lupa.cz/clanky/daniel-lessner-jednota-skolskych-informatiku-v-yuku-informatiky-ceka-po-17-letech-revoluce-jak-dopadne-zalezi-na-skolach-a-ucitelich/>.
- [5] Školský zákon. *Zákon o předškolním, základním, středním, vyšším odborném a jiném vzdělávání (školský zákon)* [online]. [cit. 2004-11-10]. Dostupný z: <https://www.zakonyprolidi.cz/cs/2004-561>.
- [6] MŠMT. *Rámcový vzdělávací program*. Dostupný z: https://cs.wikipedia.org/w/index.php?title=RC3A1mcovC3BD_vzdC49B1C3A1vacC3AD_program&oldid=20342426.
- [7] Národní pedagogický institut. *Revize rámcového vzdělávacího programu základního vzdělávání: Vzdělávací obsah vzdělávacího oboru 1. stupeň*. Dostupný z: <https://revize.edu.cz/nova-informatika-v-rvp-zv>.
- [8] MŠMT. *MŠMT spustilo web na podporu moderní výuky informatiky*. Dostupný z: <https://www.msmt.cz/msmt-spustilo-web-na-podporu-moderni-vyuky-informatiky>.
- [9] MŠMT. *Rámcový vzdělávací program pro základní vzdělávání*. Dostupný z: <https://revize.edu.cz/files/informatika-2021.pdf>.
- [10] Národní ústav vzdělávání. *Návrh revizí rámcových vzdělávacích programů v oblasti informatiky a informačních a komunikačních technologií*. Dostupný z: https://www.edu.cz/wp-content/uploads/2020/08/Navrh_revizi_ICT-1-1.pdf.
- [11] Lessner, Daniel. *Analýza významu pojmu “computational thinking”*. Dostupný z: <http://www.jtie.upol.cz/pdfs/jti/2014/01/06.pdf>.
- [12] Pedagogická fakulta Jihočeské univerzity v Českých Budějovicích. *Projekt PRIM: Co je to informatické myšlení?* Dostupný z: <https://imysleni.cz/informaticke-mysleni/co-je-informaticke-mysleni>.
- [13] Wróblewski, Piotr. *Algoritmy: dříve než začneme*. První vyd. Brno: Computer Press, 2015. 19 s. ISBN 978-80-251-4126-7.

- [14] Jandová, Radoslava. *Algoritmus a jeho vlastnosti*. Dostupný z: http://www.vrstevnice.com/akce/grandaction/vskola/2semestr/ted/yd14ted_clanek_jandoral.pdf.
- [15] Pedamkar, Priya. *Introduction to Algorithms*. Dostupný z: <https://www.educba.com/types-of-algorithms/>.
- [16] Jedu Edu! *CO JE STEM?* Dostupný z: <https://www.jeduedu.cz/stem/>.
- [17] Rodríguez Corral, José María. *A Study on the Suitability of Visual Languages for Non-Expert Robot Programmers*. Dostupný z: https://www.researchgate.net/publication/330723636_A_Study_on_the_Suitability_of_Visual_Languages_for_Non-Expert_Robot_Programmers.