

Katedra informatiky  
Přírodovědecká fakulta  
Univerzita Palackého v Olomouci

# BAKALÁŘSKÁ PRÁCE

Využití robotů Ozobot pro výuku algoritmizace  
a programování



2022

Vedoucí práce:  
Mgr. Tomáš Kühr, Ph.D.

Mgr. Zuzana Pacholíková

Studijní obor: Aplikovaná informatika,  
kombinovaná forma

## **Bibliografické údaje**

Autor: Mgr. Zuzana Pacholíková  
Název práce: Využití robotů Ozobot pro výuku algoritmizace a programování  
Typ práce: bakalářská práce  
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci  
Rok obhajoby: 2022  
Studijní obor: Aplikovaná informatika, kombinovaná forma  
Vedoucí práce: Mgr. Tomáš Kühn, Ph.D.  
Počet stran: 37  
Přílohy: 1 CD  
Jazyk práce: český

## **Bibliographic info**

Author: Mgr. Zuzana Pacholíková  
Title: Ozobots 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, combined form  
Supervisor: Mgr. Tomáš Kühn, Ph.D.  
Page count: 37  
Supplements: 1 CD  
Thesis language: Czech

## Anotace

*Práce se věnuje nově zařazené výuce algoritmizace a programování na základních školách, pro kterou lze využít edukační robotiku, která dokáže lépe demonstrovat abstraktní programování. Pro tuto práci je zvolen Ozobot Evo, protože poskytuje programování pomocí kreslení černých čar a barevných kódů na papír a blokové programování OzoBlockly. Pro vybraného robota je vytvořena sbírka úloh pro žáky 4. a 5. třídy ZŠ, která pokrývá nově požadované učivo.*

## Synopsis

*The work deals with the newly included teaching of algorithms and programming in primary schools, for which it is possible to use educational robotics, which can better demonstrate abstract programming. In this work, Ozobot Evo is chosen because it provides programming by drawing black lines and color codes on plain white paper and block programming in OzoBlockly editor. A collection of problems for pupils of the 4th and 5th grade of primary school is created for Ozobot Evo, which covers the newly required curriculum.*

**Klíčová slova:** algoritmizace; programování; robotika; Ozobot; vzdělávání

**Keywords:** algorithms; programming; robotics; Ozobot; education

Děkuji vedoucímu své bakalářské práce Mgr. Tomáši Kührovi, Ph.D. za odborné vedení a cenné rady. Dále děkuji své rodině a příteli za podporu a trpělivost.

*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

<b>1</b>	<b>Úvod</b>	<b>7</b>
<b>2</b>	<b>Teoretická část</b>	<b>9</b>
2.1	Rámcový vzdělávací program pro základní vzdělávání 2021 . . . . .	9
2.1.1	Struktura RVP ZV . . . . .	9
2.1.2	Informatika (vzdělávací oblast) . . . . .	10
2.2	Informatické myšlení . . . . .	11
2.2.1	Algoritmizace a programování . . . . .	11
2.3	Myšlení a jeho vývoj u dětí prvního stupně ZŠ . . . . .	12
2.4	Konstruktivistická teorie učení . . . . .	14
2.5	Informatika a informatické myšlení na základních školách . . . . .	14
2.5.1	Robotika ve výuce . . . . .	16
2.5.2	Využití robotů Ozobot ve výuce . . . . .	17
<b>3</b>	<b>Praktická část</b>	<b>19</b>
3.1	Cíle práce . . . . .	19
3.2	Použité metody . . . . .	20
3.2.1	Ozobot Evo . . . . .	20
3.3	Sbírka úloh . . . . .	24
3.3.1	Úlohy s barevnými kódy . . . . .	25
3.3.2	Úlohy v OzoBlockly . . . . .	28
3.3.3	Metodická příručka . . . . .	29
3.4	Diskuze . . . . .	30
3.4.1	Náměty pro další práce . . . . .	31
3.4.2	Limity práce . . . . .	31
	<b>Závěr</b>	<b>33</b>
	<b>A Obsah přiloženého CD</b>	<b>34</b>
	<b>Literatura</b>	<b>35</b>

## Seznam obrázků

1	Screenshot Karel IDE ze stránek Standfordské univerzity [26] . . .	16
2	Nákres Ozobota Evo z českého manuálu [31] . . . . .	20
3	Příklad cesty s barevnými kódy [32] . . . . .	21
4	Screenshot editoru OzoBlockly [34] . . . . .	23
5	Úroveň 1 - Nováček . . . . .	23
6	Úroveň 2 - Začátečník . . . . .	24
7	Testování úlohy 1 – část C . . . . .	26
8	Screenshot mobilní obrazovky při spuštění MetaBota . . . . .	32

# 1 Úvod

Změny ve společnosti, ovlivněné rozvojem digitálních technologií a jejich využíváním v nejrůznějších oblastech lidské činnosti, vyžadují průběžnou úpravu strategie pro vzdělávání. Dnešní generace vyrůstá v prostředí, ve kterém jsou digitální technologie všudypřítomné a jejich používání je bráno za přirozenou součást života. V důsledku těchto změn schválila vláda České republiky dne 20. března 2013 novou koncepci Digitální Česko v. 2.0, Cesta k digitální ekonomice [1]. Na základě této koncepce pak vznikl dokument Strategie digitálního vzdělávání do roku 2020 [2], který se zabývá infromatickým myšlením, a který si za prioritní cíl klade rozvoj tohoto myšlení u žáků.

Rozvoj infromatického myšlení u žáků není důležitý kvůli pokrytí nedostatku IT specialistů na trhu práce, ale z důvodu, že infromatické myšlení je vyžadováno stále více profesemi a lze jej využít mimo jiné i při řešení každodenních situací. Infromatické myšlení totiž pomáhá s osvojením si dovedností jako je řešení problémů, kreativita či schopnost strukturace [2].

Z nové strategie vzdělávání vzešla pak změna rámcového vzdělávacího programu pro základní vzdělávání (dále jako RVP ZV), podle kterého je možné učit od 1. září 2021 a nejzazším termínem pro jeho implementaci je 1. září 2023 (v případě 2. stupně ZŠ je nejzazší termín 1. září 2024) [3]. Zásadní změnou v RVP ZV je nahrazení vzdělávací oblasti Informační a komunikační technologie vzdělávací oblastí Informatika. Tato nová vzdělávací oblast cílí na rozvoj infromatického myšlení žáků a na porozumění principům digitálních technologií na rozdíl od původní oblasti Informační a komunikační technologie, která se věnovala práci s informacemi a ovládání moderních informačních technologií [4].

I přesto, že tato změna byla plánovaná, ponechala učitele bez dostatku metodiky a výukových materiálů. „Učitelé cítí výraznou absenci systémového rámce, který by jim poskytl potřebné ‘know-how’ pro realizaci“ výuky algoritmizace a programování [5]. To je logické vzhledem k tomu, že velká část učitelů 1. stupně ZŠ se ve své pregraduální přípravě nesetkala s touto vzdělávací oblastí, a je tak postavena před výzvou, jak tuto oblast zapracovat do školních vzdělávacích plánů a do svých vyučovacích hodin.

Možným řešením je edukační robotika, která je brána za jeden z nejslibnějších nástrojů pro rozvoj infromatického myšlení a programování. Její výhodou je, že dokáže žáky zaujmout, což podporuje jejich motivaci při výuce a že umožňuje vizualizaci vyučované látky [6]. Na trhu je nyní možné pořídit širokou škálu robotických pomůcek – Bee-Bot<sup>1</sup>, mBot<sup>2</sup>, Sphero<sup>3</sup>, Lego WeDo 2.0<sup>4</sup>, Lego Mind-

<sup>1</sup><https://www.tts-international.com/bee-bot-programmable-floor-robot/1015268.html>

<sup>2</sup><https://www.makeblock.com/steam-kits/mbot>

<sup>3</sup><https://sphero.com/>

<sup>4</sup><https://education.lego.com/en-us/products/lego-education-wedo-2-0-core-set/45300#wedo-20>

storms<sup>5</sup>, Ozobot Bit, Ozobot Evo<sup>6</sup>, Micro:bit<sup>7</sup>, Arduino<sup>8</sup> a spoustu dalších. Každá z těchto robotických pomůcek pokrývá různou oblast infromatického vzdělávání a je koncipována pro různé věkové kategorie a úrovně infromatického myšlení. Například Bee-Bot je robotická včelka, která má na sobě pár tlačítek (především tlačítka k udání směru) a funguje zcela bez potřeby počítače. Využívá se především pro výuku dětí v mateřských školách a v prvních třídách na základní škole. Kdežto v případě Lego Mindstorms se jedná o konstrukční stavebnici, obsahující motory, různé senzory a další součástky, z nichž lze sestavit robota. K Lego Mindstorms je nutné mít počítač, na kterém si žák naprogramuje chování postaveného robota. Stavebnice je využívána spíše pro výuku starších dětí na 2. stupni ZŠ a středních školách.

Na základě výše uvedeného se tato práce věnuje výuce algoritmizace a programování na 1. stupni ZŠ. Algoritmizace a programování jsou dva hlavní pilíře, na kterých stojí nově přidaný důraz na infromatické myšlení v rámci změn RVP pro základní vzdělávání [6]. Nejvhodnější robotickou pomůckou pro výuku těchto pilířů se jeví robot Ozobot Evo. Ten umožňuje dva hlavní způsoby ovládní – programování pomocí barevných kódů a pomocí blokového programovacího jazyka OzoBlockly. Robot je vhodný pro děti od 8 let a je možné ho využít i v jiných školních předmětech.

První část práce se věnuje teoretickému zakotvení, které slouží jako odrazový můstek pro pochopení cílů praktické části. Teoretická část vysvětluje, co přesně je RVP ZV, jaké změny v něm byly provedeny a co si představit pod infromatickým myšlením. Práce dále popisuje psychologické a pedagogické teorie, o které se opírá praktická – druhá – část práce. V neposlední řadě jsou pak zmíněny výzkumy a studie, které se zabývaly výukou infromatiky a využitím robotických pomůcek na školách.

Druhá část práce popisuje cíle práce, fungování robota Ozobot Evo a vznik sbírky úloh pro žáky 1. stupně ZŠ s metodickou příručkou pro učitele. V závěru je výsledná práce shrnuta spolu se svými limity.

---

<sup>5</sup><https://www.lego.com/cs-cz/themes/mindstorms/about>

<sup>6</sup><https://ozobot.com/>

<sup>7</sup><https://microbit.org/>

<sup>8</sup><https://www.arduino.cc/>



## 2 Teoretická část

### 2.1 Rámcový vzdělávací program pro základní vzdělávání 2021

Rámcový vzdělávací program vymezuje rámec vzdělávání pro jednotlivé typy vzdělávání – předškolní, základní a střední. Podle tohoto rámce si pak každá škola vytváří svůj vlastní školní vzdělávací plán (ŠVP).

„RVP ZV vymezuje vše, co je společné a nezbytné v povinném základním vzdělávání žáků, včetně vzdělávání v odpovídajících ročnících víceletých středních škol.“ [4] Také se snaží o komplexní přístup k vytváření vzdělávacího obsahu, tak aby se v něm objevovaly různé metody, formy výuky, vzdělávací postupy podle potřeb žáků, a aby samotný vzdělávací obsah, rozsah a zaměření výuky mohlo být modifikováno.

RVP ZV popisuje samotný koncept základního vzdělávání – jak jej lze chápat a jaké jsou jeho cíle. To, co je důležité, je, aby žákům byla dána možnost zažívat úspěch, nebát se chyby a možnost s chybou pracovat. RVP ZV popisuje mimo jiné i cíle základního vzdělávání. Jedním z nich je například podněcování žáků k tvořivému myšlení, logickému uvažování a k řešení problémů [4].

#### 2.1.1 Struktura RVP ZV

RVP ZV je orientačně rozděleno do devíti vzdělávacích oblastí. Každá oblast má vymezenou charakteristiku, týkající se jejího postavení a významu v základním vzdělávání. Každá vzdělávací oblast má popsáno cílové zaměření, jakých klíčových kompetencí má žák dosáhnout, jaké jsou očekávané výstupy, co by mělo být obsaženo v rámci učiva a je rozdělena na vzdělávací obory.

Očekávané výstupy jsou prakticky zaměřené, využitelné v běžném životě a ověřitelné. V případě 1. stupně ZŠ vymezují předpokládanou způsobilost žáků využívat osvojené učivo na konci 5. třídy. Tyto výstupy jsou závazné pro formulování výstupů v učebních osnovách ve ŠVP.

Učivo v RVP ZV zahrnuje tematické okruhy a slouží jako prostředek k dosažení očekávaných výstupů. Toto vymezení je školám pouze doporučeno. Záleží na samotné základní škole, jak se rozhodne učivo zapracovat do ŠVP. Avšak na úrovni ŠVP se již učivo stává závazným.

Jelikož u mladších žáků jsou značné vývojové rozdíly s každým přibývajícím rokem dítěte, tak v rámci 1. stupně je vzdělávací obsah členěn na 1. období (1. až 3. třída) a 2. období (4. až 5. třída).

V roce 2021 prošel RVP ZV změnami. Tou největší bylo nahrazení vzdělávací oblasti Informační a komunikační technologie oblastí Informatika [4].

### 2.1.2 Informatika (vzdělávací oblast)

Hlavním zaměřením vzdělávací oblasti Informatika je rozvoj inforatického myšlení a porozumění základním principům digitálních technologií [4].

Na 1. stupni ZŠ si žáci vytvářejí představy o způsobech, jakými je možné data a informace zaznamenávat, a to skrze hry, experimenty, diskuze a další aktivity. Postupně rozvíjejí schopnost popsat problém, analyzovat ho a najít jeho řešení. Ve vhodném programovacím prostředí si ověřují algoritmické postupy. Dále si vytvářejí základ pro pochopení inforatických konceptů prostřednictvím digitálních technologií. V neposlední řadě se pak učí bezpečnému zacházení s technologiemi a osvojení návyků, které vedou k prevenci rizikového chování.

Oblast vede žáka mimo jiné ke zkušenosti, že týmová práce spolu s technologiemi může vést k lepším výsledkům než samostatná práce, ke komunikaci pomocí formálních jazyků, kterým porozumí i stroje a mnohým dalším kompetencím.

Informatika obsahuje čtyři tematické okruhy:

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

Rámcový učební plán, který je součástí RVP ZV, požaduje minimální časovou dotaci 2 vyučovacích hodin týdně pro 1. stupeň ZŠ. Dále stanovuje, že Informatika musí být minimálně realizovaná ve čtvrté a páté třídě. Z toho vyplývá, že každá základní škola musí mít alespoň 1 vyučovací hodinu týdně ve čtvrté a následně v páté třídě, což odpovídá dle členění 2. období 1. stupně. Stejně tak jsou očekávané výstupy uvedeny pouze pro 2. období 1. stupně a pro 2. stupeň ZŠ. Zavedení inforatických předmětů do ŠVP pro mladší žáky (1. období 1. stupně) je nad rámec minimálních požadavků RVP ZV [4]. Nejnáročnějším tematickým okruhem, který obsahuje nejvíce učiva a má nejvíc očekávaných výstupů, je algoritmizace a programování.

**Algoritmizace a programování (tematický okruh)** Tematický okruh algoritmizace a programování očekává u žáka na konci 5. třídy, že bude schopen sestavit si a otestovat si symbolické zápisy postupů, dokáže popsat jednoduchý problém, sestaví program v blokově orientovaném programovacím jazyce s využitím cyklů a dokáže ověřit správnost postupu se schopností jej případně opravit. Obsahem učiva by tak mělo být řešení problému krokováním v různých formách zápisu, programování a kontrola řešení, ať už ověřením při opakovaném spuštění programu nebo diskusí a porovnáním postupu se spolužáky [4].

## 2.2 Informatické myšlení

Jak již bylo zmíněno, Strategie digitálního vzdělávání do roku 2020 klade důraz na rozvoj informatického myšlení, což se odráží i v popisu vzdělávací oblasti Informatika v RVP ZV. Toto myšlení je vhodné u žáků rozvíjet, protože informatické myšlení „rozvíjí schopnost analyzovat a syntetizovat, zevšeobecňovat, hledat vhodné strategie řešení problémů a ověřovat je v praxi. Vede k přesnému vyjadřování myšlenek a postupů a jejich zaznamenání ve formálních zápisech, které slouží jako všeobecný prostředek komunikace“ [2].

Jak uvádí Wing [7], jedná se o fundamentální schopnost pro všechny, nikoliv pouze pro počítačové vědce. Ke každodenním činnostem lze totiž analogicky najít činnost, se kterou se setkáváme v počítačových vědách. Příkladem může být dítě, které ztratí svoji rukavici a rodič mu doporučí, aby se vrátil krok za krokem zpátky, aby rukavici našel. To Wing srovnává k backtrackingu v počítačové vědě. Informatické myšlení bude důležité i kvůli tomu, že do lidských životů víc a víc prostupují různé digitální technologie (ať už jako hardware či software). Avšak tím zásadním budou právě informatické koncepty, které se stanou důležitou součástí lidských životů. Tyto koncepty pomůžou lidem řešit problémy, vést své každodenní životy a komunikovat a interagovat s ostatními. Díky tomu se budou lépe vypořádávat s výzvami 21. století [8].

Pro informatické myšlení jsou zároveň důležité jisté dispozice člověka. Jsou to dispozice, které však informatické myšlení zároveň posiluje. Jedná se o:

- sebevědomí při řešení složitostí,
- vytrvalost při práci s obtížnými problémy,
- toleranci k nejednoznačnosti,
- schopnost řešit otevřené problémy,
- schopnost komunikovat a spolupracovat s ostatními na dosažení společného cíle nebo řešení.

Všichni potřebujeme pochopit, jak, kdy a kde mohou počítače a další digitální nástroje pomoci člověku řešit problémy, a člověk potřebuje vědět, jak komunikovat s ostatními, kteří mohou pomoci s řešením podporovaným počítačem. To vše jsou důvody, proč je tak zásadní rozvíjet informatické myšlení už u dětí mladšího věku [9].

### 2.2.1 Algoritmizace a programování

Jednou z částí informatického myšlení je algoritmizace a programování. Výsledkem algoritmizace je algoritmus, což je přesný popis procesu, který má na vstupu nějakou sadu hodnot a vede k výsledku. Na algoritmus je možné také nahlížet jako na posloupnost kroků, které vedou k vyřešení problému.

Člověk při řešení problému (v rámci algoritmizace) prochází několika fázemi. Tyto fáze nejdou nutně v posloupnosti za sebou, ale jsou spolu úzce provázané. Jedná se o:

- analýzu problému,
- alternativní úvahu při analýze problému,
- volbu vhodného přístupu k řešení,
- rozložení problému na dílčí úkoly,
- vytvoření algoritmu po dílčích úkolech,
- ověření správnosti algoritmu,
- ověření efektivity algoritmu a
- reflexi pro budoucího řešení problémů [10, 11].

Ve chvíli, kdy je algoritmus přepsán do programovacího jazyka, tak se jedná o programování. Digitální zařízení aplikují algoritmy na data prostřednictvím programování. Programování je však mnohem víc. Zahrnuje i testování programu a při nalezení chyby její opravení [12].

U žáků je vhodné začít výuku nejprve od samotné algoritmizace – pochopit základy řešení problémů – a teprve pak zapojit programování, které je s algoritmizací úzce spojeno. Ve chvíli, kdy žák získá schopnosti spojené s algoritmizací a programováním, zlepší se i jeho logické myšlení, matematické schopnosti a kritické myšlení [13].

### 2.3 Myšlení a jeho vývoj u dětí prvního stupně ZŠ

Při výuce žáků prvního stupně ZŠ je však nutné brát ohled i na jejich vývojové stadium myšlení. Obecně myšlení je proces zpracování a využívání informací. Z hlediska psychologie hlavní funkce myšlení je „formování pojmů, rozpoznávání a nacházení vztahů, vyvozování závěrů z výchozích předpokladů (usuzování), řešení problémů a vytváření něčeho nového“ [14].

Účelným mentálním manipulacím s psychickými obsahy, které směřují k řešení teoretických i praktických problémů se říká myšlenkové operace. Tento pojem zavedl do psychologie Jean Piaget. Myšlenkové operace lze dělit na logické a heuristické. Při použití heuristické operace se člověk snaží najít řešení cestou nejmenšího odporu, takřkajíc jde o řešení, které přijde na mysl jako první. Kdežto logické myšlenkové operace se řídí přesnými pravidly, které nelze porušit, pokud chceme správný závěr. Tyto operace se uplatňují právě v matematice, formální logice a řadě dalších přírodních vědách. Mohou být považovány za algoritmy. Z hlediska psychologie je algoritmus „specifický myšlenkový postup vhodný pro řešení určitého typu problémů. (...) Důležitou metakomponentou lidského intelektu je schopnost vyhledat v paměti algoritmus, který se hodí

pro řešení dané úlohy.“ Logické myšlení má však ale i svoji nevýhodu, že klade značné nároky na čas a myšlenkové úsilí. Algoritmizace tak žáka učí logickému a kritickému myšlení [14].

Již výše zmíněný Jean Piaget se věnoval i vývojovým stádiím myšlení, která jsou důležitá pro představu, jaké úkoly jsou žáci ve svém věku schopni řešit. V rámci své práce rozdělil kognitivní vývoj dítěte do čtyř základních fází:

1. senzomotorické stadium (do 2 let)<sup>9</sup> – děti reagují reflexivně a poznávají svět pomocí pohybů a smyslů, ke konci stadia získávají vědomí stálosti objektů,
2. předoperační a symbolické stadium (2 – 7 let) – dítě využívá svou fantazii, používá jazyk a uvažuje egocentricky
3. stadium konkrétních operací (7 – 12 let) – dokáže logicky přemýšlet, vnímá vztahy mezi předměty a chápe stálost vlastností u předmětů
4. stadium formálních operací (od 12 let) – dokáže logicky myslet o abstraktních pojmech

Myšlení žáků na prvním stupni ZŠ odpovídá stadium konkrétních operací. V tomto období se dítě přestává v myšlení opírat pouze o vlastní zkušenost a začíná více uvažovat logicky. Při logickém uvažování postupuje od specifické zkušenosti k zobecnujícímu závěru. Dítě však stále není schopné abstraktního myšlení a nepracuje s hypotetickými koncepty.

Dítě chápe princip konzervace. To znamená, že rozumí stálosti některých vlastností u předmětů, i když mění vzhled. Dítě má schopnost klasifikace, která mu umožňuje strukturovat informace, a schopnost seriality, což je schopnost uspořádat objekty v logické sekvenci, a to následně umožňuje dítěti pracovat podle návodu, orientovat se v čase a pracovat systematicky. Pro toto stadium je také typická tranzitivní inference, jakožto schopnost logicky porovnat dva prvky skrze porovnání s třetím prvkem.

Stěžejním aspektem této fáze je však reverzibilita a decentrace. Ve chvíli, kdy si dítě osvojí reverzibilitu, tak dokáže sledovat vlastní myšlenky a vracet se na začátek procesu řešení problému. Decentrace pak dítěti umožňuje uvažovat při řešení úkolu o více aspektech najednou. Může se mentálně přesouvat z jednoho způsobu klasifikace do druhé a je schopné zaujmout více perspektiv [15].

Rozvoj těchto schopností je důležitý pro osvojení si informatického myšlení a k tomu, aby žák byl schopen úspěšně řešit problémy se správným nebo nejvhodnějším závěrem.

---

<sup>9</sup>věková hranice stadií je pouze orientační

## 2.4 Konstruktivistická teorie učení

Piaget položil i významné základy pro konstruktivistickou pedagogiku. Konstruktivismus „ukazuje dítě jako poznávající subjekt, který ve snaze optimalizovat své výměny s prostředím konstruuje sám sebe a integruje přitom jak výsledky, tak mechanismy svého myšlení“ [16]. V tomto pedagogickém modelu je učení tvořivým činem aktivního dítěte.

Konstruktivistické didaktiky se zakládají na pojmu prekoncept, což lze chápat jako individuální interpretace světa. Žákovy prekoncepty jsou pak v rámci výuky konceptuálně měněny. Nové poznatky se integrují do existujících kontextů, struktur a mozaik, které má žák k dispozici. To celé vede k tomu, že by učitel měl umožnit žákům, aby objevovali sami nové poznatky, pomocí tzv. subjektivní tvořivosti, namísto nabízení poznatků v hotové podobě [17].

U Piageta studoval i významný matematik a informatik Seymour Papert, který na základě Piagetova konstruktivismu<sup>10</sup> dal základy teorii učení konstrukcionismu<sup>11</sup>. Rozdílem mezi teorií Piageta a Paperta je ten, že Papert klade větší důraz na propojení se s prostředím, ve kterém se učení odehrává. Zabýval se více tím, jak se znalost tvoří a transformuje v konkrétních kontextech. Za neefektivnější tak považoval výuku s hmatatelnými předměty.

Jak Piaget, tak Papert, ve svých teoriích popisují způsob, jakým se dítě učí. Piaget se více zabýval zvnitřněním zkušeností, ve kterých je dítě spíše aktivním pozorovatelem. Kdežto Papert se více zabýval situací a kontextem zkušenosti, ve které je žák účastníkem. Základ obou teorií je však konstruktivistický a navzájem se tak skvěle doplňují. Jak by se totiž lidé mohli „poučit ze své zkušenosti, pokud jsou do ní zcela ponořeni. Přichází chvíle, kdy člověk potřebuje převést zážitek do popisu nebo modelu“ [18].

Je důležité, aby žák měl možnost si nové učivo odzkoušet, a to za různých situací, v různých kontextech, s různými prostředky, ale zároveň je nutné, aby měl možnost si získanou znalost zpracovat do svého individuálního vnímání světa.

## 2.5 Informatika a informatické myšlení na základních školách

Ze studie Friebroon-Yesharima a jeho kolegy [19] je možné se dozvědět, že Seymour Papert byl jedním z prvních lidí, kteří navrhovali učit programování již malé děti. Na základních školách je ideální doba s programováním začít, protože učitel učí děti různého pohlaví, pocházející z různých prostředí a sociálních skupin. Navíc roky na základní škole, jsou léta, během kterých si dítě utváří představu o svých schopnostech a dovednostech [20].

Výhodou výuky informatiky a informatického myšlení už na základní škole také je, že děti se rychle učí a utváří si postoj k programování. To však může být i jistou nevýhodou. Děti si mohou vytvořit negativní dojem z informatiky,

---

<sup>10</sup>anglicky “constructivism”

<sup>11</sup>anglicky “constructionism”

ať už kvůli nejistotě ke svým vlastním schopnostem ohledně inforatického myšlení nebo nedostatečné připravenosti učitele. Je tedy nutné zvolit přístup, který je adekvátní věku dítěte, a se kterým je učitel ztotožněn [19].

Běžným přístupem, jak mladším dětem přiblížit programovací koncepty, je za využití tzv. „unplugged“ aktivit. To jsou aktivity, které nezahrnují použití počítače. Jedná se o metodu, kdy se využívají hmatatelné věci, s kterými mohou žáci manipulovat. Sami žáci se tak mohou chovat jako roboti, kteří postupně vykonávají zadané instrukce nebo žáci mohou dokreslovat obrázek dle instrukcí za použití papíru a fix [21].

Žáci základních škol 1. stupně jsou však schopni pracovat i se zjednodušenými programovacími jazyky, např. blokové programování Scratch<sup>12</sup>, Blockly<sup>13</sup> nebo OzoBlockly. Není tak důvod, proč programování do výuky nezahrnovat, pokud se použije nástroj, který dokáže upoutat dětskou pozornost a je navržen pro používání mladšími dětmi. Stejně tak samotná výuka algoritmů by měla probíhat skrze hry a motivační úkoly vhodné pro mladší děti [22].

To, že například žáci druhé třídy jsou schopni pracovat s blokovým programovacím jazykem Scratch, se potvrdilo i ve studiích, které zmiňuje ve své práci Friebronn-Yesharim s kolegy [19].

Problém nastává však spíše na straně učitele. Učitelé často nemají dostatek zkušeností s učením programování nebo jim chybí metodické zázemí k jeho výuce. Rambousek se svými kolegy [23] udělal před deseti lety průzkum stavu inforatického předmětu na základních školách (oba stupně ZŠ). Finálně se průzkumu zúčastnilo 84 učitelů inforatických předmětů z 3500 oslovených. Ukázalo se, že učitelé považovali za důležitá témata inforaticky, která sami ovládají nejlépe, kdežto témata, která jim nejsou tolik vlastní, tak považují za nepodstatná. A právě téma algoritmizace a programování se ukázalo u většiny respondentů jako téma nejméně důležité a téma, které by dle oslovených učitelů stačilo vyučovat až na středních školách.

Za posledních deset let nedošlo z hlediska postojů učitelů k velkému posunu. Většina učitelů 2. stupně má aktuálně obavy z vlastní nepřipravenosti na výuku algoritmizace a programování a dle výzkumu Klementa a Dragona [6] by potřebovali v oblasti podpory především metodické materiály pro učitele a výukové materiály pro žáky. Přednáška Stanislava Hoška [24], založená na jeho výzkumném šetření, potvrzuje, že i učitelé 1. stupně zaujímají negativní stanovisko vůči změnám obsahu výuky inforaticky, především pak učitelé s více jak dvacetiletou praxí.

Učitelé se stále vrací k tendenci zařazovat do výuky především práci s kancelářskými balíky, daty a informacemi, což je pravděpodobně pozůstatek z předchozího nastavení vzdělávací politiky v oblasti inforatických a komunikačních technologií, a to i přesto, že nová digitální doba poskytuje mnoho skvělých pomůcek, kterými lze výuku zpestřit – jako například různé robotické hračky [25].

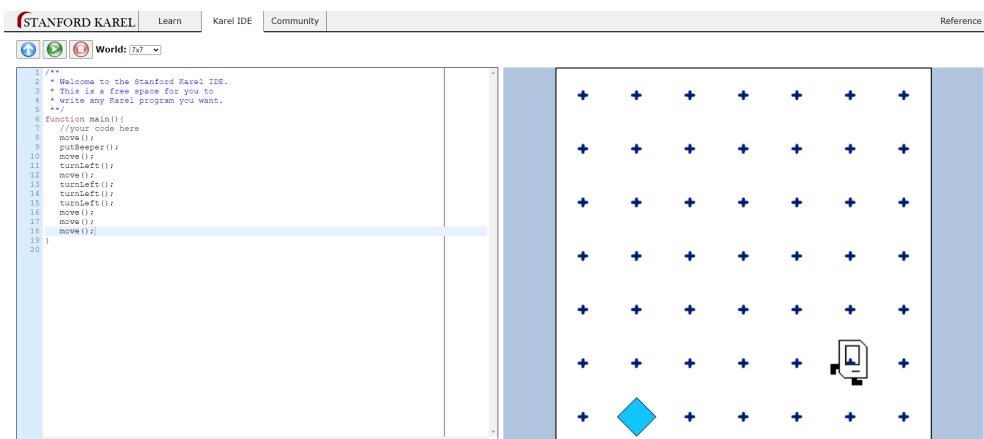
---

<sup>12</sup><https://scratch.mit.edu/>

<sup>13</sup><https://developers.google.com/blockly>

## 2.5.1 Robotika ve výuce

Od 80. let 20. století se začaly vyvíjet vzdělávací programy, které zahrnovaly robotiku. Jedním z takových prvních nástrojů, který si pohrával s myšlenkou ovládnutí robota, byl programovací jazyk Karel. Robot Karel se pohybuje po mapě (obrazovce) za pomoci pár základních příkazů, které uživatel píše do konzole, jak je možné vidět na obrázku 1. Od té doby však vzniklo spousta nových výukových programovacích jazyků a hmatatelných robotů, které lze fyzicky ovládat pomocí jejich tlačítek, například Bee-Bot nebo je lze programovat pomocí počítače, například Lego Mindstorms.



Obrázek 1: Screenshot Karel IDE ze stránek Standfordské univerzity [26]

Výhodou používání robotů ve výuce je, že umožňuje zhmotnit abstraktní chování algoritmů a programů. Robotika poskytuje praktické zkušenosti s problémy v reálném světě a dokáže snížit strach z neznámých konceptů, se kterým se žáci mohou setkat. Výhod robotiky je však daleko více. Robotika ovlivňuje sebedůvěru žáků ve vlastní schopnosti či pozitivní přístup k technickým předmětům z oblasti vědy, technologie, inženýrství a matematiky. Žáci, kteří se setkají s robotikou, si daleko častěji vyberou na střední škole technický předmět než žáci, kteří s robotikou do kontaktu nepřišli [19].

Lopez a jeho tým [13] na základě svého výzkumu také doporučují zavedení robotiky a blokového programování do základního vzdělávání, protože tyto koncepty zvyšují motivaci a nadšení žáků při výuce, což vede ke zlepšení v pochopení inženýrských konceptů.

I v případě výuky s roboty je nutné dbát ohled na věkovou kategorii žáků. Starší žáci jsou schopni porozumět a aplikovat více inženýrských konceptů (cykly, parametry, podmínky, sekvence) na rozdíl od mladších dětí. Zároveň každý robot má různý způsob, jak se ovládá a jak s ním lze pracovat. Výzkumy však ukazují, že robotické hračky lze ve výuce využívat od 3 let [19].

Jak lze vidět, robotika je při výuce vhodná jak pro rozvoj osobnosti žáka, tak pro jeho rozvoj logického myšlení, tvořivosti, systematickosti a týmové spolupráce



[27]. Kromě rozvoje žáka však edukační robotika také nenahraditelnou pomůckou pro vizualizaci vyučované látky [6]. Jedním z takových edukačních robotů je Ozobot.

### 2.5.2 Využití robotů Ozobot ve výuce

Körber, Bailey a další [21] provedly v letech 2018 až 2019 dvouhodinové workshopy programování s roboty na základních školách prvního stupně. Pro výuku zvolili robota Ozobot Evo. Ozobota Evo zvolili z důvodu, že již v jiných výzkumech se ukázal být dětmi pozitivně přijímán a proto, že představuje vhodný můstek mezi „unplugged“ programováním a běžným programováním na počítači. V rámci workshopů si výzkumnice odnesly několik důležitých poznání.

- Děti se daleko více identifikují s robotem, kterého si pojmenovali.
- Pomocí fix a papíru děti snadno pochopí, jak používat barevné kódy k programování robotů a přechod od kreslení k blokovému programování je snazší.
- Děti jsou z práce s roboty nadšené a neprojevuje se žádný rozdíl mezi chlapci a děvčaty.
- Robot vydrží bez nabíjení přibližně 90 minut, pokud jezdí pouze podle nakreslených čar. Pokud je připojený pomocí Bluetooth k tabletu či počítači jeho baterie rychle klesá.

Nadšení z Ozobotů u žáků popisuje i Picka [28] ve svém průzkumu. Picka zjišťoval, jaké jsou zkušenosti s Ozobotem v českých školách. Odpovědi získal od devíti českých učitelů s různě dlouhou profesní praxí, kteří vyučovali různé předměty. Všichni však byli učitelé prvního stupně nebo informatiky. Učitelé si při používání Ozobotů všimli, že i žáci 1. tříd, kteří se běžně soustavnou prací rychleji unaví, tak se dokážou daleko déle soustředit. Dále si učitelé pochvalovali, že Ozobot je jednoduchý na ovládání i na přípravu v porovnání třeba s Lego Mindstorms. Učitelé Ozobota používali i v jiných předmětech než pouze v informatice. Při jeho používání však dítě vždy potřebovalo zapojit informatické myšlení a využít algoritmicizaci. Učitelé také popisovali potřebu diverzifikace výuky při používání Ozobota. Schopnější žáci totiž měli úkoly velice rychle hotové, zatímco ostatní potřebovali více času. Z toho důvodu je důležité, aby učitel měl na každou takovou hodinu připravené různé varianty úkolů pro různé výkonnostní skupiny studentů.

Fojtík [29] udělal případovou studii, ve které se snažil zjistit, jak děti různých věkových kategorií reagují na výuku s Ozobotem. V této studii použil nižší verzi Ozobota, a to Ozobot 2.0. Bit. Ukázalo se, že děti velice rychle dokázaly pochopit, jak Ozobota ovládat. Díky tomu, že Ozobot se pohybuje na základě nakreslené čáry na papíře, tak se u dětí zlepšily i jejich motorické schopnosti. Některé děti však měli problém s kreslením čar, resp. Ozobot nebyl vždy schopen správně přečíst barevné kódy, které nakreslily. Problém byl s požadovanou

šířkou čáry a při přechodu z jedné barvy na druhou děti vytvářely prázdná (bílá) místa. Nejzajímavějším zjištěním této studie ale bylo, že skupina chlapců ve věku 11 až 17 let, která nikdy ani ve svém volném čase neprojevovala zájem o programování a robotiku, vykazovala vysokou míru motivace při práci s Ozobotem.

## 3 Praktická část

### 3.1 Cíle práce

S novým rámcovým vzdělávacím programem musí učitelé na základních školách začlenit do svých učebních plánů tematický okruh algoritmizace a programování. Jedná se o okruh, ve kterém si značná část učitelů není jistá a chybí jim jak dostatek výukových, tak metodických materiálů. Tato práce si klade za cíl podpořit učitele základních škol, a to především učitele 1. stupně, kteří nemají v rámci své aprobace specializovaný předmět. Většina z nich se totiž během své pregraduální přípravy na učitele 1. stupně nesetkala s infromatickým předmětem v takové formě, v jaké se dnes očekává jeho výuka.

Cílem práce je poskytnout učitelům sbírku úloh, která pokryje učivo okruhu algoritmizace a programování, a metodickou příručku ke sbírce úloh, která jim poskytne vhled do úloh a možnosti, jak dané úlohy s žáky řešit.

Úlohy budou přizpůsobeny věku žáků 2. období 1. stupně ZŠ, ve kterém je povinné předmět informatika vyučovat, což je věková kategorie 9 až 11 let. Kvůli tomu, že děti mladšího školního věku nemají ještě rozvinuté abstraktní myšlení, budou úlohy postaveny za využití robota Ozobot Evo. Ozobot Evo se zdá být nejvhodnějším robotem pro tuto práci z vícero důvodů.

- Jedná se o robota, který odpovídá věkové kategorii žáků, ale lze ho využít i pro starší žáky.
- Z výzkumů se ukazuje, že je pozitivně přijímán žáky i učiteli.
- Má dvojí způsob ovládání / programování.
- Je cenově dostupný a lze jej využívat i v jiných předmětech než pouze v předmětu informatiky.

Cílem je, aby sbírka úloh byla dostačující pro praktické naplnění plánu. Zároveň ale budou úlohy vytvořeny nezávisle na úlohách jiných, tak aby učitel mohl využít všechny nebo pouze některé úlohy – dle svého uvážení a vytvořeného učebního plánu.

V metodické příručce bude popsáno, které očekávané výstupy z RVP úloha pokrývá, řešení úlohy a případně další metodické poznámky. U některých úloh bude popsáno, jak úlohu lze zjednodušit či ztížit. Metodická příručka představí základní vlastnosti Ozobota Evo, ale nebude obsahovat podrobnější návod, jak se s Ozobotem pracuje, protože ten je zahrnutý v balení spolu s robotem.

Cílem této práce není vytvořit učební plán pro pokrytí tématu algoritmizace a programování. Ze strany učitele bude stále nutné doplnit teoretický rámec učiva. Úlohy a metodická příručka mají sloužit učiteli jako praktická pomůcka, kterou může propojit s teorií, která umožní žákům látku lépe pochopit. Cílem práce ani není dozdělení učitelů. Pokud si učitel chce být při výuce jistější, jak pracovat s žáky v předmětu informatika, tak doporučením je absolvovat některé ze školení v rámci DVPP (dalšího vzdělávání pedagogických pracovníků).

## 3.2 Použité metody

### 3.2.1 Ozobot Evo

Společnost Ozobot vytvořila dva typy robotů – Ozobot Bit a Ozobot Evo. Ozobot Bit byl vydán v roce 2014 a nemá Bluetooth připojení. Ozobot Evo byl uveden na trh v roce 2016 a na rozdíl od Bitu má možnost Bluetooth připojení, senzory vzdálenosti a reproduktor.

Oba Ozoboti přicházejí se dvěma způsoby kódování – bez obrazovky s barevnými kódy a na obrazovce s OzoBlockly. Kód OzoBlockly lze okamžitě načíst do Ozobota Evo přes Bluetooth nebo přes flashové načtení, při kterém musí být Ozobot přidržován k obrazovce, na které se rychle střídají barvy: červená, zelená, modrá, tyrkysová, fialová a žlutá [30].

Společnost Ozobot poskytuje širokou škálu podpůrných úloh, metodik a aplikací. Jejich nevýhodou bohužel je, že jsou pouze v anglickém jazyce. Velkým pomocníkem může být právě aplikace Ozobot Classroom<sup>14</sup>, která poskytuje aktualizace Ozobota a učitel zde nalezne různé úlohy s možností vytvářet i své vlastní.

Samotný Ozobot Evo je malý – má přibližně 3,5 cm na 3,5 cm. Na přední části má 5 LED diod a reproduktor pro vydávání zvuků Ozobota. Vpředu jsou také dva senzory vzdálenosti a stejně tak jsou dva vzadu. Vzadu je pak ještě nabíjecí port, fungující i jako USB připojení. Z boku je tlačítko pro zapnutí/vypnutí Ozobota. Zespodu jsou senzory pro detekci barev a čar a na vrchu Ozobota je Bluetooth anténa a 1 LED dioda. Ozobot se pohybuje pomocí dvou koleček. Grafické znázornění Ozobota lze vidět na obrázku 2.



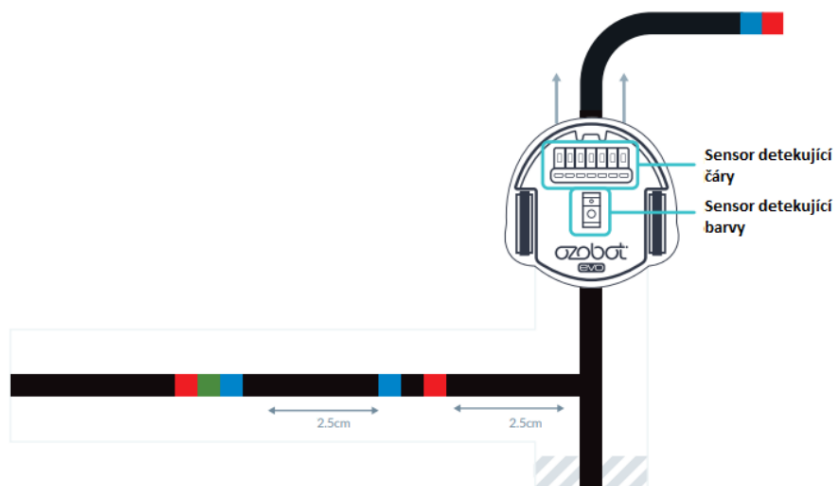
Obrázek 2: Nákres Ozobota Evo z českého manuálu [31]

<sup>14</sup>lze přeložit jako Ozobotí třída

**Barevné kódy** Primární funkcí Ozobota je sledovat 4 až 7 mm širokou černou čáru a rozpoznat barevné kódy, kterou jsou na ní umístěné. „Pokud narazí na křižovatku, vydá se náhodně vybraným směrem. Kromě černé barvy umí robot rozoznat červenou, zelenou a modrou. Různou kombinací těchto barev v přesném pořadí za sebou se vytvoří příkaz, kterým můžeme například naříditi robotovi přesný směr na příští křižovatce.“ [30].

Při koupi Ozobota, obdrží člověk jak samotného robota, tak i potřebné fixy. Je však možné využít i jiné fixy, které mají tlustou špičku ve tvaru dláta. Nedoporučují se pastelky a zvýrazňovače, protože nedokážou vytvořit plnou a plynulou širokou čáru, tak aby ji senzor rozpoznal.

Při kreslení cest je důležité brát ohled nejen na tloušťku a plnost čáry, ale také na zakřivení cest. Cesty je možné zalamovat od úhlu  $45^\circ$ . Menší úhel nemusí být pro Ozobota čitelným. Pokud jsou vedle sebe dvě paralelní cesty, tak by od sebe měly být vzdálené minimálně 2,5 cm. Stejná vzdálenost platí i pro jednotlivé barevné kódy, které jdou po sobě a pro vzdálenost barevného kódu od křižovatky. Některé barevné kódy však potřebují mít za sebou větší prostor, než je 2,5 cm – např. pohyb cik-cak potřebuje mít za sebou nejvíce místa, a to 6,35 cm. Dále žádný barevný kód by se neměl objevovat na křižovatce a ani v zatáčce. V neposlední řadě na cestě, která končí s barevným kódem, ukončující jízdu Ozobota, je nutné ponechat za čárou alespoň 2 cm prázdného místa, protože Ozobot přejede za konec čáry. Příklad, jak taková cesta s barevnými kódy vypadá, je na obrázku 3.



Obrázek 3: Příklad cesty s barevnými kódy [32]

Samotné barevné kódy jsou krátké sekvence barev, které umožňují kódovat zcela bez počítače. Barevný kód se skládá ze dvou až čtyř barev, které se mohou opakovat. Při přípravě materiálů v počítači je vhodné využít pro červenou barvu hexadecimální kód #ed1c24, pro modrou barvu #0083ca a pro zelenou barvu #48b749. Každá barva v barevném kódu by měla být ve tvaru čtverce. Neměla by se překrývat s jinými barvami, ale zároveň by neměla vznikat bílá místa mezi jednotlivými barvami. Všechny barevné kódy musí být mezi černými čarami [32].

Ozobot umí přečíst 29 barevných kódů, které lze rozdělit do 5 kategorií – rychlost, směr a speciální pohyb, časovače, výhry/ukončení a čítače. Díky tomu lze u Ozobota měnit rychlost jeho jízdy, pro křižovatky mu určit směr jízdy nebo během jízdy mu zadat zvláštní pohyb jako například, aby se dvakrát otočil kolem své osy. Ozobota lze také na 3 sekundy zastavit nebo úspěšně ukončit jeho jízdu [33].

Pokud je na cestě přidán špatný barevný kód je možné ho přelepit dodanými samolepkami s barevnými kódy nebo jednoduše přelepit cestu kouskem bílého papíru a dokreslit cestu s barevným kódem znovu.

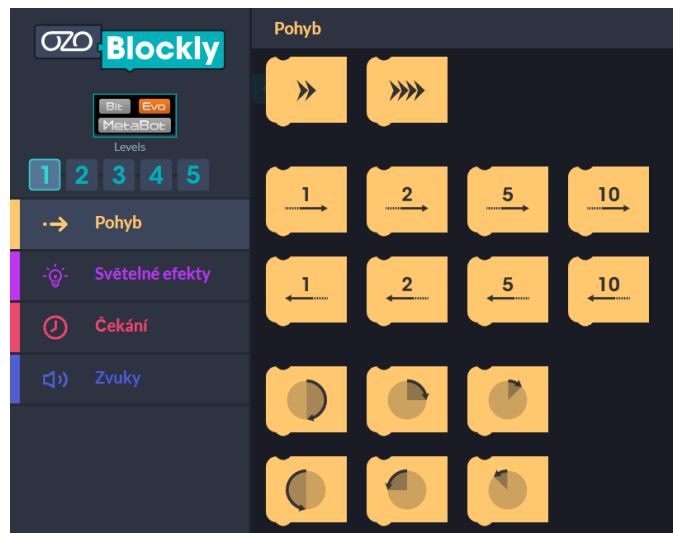
**Blokové programování – OzoBlockly** OzoBlockly je vizuální blokový programovací jazyk, který je založen na javascriptové knihovně Blockly od společnosti Google. Editor OzoBlockly běží na počítačích i tabletech s moderním prohlížečem nebo přes aplikaci Evo by Ozobot. Výhodou tohoto editoru je jeho multijazyčnost a především pak pro české žáky možnost nastavení českého jazyka.

Editor OzoBlockly má spoustu možností. V levé části obsahuje volbu Ozobota – Bit, Evo nebo MetaBota, což je Ozobot Evo v rozšířené realitě, takže jej lze využít s žáky, kteří mají mobilní telefon s fotoaparátem a připojením k internetu. Dále ke každému Ozobotovi lze vybrat úroveň programování. Každá vyšší úroveň odkrývá další složitější bloky kódu, které lze využít při programování. Tyto bloky kódu jsou vkládány pod sebe do střední části editoru. V pravé části editoru lze spustit vytvořený program přes Bluetooth, zobrazit nápovědu k práci s OzoBlockly, nastavit zobrazení editoru nebo také možnost zobrazit si pár připravených příkladů s hotovým řešením. Ve spodní části lze pak nahrát program do Ozobota přes flashové načtení, pokud Ozobot není připojen přes Bluetooth. Vzhled editoru lze vidět na obrázku 4. Co se týče úrovní pro Ozobota Evo, tak první úroveň Nováček je založena čistě na grafických blocích kódu, které je možné vidět na obrázku 5. Prvky, které lze v první úrovni využít jsou pouze z kategorie pohybu, světelných efektů, čekání a zvuků.

Druhá úroveň Začátečník přidává prvky z kategorie pro cykly. Zároveň jsou již bloky kódů uvedeny textem a je možné je více variovat, jak lze vidět na obrázku 6.



Obrázek 4: Screenshot editoru OzoBlockly [34]



Obrázek 5: Úroveň 1 - Nováček



Obrázek 6: Úroveň 2 - Začátečník

Třetí úroveň Středně pokročilý přidává nové kategorie bloky kódů – jízda po čáře, senzory a logika – a zároveň opět ještě více rozšiřuje možnosti jednotlivých bloků. Čtvrtá úroveň Pokročilý umožňuje využít i prvky z kategorie ukončení, matematika, proměnné a funkce. Pátá – poslední – úroveň Mistr umožňuje reakci na tlačítko vypnout/zapnout a práci s poli a seznamy. Vzhledem k věkové kategorii žáků a potřebám učiva bude pro vytvoření úloh využita pouze první a druhá úroveň. Třetí úroveň už je vhodná spíše pro žáky 2. stupně ZŠ.

### 3.3 Sbíрка úloh

Při vytváření úloh s Ozobotem jsem vycházela z teoretického ukotvení. Kladla jsem důraz na stupňování úloh a pokrytí očekávaných výstupů a učiva dle RVP ZV pro 2. období 1. stupně ZŠ. Ačkoliv jsem úlohy vytvářela s možností úlohy používat odděleně, poskládala jsem je tak, aby bylo možné úlohy procházet postupně a žák procvičoval případně již naučené.

Úloh bylo vytvořeno více, ne všechny však měly vhodné zadání pro žáky 1. stupně ZŠ. Dále jsem se potýkala s problémem, kdy vymyšlenou mapu nebylo možné dostat na papír velikosti A4, tak aby po ní Ozobot úspěšně jezdil a měl mezi barevnými kódy a křížovatkami patřičné rozestupy.

Úvodní obálky pro sbírku úloh a metodickou příručku jsem vytvořila v online nástroji Canva<sup>15</sup> s obrázkem získaným přes Obrázky Google<sup>16</sup>.

<sup>15</sup><https://www.canva.com/>

<sup>16</sup><https://images.google.com/>



### 3.3.1 Úlohy s barevnými kódy

První překážkou při vytváření úloh s barevnými kódy, na kterou jsem narazila, byla neexistence nástroje, ve kterém by bylo možné vytvářet mapy pro Ozobota. Z videí a diskuzí na internetu jsem nabyla dojmu, že mám před sebou dva možné způsoby, jak je vytvořit – použít nějaké grafické studio nebo využít tabulkového nástroje jako je Excel [35]. Já jsem pro svoji práci zvolila online grafický editor Photopea<sup>17</sup>, protože jsem chtěla mít možnost doplňovat nejen čáry a barevné kódy, ale i obrázky, které úlohu zpestří a zasadí do příběhu.

Nejprve jsem si připravila čtvercové obrázky o velikosti 300 x 300 pixelů pro rovnou černou cestu, začátek cesty, barevné kódy na černé čáře a prázdná políčka pro vybarvení na černé cestě. Šířka cesty je 50 pixelů. Tyto obrázky jsem pak vkládala na bílé plátno o velikosti papíru A4 (2480 x 3508 pixelů) a skládala z nich jednotlivé cesty. Do úloh jsem pak doplňovala obrázky, které jsem získala ze stránky Pixabay<sup>18</sup>.

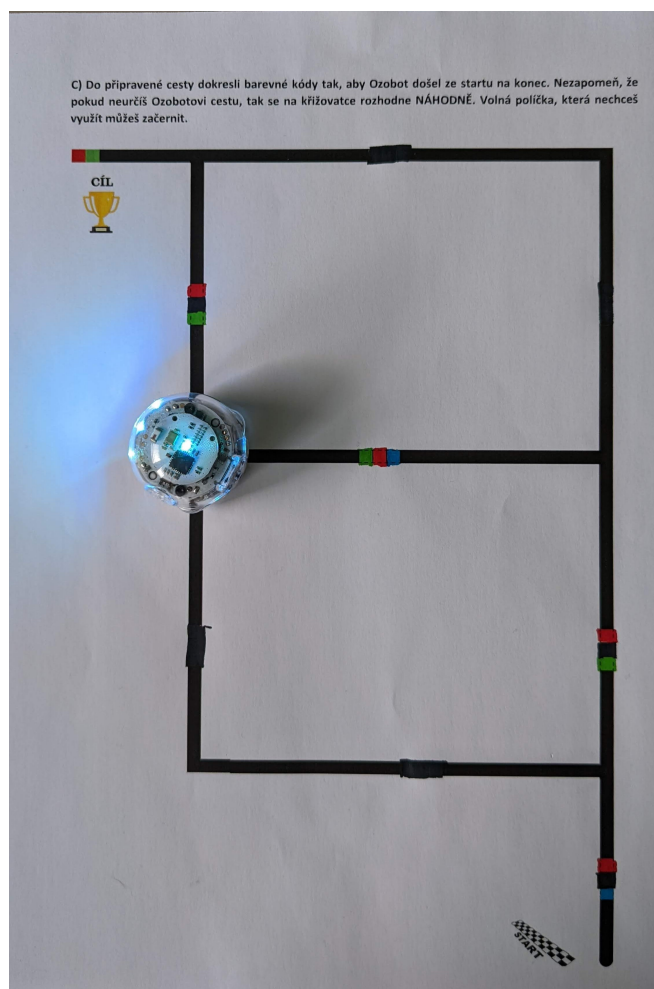
Výsledné obrázky pak byly použity v textovém procesoru Microsoft Word, ve kterém jsem vytvořila finální podobu sbírky úloh. Při vložení obrázku docházelo k jeho oříznutí a posunům při umístění na stránce. Musela jsem však kontrolovat, že čáry na mapě mají alespoň 4 mm na šířku. S tím pomohla možnost zobrazení pravítka v procesoru MS Word. Ještě před tiskem úloh jsem byla schopna ověřit, že čára bude dostatečně široká pro přečtení Ozobotem. Zároveň jsem potřebovala kontrolovat vzdálenost barevných kódů od sebe a od křižovatek. U některých úloh bylo nutné obrázek s mapou předělat, protože Ozobot nebyl schopný na křižovatce správně reagovat při kratších vzdálenostech do 2,5 cm. Všechny úlohy byly otestovány vytisknuté, jak je možné vidět na obrázku 7.

První úloha se věnuje seznámení žáků s Ozoboty a s tím, jak fungují. Na místo běžného předložení tabulky, ve které by si žáci mohli přečíst všechny barevné kódy, které Ozobot umí, je žák vyzván, aby sám na základě pozorování zjistil, co který barevný kód dělá. Tuto metodu jsem zvolila, aby pro žáka byla úloha zajímavější a žák mohl být jakýmsi objevitelem dle konstruktivismu [17]. V úloze nejsou využity všechny barevné kódy, které Ozobot umí. Zahrnula jsem pouze kódy, se kterými žák bude v dalších úlohách pracovat. V poslední části úlohy má žák nově poznané barevné kódy aplikovat, aby své poznání upevnil.

Druhá úloha je již zaměřená na popsání problému a vytvoření algoritmu na základě dostupných možností – fixně daná prázdná místa, která je nutné správně barevně doplnit. Úlohu jsem postavila tak, aby měla pouze jedno řešení a Ozobot mohl jezdit po okruhu pouze jedním správným směrem, a to po směru hodinových ručiček. Pokud žák navrhne řešení, ve kterém by Ozobot měl jezdit v protisměru hodinových ručiček, tak se mu nepodaří dané řešení zakreslit do připravených prázdných políček. Ozobot by se totiž na některých křižovatkách mohl rozhodnout náhodně. V druhé části úlohy je pak kladen důraz na seznámení se s barevnými kódy, a aby žák znal rozdíl mezi jednotlivými rychlostmi,

<sup>17</sup><https://www.photopea.com/>

<sup>18</sup><https://pixabay.com/cs/>



Obrázek 7: Testování úlohy 1 – část C

keré lze do cesty dobarvit. Tuto znalost mohl získat z první úlohy nebo v této úloze z experimentování s barevnými kódy.

Na první dojem se může zdát, že třetí úloha je velice snadná, protože žák doplňuje pouze barevné kódy dle ikonky v tabulce. Úlohu jsem však tvořila tak, aby obsahovala spoustu prázdných políček pro barevné kódy, které nejsou vodorovně a Ozobot je nebude při své cestě číst zleva doprava – tedy tak, jak jsou zakresleny v tabulce. V úloze jsem se snažila, aby si žák vyzkoušel prostorovou představivost a procvičil své úvahy v tom, v jakém pořadí bude Ozobot barevné kódy číst.

Pro čtvrtou úlohu jsem chtěla, aby žák musel pracovat s informacemi v zadání – uvědomit si, kde bude Ozobot startovat, co do kterých prázdných políček doplnit, aby se Ozobot zachoval podle popisu a zároveň uvažovat nad potřebou ošetření náhodných rozhodnutí na některých křižovatkách. Úlohu jsem složila ze dvou částí, aby v první části došlo k poznání a v druhé části k upevnění. V této úloze bylo mimo jiné náročnější, jak vykreslit přechod pro chodce, aby

jej Ozobot nečetl jako čáru, po které má jet. Vhodným řešením se ukázala velmi světle šedá barva (hexadecimálně #f2f2f2), na kterou při testování Ozobot nereagoval.

Pokud učitel bude úlohy brát postupně, tak v této chvíli již žák poznal základní tvoření kódu pomocí čar a barevných kódů a základní úvahy při vytváření postupu k vyřešení problému. Další pátou úlohu jsem tedy zaměřila na testování a hledání chyby. Žák má v této úloze před sebou již hotovou mapu s barevnými kódy, ve které se Ozobot dostane do nekonečné smyčky. V úloze jsem připravila místa, ve kterých může žák změnit barevný kód a teoreticky na základě náhody se Ozobot může dostat do cíle. Jeho řešení však nebude správné, protože v mapě existuje pouze jedno místo s barevným kódem, které zaručuje vždy stejný výsledek, a to dojetí Ozobota k hořícímu domu. Zároveň jsem úlohu postavila tak, aby bylo již možné žáka seznámit s pojmem cyklu a opakování.

Šestá úloha pomůže utvrdit učivo z páté úlohy, nebo naopak žáka seznámit prvotně s testováním. Díky náhodnosti Ozobota na křižovatkách, před kterými není stanovený směr pomocí barevného kódu, vzniká několik možností, jak projet zoo. Učitel může s žáky v této úloze vytvořit orientovaný graf pro průchod Ozobota mapou, kdy vrcholy grafu jsou křižovatky a konce cest. V takovém grafu pak může učitel žákům zobrazit i cykly, které se v mapě skrývají a zároveň si i ukázat, do kterých částí zoo se Ozobot nikdy nemůže dostat. V úloze jsem chtěla, aby žák nejprve vyvodil závěr z dostupných informací a teprve poté ověřil svoji hypotézu testováním. Cílem bylo, aby žák uměl nejen testovat vzniklá řešení, ale porozumět formě zápisu, tak aby věděl, co daný program bude dělat.

V předposlední – sedmé – úloze jsem chtěla zacílit na potřebu spolupráce při vytváření programu. Při řešení problémů se totiž člověk nepodílí nutně na všech fázích řešení problému, a tak výstup z každé fáze musí být natolik srozumitelný, aby s ním mohl pracovat i někdo jiný. V úloze si žák vyzkouší roli navrhovatele a řešitele dle cizího návrhu. Žák bude sestavovat popis svého algoritmu nejprve pomocí symbolů barevných kódů přímo do mapy a následně jej bude muset popsat pomocí textu, který předá spolužákovi. V druhé části pak procvičí sestavení funkčního programu na základě popsaného návrhu (algoritmu) od spolužáka.

Poslední osmá úloha je taktéž o spolupráci se spolužáky. Je však zaměřena na nalezení společného řešení a jeho aplikování pomocí barevných kódů do mapy. První část úlohy lze vyřešit vícero způsoby. Žák si v této části vyzkouší spolupráci na jednodušší úloze – za předpokladu, že žáci objeví jednoduché řešení problému. V druhé části jsem se snažila vytvořit situaci, ve které se Ozoboti na mapě střetnou a bude nutné najít řešení, které tomuto střetnutí zabrání. V této části je nutné daleko větší spolupráce než v první části. Cílem také bylo, aby žáci při používání především směrových barevných kódů neopomněli i na možnost využití barevných kódů, týkajících se například rychlosti jízdy Ozobota.

V rámci úloh s barevnými kódy jsem pokryla tři ze čtyř očekávaných výstupů RVP ZV. Zbývalo mi vytvořit úlohy, které se budou věnovat i blokovému programování, jakožto poslednímu chybějícímu očekávanému výstupu. K tomuto jsem využila blokového programování v OzoBlockly.

### 3.3.2 Úlohy v OzoBlockly

Při vytváření úloh v OzoBlockly jsem se nesečkala s žádnými výraznými problémy jako u úloh s barevnými kódy, u kterých byla potřeba většího ladění. OzoBlockly jsem ve sbírce zařadila až po barevných kódech, aby se žák mohl nejprve seznámit se samotným Ozobotem. Programování pomocí barevných kódů je jednodušší, a jak ukázal výzkum [21], tak přechod z barevných kódů na OzoBlockly je snazší, než začít rovnou s OzoBlockly.

**Úroveň Nováček** Stejně jako u úloh s barevnými kódy jsem první úlohu vytvořila s cílem seznámit žáka s OzoBlockly. Žák má porovnat vždy dva kódy, které jsou si často velice podobné, ale Ozobot vykoná každý kód nepatrně jinak. Žák musí každý kus kódu, zobrazený v úloze, naklikat do editoru OzoBlockly, díky čemuž se naučí základy práce s editorem. Poznává, jak Ozobota připojit pomocí Bluetooth, příp. jak kód načíst pomocí flashového načítání. Po spuštění se pak dozví, co každý blokový kód, reprezentovaný nějakými symboly znamená, což mu ulehčí řešení dalších úloh.

V druhé úloze jsem chtěla dát žákovi prostor pro experimentování a objevování v blokově orientovaném programovacím prostředí, aby ho programování bavilo a mohl využít své kreativity. Rozvoj kreativity je totiž také důležitou součástí při řešení problémů a vytváření algoritmů. Zároveň úloha není postavená na úplně volném zadání a žák tak musí pracovat s danými vstupy – využití všech kategorií bloků, trvání tance (jízdy) alespoň 15 sekund a využití pouze vyznačené plochy. K těmto omezením jsem přistoupila ze dvou důvodů – aby žák uměl pracovat se vstupními podmínkami úloh a také aby žákovo řešení nebylo nejjednodušší možné řešení jako např. použití pouze jednoho bloku pro 5 kroků dopředu.

Třetí úloha se týká sestavení programu na základě stanoveného algoritmu. U počtu kroků či pozastavení Ozobota jsem volila číselné hodnoty, které lze v OzoBlockly nalézt jako jeden blok, ale také hodnoty, které je naopak potřeba složit z vícero bloků. Je vhodné, aby úloha byla doplněna o diskuzi a vysvětlení ze strany učitele, co to je efektivita programu. Žák sice může poskládat za sebou třikrát 1s čekání, ale měl by se naučit využívat, co nejméně bloků, když je to možné. S tímto principem se totiž pak setká v další úrovni při cyklech.

V poslední úloze v úrovni Nováček se symbolickými bloky jsem chtěla, aby došlo k prohloubení experimentování v blokovém programovacím jazyce OzoBlockly a znalosti pro sestavování programů. Při programování číselnic může žák přistoupit k řešení z různých stran. Díky tomu, jednotlivá řešení žáků mohou být odlišná a přesto správná. Učitel má tak možnost využít porovnání postupu mezi žáky a zavést diskuzi o jednotlivých řešeních jako je navrhováno v RVP ZV pro algoritmizaci a programování [4].

**Úroveň Začátečník** Pro prvního úlohu v úrovni Začátečník jsem využila dvou zadání úloh z úrovně pro Nováčky. Díky tomu, že zadání úlohy je popsáno, tak je možné úlohu využít i v případě, kdy žák neprogramoval v úrovni Nováček. Stejně zadání jsem využila z důvodu možnosti porovnat programování mezi úrovní Nováček a Začátečník. Učitel s žáky může probrat i dojmy z nové úrovně – která úroveň programování je pro ně snazší nebo co jim nová úroveň přináší navíc. Vytvoření těchto úloh má samozřejmě také stejné důvody popsané v kap. 3.3.2.

Druhá úloha seznamuje s cykly – přesněji s dvěma bloky pro cyklus, které lze v úrovni Začátečník využít. Tuto úlohu jsem zařadila jako druhou, protože ji lze doplnit k úloze první, ve které mohl žák cyklus využít nebo vytvořit kód, ve kterém jsou bloky, které se opakují, a pro které by cyklus byl vhodný. Zároveň ve třetí úloze se již pracuje s cyklem a je vhodné žáka s blokem cyklu seznámit předtím, pokud se pracuje s celou sbírkou úloh.

Třetí úlohu jsem vytvářela s cílem, aby žák testoval již naprogramovaná řešení i v rámci OzoBlockly. Podstatou úlohy je nalezení chyby a oprava kódu. Žák by v této chvíli měl mít již nějaké zkušenosti s programováním, pokud řešil předešlé úlohy z příručky, takže úlohu jsem vytvořila trochu těžší, aby oprava kódu nebyla pouze oprava jednoho nebo pár bloků. Žák musí totiž opravit značnou část kódu, která následuje po cyklu.

Stejně jako pro úlohy s barevnými kódy jsem vytvořila i pro úlohy v OzoBlockly alespoň jednu úlohu zaměřenou na spolupráci žáků. Na řešení čtvrté úlohy pracuje skupina třech, příp. čtyř žáků, kdy každý žák vytváří svoji část kódu, tak aby jeho Ozobot následoval jednu z barevných cest. Cesty v zadání jsem vytvořila tak, aby se křížily a docházelo případně ke kolizím mezi Ozoboty.

### 3.3.3 Metodická příručka

Po vytvoření úloh jsem ještě vytvořila metodickou příručku. V příručce jsem se snažila doplnit myšlenky, které mě vedly k vytvoření úlohy, a které by mohly být vhodné pro učitele při práci s úlohou. Každá úloha má popis, podporovaný výstup dle RVP ZV [4], náročnost, dodatečné poznámky a řešení.

Kromě popisu a doplnění úloh jsem se rozhodla doplnit i krátké seznámení s Ozobotem, a to především popis chování Ozobota na různých čarách z tipů, které jsou uvedeny na oficiálních Ozobot stránkách, ale jsou pouze v anglickém jazyce [32]. Dále jsem doplnila i doporučení, která se osvědčila mě nebo která jsem se dočetla ve studiích.

Do příručky jsem mimo jiné na závěr doplnila i český překlad tří kategorií barevných kódů, které jsem v úlohách použila – rychlost, směr pohybu a speciální pohyby a časovače –, a kterou jsem vytvořila původně pro třetí úlohu s barevnými kódy. Grafiku této tabulky jsem ponechala stejnou, jak je prezentována na oficiálních Ozobot stránkách [33].

### 3.4 Diskuze

Tematický okruh algoritmizace a programování je nově povinnou součástí základního vzdělávání. Tuto povinnost jsem přetvořila do sbírky 16 úloh, kterou mohou využít učitelé 1. stupně ZŠ, a která pokrývá plně výuku nového okruhu. Vzhledem k tomu, že výuka informatiky musí probíhat minimálně 1 vyučovací hodinu týdně ve 4. a 5. třídě ZŠ, učitel má před sebou přibližně 64 vyučovacích hodin [36], během kterých musí zvládnout žákům předat znalosti a schopnosti ze čtyř tematických okruhů. Pokud vezmeme v úvahu, že největším okruhem je právě algoritmizace a programování a učitel by ji věnoval třetinu dotovaného času, tak na výuku tohoto okruhu má přibližně 21 vyučovacích hodin. S potřebou seznámit žáky se samotným robotem, s teorií a s možností přidání vlastních úloh se zdá výsledný počet 16 úloh dostatečný.

Sbírka úloh s příručkou může také přinést zlepšení postojů učitelů ke změnám ve výuce informatiky. Sbírka totiž rozšiřuje množinu dostupných výukových materiálů a metodických příruček, kterých je pomálu a na jejichž nedostatek si učitelé stěžují [5, 24].

V příručce doporučuji, aby i sám učitel vyzkoušel úlohy, které chce ve výuce použít. Důvodem je nejen jeho příprava na výuku, ale aby si i učitel byl v algoritmizaci a programování jistější a mohl se příp. sám v ní zdokonalovat, pokud se s oblastí prozatím moc nesetkal. Učitelé totiž považují oblasti informatiky, ve kterých nejsou moc zběhlí, za nedůležité [23]. Pokud učitel využije celou sbírku úloh může si být jistý, že pokryl všechny očekávané výstupy a učivo dle RVP ZV [4]. Z hlediska čtyř očekávaných výstupů:

- sestavení a testování symbolických zápisů postupů lze naléznout ve všech úlohách s barevnými kódy a v úlohách v OzoBlockly v úrovni Nováček;
- popsání jednoduchého problému, navrhnutí a popsání jednotlivých kroků řešení lze naléznout v polovině úloh ve obou způsobech programování;
- sestavení programu v blokově orientovaném programovacím jazyce, rozpoznání opakujících se vzorů, používání opakování a připravených podprogramů lze naléznout ve všech úlohách v OzoBlockly;
- ověření správnosti navrženého postupu či programu, nalezení a opravení v něm případnou chybu lze naléznout v části úloh s barevnými kódy a ve většině úloh v OzoBlockly.

V úlohách je využito řešení problému krokováním a sestavováním funkčního postupu řešící danou situaci. Žák v úlohách má možnost experimentovat a objevovat v blokově orientovaném programovacím prostředí OzoBlockly a naučit se, jak pracovat s opakovanými částmi programu. Úlohy jsou také zaměřeny na nalezení chyb a jejich následné opravení. Pro některé úlohy je důležité opakované spuštění pro ověření funkčnosti programu. Všechny tyto činnosti jsou prostředky navrhovaného učiva v RVP ZV.

### 3.4.1 Náměty pro další práce

Další práce věnující se tématu edukační robotiky a Ozobotům by mohla být zaměřena na vývoj aplikace pro vytváření map s barevnými kódy pro Ozoboty, ve které by uživatel byl schopný zvolit velikost papíru, na který se úlohy budou tisknout. Na základě velikosti by byla přizpůsobena velikost čar, tak aby byla pro Ozobota vždy čitelná. Dále by aplikace obsahovala už připravenou sadu prvků, ze kterých by bylo možné vytvořit mapu doplněnou barevnými kódy, ale i mapu s prázdnými místy, připravenými pro doplnění v rámci úlohy. Aplikace by kontrolovala i parametry uvedené na oficiálních stránkách Ozobota v tipech pro kreslení čar a barevných kódů [32].

Vhodným rozšířením této práce je také využití Ozobotů nejen pro výuku algoritmizace a programování, ale pro další vzdělávací oblasti Informatiky, tak aby škola využila Ozobota naplno. Žák může vytvářet modely, ve kterých bude vystupovat naprogramovaný Ozobot a následně popisovat, s jakými daty Ozobot pracuje. Nejširší využití má Ozobot v algoritmizaci a programování, ale má potenciál být hravým pomocníkem pro výuku i jiných témat.

### 3.4.2 Limity práce

Limitem sbírky úloh a obecně využívání edukační robotiky ve výuce je nutnost mít přístup k dostatku robotům. Pokud se podíváme na nedávné události, kdy na jaře 2020 zasáhla celý svět pandemie koronavirové choroby covid-19 a veškerá výuka žáků se přesunula do formátu dálkové výuky, tak tato sbírka není schopna v dálkové výuce plně obstát. Pouze některé úlohy v OzoBlockly je možné využít pro dálkovou výuku, a to jen za předpokladu, že každý žák má přístup k chytrému telefonu, což by ve většině případů neměl být problém, ale je potřeba na tento faktor myslet. S chytrým telefonem je totiž možné programovat v OzoBlockly a spouštět jízdu Ozobota přes MetaBota. MetaBot je Ozobot v rozšířené realitě, kterého je možné si zobrazit např. na svém pracovním stole pomocí fotoaparátu v mobilu, jak je zobrazeno v obrázku 8.



Obrázek 8: Screenshot mobilní obrazovky při spuštění MetaBota



## Závěr

Výsledkem práce je sbírka úloh Programování s Ozobotem, ke které vznikla i metodická příručka pro učitele. Sbírnka úloh obsahuje celkem 16 úloh, ve kterých se žák učí programovat pomocí barevných kódů na černých čarách a pomocí blokového programování v OzoBlockly. Sbírnka úloh poskytuje učitelům potřebný výukový materiál, ale nezajišťuje jejich potřebné dozdělení v oblasti teorie algoritmizace a programování, které jim ve většině případů aktuálně chybí. Metodická příručka pokrývá vhlad do jednotlivých úloh, aby učitel rozuměl cílům a smyslu úlohy.

Úlohy vychází z potřeb očekávaných výstupů uvedených v RVP ZV, aby si učitel mohl být jistý, že se žák naučí vše, co je dáno koncepčně. Dále však záleží na školních vzdělávacích plánech, které si vytváří každá škola zvlášť, a které mohou obsahovat učivo nad rámec sbírky úloh. Úlohy však mohou sloužit i jako inspirace pro vytváření dalších úloh.

## A Obsah přiloženého CD

### **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.

### **Sbírka úloh – Programování s Ozobotem.pdf**

Výsledná sbírka úloh Programování s Ozobotem, která obsahuje 16 úloh pro žáky 4. a 5. třídy.

### **Příručka – Programování s Ozobotem.pdf**

Metodická příručka k sbírce úloh Programování s Ozobotem, která poskytuje řešení a poznámky k jednotlivým úlohám. Příručka je určena učitelům.

## Literatura

- [1] MPO ČR. *Digitální Česko v. 2.0: Cesta k digitální ekonomice* [online]. 2013 [cit. 2022-4-5]. 67 s. Dostupný z: [https://www.mpo.cz/assets/cz/e-komunikace-a-posta/Internet/2013/4/Digi\\_esko\\_v.2.0.pdf](https://www.mpo.cz/assets/cz/e-komunikace-a-posta/Internet/2013/4/Digi_esko_v.2.0.pdf).
- [2] MŠMT ČR. *Strategie digitálního vzdělávání do roku 2020* [online]. 2014 [cit. 2022-4-5]. 50 s. Dostupný z: <https://www.msmt.cz/uploads/DigiStrategie.pdf>.
- [3] MŠMT ČR. *Celkové vyhodnocení strategie digitálního vzdělávání do roku 2020* [online]. [cit. 2022-4-5]. Dostupný z: <https://www.msmt.cz/vzdelavani/skolstvi-v-cr/celkove-vyhodnoceni-strategie-digitalniho-vzdelavani-do-roku/>.
- [4] NÁRODNÍ ÚSTAV PRO VZDĚLÁVÁNÍ. *RVP ZV 2021 s vyznačenými změnami* [online]. 2021 [cit. 2022-4-5]. Dostupný z: [https://www.nuv.cz/file/4982\\_1\\_1/](https://www.nuv.cz/file/4982_1_1/).
- [5] KLEMENT, Milan; DRAGON, Tomáš. Preference učitelů informatiky při implementaci tematického celku algoritmizace a programování do výuky. *Trendy ve vzdělávání 2020*. 2020, roč. 13, č. 2, s. 68–77. ISSN 1805-8949.
- [6] KLEMENT, Milan. Edukační robotika z pohledu žáků základních škol. In KLEMENT, Milan (sest.). *Trendy ve vzdělávání 2021: sborník abstraktů mezinárodní konference*. Olomouc: Univerzita Palackého v Olomouci, 2021, s. 19–20.
- [7] WING, Jeannette M. Computational thinking. *Communications of the ACM*. 2006, roč. 49, č. 3, s. 33–35.
- [8] ANGELI, Charoula; GIANNAKOS, Michail. Computational thinking education: Issues and challenges. *Computers in Human Behavior*. 2020, roč. 105, s. 106185. ISSN 0747-5632.
- [9] BARR, David; HARRISON, John; CONERY, Leslie. Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*. 2011, roč. 38, č. 6, s. 20–23.
- [10] HAZZAN, Orit; LAPIDO, Tami; RAGONIS, Noa. *Guide to Teaching Computer Science: An Activity-Based Approach*. London: Springer, 2014. 296 s. ISBN 978-1-4471-6629-0.
- [11] CORMEN, Thomas H.; LEISERSON, Charles E.; RIVEST, Ronald L.; STEIN, Clifford. *Introduction to Algorithms*. 3rd. 2009. ISBN 978-0-262-53305-8.
- [12] DUNCAN, Caitlin; BELL, Tim. A Pilot Computer Science and Programming Course for Primary School Students. In. *Proceedings of the Workshop in Primary and Secondary Computing Education*. 2015, s. 39–48. Dostupný také z: <http://dx.doi.org/10.1145/2818314.2818328>.
- [13] SÁEZ LÓPEZ, José Manuel; BUCETA OTERO, Rogelio; LARA GARCIA-CERVIGÓN, Sebastián De. Introducing robotics and block programming in elementary education. *RIED. Revista Iberoamericana de Educación a Distancia*. 2021, roč. 24, č. 1, s. 95–113.
- [14] PLHÁKOVÁ, Alena. *Učebnice obecné psychologie*. Praha: Academia, 2011. 472 s. ISBN 978-80-200-1499-3.

- [15] THOROVÁ, Kateřina. *Vývojová psychologie: proměny lidské psychiky od početí po smrt*. Praha: Portál, 2015. 578 s. ISBN 978-80-262-0714-6.
- [16] BAŠTECKÁ, Bohumila (sest.). *Psychologická encyklopedie: aplikovaná psychologie*. Praha: Portál, 2009. 520 s. ISBN 978-80-7367-470-0.
- [17] KOHOUTEK, Rudolf. Kognitivní vývoj dětí a školní vzdělávání. *Pedagogická orientace*. 2008, roč. 18, č. 3, s. 3–22.
- [18] ACKERMANN, Edith. Piaget's constructivism, Papert's constructivism: What's the difference. *Future of learning group publication*. 2001, roč. 5, č. 3. Dostupný také z: <http://www.sylvia stipich.com/wp-content/uploads/2015/04/Coursera-Piaget--Papert.pdf>.
- [19] FRIEBROON-YESHARIM, Mor; BEN-ARI, Mordechai. Teaching computer science concepts through robotics to elementary school children. *International Journal of Computer Science Education in Schools*. 2018, roč. 2, č. 3.
- [20] PROTTSMAN, Kiki. Computer science for the elementary classroom. *ACM Inroads*. 2014, roč. 5, č. 4, s. 60–63.
- [21] KÖRBER, Nina; BAILEY, Lisa; GREIFENSTEIN, Luisa aj. An Experience of Introducing Primary School Children to Programming using Ozobots (Practical Report). In. *The 16th Workshop in Primary and Secondary Computing Education*. 2021, s. 1–6.
- [22] PETRUŠKOVÁ, Hedviga; LEHOTSKÝ, Milan; MAJHEROVÁ, Janka; ČERNÁK, Igor. Projects for Teaching Algorithmization in Primary Schools. In. *The Third International Conference on Computer Science, Computer Engineering, and Education Technologies (CSCEET2016) September 19-21, 2016 Lodz University of Technology, Lodz, Poland*. 2016, s. 32–39.
- [23] RAMBOUSEK, Vladimír; ŠTÍPEK, Jiří; WILDOVÁ, Radka. ICT Competencies and their Development in Primary and Lower-secondary Schools in the Czech Republic. *Procedia-Social and Behavioral Sciences*. 2015, roč. 171, s. 535–542.
- [24] HOŠEK, Stanislav. Názory učitelů prvního stupně základních škol na změny obsahu výuky informatiky. In DOSTÁL, Jiří (sest.). *Trendy ve vzdělávání 2018: Technika, informatika a inovace ve vzdělávání napříč obory*. Olomouc: Univerzita Palackého v Olomouci, 2018, s. 37.
- [25] BRYNDOVÁ, Lucie. Přístup učitelů informatiky ke koncepci informatického myšlení a implementaci jeho rozvoje na základních školách. In KLEMENT, Milan (sest.). *Trendy ve vzdělávání 2021: sborník abstraktů mezinárodní konference*. Olomouc: Univerzita Palackého v Olomouci, 2021, s. 12.
- [26] Karel IDE. [online]. [cit. 2022-4-18]. Dostupný z: <https://stanford.edu/~cpiech/karel/ide.html>.
- [27] HYKSOVÁ, Hana. Programování robotů na základní škole. In KLEMENT, Milan (sest.). *Trendy ve vzdělávání 2021: sborník abstraktů mezinárodní konference*. Olomouc: Univerzita Palackého v Olomouci, 2021, s. 17.

- [28] PICKA, Karel. Ozoboti na českých školách: zkušenosti učitelů základních škol. 2021. Dostupný také z: <https://www.ped.muni.cz/komensky/clanky/ozoboti-na-ceskych-skolach-zkusenosti-ucitelu-zakladnich-skol>.
- [29] FOJTIK, Rostislav. The Ozobot and education of programming. *New Trends and Issues Proceedings on Humanities and Social Sciences*. 2017, roč. 4, č. 5.
- [30] OZOBOT. *FAQs* [online]. [cit. 2022-4-5]. Dostupný z: <https://ozobot.com/support/faq>.
- [31] ROBOTWORLD. *Manuál Ozobot Evo - český návod* [online]. [cit. 2022-4-5]. Dostupný z: <https://www.robotworld.cz/downloads/manual-ozobot-evo-cs.pdf>.
- [32] OZOBOT. *Brand-Color-Codes-Map-Guide Template* [online]. [cit. 2022-4-5]. Dostupný z: <https://files.ozobot.com/stem-education/ozobot-tips.pdf>.
- [33] OZOBOT. *Ozobot-Color-Codes-Chart* [online]. [cit. 2022-4-5]. Dostupný z: <https://play.ozobot.com/print/guides/ozobot-ozocodes-reference.pdf>.
- [34] OZOBOT. *OzoBlockly* [online]. [cit. 2022-4-5]. Dostupný z: <https://ozoblockly.com/editor?lang=cs%5C&robot=evo%5C&mode=1>.
- [35] HANKA. *Kreslíme vlastní mapy* [online]. [cit. 2022-4-5]. Dostupný z: <http://ozobot.sandofky.cz/kreslime-vlastni-mapy/>.
- [36] NÁRODNÍ ÚSTAV ODBORNÉHO VZDĚLÁNÍ. *Učební plán* [online]. [cit. 2022-4-5]. Dostupný z: <http://www.nuov.cz/kurikulum/ucebni-plan>.