

Katedra informatiky  
Přírodovědecká fakulta  
Univerzita Palackého v Olomouci

# BAKALÁŘSKÁ PRÁCE

Využití LEGO MINDSTORMS pro výuku algoritmizace a  
programování



2022

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

Noe Švanda

Studijní obor: Aplikovaná informatika,  
prezenční forma

## **Bibliografické údaje**

Autor: Noe Švanda  
Název práce: Využití LEGO MINDSTORMS 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, prezenční forma  
Vedoucí práce: Mgr. Tomáš Kühn, Ph.D.  
Počet stran: 40  
Přílohy: 1 CD/DVD  
Jazyk práce: český

## **Bibliographic info**

Author: Noe Švanda  
Title: Use of LEGO MINDSTORMS for teaching 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: 40  
Supplements: 1 CD/DVD  
Thesis language: Czech

## **Anotace**

*Práce popisuje konkrétní případ komplexního výukového plánu, využívajícího stavebnici LEGO MINDSTORMS EV3 Education. V úvodu jsou analyzovány možnosti využití stavebnice pro výuku. Následuje popis výukového plánu a jeho zamýšlené začlenění do RVP. V neposlední řadě jsou v práci zdokumentovány podpůrné materiály, jež byly vytvořeny jako součást plánu.*

## **Synopsis**

*The thesis describes a specific case of a comprehensive curriculum using the LEGO MINDSTORMS EV3 Education. The introduction analyzes the possibilities of using the kit for teaching. The following is a description of the curriculum and its intended incorporation into the FEP. Last but not least, the work documents the supporting materials that were created as part of the plan.*

**Klíčová slova:** Výukový plán; programování; stavebnice; robot

**Keywords:** Curriculum; programming; kit; robot

Děkuji mému vedoucímu Mgr. Tomáši Kührovi, Ph.D. za asistenci při vedení této práce.

*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>8</b>
<b>2</b>	<b>Stavebnice LEGO MINDSTORMS EV3 Education</b>	<b>9</b>
2.1	Obsah stavebnice	9
2.1.1	Řídící jednotka Brick	9
	Operační systém	10
2.1.2	Servomotory	10
2.1.3	Senzory	10
2.1.4	Kabely	11
2.1.5	Díly Lego TECHNIC	11
2.1.6	Ostatní příslušenství	11
2.2	Vybrané technologie použití LEGO MINDSTORMS EV3	12
2.2.1	LEGO MINDSTORMS EV3 Lab	12
2.2.2	LEGO MINDSTORMS EV3 Classroom	12
	Jazyk EV3	12
2.2.3	Operační systém EV3dev-stretch a MicroPython	13
2.2.4	Microsoft MakeCode	14
2.2.5	Open Roberta Lab	15
2.2.6	Shrnutí	16
<b>3</b>	<b>Tvorba výukového plánu</b>	<b>17</b>
3.1	Kurikulární dokument	17
3.2	Vzdělávací program	17
3.3	Výukový plán	17
3.3.1	Postup tvorby výukového plánu	17
	Identifikace vzdělávacích potřeb	18
	Fáze tvorby	18
3.3.2	Faktory ovlivňující tvorbu výukového plánu	18
3.4	Zařazení výukového plánu do vzdělávacího programu	19
3.5	Třífazový model učení	19
3.6	Využití robotických pomůcek při výuce programování	20
<b>4</b>	<b>Plán výuky programování pro 1. ročník SŠ</b>	<b>21</b>
4.1	Účastníci výuky	21
4.2	Prerekvizity	21
4.3	Návaznost	22
4.4	Struktura plánu	23
4.4.1	Pomůcky výuky	24
4.4.2	Výukový plán	24
4.5	Výuka programování s pomocí stavebnice LEGO MINDSTORMS EV3 Education	25
4.6	Nástroje a technologie výuky	25
4.6.1	Diagram FlowChart	25

4.6.2	Vizuální jazyk EV3 . . . . .	26
4.6.3	Python 3 . . . . .	28
4.6.4	MicroPython . . . . .	29
4.6.5	EV3Dev . . . . .	29
4.6.6	Vývojová prostředí . . . . .	30
4.7	Cíl výuky . . . . .	30
<b>5</b>	<b>Webová aplikace RoboKurz</b>	<b>31</b>
5.1	Popis webu . . . . .	31
5.2	Použité technologie . . . . .	31
5.2.1	HTML . . . . .	32
5.2.2	CSS . . . . .	32
5.2.3	JavaScript . . . . .	33
5.2.4	React . . . . .	33
5.2.5	Node.js . . . . .	34
5.2.6	Docker . . . . .	34
5.2.7	Markdown . . . . .	34
	<b>Závěr</b>	<b>35</b>
	<b>Conclusions</b>	<b>36</b>
	<b>A Obsah přiloženého CD</b>	<b>37</b>
	<b>Literatura</b>	<b>38</b>

## Seznam obrázků

1	Jednotka Brick . . . . .	9
2	Motory a senzory stavebnice . . . . .	10
3	Ukázka programového staku . . . . .	13
4	Mapa výuky . . . . .	22
5	Programový stack . . . . .	26
6	Prostředí EV3 Classroom . . . . .	28
7	Webová aplikace RoboKurz - výběr lekcí . . . . .	31
8	Webová aplikace RoboKurz - ukázka lekce . . . . .	32

## Seznam tabulek

1	Základní obrazce diagramu FlowChart . . . . .	26
2	Kategorie bloků EV3 Classroom . . . . .	27

# 1 Úvod

Využití robotických pomůcek má v moderní výuce programování značný potenciál. Vnášejí do výuky netradiční prvek. Poskytují vysokou míru interaktivity. Výstup práce s robotem je hmatatelný v reálném světě. Zároveň ale podléhá vyšší míře rozpytlu zaměření výuky. Programování robota je vždy z části tvorba programu, a z části ovládání a ladění robota. A je důležité, aby prostředky s účelem výuky aspektů programování nebyly zbytečně uplatňovány pro výuku práce s konkrétním modelem robota. Použití robotických pomůcek v rámci výuky tedy vyžaduje důkladnou analýzu, závisí na možnostech zvolených pomůcek a obsahu cílového učiva.

Předmětem této práce je návrh komplexního výukového plánu programování, jenž využívá stavebnice LEGO® MINDSTORMS® EV3 Education.[1] Zmíněná stavebnice umožňuje vytvářet programy v různých programovacích jazycích, je doprovázena vlastním vývojovým prostředím a vizuálním programovacím jazykem. Pomůcka se rovněž těší aktivní komunitě, vytvářející další open-source funkcionalitu. Cílem plánu je s pomocí stavebnice naučit fundamentální základy programování. Stavebnice zastává roli interaktivní pomůcky a média pro přenos a využití znalostí napříč různými formami zápisu programu. V rámci plánu je snaha o co nejefektivnější využití získaných znalostí, jak pro všeobecný rozvoj studenta, tak i pro navazující výuku. Začlenění plánu bere ohled, jak na budoucí programátory, tak i na potencionální využití znalostí programování v jiných oborech.

Bakalářská práce je rozdělena na dvě části. V první části je zaměřena na informace získané studiem literárních a internetových zdrojů. Jsou zde zmíněny běžné postupy pro výuku programování, její členění. Do podrobnosti je rozebrána stavebnice LEGO MINDSTORMS EV3 Education, jaké jsou její technické možnosti a jakými způsoby lze s ní pracovat. Pozornost je věnována postupům pro tvorbu výukového plánu a jeho zapojení do vzdělávacích programů. V druhé, praktické části se práce zabývá konkrétními prvky studijního plánu. Jsou zde představeny klíčové aspekty plánu jako strukturální rozdělení, účastníci výuky nebo použité technologie. Zvolená forma plánu bude důkladně obhájena. Rovněž budou zmíněny a přiblíženy podpůrné materiály výuky. Na závěr bude provedeno komplexní zhodnocení sestaveného výukového plánu.



## 2 Stavebnice LEGO MINDSTORMS EV3 Education

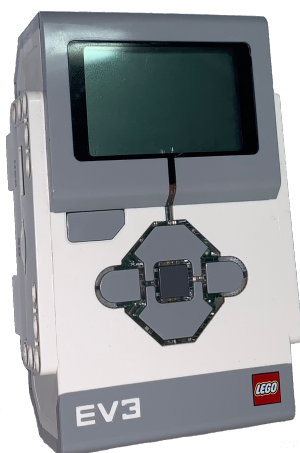
LEGO MINDSTORMS EV3 Education je robotická stavebnice od společnosti LEGO.[1] Je součástí řady produktů Mindstorms, zaměřující se na robotiku a programování. Byla vydána ve dvou edicích, Home a Education. Pro využití ve výuce je vhodnější verze Education, která disponuje specializovanou sadou vybavení a rozšíření. Poskytuje možnost stavby jiných doporučených modelů, než je tomu v edici Home.

### 2.1 Obsah stavebnice

Následující sekce přibližuje jednotlivé části stavebnice LEGO MINDSTORMS EV3 Education.

#### 2.1.1 Řídící jednotka Brick

Jedná se o kontrolní jednotku stavebnice. Brick je mikro počítač, slouží ke komunikaci s uživatelem, vykonávání zadaných instrukcí a k obsluze ostatních komponent stavebnice. Základem jednotky je procesor na architektuře ARM9. Výstup jednotky představuje monochromatický displej bez podsvícení. Vizuální výstup je doplněn reproduktorem. Přímý vstup a manipulace s operačním systémem jednotky je umožněna 6 tlačítky pod displejem. Jednotka je vybavena spojovými otvory pro upevnění ke stavebním dílům.[2]



Obrázek 1: Jednotka Brick

Další vstupně výstupní rozhraní jsou:

- 8 portů ve standardu RJ14, rozdělené na sekci A-D určenou k připojení motorů a sekci 1-4 pro senzory
- USB A a MINI USB port

- BlueTooth a WIFI rozhraní pro bezdrátové připojení
- MicroSD slot na paměťovou kartu

Jednotka je poháněna buďto přiloženou dobíjecí baterií nebo šesti tužkovými bateriemi standartu AA.

**Operační systém** Operační systém jednotky je upravená verze GNU/Linux, určená pro mikrokontrolery. Jednotka však umožňuje bootování i jiných operačních systému z externích úložišť. Grafické rozhraní představuje jednoduché, primárně textové prostředí se základní funkcionalitou. V rámci tohoto prostředí lze upravit nastavení, připojit jednotku k jinému zařízení, spravovat a spouštět programy uložené ve vnitřím úložišti.

### 2.1.2 Servomotory

Pohyb stavebnice je umožněn třemi servomotory. Ty se dělí na větší, jenž jsou ve stavebnici po dvou kusech, a střední. Větší motory poskytují vyšší výkon vhodný pro pohyb robota. Střední motor umožňuje přesnější pohyby a je vhodný především pro delikátní manipulaci. Všechny motory jsou vybaveny vestavěnými snímači, regulujícími rychlost otáčení a monitorujícími počet otáček.

### 2.1.3 Senzory

Vnímání a interakci okolního světa pro robota poskytují senzory. V edici stavebnice Education se setkáme s následujícími senzory:



Obrázek 2: Motory a senzory stavebnice

- **Ultrazvukový senzor**  
Senzor funguje na bázi ultrazvukových vln (sonar). Jeho jedinou funkcionalitou je detekce odražených vln od překážky, které přepočte na vzdálenost.

Pracuje v rozsahu 3 cm až 250 cm s přesností na 1 cm. V rámci těla senzoru je obsažen jak vysílač vln, tak i přijímač. Jednotka Brick poskytuje při zapojení senzoru základní řešení kolizí, i tak se ale více pracujících senzorů navzájem ovlivní. Výstupem senzoru je vzdálenost od detekované překážky. Z principu odražení zvukových vln je robot schopen nejlépe detekovat překážky jenž disponují velkými a rovnými plochami.

- **Kontaktní senzor**

Vydává signál na základě stlačení či uvolnění kontaktní plochy v přední části senzoru. Dává jednoduchou informaci o tom, zda se nachází v jednom ze dvou stavů, stlačen (pressed) a uvolněn (released).

- **Světelný senzor**

Obsahuje snímač intenzity světla z okolí a zároveň disponuje vlastními světelnými diodami. Výstupem snímání je hodnota intenzity odraženého světla. V pracovních režimech senzor umožňuje prostor před snímačem navesvit. Z detekovaného světla je schopen rozlišit odstín, případně barvu pro blízké předměty nebo jas okolního prostředí.

- **Gyroskopický senzor**

Senzor umožňuje snímat svůj rotační pohyb. Rotace je snímána pouze v rámci jedné osy. Ta je určena podle vyznačených šipek na vrchní straně senzoru. Rotace je snímána v jednotkách stupně za vteřinu a senzor předává informaci nejen o celkovém otočení, ale i o jeho rychlosti.

#### 2.1.4 Kabely

V stavebnici nalezneme přiloženy 2 druhy kabelů. USB 2.0 Type B kabel, jenž slouží k připojení kontrolní jednotky Brick k počítači a propojovací RJ14 kabely na propojení jednotky Brick a servomotorů a senzorů. Přiložené propojovací kabely jsou v délkách 25/35/50 cm.

#### 2.1.5 Díly Lego TECHNIC

Soubor 541 stavebních a pohybových dílů ze série LEGO Technic. Stavební díly jsou tvořeny téměř exkluzivně z tvrzeného plastu a bytelné. Pro lepší organizaci jsou díly barevně označeny a rozděleny do jednotlivých balení.

#### 2.1.6 Ostatní příslušenství

Stavebnice obsahuje rovněž dobíjecí baterii s kapacitou 2200 mAh a nabíjecí adaptér. Nechybí ani stavební manuál, který popisuje základní model robota označený jako Driving Base. Stavebnice jako celek je uložena v prostorné plastové bedně, která nabízí ideální protor pro organizaci dílů a skladování sestavených modelů.

## 2.2 Vybrané technologie použití LEGO MINDSTORMS EV3

Tato část naváže na seznámení s technickými možnostmi a díly stavebnice LEGO MINDSTORMS EV3. Bude provedena analýza a porovnání programů pracujících se stavebnicí, jejich vhodnost pro použití ve výukovém plánu. V konečné fázi bude provedeno souhrnné zhodnocení s výběrem technologií pro výukový plán.

### 2.2.1 LEGO MINDSTORMS EV3 Lab

Jedná se o dnes již řídce využívané, nicméně neustále dostupné oficiální vývojové prostředí od společnosti LEGO.[6] Je zajímavé především svým editorem, jenž je inspirován laboratorním software labVIEW. labVIEW je běžně používaným software v přírodovědných odvětvích a využívá jej například NASA<sup>1</sup>. [10] Editor EV3 Lab nabízí pokročilejší nástroje pro tvorbu programů. Samotné programy jsou však méně přehledné a přívětivé pro začínající programátory. Podpora prostředí navíc byla oficiálně ukončena a není kompatibilní s MAC OS od verze 10.15. Nebude dále hodnocen jako potenciální prostředek výuky programování. Dříve však byl primárním nástrojem práce se stavebnicí LEGO MINDSTORMS EV3 a nespočet technické a odborné literatury se zaměřuje na toto prostředí.

### 2.2.2 LEGO MINDSTORMS EV3 Classroom

Proprietární vývojové prostředí od společnosti Lego. Tak jako samotná stavebnice je rozdělena na verze Home a Education, existují i odpovídající verze prostředí Home a Classroom. Zaměření těchto vývojových prostředí je na konkrétní verzi stavebnice a pro účely výuky s pomocí stavebnice EV3 Education je doporučeno použít EV3 Classroom.[1]

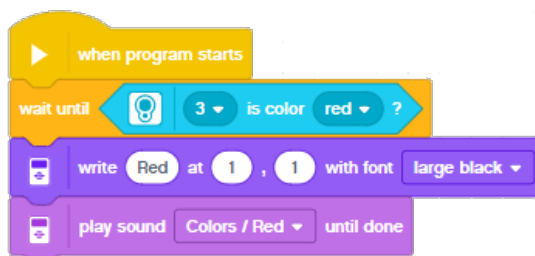
Toto prostředí je na míru vytvořeno pro práci se stavebnicí EV3 Education, odpovídají plány modelů i dostupná funkcionalita pro obsažené díly. V editoru Classroom je možné vytvářet programy pomocí jazyka EV3. Ten je inspirován vizuálním programovacím jazykem Scratch.[8] Prostředí obsahuje několik kompletních robotických řešení, včetně modelů a programů. Spolu s modely jsou zde obsaženy interaktivní lekce s úlohami. Prostředí neobsahuje virtuálně žádné možnosti přizpůsobení. Základní jednotkou programu, je programový soubor ve formátu .lsmt. V rámci prostředí neexistuje adresářová struktura a orientace mezi více programy spočívá v uspořádání souborů.

**Jazyk EV3** EV3 je vizuální programovací jazyk, což znamená, že využívá vizuálních komponent pro strukturu programu. Tyto komponenty mají podobu dílku puzzle. Každý program je vymezen na jeden programový soubor. Základní jednotkou programu je tzv. programový stack. Jedná se o sérii po sobě jdoucích bloků. Blok je základní funkční jednotka programu, pomocí bloků jsou prováděny

---

<sup>1</sup>National Aeronautics and Space Administration (USA)[9]

veškeré operace. Jazyk disponuje základními bloky, jenž se zaměřují na běžnou programovou funkcionalitu, kontrolu průběhu programu, ukládání hodnot atp. Jsou zde ale také specializované bloky pro správu připojených částí stavebnice. Celkově je EV3 značně abstraktní a jednotlivé funkční bloky často vykonávají souhrn komplexních operací.



Obrázek 3: Ukázka programového staku

Vykonání programu probíhá propojením robota s prostředím, pomocí integrovaného rozhraní. K dispozici je buďto bezdrátové připojení (wifi, bluetooth) nebo usb kabelem.

- Výhody**
- Jednoduchá a zřejmá forma zápisu programu
  - Přehledný editor
  - Stavební návody a výukové kurzy s úkoly jsou součástí prostředí

- Nevýhody**
- Špatná přehlednost u rozsáhlých programů
  - Jazyk EV3 má praktické využití čistě v kombinaci se stavebnicí

**Závěr** Práce s robotem v prostředí EV3 Classroom s využitím zmíněného jazyka EV3 je vhodná volba pro začínající programátory. Jednoduchá vizuální forma jazyka udržuje řád v programu a intuitivně diktuje jeho strukturu. V EV3 však poměrně brzy nastane situace, kdy komplexnost tvořeného programu přesáhne únosnou mezi a jeho vizuální zápis začne být detrematální pro přehlednost. EV3 Classroom je v první řadě nástrojem pro ovládání robota a až v druhé řadě programovacím jazykem. Přesto nelze popřít vhodnost jazyka EV3 pro začátečnickou práci s robotem čistě pro svoji deskriptivnost a přehlednost ve stručných programech.

### 2.2.3 Operační systém EV3dev-stretch a MicroPython

EV3dev-stretch je komunitně vytvořený operační systém založen na GNU/Linux distribuci Debian.[19] Nabízí širokou podporu programovacích jazyků a externího příslušenství pro ovládání a správu téměř všech aspektů robota. Systém disponuje rozsáhlou uživatelskou podporou a více než 50 000 softwarovými balíčky. Mezi jazyky podporované v EV3dev-stretch patří C#[27], Java[28], Python[17], JavaScript[30] a další. S rozsáhlou podporou souvisí značná různorodost v použitelnosti a přívětivosti práce s robotem. Pro účely výukového plánu bude brána

v úvahu kombinace operačního systému EV3dev-stretch, programovacího jazyka MicroPython a prostředí Visual Studio Code.[23]

Práce s robotem v tomto systému obnáší značnou míru příprav. Samotné spuštění operačního systému vyžaduje zapsat jeho obraz na předem připravenou MicroSD kartu, jenž bude sloužit pro bootování robota. Následné připojení jednotky Brick do prostředí VS Code je jednoduše umožněno díky rozšíření ev3dev-browser.[20] V rámci doplňujícího rozšíření MicroPython[21] je k dispozici předpřipravený formát vytvoření projektu pro robota a dokumentace ke knihovnam, ovládajícím robota. Samotný jazyk MicroPython je zjednodušená verze Pythonu určená pro programování mikrokontrolerů. V rámci prostředí není pro MicroPython k dispozici našeptávač. Příložená uživatelská dokumentace je však přehledná a obsahuje několik praktických ukázek.

- Výhody**
- Práce v editoru VSCode
  - Motivace k aktivnímu využití dokumentace
  - Přenesitelnost praktických znalostí z jazyka MicroPython
  - Podpora příslušenství

- Nevýhody**
- Bez našeptávače
  - Nutná základní znalost jazyka Pythonu
  - Nutnost extra HW (microSD karta)

**Závěr** Kombinace technologií EV3dev, VSCode a MicroPython umožňuje programovat robota v prakticky využitelném jazyce a získat zkušenosti s prací v jednom z nejrozšířenějších vývojových prostředí. Oproti ostatním možnostem ovládání robota se jedná o plnohodnotné kódové programování. Absence listeneru a snadno dostupných řešení na internetových fórech směruje studenty k aktivní práci s dokumentací. Programování samotné avšak téměř vyžaduje předchozí znalosti, ať už samotného jazyka MicroPython (Python) nebo použití stavebnice.

#### 2.2.4 Microsoft MakeCode

Microsoft MakeCode[5] je open source platforma určená k výuce programování a robotiky. Jedná se o webovou aplikaci umožňující tvorbu programů vizuálně kódovými bloky anebo skrz programovací jazyk Javascript. Disponuje editorem s částí zaměřenou na tvorbu programu skrz vizuální jazyk, téměř identický jazyku EV3Dev. Druhá část editoru je zaměřena na tvorbu kódového programu v jazyce JavaScript. Tvorba programu je mezi jednotlivými částmi editoru provázána. Vytváření programu v jednom jazyce, generuje program i v jazyce druhém. To umožňuje šikovné srovnání kódové a vizuální podoby a nabízí výpomoc pro situace, kdy si student neví rady. Obodně jako v EV3 Classroom i zde je editor vizuálního jazyka vybaven výčtem všech funkčních bloků.

V rámci webové aplikace se nachází jednoduchý simulátor EV3 Brick a připojených servomotorů a senzorů. Jedná se o tester poskytující strohý pohled na funkčnost psaného programu. Nesimuluje však jakkoli okolní prostor, či pohyb robota. Web disponuje rozhraním pro připojení jednotky Brick a následné spuštění nebo stahování vytvořených programů.

Součástí webové aplikace je rovněž knihovna kurzů a tutoriálů. Ty poskytují široké možnosti inspirace a procvičování pro studenty. Od jednoduchých úloh a hříček až po komplexní kurzy.

- Výhody**
- Bez nutnosti instalace
  - Kombinace a propojení vizuálního programování s kódovým
  - Bohatá knihovna tutoriálů a úloh

**Nevýhody** – Využití JavaScriptu, jakožto doplňujícího kódového jazyka

**Závěr** MakeCode poskytuje obsáhlou platformu pro práci se stavebnicí a výuku programování. Knihovna cvičení a tutoriálů poskytuje bohaté rozšíření výuky. Okamžitá interpretace kódu psaného bloky do JavaScript a zpět zase umožňuje pohlédnout novým způsobem na tvorbu programu. Syntaxe JavaScriptu je však méně přehledná pro nové programátory. Samotný JavaScript má užší oblast využití, ve srovnání s populárními konkurenčními jazyky a jeho vhodnost pro výuku začátečníků je diskutabilní.

### 2.2.5 Open Roberta Lab

Open Roberta Lab je internetové programovací prostředí zaměřené na robotiku.<sup>[7]</sup> S důrazem na co nejnižší bariéru přístupu, tato webová aplikace poskytuje vývojové prostředí a zároveň simulátor pro širokou škálu robotických stavebnic a pomůcek. Pro stavebnici Lego Mindstorms EV3 existuje hned několik variant vývojových prostředí. Programy v rámci všech prostředí EV3 jsou vytvářeny ve vizuálním programovacím jazyku podobném EV3, nazývaném NEPO. Zároveň, každé vývojové prostředí kód sestojený v NEPO interpretuje do dalšího programovacího jazyka, v závislosti na volbě konkrétního prostředí. Editace programu je možná i v rámci kódového jazyka, avšak narozdíl od MakeCode, nedochází k automatické interpretaci. Export programu je možný do jednotkou Brick akceptovaného formátu UF2 nebo do programového souboru vybraného kódového jazyka (pro využití například s ev3dev-stretch). Roberta Lab taktéž poskytuje možnost přímého linku mezi webovou aplikací a robotem EV3.

- Výhody**
- Simulátor robota
  - Forma webová aplikace (přenositelnost)
  - Přímé propojení robota s webovou aplikací

**Nevýhody** – Motivace a možnosti použití simulátoru

- Editor pro psaní kódového programu není pro danou funkcionalitu patřičně stavěný

**Závěr** Open Roberta Lab poskytuje řešení pro situace, kdy by nedostatek robotů brzdil či dokonce neumožňoval výuku. Interpretace simulátoru nabízí k dispozici prostředky pro vytváření jednoduchých simulací prostředí. V těchto prostředích poté virtuální robot vykonává program. Nicméně simulátor ztrácí záživnost fyzického robota a jeho použití oproti stavebnici postrádá aspekt interaktivity. Webová aplikace není stavěna na psaní kódového jazyka a ve srovnání s MakeCode chybí přímá interpretace mezi jednotlivými formami zápisu.

### 2.2.6 Shrnutí

Všechny uvedené technologie, respektive prostředí mají výhody a určité nedostatky. Nejdůležitějšími kritérii pro výukový plán byly jednak motivace studentů a vytvoření nízkoprahového úvodu do programování, ale i zakomponování kódového programování. Nejvhodnější variantou pro výukový plán, jenž je předmětem této práce, byla zvolena kombinace EV3 Classroom a EV3Dev + MicroPython. Jazyk MicroPython má téměř identickou formu zápisu jako Python 3. Díky tomu v této kombinaci technologií může výuka poskytnout výstup znalostí v jednom z nejrozšířenějších programovacích jazyků. Zároveň je díky použití Classroom postaráno o jednoduchý úvod pro pochopení ovládacích prvků robota a základních programovacích konceptů.



## 3 Tvorba výukového plánu

Tato sekce přiblíží náležitosti tvorby komplexního výukového plánu. Představí jednotlivé faktory, které jej ovlivňují. Současně nastíní pojem vzdělávací program a jaký má vztah k výukovému plánu.

### 3.1 Kurikulární dokument

Jedná se o pedagogický dokument, jenž definuje koncept, cíle a obsah vybraného vzdělávacího období.[13] Dokument vzniká ve dvou úrovních. První úroveň je státní, v rámci které vznikají tzv. rámcové vzdělávací programy (RVP). Ty jsou vydávány ministerstvem školství, mládeže a tělovýchovy (MŠMT).[16] Jednotlivé RVP jsou členěny pro navazující období od předškolního věku až po vysoké školství a specializované vzdělávání.

Na druhé úrovni vznikají školní vzdělávací programy (ŠVP). Ty představují vzdělání v rámci konkrétní školy, s ohledem na její specializaci. ŠVP představují implementaci RVP a uskutečňuje se podle nich výuka. Reprezentují jeden z povinných prvků dokumentace školy.

### 3.2 Vzdělávací program

Obecně se jedná se kurikulární dokument normativního charakteru, který byl schválen nebo vydán centrálními orgány školství. Vzdělávací program obsahuje souhrnné vymezení cílů, koncepce a obsahu parametrů vzdělávání pro specifický stupeň nebo druh školy. Základním prvkem školního vzdělávacího programu je implementace konkrétních výukových plánů. Ty slouží jako dílčí realizační cíle. Výstupem programu je přemostění výkonnostních a vědomostních mezer jedince, mezi tím jak danou problematiku ovládá v současnosti a jak je požadováno aby danou problematiku ovládal v budoucnu. Pro získání kvalitního vzdělávacího programu je nutná průběžná kontrola, revize a testování. Je potřeba identifikovat nedostačující části a program postupně iterativně vylepšovat.

### 3.3 Výukový plán

Plán popisující konkrétní úsek vzdělávání. Obsahuje definici požadavků na formu výuky, její strukturu, účastníky a obsah. Jsou v něm detailně vymezeny jednotlivé dílčí a komplexní cíle a předpoklady pro jejich splnění. Výstupem výukového plánu by měl být detailní strukturovaný popis, jak budena samotná výuka probíhat. Kdo se jí bude účastnit, jaké pomůcky v ní budou využity a jaké časové prostředky vyžaduje.

#### 3.3.1 Postup tvorby výukového plánu

Základem tvorby výukového plánu je identifikace vzdělávacích potřeb studentů. Ty zjistíme analyzováním současného stavu studenských znalostí a jejich porov-

náním s požadovaným stavem a výsledky. S tím souvisí stanovení, koho se budou vzdělávací potřeby týkat a ve které fázi studia by měly být doplněny. Následuje formulace jednotlivých cílů výukového plánu a jejich rozdělení do realizačních celků. Pro jednotlivé celky jsou zvoleny metody hodnocení jejich dosažení. V neposlední řadě jsou vybrány pomůcky a postupy dosažení stanovených cílů.

**Identifikace vzdělávacích potřeb** Zjišťování vzdělávacích potřeb probíhá na základě firemní praxe, případně navazujících akademických požadavků. Samotná potřeba vzdělání se určí jako cílová množina znalostí mínus množina současných znalostí cílových účastníků výukového plánu. Potřeba se v průběhu času může měnit, obzvláště u dynamických a neustále se vyvíjejících oborů. Je proto důležité sledovat vývoj potřeby a průběžně kontrolovat, zda se nezměnila.

**Fáze tvorby** Úvodní fázi tvorby výukového plánu představuje analýza. Dochází k zhodnocení profilu absolventa kurzu, jakými praktickými a teoretickými znalostmi má disponovat. Následuje návrh plánu, kdy se zformulují jednotlivé cíle pro výuku, popíše se prerekvizity a kontrolní metrika plánu. V další fázi nastává vypracování plánu, dochází k určení výukových pomůcek, podpůrných materiálů a organizační struktury výuky. Dalším krokem je uskutečnění výuky plánu. V závěrečné fázi dochází k hodnocení vnějšimu, od absolventů výuky, a vnitřnímu, sebereflexi a zhodocení ze stran spolupracovníků.

### 3.3.2 Faktory ovlivňující tvorbu výukového plánu

Tvorba každého výukového plánu je ovlivňována celou řadou faktorů. Formalizace a strukturování myšlenek do jednotlivých kategorií je klíčová. Plán by měl být komplexní a co nejvíce konkrétní. Mezi faktory, které do něj zasahují patří:[14]

- **Východisko**

Na základě jakého popudu má být výukový plán vytvořen. Z čeho vychází informace, na kterých je potřeba plánu založena

- **Vzdělávací cíl**

Určení jakého stavu má být dosaženo po splnění výukového plánu. V této části je klíčové konkretizovat vzdělávací potřeby na základě znalostních a výkonostních mezer. Je nutné určit potřebné dílčí znalosti a ty zorganizovat do hierarchie dle důležitosti. Organizace má zpravidla formu pyramidy, kdy nižší úroveň reprezentuje základní/obecné znalosti, vyšší úroveň naopak značí klíčové znalosti, na kterých stojí splnění vzdělávacího cíle.

- **Účastníci akce**

Pro koho je program určen a kdo další se výuky účastní. Identifikace studentů konkrétního ročníku, pro něž je výuka vhodná a stanovení lektorů a jejich kvalifikace.

- **Forma a druh akce**  
O jaký typ výukové akce se jedná, individuální nebo skupinová.
- **Rámcová osnova**  
Struktura akce, časové rozdělení a strukturální rozdělení, obsah jednotlivých tématických celků a lekcí.
- **Metody a prostředky**  
Použité pedagogické metody výuky a prostředky, využity k její podpoře.
- **Organizace výuky**  
Určení, jak velké skupiny studentů se budou výuky účastnit. Zároveň tento faktor ovlivňuje průběh a časový harmonogram. Při organizaci je vhodné počítat s určitými rezervami a výpadky.
- **Výstupy, ověření znalosti a dovedností**  
Výběr způsobu a samotné ověření získaných znalostí účastníků vzdělávacího programu.
- **Studijní materiály**  
Povinné a doporučené materiály k výuce. Zvláště je rozdělena skupina materiálů určená pro studenty a pro lektory.

### 3.4 Zařazení výukového plánu do vzdělávacího programu

Výukový plán může být zařazen do vybraného vzdělávacího programu, pokud se shodují v rámci požadavků a cílů. Požadavky výukového plánu nesmí přesahovat prostředky, které jsou k dispozici v programu. Výukový plán musí splňovat dílčí cíle stanovené v programovém celku. Klíčová je rozumná a efektivní návaznost mezi plány ve výukovém programu, pokud jich je zakomponováno více.

### 3.5 Třífázový model učení

Model je založen na neustále se opakujícím procesí, při kterém si student vytváří pevné vazby na naučenou látku v paměti.[12] Jednotlivé aspekty spolu plně souvisí a je vhodná jejich kombinace. Nové učivo by mělo plynule navazovat na předchozí související.

- **Evokace**  
Student získává přehled o tom, co z daného učiva již ovládá. Naladí se na téma učiva a vytvoří si vnitřní zájem, jenž jej motivuje k dalšímu studiu. Zájem je stavěn jak na charakterových predispozicích studenta, tak i na jeho motivaci průběhem a cílem výuky.
- **Uvědomění si významu**  
Nové znalosti a myšlenky jsou předloženy studentovi. Klíčovou roli v této fázi hraje studentův aktivní přístup.

- **Reflexe**

Fáze upevňování poznatků. Student vyjadřuje nově nabyté informace. Dochází k transformaci starých myšlenkových pochodů v nové, ovlivněné získanými znalostmi.

### **3.6 Využití robotických pomůcek při výuce programování**

Robotické pomůcky vnášejí do klasické výuky programování zásadní změny. Práce s robotem je náročnější. Z programovacího hlediska, se výuka do značné míry odvíjí okolo pochopení a získání kontroly nad ovládním robota. Z praktického hlediska je testování jednotlivých programů vytvořených pro robota, obzvlášť pohybových, v porovnání s čistě digitálním výstupem běžného programování, mnohem časově i prostorově náročnější. Student musí zvládnout organizovat volný čas mezi vstřebáváním nových informací, tvorbou programů a jejich následným testováním. Student je motivovaný méně pracovat stylem pokus omyl a více se zamýšlet, jak bude vypadat výstup jeho programu. Robotická část výuky, tedy ovládním robota, může být velmi prospěšná, pro lepší celkové pochopení výuky. Zároveň, při správném využití, dokáže robotická pomůcka působit jako významný motivační faktor. Důležité je nalézt správnou rovnováhu mezi praktickým a teoretickým pojetím výuky a její obtížností.

## 4 Plán výuky programování pro 1. ročník SŠ

Výukový plán vznikl na základě vědomostní mezery v oblasti programování u studentů netechnických oborů středních škol. Plán cílí na stručný úvod do programování. Snaží se o vytvoření znalostí, které najdou uplatnění v oborech, jenž se prolínají s programováním. Výuka stojí na využití svatebnice LEGO EV3 Mindstorm Education. Plán se skládá z teoretického popisu lekcí. Je doplněn o podpůrné materiály ve formě webové aplikace. Ta obsahuje základní teorii, návody, ukázky a cvičné příklady. Plán cílí na obecnou vazbu na RVP pro střední vzdělávání.[15] Cílí na začlenění do implementace školního vzdělávacího plánu informačních a komunikačních technologií.

Mezi základní cíle výukového plánu patří:

- Rozvíjet logické myšlení
- Podporovat schopnost analyzovat problém a dostupné prostředky, navrhnout a sestavit řešení
- Získat přehled v práci s robotem Lego EV3 Mindstorms
- Seznámit se s fundamentálními základy programování
- Získat zkušenosti s programátorskou praxí

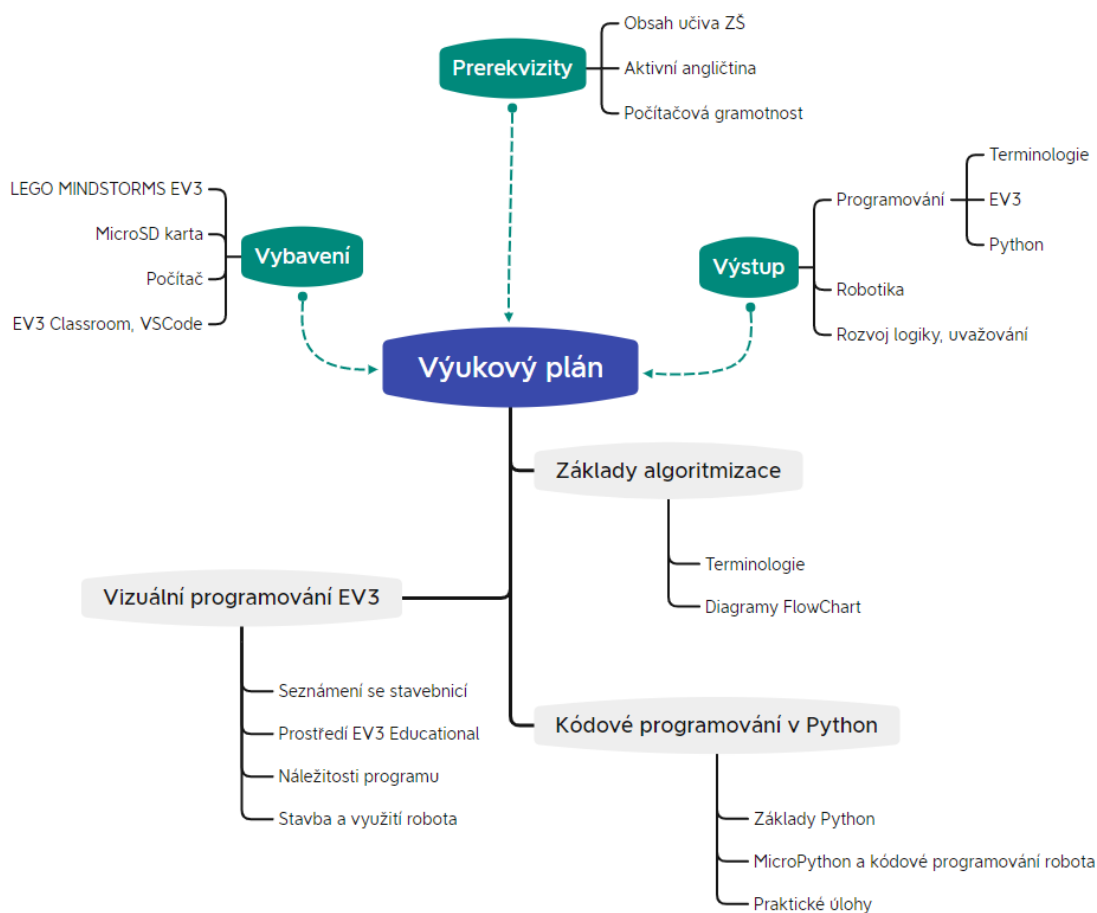
### 4.1 Účastníci výuky

V rámci plánu se budou výuky účastnit studenti 1. ročníku SŠ s maturitním zakončením a odpovídající ročníky gymnázií. Výuka je zaměřena na obecné znalosti a její výstup je užitečný jak pro technické, tak i obecně zaměřené obory. Ze strany vyučujících, počítá se s účastní jednoho a více lektorů, dle dostupných stavebnic a časových prostředků. Vyučující musí být seznalí s technologiemi, jenž jsou použity ve výuce. Zároveň je zde předpoklad pro znalost problematiky programování nad rámec toho, co je obsaženo v lekcích. Požadovány jsou dobré organizační schopnosti a zručnost při manipulaci se stavebnicemi.

### 4.2 Prerekvizity

Základním faktorem výukového plánu je jeho přístupnost. Jelikož výuka programování v plánu je úvodem do dané oblasti, z tohoto hlediska plán nemá překážky. Respektive, předpoklady pro výuku jsou nastaveny tak, aby vstupní kvalifikací bylo splnění základní školní docházky.

Existuje avšak několik nepřímo zřejmých prerekvizit, obzvláště v oblasti schopností, které jsou nezbytné pro zvládnutí výuky:



Obrázek 4: Mapa výuky

- Schopnost pracovat samostatně a logicky uvažovat
- Počítačová gramotnost
- Porozumění technického textu
- Aktivní znalost angličtiny

### 4.3 Návaznost

Znalosti získané absolvováním plánu umožňují plynule navázat v další výuce programování nebo robotiky. Znalost základů programování a jazyka Python umožňuje pokračovat také mezioborovou výukou v tradičních předmětech. Ta nachází v dnešní době stále větší využití a vysokoškolské obory jako chemie, biologie nebo fyzika stále častěji vyžadují základní znalosti programování. Navázat takovýmto způsobem na výuku již v rámci střední školy je prozatím obtížně realizovatelné a závisí to na vůli a spolupráci ve výukových plánech daných oborů. V případě, že by možnost takovéto návaznosti existovala, vzdělávací potřeba plánu by se mnohonásobně zvýšila.

## 4.4 Struktura plánu

Strukturálně je plán rozdělen do 12 lekcí začleněných v rámci 3 celků. Jednotlivé lekce jsou nastaveny tak, aby byly splnitelné za 45 minut výuky ve svižném tempu a s přeskokováním vybraných úloh. Ideální časová dotace je 90 minut na lekci. Jednotlivé lekce spoléhají primárně na individuální práci. Konzultace mezi spolužáky jsou povoleny, v některých situacích dokonce podporovány. Společným prvkem napříč plánu je stebnice LEGO MINDSTORMS EV3. Výuka začíná více abstraktní formou reprezentace algoritmu, kdy se první celek zabývá zápisem pomocí diagramů FlowChart.[22] Navazuje celek vizuálního programování v prostředí EV3 Classroom. Plán kulminuje celkem programování v jazyce Python. Každý celek se zabývá specifickým způsobem zápisu algoritmu, respektive programu. Jednotlivé celky mají vyčleněny vlastní cíle a účel, které ovšem chronologicky a obsahově navazují.

- **Diagramy FlowChart**

Celek se zabývá naprostým úvodem do algoritmizace. Proběhne definování základních pojmů a formy zápisu algoritmu. Zahnut je přechod od volného zápisu algoritmu k využití diagramu FlowChart. V rámci FlowChart dojde k seznáání studentů s termíny proměnná, operátor, pole. Rovněž bude studentům představen koncept cyklů a podmíněného vyhodnocení algoritmu.

- **Lekce 1** - Algoritmy

- **Lekce 2** - Procvičování FlowChart

- **Vizuální programování EV3**

Práce v rámci prostředí EV3 Classroom. Nastává plynulý přechod z diagramů do vizuálního programování pomocí bloků (jazyk EV3). Součástí celku je stavba a aktivní využití robota. Nejdříve po částech, s primárním využitím kontrolní jednotky Brick a vybrané funkcionality nebo příslušenství. V závěrečných lekcích probíhá práce s komplexním modelem robota Driving Base. V průběhu celku dochází k opakování konceptů známých z diagramů FlowChart, pouze s novou formou zápisu. Znalosti jsou rozšířeny o ovládání robota a využití vlastních bloků, které představují reprezentaci funkcí v EV3.

- **Lekce 3** - Úvod do vizuálního programování v EV3

- **Lekce 4** - Programujeme s jednotkou Brick

- **Lekce 5** - Podmíněný běh kódu a proměnné

- **Lekce 6** - Vlastní bloky

- **Lekce 7** - Ovládání robota

### **Programování v Python**

Úvod do kódového programování v jazyce Python. Spoléhá se na přemostění znalostí a konceptů programování z EV3. Výuka z počátku probíhá

bez využití stavebnice, pouze s terminálovým výstupem v rámci užitého vývojového prostředí Visual Studio Code. Sekce se zaměřuje na základní koncepty, jež jsou z velké části opakováním s novým zápisem. Nově přibývá základní terminologie a praktické využití tříd. V závěru celku je předmětem práce ovládání robota pomocí EV3 MicroPython. S tím souvisí využití operačního systému ev3dev-stretch.

- **Lekce 8** - Úvod do programování v Python
- **Lekce 9** - Opakování v programu
- **Lekce 10** - Třídy
- **Lekce 11** - Ovládání robota v MicroPython
- **Lekce 12** - Komplexní úlohy s robotem

#### 4.4.1 Pomůcky výuky

**Hardwarové pomůcky výukového plánu:**

- LEGO Mindstorms EV3 Education
- Počítačové zařízení
- MicroSD karta - je nutné, aby byla maximální kapacita karty 32GB, je povolena varianta SDHC, varianta SDXC není podporována

**Software nástroje:**

- LEGO EV3 Classroom
- Python 3
- Visual Studio Code - rozšíření: Python, EV3 Education MicroPython, ev3dev-browser
- ev3dev-stretch - operační systém určený pro umístění na MicroSD kartu k bootování robotapodporována

#### 4.4.2 Výukový plán

Primární materiálem pro vedení výuky je dokument výukového plánu. Jedná se o popisný dokument určený pro vyučující. V rámci dokumentu je obsaženo:

- Strukturální popis výuky rozdělen na jednotlivé celky a lekce.
- Motivace a předpoklady pro výuku.
- Požadavky na studenty, cílová skupina.



- Soubor hardwarových a softwarových pomůcek, jenž budou při výuce potřebné.
- Popis jednotlivých lekcí, struktura výuky v rámci lekce.
- Koncepty a cíle jednotlivých celků a lekcí.
- Komentáře a ukázková řešení k úlohám.

## 4.5 Výuka programování s pomocí stavebnice LEGO MINDSTORMS EV3 Education

Pro potřeby ovládnutí robotické stavebnice LEGO MINDSTORMS EV3 je nutná znalost základů algoritmického řešení konkrétních problémů. Činnosti a úkoly, které jsou po robotu požadovány musí být nejprve analyzovány a následně zapsány do programu. Musí být kladen důraz na správnou formu zápisu. Vyučující důkladně kontroluje vytvořená programová řešení. Pro úspěšné programování jednotky je důležité pochopit a upřesnit některé fundamentální koncepty, jako uložení dat, funkce, podmínky a opakování cykly. Ty by měly být získané úvodem v diagramech FlowChart.<sup>[22]</sup>

Konečné sestavení programu pro robota probíhá v programovacích jazycích, se kterými by se měl student seznámit ještě před jakoukoli prací s robotem. Orientace v UI<sup>2</sup>, spouštění, ukládání a export programů, stejně jako seznámení a práce s dokumentací, ale také tipy, jak řešit problémy, by měly být studentům známy před počátkem vývoje programu.

## 4.6 Nástroje a technologie výuky

V této části práce jsou zmíněny obecné znalosti a prostředky, které jsou nedílnou součástí výukového plánu.

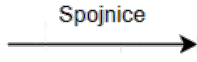
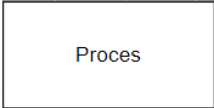



### 4.6.1 Diagram FlowChart

FlowChart je druh diagramu, který slouží k popisu dílčích kroků algoritmu. Je složen z obrazců, představující kroky a rozhodnutí v rámci algoritmu, a spojnic ukončených šipkou, které udávají směr vývoje algoritmu. V rámci obrazců je obsažen text upřesňující vykonávanou činnost nebo operaci.

Zápis pomocí diagramu FlowChart nabízí přirozený způsob, jakým se naučit zaznamenávat algoritmy. Vytváří logickou sérii kroků, nastolí jejich první zobecnění a kategorizování. Umožňuje pochopit a reprezentovat podmíněné a cyklické vykonávání algoritmu.

---

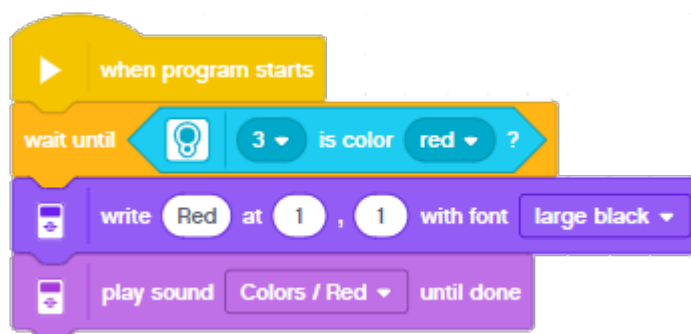
<sup>2</sup>Uživatelské rozhraní aplikace.

Obrázek	Název	Popis
	Spojnice	Spojuje jednotlivé obrazce diagramu. Její směr udává postup kroků algoritmu.
	Proces	Dílčí krok algoritmu. Představuje srozumitelnou operaci.
	Terminál	Reprezentuje počátek nebo konec algoritmu.
	Rozhodnutí	Obrazec rozhodnutí obsahuj podmíněný výraz. Na základě vyhodnocení výrazu se algoritmus vyvíjí dále.
	Vstup/výstup	Představuje krok, ve kterém do algoritmu vstupují data anebo z něj vystupují.

Tabulka 1: Základní obrazce diagramu FlowChart

#### 4.6.2 Vizualní jazyk EV3

Prostředí LEGO MINDSTORMS EV3 Classroom využívá programovacího jazyka EV3. Lze jej definovat jako imperativní, vizuální a vycházející z jazyka Scratch. Veškeré programování je prováděno pomocí blokových struktur.[3]



Obrázek 5: Programový stack

Bloky můžeme obecně rozdělit na 3 druhy:

- **Běžné bloky**

Běžné bloky nabývají typické formy připomínající puzzle. Tato podoba je reprezentována buď výřezem na vrchní straně, anebo výstupkem na straně spodní, popřípadě kombinací obou případů. To umožňuje jednotlivé bloky do sebe skládat a vytvářet stacky souvislými instrukcemi složenými směrem shora dolů. Bloky, které mají typicky pouze výstupek na spodní straně jsou

bloky startovní. Jedná se o bloky výhradně z kategorie Events, obsahují podmínku a vždy musí začínat stack.

- **Podmínkové bloky**

Bloky určující podmínku mají jednotný tvar šestiúhelníků. Každý podmínkový blok obsahuje tvrzení, jenž po vyhodnocení vydá pravdu (true) nebo lež (false). Některé běžné bloky (převážně kontrolní) mají ve své struktuře šestiúhelníkový otvor, určený k osazení podmínkovým blokem. Při běhu programu pak tyto bloky vyhodnotí osazený podmínkový blok a na základě výsledku určují další vývoj programu.

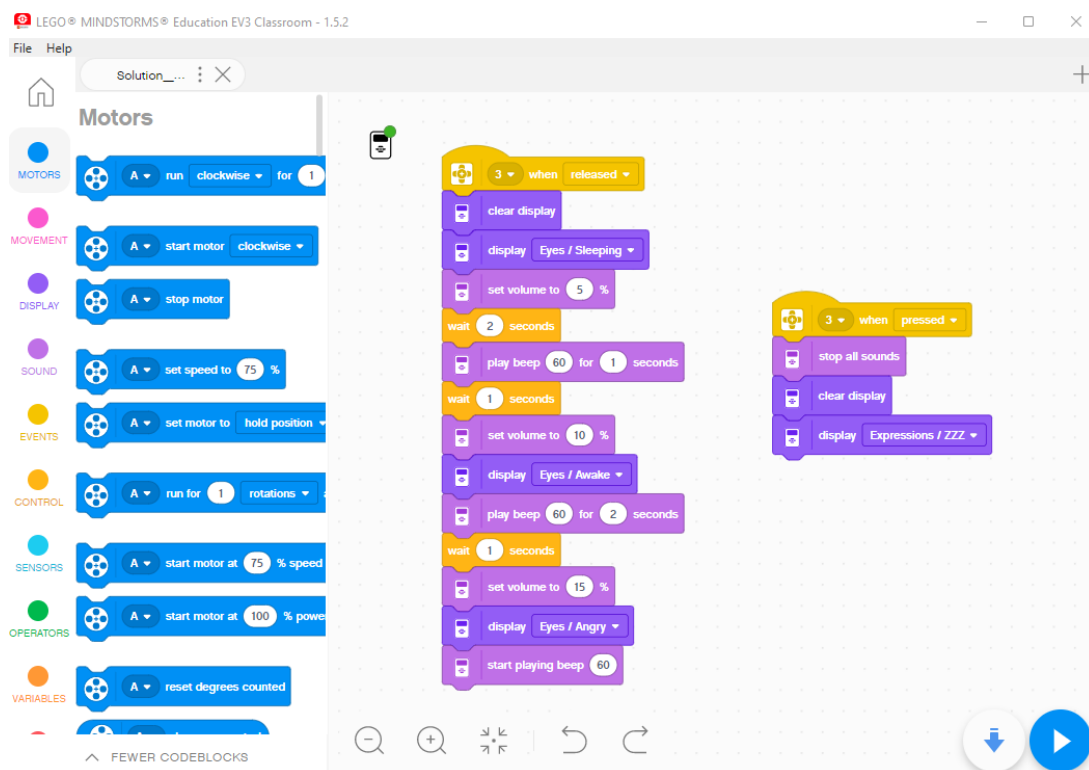
- **Bloky hodnot**

Tyto bloky oválné struktury, udávají numerickou nebo textovou hodnotu. Tato hodnota je určena podle pravidel popsanych v rámci bloku, případně je určena kontextem.

Název sekce	Popis bloků
Motors	Obsluhují konkrétní motor identifikovaný podle připojeného portu.
Movement	Poskytují komplexní pohybové úkony nad dvojicí paralelně zapojených motorů.
Display	Pracují s displejem kontrolní jednotky EV3 Brick.
Sound	Umožňují jednotce vydat zvukový výstup. K dispozici je list předpřipravených zvuků a not.
Events	Korigují spouštění jednotlivých programových stacků. S výjimkou bloku pro zaslání zprávy (broadcast), se jedná o bloky startovní.
Control	Korigují průběh programu podmíněným vykonáváním kódu. Mohou obsahovat vlastní podmínku, případně slot na podmíněný programový blok. Příklad operací: podmínka proved pokud (if) jinak (else), opakuj X krát, . . .
Senzors	Slouží k práci se senzory připojenými k robotovi. Vybraný senzor, se kterým blok může pracovat, je graficky znázorněn ve struktuře bloku.
Operations	Bloky pro matematické, logické a datové operace. Na základě vyhodnocení blok vrací buďto alfanumireckou nebo logickou hodnotu.
Variables	Zaštitují uchovávání a následnou manipulaci dat v rámci programu. Konkrétní proměnné musí být definován uživatelem.
My blocks	Definují sérii operací jako nový uživatelský blok. Představují funkce v rámci programového zápisu EV3.

Tabulka 2: Kategorie bloků EV3 Classroom

Vytvořený program je možno okamžitě spustit v jednotce, pokud došlo k jejímu připojení v UI vývojového prostředí EV3 Classroom. Popřípadě je možné program jako soubor ve formátu .lmst uložit do vnitřního úložiště jednotky Brick. Následné spuštění programu proběhne zvolením programu v adresáři. Jednotlivé soubory jsou identifikovatelné podle jména. Je vhodné každý program stručně a zřetelně pojmenovat.



Obrázek 6: Prostředí EV3 Classroom

### 4.6.3 Python 3

Python je vysokoúrovňový programovací jazyk. Podporuje objektově orientované programování, dynamické typování, volání metod z knihoven z jazyka C/C++. Nejen díky tomuto nachází široké uplatnění, jak ve vývoji moderního software, tak i v integraci starých či primitivních zařízení a systémů. Vývoj Pythonu je open source a většina jeho knihoven je k dispozici zdarma. Jedná se o velice populární jazyk, jak ve vědecké, tak i v komerční sféře.[17]

Základem, respektive poznávacím znakem Pythonu je jednoduchý, silně abstraktní zápis programu a jeho nezaměnitelná struktura. Dělení jednotlivých vnořených sekcí kódu Pythonu je založeno na jeho tabulaci, tedy odsazení.

Python přináší do výuky jednoduchý programovací jazyk, ve kterém není potřeba téměř vůbec přemýšlet nad typováním. Díky své syntaxi do značné míry připomíná vizuální zápis jazyka EV3. Práce s poli a proměnnými je pro studenty

kurzu po předchozích zkušenostech známá. Stejně tak známé programové konstrukce představují podmíněné výrazy a cykly. Definice funkcí je prostá a její výuka se opírá o znalost "my blocks", které byly přímou reprezentací funkcí v EV3. Zápis a využití tříd je nejkompaktnější část jazyka, s kterou se studenti setkají. Zde je ovšem značnou nevýhodou, vůči které musela být výuka přizpůsobena, rozdělení metod na instancí a metody tříd. Rovněž simulovaný zápis *privátních* atributů třídy není zmíněn a vysvětlen v základní formě výukového plánu.

#### 4.6.4 MicroPython

Jazyk MicroPython je implementace jazyka Python určená specificky pro tzv. mikrokontrolery. MicroPython se skládá z kompilátoru Pythonu pro bajtkód a běhového interpretu tohoto bajtkódu. [18] Jelikož se jedná o implemetaci, syntaxe programu je velice blízká standartnímu Pythonu 3. V rámci MicroPythonu bude využita knihovna *pybricks*, která poskytuje veškerou funkcionalitu potřebnou k výuce.

Vytváření a spouštění programu na robotu je usnadněno díky rozšíření ev3dev-browser do Visual Studio Code. Díky němu je možné jednoduše připojit jednotku Brick operující na systému Brickman (ev3dev-stretch). Připojení probíhá pomocí tlačítka "Connect here to connect device" a zvolením jednotky. Propojení je signalizováno zelenou kontrolkou v prostředí. Následné spuštění programu spočívá ve spuštění debugovací sekvence, případně zmáčknutí tlačítka *F5*. Pro tvorbu programů je využito rozšíření LEGO MINDSTORMS EV3 MicroPython, jenž poskytuje vytvoření nového vzorového projektu. V rámci projektu je vytvořen soubor *main.py* s importovanou funkcionalitou robota.

#### 4.6.5 EV3Dev

Operační systém pro jednotku. Jeho instalace probíhá pomocí návodu, který je k nalezení na stránkách dokumentace MicroPython.[4] Zde uvedená verze operačního systému má garantovanou podporu od společnosti Lego. Nicméně je zde využita alternativa ve formě ev3dev-stretch. S ohledem na potencionální využití jednotky v rámci jiných výukových plánů, je praktické, aby byla poskytnuta co nejširší funkcionalita. Zmíněný systém je k dispozici na komunitním webu <https://www.ev3dev.org/downloads/> jako ev3dev-stretch.[19] Orientace v spuštěném operačním systému na jednotce je intuitivní a do značné míry kopíruje výchozí operační systém. Spuštění operačního systému ev3dev-stretch vyžaduje spuštění jednotky Brick s vloženou MicroSD kartou obsahující zmíněný systém. Po prodlevě a statusovém výpisu na displeji jednotky se spustí systém s označením BrickMan a jednotku lze používat. Je důležité ponechat kartu po dobu používání kartu v jednotce. Po vypnutí je možné kartu vyjmout. Bootování do výchozího operačního systému se provede automaticky, vždy když není v jednotce vložena bootovací karta.

#### 4.6.6 Vývojová prostředí

V rámci výukového plánu jsou využity dvě vývojové prostředí, EV3 Classroom a Visual Studio Code.[23] Pro plynulý průběh výuky musí být obě prostředí nainstalována ve funkčním stavu pro studenty. Dle vlastního uvážení je možnost se studenty provést instalaci prostředí případně i rozšíření. To je obzvláště doporučeno u prostředí Visual Studio Code, které pravděpodobněji najde uplatnění v domácím programování studentů. Seznámení s prostředím a orientace v něm je vítální pro úspěšnou práci v rámci patřičných lekcí. Je potřeba dbát důraz ve Visual Studio Code, které může oproti EV3 Classroom studenty zahltit informacemi a dostupnou funkcionalitou.

#### 4.7 Cíl výuky

Výukový plán cílí na několik výstupů po jeho absolvování. Hlavní cíl představuje zaujmout studenty. Pomocí robotické stavebnice LEGO MINDSTORMS EV3 Education a interaktivního výstupu, motivovat k dalšímu získávání znalostí z oblasti programování a robotiky. Dalším cílem je seznámit se s základy programování. Jedná se sice o opravdové základy, ale zase důkladně prakticky vyzkoušené napříč zápisy a reprezentacemi programu. V neposlední řadě se plán snaží o rozvoj kritických schopností logického uvažování, analýzy problémů a samostatné práce podle přiložené dokumentace a materiálů.

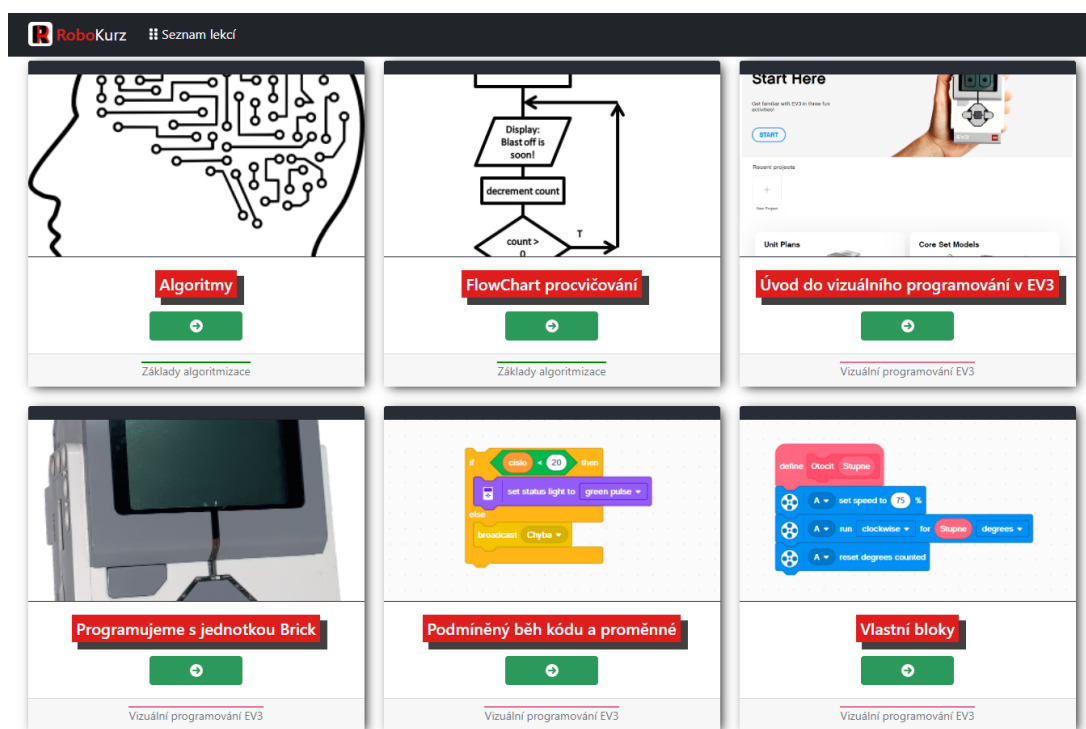
Student, který projde výukou, má přehled o stavebnici LEGO MINDSTORMS EV3 Education, dokáže zapsat algoritmus pomocí diagramu FlowChart. Rozumí základním programovacím konceptům jako jsou proměnné, operátory, pole, cykly, podmíněné výrazy, funkce a má strohý pojem o třídách a instancích. Orientuje se v prostředích EV3 Classroom a Visual Studio Code. Dokáže ovládat robota pomocí programovacích jazyků EV3 a MicroPython. Zároveň umí vyřešit jednoduché úlohy v jazyce Python a pracovat s terminálem.

## 5 Webová aplikace RoboKurz

Webová aplikace vytvořená jako podpůrný materiál pro výukový plán. Její forma i účel supplementuje interaktivní elektronický dokument. Slouží k prezentaci základní podpůrné teorie, návodů, praktických ukázek a úloh pro studenty.

### 5.1 Popis webu

Základní kostrou webové aplikace je obsahová plocha, horní navigační lišta menu a spodní panel, tzv. footer. V rámci webové aplikace jsou k dispozici 2 vizuální stavy: výběr lekcí a zobrazení konkrétní lekce. Výběr lekcí je výchozí stav obsahové plochy načteného webu. Lekce jsou zobrazeny jako dlaždice. Po výběru konkrétní dlaždice zmáčknutím tlačítka, je uživateli zobrazen obsah dané lekce. Zobrazení lekce má jednotnou strukturu. Skládá se z nadpisu, textu lekce, popisu úloh a stránkovací lišty. Obsah jednotlivých lekcí a úloh je místy doplněn obrázky případně programovým kódem vloženým ve webové komponentě.

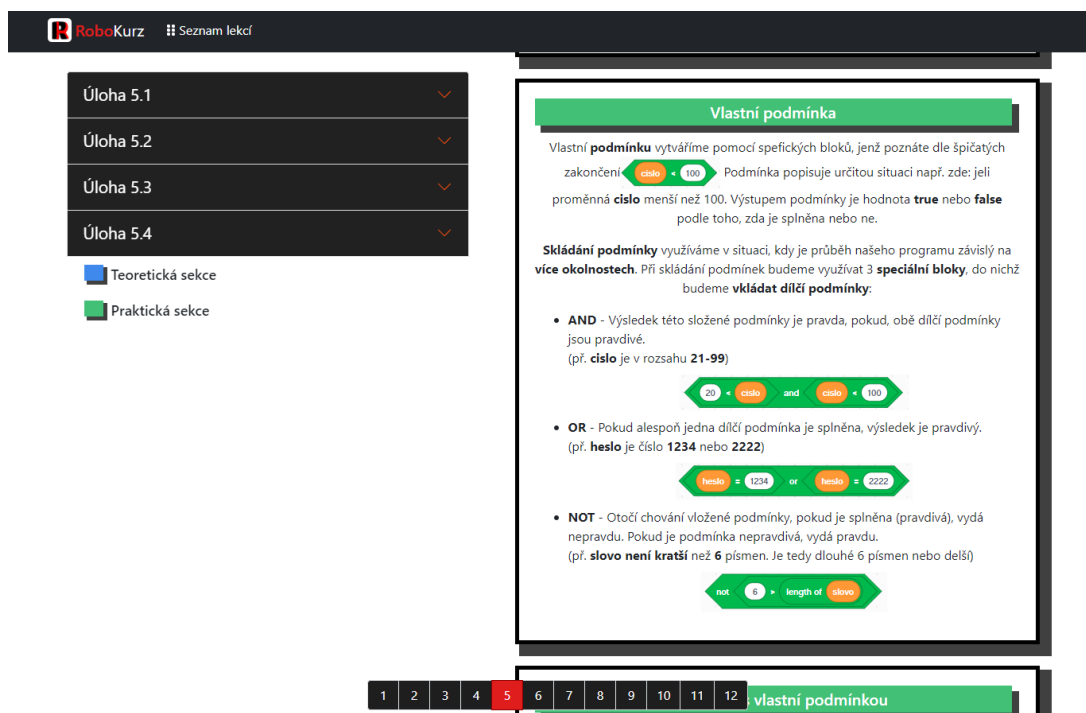


Obrázek 7: Webová aplikace RoboKurz - výběr lekcí

### 5.2 Použité technologie

Po technologické stránce je webová aplikace RoboKurz tzv. single page application<sup>3</sup>, vytvořená s pomocí knihovny react v prostředí node.js.

<sup>3</sup>Aplikace dynamicky překresluje obsah v rámci jedné webové stránky.



Obrázek 8: Webová aplikace RoboKurz - ukázka lekce

### 5.2.1 HTML

Jedná se o značkovací jazyk, který popisuje sémantiku webové stránky. HTML soubor se skládá z množiny párových značek. Uzavřením obsahu mezi dvě značky dosáhneme přiřazení významu. Strukturálně se standardní HTML soubor skládá z hlavičky s popisem souboru a těla s jeho obsahem. Jelikož je aplikace jednostránková, je v rámci zdrojů využit pouze jediný soubor ve formátu html, reprezentující celý web. Nicméně kód zapsaný ve formě HTML je využit skrz jednotlivé komponenty a obsahy lekcí v rámci React komponent a JavaScriptových souborů. Tento kód je poté dynamicky vkládán do zmíněného souboru.[24]

Dynamické vkládání kódu do HTML je umožněno díky DOM neboli domain object model. Jedná se o objektově orientované rozhraní, která reprezentuje strukturu značkovacího jazyka (HTML nebo XML) jako strom.[25] Tato stromová reprezentace má kořen v tzv. root elementu, což je označení souboru dvojicí značek html. Struktura se následně větví do uzlů, ze kterých mohou vycházet zase větve. Každá větev přitom musí končit v uzlu. Pojem uzel reprezentuje objekt dokumentu, kterým může být pár značek, hodnota nebo například atribut.

### 5.2.2 CSS

Jazyk popisující vizuální vzhled webové stránky. CSS styluje jednotlivé html elementy, jenž jsou vybrány pomocí různých HTML značek, jejich atributů a souvisejících vztahů.[26] CSS se stará o grafický vzhled, umožňuje ovlivnit barvu,



velikost, ohraničení nebo dokonce animace. Stylovat lze buďto přímým stylováním, kdy jsou styly umístěny v html souboru nebo elementu (případně formou vlastnosti pro React kompenntu). Anebo referencí stylového souboru ve formátu CSS.

### 5.2.3 JavaScript

JavaScript je objektově orientovaný programovací jazyk. Poskytuje dynamickou funkcionalitu na webové stránce. Pracuje jak na klientské stanici, tak i na webovém serveru. Dodává logiku, díky JavaScript lze v rámci webové stránky reagovat na stisk tlačítka změnou změnou obsahu, zasláním formuláře atp. Funkcionilta JavaScriptu je dodána buď přímým vložením skriptů nebo referencováním skriptových souborů.[30]

### 5.2.4 React

Knihovna postavená na jazyku JavaScript, jenž umožňuje vytvářet znovupoužitelné komponenty, obstarává cílovou podobu webu a dynamicky překresluje DOM webové stránky. Programátor pracuje s tzv. virtuálním DOM jenž je reprezentací požadové struktury DOM webu a je uložen v paměti. Knihovny React detekují rozdíly mezi virtuálním DOM a DOM webu. Při detekování rozdílu se provede aktualizace DOM webu v patřičných částech jeho stromové struktury.[29]

#### Komponenty v rámci aplikace

- Code-block - Umožňuje stručnější použití komponenty CodeBlock z knihovny react-code-blocks s konkrétním nastavením. Vytvoří objekt se zapsaným strukturovaným zdrojovým kódem a tlačítkem pro zkopírování.
- Lesson - Vytváří strukturovaný zápis lekce a rozmístění dat na základě vstupních atributů. Vykreslí nadpis s označením, tzv. Accordeon s jednotlivými úkoly a zobrazí teorii lekce.
- Menu - Navigační lišta aplikace. Obsahuje logo a tlačítko pro zobrazení seznamu lekcí.
- Footer - Jednoduchá statická komponenta pro zobrazení patičky webu.
- Title-card - Reprezentuje kartu lekce v rámci přehledového výpisu. Na základě vstupu zobrazí příslušný obrázek, nadpis a tlačítko s přesměrováním na konkrétní lekci.
- Download-card - Prostá komponenta pro stručnější reprezentaci ikony stažení s popisem.

- Body-holder - Vykresluje tělo aplikace. Zpracovává stránkovací logiku. Na základě vnitřních hodnot zobrazí buďto seznam lekcí nebo obsah konkrétní lekce.

### 5.2.5 Node.js

Prostředí pro tvorbu webových aplikací v jazyce JavaScript. Aplikace vytvořeny pomocí node.js umožňují tvorbu webu s dynamickým obsahem. Kód generovaný pomocí node.js pracuje na straně serveru. Webová aplikace pracuje na jednom vlákně v cyklu, jenž spravuje připojení a na základě nových připojení vyvolává události. Primární výhodou node.js je, že aplikace vytvořené v prostředí jsou asynchronní a neblokující.[31]

Knihovny využité pro webovou aplikaci:

**react**

**react-bootstrap**

**react-code-blocks**

**react-dom**

**react-scripts**

**knihovny ikon z fontawesome**

**bootstrap**

Testovací:

**web-vitals**

**testovací knihovny z testing-library**

Webová aplikace je sestavena pomocí node.js v produkční verzi. Tím je zdrojový kód transformován a optimalizován.

### 5.2.6 Docker

Docker je virtualizační platforma. Umožňuje spouštění minimalistických virtuálních strojů (Linux/Windows) s předpřipravenou konfigurací, aplikacemi a daty.[32] Pro testovací účely webové aplikace byl vytvořen obraz pro Linuxový virtuální stroj. Základ virtuálního stroje tvoří lts verze alpine node.js.[33]

### 5.2.7 Markdown

Markdown je jednoduchý značkovací jazyk. Základá si na čitelné syntaxi pomocí intuitivních značek, umožňuje však zakomponovat v rámci kódu také značky HTML. Rozšířeným použitím Markdown je tvorba textových souborů, primárně dokumentací, které jsou posléze snadno převeditelné do HTML a publikovatelné.[34]

## Závěr

Výstupem bakalářské práce je komplexní výukový plán: *Plán výuky programování s pomocí Lego Mindstorms EV3 pro 1. ročník SŠ* a pomocná webová aplikace: *RoboKurz*.

Výukový plán se zaměřuje na úvod do programování. Studenty provede skrz vývojová prostředí *EV3 Classroom* a *Visual Studio Code*. Studenti se seznámí se zápisem programu resp. algoritmu pomocí diagramů *FlowChart* a programovacích jazyků *EV3* a *Python*. Ve prospěch výuky je aktivně využita stavebnice *LEGO MINDSTORMS EV3 Education*. Plán prošel před vznikem důkladnou analýzou, zhodnoceny byly možnosti využití stavebnice, požadavky na výukový plán a jeho začlenění. Byly stanoveny klíčové znalosti a oblasti, kterými se plán zabývá. Spolu s plánem byly vytvořeny pomocné materiály a cvičební úlohy. Byla analyzována a zvolena vhodná forma prezentace vytvořených materiálů. Webová aplikace *RoboKurz* slouží jako základní opora jednotlivých lekcí plánu. Poskytuje prostor pro teoretickou část lekcí, návody, ukázky a zadání úloh.

Za hlavní přínosy práce považuji zaměření plánu na praktické úkoly s interaktivním výstupem díky LEGO stavebnici. A také získané znalosti prostředí *Visual Studio Code* a programovacího jazyka *Python*. Mezi zápory se řadí absence praktického vyzkoušení výuky na základě plánu, alespoň v rámci školního či mimoškolního kroužku. Dalším záporem je neanalyzovaná možnost virtuální práce s robotem, při nedostatku stavebnic pro výuku.

Jak samotný plán, tak i pomocné materiály nabízejí rozšíření. V rámci výukového plánu může být upraven počet lekcí a jejich pořadí, za účelem vzájemně výlučného využití stavebnice mezi dvě rozdělené skupiny při půlení výuky. Příklady použité v lekcí mohou být doplněny vlastními či ze zdrojů třetí strany. Rovněž se nabízí rozšíření výuky ostatních přírodovědeckých předmětů s návazným využitím programovacích znalostí.

## Conclusions

The output of the bachelor's thesis is a comprehensive teaching plan: *Plan for teaching programming with the help of Lego Mindstorms EV3 for the 1st year of high school* and an auxiliary web application: *RoboKurz*.

The curriculum focuses on an introduction to programming. Students will be guided through the development environments *EV3 Classroom* and *Visual Studio Code*. Students will get acquainted with the enrollment of the program or algorithm using *FlowChart* diagrams and *EV3* and *Python* programming languages. The kit *LEGO EV3 Mindstorms* is actively used for the benefit of teaching. The plan was thoroughly analyzed before its creation, the possibilities of using the kit, the requirements for the curriculum and its incorporation were evaluated. Key knowledge and areas covered by the plan have been identified. Auxiliary materials and exercise tasks were created along with the plan. An appropriate form of presentation of the created materials was analyzed and chosen. The web application *RoboKurz* serves as a basic support for individual lessons of the plan. Provides space for the theoretical part of lessons, tutorials, demonstrations and assignments.

I consider the main benefits of the work to focus the plan on practical tasks with an interactive output thanks to the LEGO kit. And also the acquired knowledge of the Visual Studio Code environment and the Python programming language. Disadvantages include the absence of practical testing of teaching on the basis of a plan, at least within a school or out-of-school circle. Another negative is the unanalyzed possibility of virtual work with the robot, in the absence of kits for teaching.

Both the plan itself and the auxiliary materials offer extensions. Within the curriculum, the number of lessons and their order can be adjusted in order to use the kit between the two divided groups differently when halving the lesson. The examples used in the lesson can be supplemented by own or third party sources. There is also an offer to expand the teaching of other science subjects with the subsequent use of programming knowledge.

## A Obsah příloženého CD

### **doc/**

V této složce je uložen pdf soubor s textem bakalářské práce (BP-svanno00.pdf). Zároveň jsou zde obsaženy zdrojové soubory použité pro vygenerování tohoto souboru.

### **sylabus/**

Obsahuje dokument výukového plánu spolu s programovými soubory řešení jednotlivých úloh, instalačními návody a 3D modely z výuky.

### **webapp/source/**

Zdrojové soubory webové aplikace RoboKurz, určené pro vývoj a úpravy.

### **webapp/build/**

Statically vygenerované soubory webové aplikace RoboKurz pro okamžité nasazení na webovém serveru. Soubory jsou nakonfigurovány pro vykopírování do kořenového adresáře.

### **readme/**

Popis a obsah jednotlivých složek. Instrukce pro použití u složek, kde je to žádané.

## Literatura

- [1] LEGO MINDSTORMS Education EV3 [online]. The LEGO Group [cit. 2022-04-28]. Dostupné z: <https://education.lego.com/en-us/product-resources/mindstorms-ev3/teacher-resources/classroom-materials>
- [2] PARK, Eun Jung. Lego Mindstorms EV3: stavíme a programujeme roboty. Brno: Computer Press, 2015. ISBN 9788025143858.
- [3] LEGO Education kurzy [online]. The LEGO Group [cit. 2022-04-10]. Dostupné z: <https://makecode.mindstorms.com/tutorials>
- [4] LEGO Education EV3 MicroPython [online]. The LEGO Group [cit. 2022-04-10]. Dostupné z: <https://pybricks.com/ev3-micropython/>
- [5] Microsoft MakeCode [online]. The LEGO Group [cit. 2022-04-28]. Dostupné z: <https://makecode.mindstorms.com/>
- [6] SKRZYPEK, Dominika a Ola SYROCKA. EV3 Classroom or EV3 Lab? MINDSTORMS Programming Apps Comparison [online]. RoboCamp [cit. 2022-04-28]. Dostupné z: <https://www.robocamp.eu/en/blog/lego-mindstorms-ev3-classroom-app/>
- [7] Open Roberta Lab [online]. Roberta® – Learning with Robots [cit. 2022-04-28]. Dostupné z: <https://lab.open-roberta.org/>
- [8] Scratch [online]. Scratch Foundation [cit. 2022-05-01]. Dostupné z: <https://scratch.mit.edu/about>
- [9] NASA [online]. National Aeronautics and Space Administration [cit. 2022-04-28]. Dostupné z: <https://www.nasa.gov/about/index.html>
- [10] HEBERT, Phillip, Sr. a Alex C. ELLIOT. NASA Data Acquisition System Software for Rocket Propulsion Testing [online]. NATIONAL INSTRUMENTS [cit. 2022-04-28]. Dostupné z: <https://www.ni.com/cs-cz/innovations/case-studies/19/nasa-data-acquisition-system-software-for-rocket-propulsion-testing.html>
- [11] PRAHA, VÚP. ICT v matematice. Metodický portál: Články [online]. 10. 09. 2008, [cit. 2022-04-15]. Dostupný z WWW: <https://clanky.rvp.cz/clanek/2500/ICT-V-MATEMATICE.html>. ISSN 1802-4785.
- [12] ZORMANOVÁ, Lucie. Výukové metody v pedagogice: Třífázový model učení. Metodický portál: Články [online]. 09. 07. 2012, [cit. 2022-04-28]. Dostupný z WWW: <https://clanky.rvp.cz/clanek/16247/VYUKOVE-METODY-V-PEDAGOGICE:-TRIFAZOVY-MODEL-UCENI.html>. ISSN 1802-4785.

- [13] SKALKOVÁ, Jarmila. Obecná didaktika: vyučovací proces, učivo a jeho výběr, metody, organizační formy vyučování. Praha: Grada, 2007. Pedagogika (Grada). ISBN 978-80-247-1821-7.
- [14] POKORNÁ, Věra. Rozvoj vnímání a poznávání. Praha: Portál, 2000. ISBN 80-7178-400-1.
- [15] Rámcové vzdělávací programy pro střední vzdělávání [online]. Ministerstvo školství, mládeže a tělovýchovy ČR [cit. 2022-04-29]. Dostupné z: <https://www.msmt.cz/vzdelavani/stredni-vzdelavani/ramcove-vzdelavaci-programy>
- [16] Ministerstvo školství, mládeže a tělovýchovy ČR [online]. [cit. 2022-04-29]. Dostupné z: <https://www.msmt.cz/ministerstvo/zakladni-informace>
- [17] Python Documentation [online]. Python Software Foundation [cit. 2022-04-10]. Dostupné z: <https://docs.python.org/3/>
- [18] MicroPython Documentation [online]. Damien P. George, Paul Sokolovsky, and contributors. [cit. 2022-04-24]. Dostupné z: <https://docs.micropython.org/en/latest/#>
- [19] Ev3dev [online]. Open community [cit. 2022-04-28]. Dostupné z: <https://www.ev3dev.org/>
- [20] Rozšíření pro VS Code: ev3dev-browser [online]. Microsoft [cit. 2022-05-01]. Dostupné z: <https://marketplace.visualstudio.com/items?itemName=ev3dev.ev3dev-browser>
- [21] Rozšíření pro VS Code: LEGO MINDSTORMS EV3 MicroPython [online]. Microsoft [cit. 2022-05-01]. Dostupné z: <https://marketplace.visualstudio.com/items?itemName=lego-education.ev3-micropython>
- [22] WHAT IS A FLOWCHART? [online]. American Society for Quality, Adapted from The Quality Toolbox, Second Edition, ASQ Quality Press [cit. 2022-04-28]. Dostupné z: <https://asq.org/quality-resources/flowchart>
- [23] Visual Studio Code Documentation [online]. Microsoft [cit. 2022-04-28]. Dostupné z: <https://code.visualstudio.com/docs>
- [24] W3C HTML Living standart [online]. [cit. 2022-04-11]. Dostupné z: <https://html.spec.whatwg.org/multipage/>
- [25] W3C DOM Living standart [online]. [cit. 2022-04-11]. Dostupné z: <https://dom.spec.whatwg.org/>
- [26] Mozilla Developers CSS [online]. [cit. 2022-04-11]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

- [27] Dokumentace k jazyku C# [online]. Microsoft [cit. 2022-04-28]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/csharp/>
- [28] Java Documentation [online]. Oracle [cit. 2022-04-28]. Dostupné z: <https://docs.oracle.com/en/java/>
- [29] React Docs [online]. Meta Platforms, Inc. [cit. 2022-04-07]. Dostupné z: <https://reactjs.org/docs/getting-started.html>
- [30] PEHLIVANIAN, Ara a Don NGUYEN. JavaScript okamžitě. Brno: Computer Press, 2014. ISBN 978-80-251-4163-2.
- [31] Node JS Documentation [online]. OpenJS Foundation [cit. 2022-04-25]. Dostupné z: <https://nodejs.org/en/docs/>
- [32] Docker Documentation [online]. Docker [cit. 2022-04-25]. Dostupné z: <https://docs.docker.com/>
- [33] DockerHub - node [online]. Docker [cit. 2022-05-01]. Dostupné z: [https://hub.docker.com/\\_/node](https://hub.docker.com/_/node)
- [34] Markdown Guide [online]. A Matt Cone project [cit. 2022-05-01]. Dostupné z: <https://www.markdownguide.org/>