

# VÝUKA INFORMATIKY A PODPORA INFORMATICKÉHO MYŠLENÍ POMOCÍ LEGOROBOTŮ NA GYMNÁZIÍCH

## TEACHING COMPUTER SCIENCE AND SUPPORT OF COMPUTATIONAL THINKING BY LEGO ROBOT AT HIGH SCHOOL

Filip Frank

### Abstrakt

Příspěvek se zabývá výukou informatiky a možnostmi podpory informatického myšlení pomocí legorobotů na gymnáziích. Zabývá se definováním informatického myšlení a podmínkami pro jeho podporu. Došlo k vysvětlení pojmu „legorobot“ a k návrhu jeho možností, výhod a nevýhod využití ve výuce. Na závěr byla navržena metodika pro podporu informatického myšlení pomocí legorobotů. Tato metodika byla otestována na gymnáziu v Aši a výsledky popsány v případové studii.

**Klíčová slova:** *Informatické myšlení, lego, legorobot, metodika*

### Abstract

The paper deals with the teaching of computer science and the possibility of support of computational thinking by usage of legorobots at high schools. It deals with the definition of computational thinking and the ways for its support. The paper explains the term "legorobot" and suggests its possibilities, advantages and disadvantages during computer science lessons. Finally, paper tries to make methodology. Methodology suggests ways how to support computational thinking during computer science lessons by legorobots. This methodology was tested at high school at Aš and the results were described in a case study.

**Key words:** *Computational thinking, lego, legoroboti, methodology*

## 1 Úvod

Příspěvek vychází z diplomové práce s názvem „Výuka informatiky a podpora informatického myšlení pomocí legorobotů na gymnáziích“, která vznikla v roce 2018 na Katedře výpočetní a didaktické techniky Fakulty pedagogické Západočeské univerzity v Plzni, a je věnován výuce informatiky a podpoře informatického myšlení pomocí legorobotů u studentů středních škol. Zmíněná diplomová práce se podrobně zabývá otázkou informatického myšlení, jeho možnými definicemi, podmínkami a možnostmi podpory na základě odlišných přístupů a podmínek stanovených různými autory. Z těch je nutné zmínit Paula Wanga [6] a Jeanet M. Wingovou [7]. Jednou z možných definic informatického myšlení je též společná definice ISTE a CSTA. Zmíněné tři přístupy byly porovnány, aby vznikla jedna definice a společné podmínky informatického myšlení a jeho rozvoje.

Dále jsou představeni legoroboti. Je třeba vymezit si nejen jejich možnosti jako takové, ale i jejich možnosti ve výuce a případné výhody a nevýhody jejich použití. Opět jsou legoroboti hodnoceni s ohledem na informatické myšlení s cílem stanovit, zda je vůbec možné jejich využití při podpoře informatického myšlení.

V rámci diplomové práce vznikla též metodika, která prezentuje způsob, jakým je možné používat legoroboty ve výuce informatiky a podpoře inforatického myšlení u studentů středních škol. Metodika byla testována na gymnáziu v Aši. Z testování byla následně sepsána případová studie, která hodnotila nejen metodiku, ale i možnosti podpory inforatického myšlení pomocí legorobotů.

Vzhledem k omezenému rozsahu příspěvku bylo nutné některé části původní diplomové práce zestručnit, případně vynechat. Celý text, podrobné informace i kompletní výsledky testování nalezne čtenář v [3].

## 2 Inforatické myšlení

Protože definic, co to inforatické myšlení je můžeme nalézt nespočet, nabízí se možnost srovnání a z jednotlivých přístupů se pokusit formulovat ten nejlepší závěr. Pro potřeby práce se spokojíme se třemi přístupy. Dva z přístupů budou autorské a jeden reprezentuje konsensus celého společenství.

### 2.1.1 DEFINICE DLE PAULA WANGA

Paul Wang v knize „From computing to computational thinking“ předkládá definici pomocí definování slova „Computize“. „Computize, verb. To apply computational thinking. To view, consider, analyze, design, plan, work, and solve problems from a computational perspective [6].“ Znamená to tedy, že pokud chceme využívat inforatické myšlení, budeme podle Wanga zkoumat, analyzovat, navrhovat, plánovat, pracovat a řešit problémy z výpočetní perspektivy.

Je zde patrné logické uspořádání kroků a rozložení problému na drobnější podproblémy, které se řeší snadněji.

Zkoumáním a analyzováním myslíme, že je potřeba nahlédnout na problém ze všech možných úhlů a objevit všechny možnosti, které nám problém nabízí. To, že nebyla nějaká možnost řešení problému zmíněna, ještě neznamená, že nemůžeme toto řešení použít.

Navrhování pak spočívá v hledání právě zmíněných řešení. Zároveň tato řešení musíme podrobovat kritickému myšlení a rozhodnout, které řešení je nejlepší.

Plánováním myslíme posloupnost kroků, která je dopředu dána a kterou bychom měli dodržet. Nakonec naplánovaný postup zrealizujeme a problém vyřešíme.

Výpočetní perspektivou je pak právě náš celý nastíněný postup, kdy se snažíme předvídat všechny, i skryté, problémy, které mohou nastat, předem se na ně připravujeme a dokážeme je efektivně vyřešit.

Wang pak také předkládá seznam důležitých aspektů pro inforatické myšlení [6]:

- Zjednodušení a abstrakce – schopnost ignorovat nepodstatné detaily. (Wang předkládá příklad, kdy řidiče automobilu nezajímá, jak automobil funguje, ale zajímá ho, jak se řídí)
- Automatizace.
- Znovuvyužití postupu.
- Pozornost k detailům.
- Srozumitelné a přesné instrukce.
- Objektivní, až chladná logika.

- Opustit bublinu – komunikovat na takové úrovni, aby nám rozuměl příjemce.
- Předvídat problémy.

### 2.1.2 DEFINICE DLE JEANETT M. WING

Jeanett M. Wing, která popsala infromatické myšlení už ve svém článku z roku 2006, jej přináší do nových souvislostí. Zatímco Wang orientoval celé myšlení na odborníky z oboru IT, Wingová problematiku rozšiřuje na problémy běžného života a generalizuje tak myšlení na celý život [7].

Originální znění definice Wingové zní takto: „Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information processing agent [7].“

Volně přeloženo pak definice říká, že infromatické myšlení je složitý myšlenkový proces zabývající se formulací problémů a formulací řešení problémů takovým způsobem, aby tato řešení mohla být využita zvoleným agentem pro jejich zpracování.

Řešení, která Wingová zmiňuje, mohou dle ní být provedena počítačem, člověkem, nebo jejich kombinací.

Nicméně Wingová upozorňuje, že jí předkládané problémy nevztahuje pouze na matematické problémy, které je možné propočítat. Problémy mohou být daleko komplexnější a je možné říci i hmatatelnější. Z toho důvodu se stává nedílnou součástí infromatického myšlení algoritmicke myšlení a paralelní myšlení.

Nedílnou součástí infromatického myšlení je abstrakce, která nám dává možnost představit si problém a tím ho zjednodušit pro naši mysl. Například algoritmus je abstrakce výstupů, vstupů, výpočetních operací a podmínek, které nakonec vedou k požadovanému cíli. Programátoři používají abstrakci kdykoli si představují jednotlivé vrstvy svého programu, aby tak dokázaly přemýšlet nad konkrétním problémem svojí práce a nebyli rušeni v tu chvíli nepotřebnými záležitostmi [7].

Při pohledu na první dvě definice vidíme podobnosti v přístupu k problému. Oba autoři Přístupují k problému tak, že je potřeba jej analyzovat a zvolit řešení. Rozdíl Wingové oproti Wangovi však tkví především v tom, že Wang se zcela nezabývá využitelností infromatického myšlení širší veřejností a zůstává především v odborné společnosti.

### 2.1.3 DEFINICE CSTA A ISTE

Computer Science Teachers Association (CSTA) je členská organizace založena roku 2004, která si klade za cíl všestranně pomoci učitelům, zejména učitelům informatiky. Pomáhá jim dále se vzdělávat a navrhuje náplně hodin. Momentálně má CSTA zhruba 25 000 členů ve 145 zemích. Členy jsou učitelé ze všech stupňů vzdělávání, od základních škol po vysoké. Zároveň spolupracuje organizace s celými fakultami, nebo školami [1].

International Society for Technology in Education (ISTE) je mezinárodní společnost zabývající se využitím technologií ve vzdělávání. Jedná se o globální společenství pedagogů, kteří podporují využívání technologií ve výuce. Společnost vytváří návrhy, jakým je možné využívat technologie a pomáhá pedagogům se zařazením technologií do své výuky. Nabízí příručky, fóra, nebo účast na akcích [4].

CSTA ve společné práci s ISTE definuje informatické myšlení, jako postup řešení problému založený na daných bodech, jejichž počet a náplň však nemusí být konečný a je možné je upravit [2].

Body postupu vedoucí k řešení problému dle CSTA a ISTE [2]:

- Formulace problému umožňující využít počítač k pomoci s jeho řešením.
- Logická organizace dat a jejich analýza.
- Abstraktivní reprezentace dat pomocí modelů, nebo simulací.
- Automatizace řešení pomocí algoritmického myšlení.
- Identifikovat, analyzovat a implementovat možná řešení tak, aby dosažení výsledku bylo co možná nejefektivnější.
- Generalizovat a přenést řešení daného problému na širší rámec problémů.

Zároveň však CSTA a ISTE přidávají ještě seznam osobních vlastností a dispozic, které jsou součástí informatického myšlení [2]:

- Schopnost rychle se adaptovat na změnu.
- Sebevědomí poradit si i se složitými problémy.
- Vytrvalost při práci na složitých problémech.
- Tolerance pro nejednoznačnost.
- Schopnost vyrovnat se s problémem, který nemá jednoznačné řešení.
- Schopnost komunikace s ostatními za účelem dosažení společného cíle, nebo řešení.

V originálním znění se dělení omezilo pouze na pět vlastností [2]:

- „Confidence in dealing with complexity.
- Persistence in working with difficult problems.
- Tolerance for ambiguity.
- The ability to deal with open-ended problems.
- The ability to communicate and work with others to achieve a common goal or solution.”

Při překladu bodů, bylo nutné rozdělit některé z nich na dva, protože čeština nedokázala vyjádřit správný význam bodu.

Body vedoucí k řešení podle CSTA a ISTE jsou velice podobné těm předchozím. Ačkoli nutno dodat, že se opět dle názoru autora přiblížili k odbornější společnosti čili k Wangovu přístupu. Nenacházíme zde tedy na první pohled obecné vyjádření informatického myšlení, jaké nabídla Wingová.

Oproti předchozím autorům však rozšířila CSTA a ISTE seznam o vlastnosti a osobnostní rysy, které napomáhají informatickému myšlení. Těmito osobnostními rysy se informatické myšlení dostalo do obecnější roviny. Je tak možné definovat potřeby informatického myšlení ve spojení s běžnou populací. Díky těmto osobnostním rysům je mimo jiné daleko snazší uvědomit si, jakým způsobem může pedagog ve výuce rozvíjet samotné informatické myšlení.

### 2.1.4 ZÁVĚR DEFINIC

Byly předloženy tři definice, které se snažily objasnit pojem informatické myšlení. Všechny tři mají společné, že problém by měl být analyzován a poté vyřešen nejefektivnější cestou. Rozdíly jsou patrné ve chvíli, kdy se zaměříme na oblast využití informatického myšlení. První z autorů Wang celou věc chápe pouze v kruhu odborné veřejnosti. Jeho příklady abstrakce sice lehce zasahují do reálného světa, ale celkově vzato se vždy vrátí k informatikům.

Wingová a CSTA ve spojení s ISTE vnášejí do problematiky obecnější náhled. Popisují vlastnosti lidí a postupy jsou obecnější.

Za definice a vlastnosti informatického myšlení, které budeme využívat v této práci, bychom tedy při výběru, souhrnu a průniku těch nejlepších návrhů měli považovat člověka s danými vlastnostmi. Těmito vlastnostmi jsou cílevědomost, zodpovědnost, kreativita v řešení problému, rychlá adaptace na změnu, týmovost, odvaha pustit se do problému.

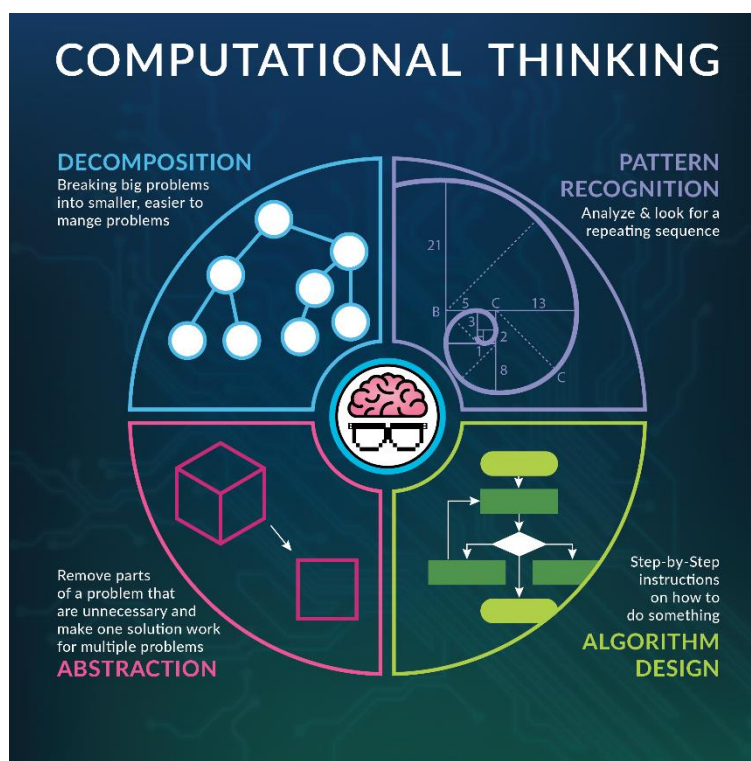
Definici pak využijeme tak, jak ji předkládá Wingová, tedy že informatické myšlení je složitý myšlenkový proces zabývající se formulací problémů a formulací řešení problémů takovým způsobem, aby tato řešení mohla být využita zvoleným agentem pro jejich zpracování. Přičemž nesmíme zapomenout na dodatek Wingové ohledně zaměření problémů, které není pouze striktně matematické, ale může se jednat i o problémy z reálného života [7].

## 2.2 PODMÍNKY INFORMATICKÉHO MYŠLENÍ

Podmínky informatického myšlení jsme si rozebrali už v předchozí kapitole. Obsah té následující především shrnuje jeho zásadní podmínky, kterými jsou:

- Náhled na řešený problém ze všech stran.
- Rozložení složitého problému na více jednodušších problémů.
- Hledání možných opakovaných postupů v řešení problému.
- Očekávání jakékoli nové okolnosti a její vyřešení.
- Schopnost abstrakce.
- Vyřešení problému v relevantním čase.
- Schopnost komunikovat s ostatními tak, aby informacím porozuměli.
- Formulace problému umožňující využít počítač k pomoci s jeho řešením.
- Logická organizace dat a jejich analýza.

Pro lepší představu postupu, který odpovídá informatickému myšlení, je přiložen obrázek (Obrázek 1).



Obrázek 14 – Podmínky informatického myšlení (COMPUTATIONAL THINKERS. [www.computationalthinkers.com](http://www.computationalthinkers.com) [online]. [cit. 17.1.2018]. Dostupný z: <https://www.computationalthinkers.com/wp-content/uploads/2016/01/ComputationalThinkingProductLogo.png>)

Při překladu jednotlivých částí obrázku využijeme určení pomocí kvadrantů, známé z kartézské soustavy, začneme druhým kvadrantem a budeme postupovat po směru hodinových ručiček. Modrá sekce „decomposition“ (II), zachycuje snahu o rozložení složitějšího problému na více jednodušších problémů, jejichž společným vyřešením získáme řešení původního složitějšího problému.

Fialová sekce „patern recognition“ (I) nelze přeložit doslovně. Jedná se o hledání opakujících se kroků řešení. Tento postup můžeme přirovnat k jedné z výhod objektově orientovaného programování. Zmíněnou výhodou je znovuvyužití kódu. V informatickém myšlení však nebudeme mít na mysli pouze naprogramovaný kód, ale jakýkoli stejný sled kroků.

Světle zelená sekce „algorithm design“ (IV) v překladu znamená algoritmický design. Popisuje algoritmický postup, který informatické myšlení využívá. Při řešení problému si totiž jednotlivé kroky musíme vhodně seřadit tak, abychom na konci algoritmu získali řešení problému. Navíc pokud si algoritmus vyjádříme například vývojovým diagramem, můžeme v případě špatného řešení problému poměrně snadno najít v atomizovaných krocích chybu a následně ji opravit.

Poslední částí obrázku je „abstraction“ (III). Českým překladem je abstrakce, kterou zmiňoval už Wang. Jedná se o odstranění částí problému, které pro naše řešení nehrají roli. V případě, že bychom tedy řešili problém, jakým způsobem zapnout domácí spotřebič, například vysavač. Řešení tohoto problému je snadné, zapojíme vysavač do elektrické zásuvky a zmáčkne vypínač na přístroji. Už nás ale nemusí zatím při řešení problému, kde se vzala v zásuvce elektřina, nebo jakým způsobem funguje tlačítko na vysavači.

## 2.3 PODPORA INFORMATICKÉHO MYŠLENÍ

Podpora informatického myšlení probíhá podobně, jako ovlivňování postojů žáků. Není tedy možné rozhodnout se, že se bude v průběhu prvního pololetí školního roku probírat informatické myšlení. Je potřeba dodržovat postupy a podmínky informatického myšlení samotným vyučujícím a vést k nim i žáky. Je tedy jasné, že není možné informatické myšlení podporovat pouze v hodinách informatiky. Zapojit by se měly, pokud možno všechny obory.

Můžeme si všimnout, že některé obory už bezděčně ve své výuce podmínky informatického myšlení dodržují, aniž by to někomu přišlo divné. Například v tělesné výchově si můžeme všimnout rozkladu komplexní pohybové dovednosti na několik menších. Tento rozklad si můžeme ilustrovat například na přeskoce. Máme zde několik fází, rozběhovou, odrazovou, letovou, dopadovou na můstek, odrazovou z můstku, letovou přes překážku, doskokovou. Každou z fází je možné trénovat odděleně a postupně je spojit a vyřešit tím problematiku pohybové dovednosti jako celku. Navíc naučené dílčí dovednosti je možné využít v jiných komplexních dovednostech. V tělesné výchově se pro tento přenos dovedností používá označení transfer. V informatických kruzích bychom mohli tento transfer připodobnit k znovu využitelnosti kódu. Podobným způsobem mohou využívat informatické myšlení i obory, jako čeština nebo matematika a samozřejmě další.

### 2.3.1 PODPORA INFORMATICKÉHO MYŠLENÍ V INFORMATICE

Pro podporu informatického myšlení v hodinách informatiky je možné využít nejrůznější aplikace zabývající se buď samotným programováním, kdy bychom ale raději volili aplikace jako scratch, stencyl, nebo programovací prostředí pro legoroboty. Legoroboty se s ohledem na zaměření práce budeme zabývat později a šířeji.

Všechny tři aplikace, respektive programovací prostředí nabízejí bloky kódu, které žáci mohou propojovat a mísit. Vývojáři se pokusili usnadnit uživatelům práci tím, že tvar a barva jednotlivých bloků napovídají, které další bloky je možné k nim připojit.

Stencyl a scratch oproti prostředí pro lego pak pracují s prvky v podobě obrázků. Uživatelé nastavují objektům vlastnosti, události, aktivity a případně reakce na vnější zásah nebo střetnutí s jiným prvkem.

Jedním z prvních a nejjednodušších příkladů by tedy mohl být úkol, aby žáci přiměli nějaký prvek k posunu až na pravý okraj, kde by se měl zastavit a pomocí zvoleného způsobu informovat uživatele, že dosáhl konce. Uživatel by pak měl dostat na výběr, kam chce jít dál.

Další z možností, jak rozvíjet informatické myšlení mohou být nejrůznější hry. Poměrně populární se stala v tomto směru hra Minecraft. Žáci hru většinou dobře znají. Nicméně už méně žáků ví, že není nutné, aby stavěli všechny své výtvořky manuálně. Žáci si mohou pomoci vyznačením prostoru, který chtějí vyplnit zvoleným materiálem a ušetřit si tak spoustu času. Zadání úkolu by pak mohlo mít podobu požadavku na vystavení budovy školy, nebo jiné veřejné budovy.

## 3 Legoroboti

První model počítačem řízeného lega se objevil už v roce 1986. O významnějším kroku lega můžeme však mluvit v roce 1988, kdy společnost Lego začala spolupracovat s Massachusetským technologickým institutem. Společně začali pracovat na programovatelné „kostce“, kterou by bylo možné propojit s dalšími

součástkami a naprogramovat chování celku počítačem. V lednu v roce 1998 pak bylo představeno „LEGO MINDSTORMS RCX Intelligent Brick and Robotics“ v muzeu moderního umění v Londýně. Ve stejném roce v září pak byl spuštěn robotický výzkum souběžně ve spojených státech a spojeném království. Zároveň vznikla dvě rozšíření původní stavebnice a sice „RoboSports“ a „Extreme Creatures“, což můžeme přeložit jako robosporty a extrémní příšery. Po těchto začátcích pokračoval vývoj poměrně rychle. V prosinci roku 1998 vznikla soutěž mezi středními školami ve využití legoroborů. Prvního ročníku se zúčastnilo 200 studentských týmů. První celosvětovou soutěží „FIRST LEGO League World Championship“ se konal v roce 2005 v Atlantě. V roce 2008 se Lego Mindstorms dostalo do robotické síně slávy Carnegie Mellon University [5].

Dalším významným vývojem prošli legoroboti v srpnu roku 2009, kdy se objevila nová verze Lego Mindstorms NXT 2.0. V lednu roku 2013 oslavili legoroboti 15. výročí uvedením své další verze Lego Mindstorms EV3, jejíž celosvětový prodej byl zahájen v září téhož roku [5].

V praktické části této práce, kde budeme využívat legoroboty, bude využita verze NXT. Zmíněná verze byla využita kvůli většímu počtu stavebnic, které byly v době testování k dispozici k zapůjčení na katedře výpočetní a didaktické techniky při fakultě Pedagogické Západočeské univerzity v Plzni. V programovatelných kostkách byl aktualizován firmware, aby bylo možné používat nejnovější programovací prostředí. Další výhodou využití stavebnice NXT je šance ukázat, že při aktualizaci firmwaru, která je k dispozici po připojení kostky k počítači přímo v programovacím prostředí, je i starší verze stále standardně použitelná. V případě, že škola vlastní verzi NXT není bezpodmínečně nutné zakupovat nové stavebnice.

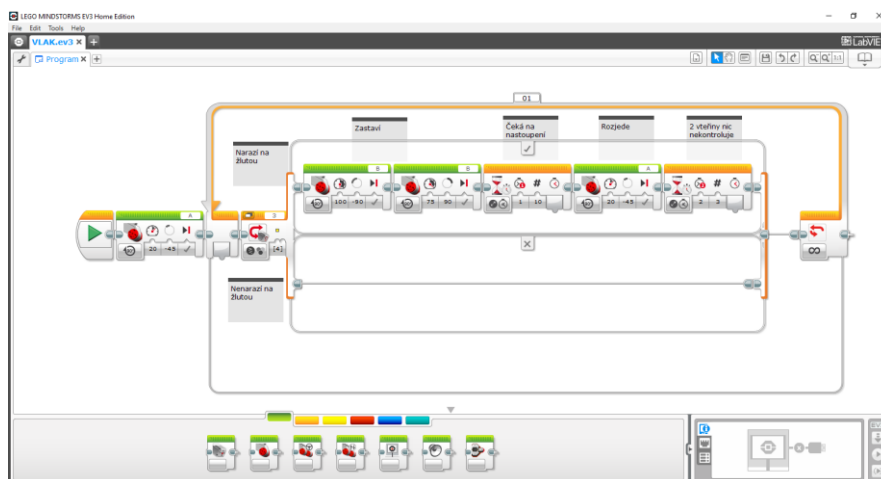
### **3.1 MOŽNOSTI LEGO ROBOTŮ VE VÝUCE A JEJICH MOŽNÉ VÝHODY A NEVÝHODY**

Legoroboti, umožňují rozvíjet nejen programovací dovednosti žáků. Žáci se musí vypořádat s komplexním problémem, podobně jako tomu bude v budoucím životě. Na základě zadaného problému musí projít přes návrh robota, jeho zkonstruování až k jeho naprogramování. S ohledem na cenu stavebnic se navíc dá očekávat, že budou muset žáci spolupracovat v týmech a řešit nedostatek stavebních dílů. Budou tedy muset kreativně a netradičně využít některé díly, které by se standardně použili jinde.

Zásadní výhodou využití legorobotů se může jevit jejich relativně jednoduché zkonstruování a jejich názornost. Žáci budou mít díky široké nabídce různých dílků rychle hotový návrh konstrukce robota. Seznámí se s nejrůznějšími čidly, jako je ultrazvukový senzor, dotykový senzor, světelný senzor, nebo zvukový senzor. Dále je možné rozšiřovat sadu o čidlo rozpoznávající barvy, infra závora nebo gyroskop.

Samostatnou částí Lego mindstorms je programovací prostředí. Žáci se v něm seznámí s objektově orientovaným programováním. Jsou zde pro ně připravené bloky, které je možné na sebe napojovat a nastavovat. Pokročilejší žáci si mohou vytvářet i vlastní bloky. To využijí zejména v případě, že využívají nějakou skupinu příkazů. Dá se říct, že si vytvoří vlastní proceduru, kterou poté znovu využívají. Krom tvorby vlastních bloků jsou přítomny i matematické operace a možnost používat klasické proměnné.





Obrázek 15 – Programovací prostředí (zdroj: vlastní)

Seznam možných výhod využití legorobotů ve výuce:

- názornost,
- relativně snadná konstrukce,
- zábavná výuka,
- řešení problémů z běžného života pomocí legorobota (tvorba robotického vysavače),
- programování hmatatelného objektu.

Seznam možných nevýhod využití legorobotů ve výuce:

- vysoká pořizovací cena (sada EV3 obchod robotworld.cz 8 499 Kč s DPH) [8],
- nesoustředěnost žáků,
- žáci si mohou hrát, ale nic se nenaučí,
- umezenost základní sady,
- vybitá programovatelná kostka.

#### 4 Metodika výuky informatiky podporující informatické myšlení na gymnáziích

Mezi cíle budou jednoznačně patřit dílčí kroky vedoucí ke schopnosti žáků pracovat s legoroboty. Žáci budou muset nejprve poznat, jaké mají možnosti při práci se stavebnicí. Následně se budou muset seznámit s programovacím prostředím robota a pochopit základní konstrukce při programování, jako jsou cykly, podmínky, nebo přepínače. Pro konkrétní práci s programovacím prostředím budou potřebovat poznat, samotnou práci s prostředím. Jakým způsobem mohou stavební bloky propojovat, odstraňovat nebo nastavovat. Pro lepší přehlednost vypíšeme nyní cíle metodiky do seznamu.

- Žáci vysvětlí a použijí základní konstrukce z programování, jako jsou cykly, podmínky nebo přepínače.
- Žáci používají části stavebnice a propojují jednotlivé díly tak, aby dosáhli svého cíle.

- Žáci používají programovací prostředí pro legoroboty k vyřešení zadaného úkolu, ale i k dosažení vlastních cílů s nimi.
- Žáci naprogramují legorobota pomocí programovacího prostředí tak, aby robot plnil svou funkci.

První z cílů se zabývá základními konstrukcemi a prvky v programování. Těmi myslíme cykly, podmínky, konstanty a proměnné. Nabyté znalosti si žáci okamžitě vyzkouší při řešení teoretických problémů ve skupinách. Vzhledem ke snaze podporovat informatické myšlení za využití legorobotů není nutné trvat na využití správných označení ve vývojovém diagramu, ani na jeho využití jako takovém. Vzhledem ke snaze o pochopení nám postačí popis běžným jazykem.

Žáci mají za úkol zvolit si ve skupině svého zástupce. Poté popíše jeho cestu do školy pomocí základních pohybů, které nejsou složeny z dalších činností. Budou muset používat především podmínky. Na příklad, pokud přijde žák k silnici, nestačí napsat „Rozhlédnu se!“, je potřeba rozhlédnutí buď definovat někde jinde, abychom jej mohli používat, jako funkci, nebo pokaždé vypisovat, co je rozhlédnutí a za jakých podmínek žák přejde. Žáci po dokončení úkolu budou postupně před třídou prezentovat svůj postup cesty do školy. Ostatní žáci reagují na případné chyby a pokouší se je opravit. Vyučující poukazuje na chyby, kterých si nevšimli ostatní žáci, případně odůvodňuje, proč se o chybu nejedená.

I když by se mohlo zdát, že v této části metodiky nedochází k podpoře informatického myšlení, není tomu tak. Žáci ve skupinách rozebírají problém docházky do školy ze všech stran. Nejenže musí očekávat příjezdící auto při přecházení, ale už volba cesty, kterou budou popisovat, může být nezvyklým úhlem pohledu na věc. Žáci se budou muset dopředu rozhodnout, která cesta bude pro jejich popis nejvýhodnější. Rozklad složitěho na jednodušší je jasně patrný samotným zadáním. Dále je rozvíjena schopnost komunikace, vyřešení problému v relevantním čase i hledání možných opakovaných postupů v řešení problému.

Pokud žáci budou schopni vyřešit zadaný problém, můžeme považovat cíl „Žáci vysvětlí a použijí základní konstrukce z programování, jako jsou cykly, podmínky nebo přepínače.“ za splněný a je možné pokračovat k dalším cílům.

Zbylé cíle by byly naplňovány v podobném duchu. Pokud čtenáře zajímá konkrétní znění naplňování všech cílů, nachází se v diplomové práci, ze které příspěvek vychází.

## 5 Závěr

V příspěvku jsme se seznámili s definicemi informatického myšlení a možnostmi jeho podpory. Z možností vyplývá, že je možné podporovat informatické myšlení na školách již dnes, a to bez rozsáhlých investic do vybavení. Jedinou podmínkou je změna přístupu vyučujících k vedení hodin. Zároveň jsme objasnili pojem legorobot, jaké jsou jeho možnosti a navrhli případné výhody a nevýhody jeho využití ve výuce.

Závěrem jsme se pokusili nastínit metodiku, která by podporovala informatické myšlení za využití zmíněných legorobotů. Je uvedeno konkrétní naplnění jednoho z cílů metodiky.

## Použitá literatura

1. **CSTA.** CSTA. [Online] CSTA. [Citace: 29. 11. 2017.] Dostupné z:<<https://www.csteachers.org/page/About>>.
2. **CSTA a ISTE.** Computational thinking teachers resources second edition. *CSTEACHERS*. [Online] 2011. [Citace: 29. 11. 2017.] Dostupné z:<[https://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/472.11CTTeacherResources\\_2ed.pdf](https://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/472.11CTTeacherResources_2ed.pdf)>.
3. **Frank, Filip.** *Výuka informatiky a podpora informatického myšlení pomocí legorobotů na gymnáziích*. Plzeň : Západočeská univerzita v Plzni, 2018.
4. **ISTE.** About. *Iste*. [Online] [Citace: 24.. 4. 2018.] Dostupné z:<<https://www.iste.org/about/about-iste>>.
5. **Lego.** Lego Mindstorms EV3. *Lego*. [Online] [Citace: 12. 2. 2018.] Dostupné z:<<https://www.lego.com/cs-cz/mindstorms/history>>.
6. **Wang, Paul.** *From computing to computational thinking*. Ohio : CRC Press, 2016. ISBN: 978-1-4822-1766-7.
7. **Wing, Jeanett M.** *Carnegie Mellon University*. [Online] 17. 11. 2010. [Citace: 27. 11. 2017.] Dostupné z:<<https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>>.
8. **Robot world.** *Robot world*. [Online] [Citace: 13. 2. 2018.] Dostupné z:<[https://www.robotworld.cz/lego-mindstorms-ev3?gclid=CjwKCAiAatorUBRBnEiwAfcP\\_YzWm135iFNW36OTxmGRvzf1sKmH6eQJQ1IlvPoOVHlvtXiD4TtwBoCMpEQAvD\\_BwE](https://www.robotworld.cz/lego-mindstorms-ev3?gclid=CjwKCAiAatorUBRBnEiwAfcP_YzWm135iFNW36OTxmGRvzf1sKmH6eQJQ1IlvPoOVHlvtXiD4TtwBoCMpEQAvD_BwE)>.

## Kontaktní údaje

Bc. Filip Frank  
 Západočeská univerzita v Plzni, Fakulta pedagogická  
 Klatovská tř. 51, 306 19 Plzeň  
 E-mail: frankf@students.zcu.cz