Univerzita Hradec Králové
Pedagogická fakulta
Katedra informatiky Přírodovědecké fakulty

Možnosti rozvíjení algoritmického myšlení s využitím projektů Hour of Code a Scratch

Diplomová práce

Autor: Bc. Tomáš Hornik
Studijní program: Anglický jazyk se zaměřením na vzdělávání
Informatika se zaměřením na vzdělávání
Vedoucí práce: PhDr. Michal Musílek, Ph.D.

Hradec Králové
2016

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval pod vedením vedoucího diplomové práce samostatně a uvedl jsem všechny použité prameny a literaturu.

V Hradci Králové dne

Poděkování

Děkuji PhDr. Michalu Musílkovi, Ph.D. za odborné vedení diplomové práce, poskytování rad a materiálových podkladů k mé práci a v neposlední řadě za veškerou projevenou vstřícnost při tomto počínání. Dále děkuji všem učitelům a vedení ZŠ Chlumec nad Cidlinou za neméně vstřícnou spolupráci v rámci mého výzkumu.

Anotace

HORNIK, Tomáš. (2016). *Možnosti rozvíjení algoritmického myšlení s využitím projektů Hour of Code a Scratch*. Hradec Králové: Pedagogická fakulta Univerzity Hradec Králové. 135 s. Diplomová práce.

Cílem této diplomové práce je zmapovat možnosti výuky programování na vybrané základní škole pomocí vizuálních programovacích jazyků, které nevyžadují znalost příkazů a syntaxe konkrétního vyššího programovacího jazyka. Podstatou výuky programování na základních školách by mělo být předat žákům schopnost pracovat s algoritmy a plně využívat algoritmického způsobu myšlení. Žáci by měli používat sekvence příkazů, cykly a podmínky, procedury a funkce, proměnné, zasílání zpráv ve stylu objektově orientovaného programování apod. Protože je výuka programování na českých školách silně opomíjena, začátek teoretické části práce je věnován projektu Hour of Code, který nabízí všem zájemcům, a tedy i učitelům informatiky, jasná a jednoduchá východiska pro výuku programování. Ve druhé části je rozebírán vizuální programovací jazyk Scratch, jeho základní i vybrané pokročilé funkce a možnosti jeho využití na školách. Praktická část této práce sestává z několika na sebe navazujících kroků. Na vybrané škole, která programování běžně nevyučuje, byl nejprve žákům i učitelům informatiky rozdán jednoduchý dotazník zjišťující stávající názory na programování, možné předsudky, představy a předpoklady. Poté si tito žáci vyzkoušeli ukázkovou úvodní hodinu programování postavenou na výše zmiňovaných projektech a po jejím skončení vyplnili druhý dotazník, zjišťující jestli došlo k nějakým změnám v jejich názorech a zda by chtěli ve studiu programování pokračovat. S učiteli informatiky byly na závěr vedeny řízené rozhovory, jejichž celkovým výstupem je jednoduchý kvalitativní výzkum.

Klíčová slova: algoritmizace, programování, výzkum, základní škola, Hour of Code, Scratch

Annotation

HORNIK, Tomáš. (2016). *Možnosti rozvíjení algoritmického myšlení s využitím projektů Hour of Code a Scratch*. Hradec Králové: Pedagogical Faculty University of Hradec Králové. 135 s. Master Degree Thesis.

The goal of the diploma thesis is to chart the possibilities of teaching of programming at the selected primary school with the use of visual programming languages that do not require knowledge of instructions nor of a syntax belonging to a specific higher complex programming language. The essence of teaching of programming at the primary schools should be to pass to a student an ability to work with algorithms and to fully utilise the algorithmic way of thinking. Pupils should work with sequences of instructions, loops and conditions, procedures and functions, variables, sending of messages in the OOP style, etc. Because of the fact that the teaching of programming is often omitted in Czech schols, the beginning of the theoretical part of the thesis is dedicated to the project Hour of Code, that offers to everyone interested in it, and thus also to teachers of IT, clear and simple starting point for the teaching of programming. The visual programming language Scratch is analysed in the second part of the thesis with its basic and some selected more advanced functions and possibilities of its use at schools. Practical part of the thesis consists of several sequential steps. At the selected school, that currently do not teach programming at all, pupils and teachers were given simple questionnaire investigating current opinions on programming, possible prejudices, concepts and presumptions. After that, the pupils tried a sample introductionary lesson of programming based on aforementioned projects and then they were given the second questionnaire investigating whether there were some changes in their opinions and whether they would like to continue in the studies of programming. At the very end, there is simple qualitative research based on controlled interviews.

Keywords: algorithmization, programming, research, primary school, Hour of Code, Scratch

OBSAH

ÚVOD

Jedním z hlavních cílů této práce je otestovat možnost zařazení výuky algoritmizace a programování do standardního kurikula základních škol. V současné době se jedná o problematiku učiteli opomíjenou (viz kapitola 2) a samotné téma vyvolává značnou kontroverzi mezi odborníky jak z pedagogických, tak z programátorských řad (podkapitoly 2.1 a 2.2). Vše je dále komplikováno absencí zákonné povinnosti zaměstnávat na daný předmět pouze učitele s odpovídající aprobací. V praxi se tak stává, že učitelé informatiky tento předmět jako svou aprobaci nevystudovali, a jestliže se nejedná o zapáleného jedince, programování takovému učiteli zpravidla nic neříká. Stejně tak v ČR neexistuje žádná ucelená aktuální metodická příručka pro učitele k výuce programování a algoritmizace.

Výše jmenované nedostatky vedly k nutnosti hledání řešení, které by bylo možné nasadit bez větších komplikací v širokém měřítku, a to zejména proto, že schopnost algoritmizace (neboli vytváření postupů řešení různých problémů) není vrozená a nerozvíjí se samovolně s věkem daného žáka či žákyně. Tato dovednost by tedy měla být rozvíjena v rámci povinné školní docházky jako jedna ze základních kompetencí a to právě v rámci předmětu informatika a výpočetní technika.

Jedná se tedy o závažné téma, jehož řešení se nabízí v podobě zakomponování projektu Hour of Code do běžné výuky. Tento projekt umožňuje žákům pracovat vlastním tempem a prakticky bez zásahu vyučujícího. Hlavním cílem této práce je nasazení ukázkové hodiny Hour of Code v běžných hodinách informatiky doprovázené vstupními a výstupními dotazníky. Vstupní dotazník zachycuje dosavadní zkušenosti a názory žáků týkající se programování a informatiky obecně, výstupní dotazník je poté zaměřen na názory žáků na absolvovaný Hour of Code a obsahuje otázky týkající se zájmu o další pokračování takovéto výuky a případně zájmu o kroužek zabývající se programováním. V závěru celé práce se nachází kvalitativní výzkum založený na vstupních a výstupních rozhovorech se dvěma vyučujícími informatiky na dané základní škole.

Jádrem teoretické části této práce je představení možných východisek pro učitele informatiky vhodných pro výuku programování. Jmenovitě sem spadají vybrané projekty představené v kapitole 3, a to Hour of Code, Scratch, LEGO Mindstorms, CodeCombat a Khan Academy.

Úvodní hypotézou práce je, že absolvováním alespoň jedné hodiny Hour of Code se u žáků zvýší schopnost algoritmizace. Tato hypotéza je ověřována na dvou elementárních příkladech, kdy v prvním z nich mají žáci popsat myšce cestu k sýru a ve druhém z nich popsat jak přejít silnici. Obě tyto úlohy jsou součástí jak vstupního, tak výstupního dotazníku. Odpovědi žáků jsou v podkapitolách 5.2.2 a 5.2.3 vyhodnoceny a je zjišťováno, zda u jednotlivých žáků došlo ke zlepšení.

Dílčí hypotézy ověřují tvrzení tvůrců projektu Hour of Code, že výuka programování je vhodná pro všechny. V rámci následujících hypotéz jsou zkoumány vstupní předpoklady žáků:

- Míra významu, kterou žáci přikládají informatice jakožto oboru, přímo souvisí s jejich úspěšností v úloze na definování programování.
- Čím vyšší je šíře informaticky orientovaných zájmů žáků, tím lepší je jejich schopnost algoritmizace.
- Věk žáků má přímou pozitivní souvislost s kvalitou jejich odpovědí.

Co se týče otázky věku, testování projektu Hour of Code v rámci této práce si dále dalo za cíl odhadnout nejvhodnější věkovou kategorii pro začátek výuky programování. Abstraktní myšlení, které je nutné pro tvorbu algoritmů, se dle J. Piageta u žáků rozvíjí až kolem 12 roku, tedy kolem 6. a 7. třídy, ale samotní tvůrci projektu Hour of Code tvrdí, že je jejich projekt *„Určen pro věk od 4 do 104.“* (Code.org 2016a). Na omezeném vzorku a bez možnosti longitudiální studie není možné nejvhodnější věk stanovit přesně a jednoznačně, jedná se však o odhad, který může být výchozím bodem pro případné další studie.

1 Obecná problematika algoritmizace a programování

Programování a algoritmizace jsou ve své podstatě dva navzájem neoddělitelné pojmy, protože programování (ve smyslu psaní kódu) musí vždy bezpodmínečně následovat po algoritmickém zpracování daného problému. K pochopení celé problematiky je potřeba si nadefinovat alespoň vybrané základní termíny z oblastí algoritmizace, programování a datových struktur, kterými se zabývá tato kapitola a které by měli učitelé informatiky znát a důkladně chápat. Teprve poté je možné se zamýšlet, je-li výuka programování vhodná již na základní škole a je-li současný systém vyučování této problematiky dostatečně kvalitní a odpovídající specifickým nárokům vyučování žáků ZŠ.

1.1 Definice algoritmizace a možnosti zápisu algoritmů

Samotné slovo algoritmus datuje svůj původ k přelomu 7. a 8. století našeho letopočtu, kdy arabský matematik Abū ʻAbd Allāh Muhammad ibn Mūsā al-Chwārizmī vydal dvě knihy s postupy výpočtů lineárních a kvadratických rovnic a poslední část jeho jména *al-Chwārizmī* byl v latinském překladu změněn na Algorismi (Harper 2016).

Přesné vymezení pojmu algoritmus je dle doc. Ing. Miroslava Viriuse, CSc. nemožné, protože *„Algoritmus je základní matematický pojem. To znamená, že jej nelze definovat* […] *Pojem algoritmus se dá formalizovat např. pomocí matematické konstrukce, označované jako Turingův stroj, nebo pomocí teorie parciálně rekurzivních funkcí.“* (1996, s. 9) Pro účely této práce je ale naprosto postačující zjednodušeně definovat **algoritmus** jako postup řešení nějakého problému v podobě sekvence na sebe navazujících dílčích kroků či instrukcí. Takto chápaný algoritmus je velice jednoduché ilustrovat na každodenních činnostech člověka (uvaření jídla, přecházení přes silnici, oprava židle, atd.), přičemž člověk je chápán jako autonomní bytost se schopností rozhodovat se dle vlastního uvážení. Tuto schopnost počítač jakožto stroj prozatím postrádá a je schopen jen toho, čemu ho člověk-programátor naučí.

Hlavní výhodou (a zároveň původním účelem) počítače je astronomická rychlost výpočtů a zpracovávání dat obecně. Veškeré postupy výpočtů ale musí počítači zpracovat programátor, který je při programování omezován mimo jiné právě i absencí jakékoliv flexibility počítače v oblasti autonomního rozhodování. Z hlediska tvorby

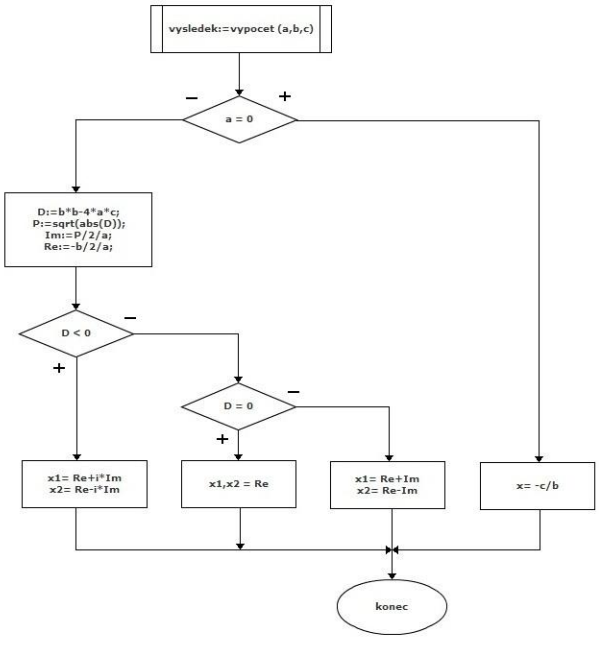
algoritmů je tedy nutné splňovat určité podmínky definující základní vlastnosti algoritmů, kterými jsou:

- **jednoznačnost neboli determinovanost** – v rámci algoritmu musí být jednoznačně určeno, který konkrétní krok/instrukce následuje, přičemž v případě rozhodování mezi vícero možnostmi musí na základě aktuálních dat opět existovat jen jediná správná možnost (počítač se jinak není schopen rozhodnout podle sebe);
- **resultativnost** – algoritmus vede ke správnému výsledku, cíli či řešení;
- **konečnost** – algoritmus se nesmí dostat do nekonečného zacyklení a výsledek musí být poskytnut ve smysluplném časovém intervalu (kdyby výpočet trval milion let, algoritmus by neměl smysl);
- **opakovatelnost** – při opakování algoritmu se zadáním stejných vstupních dat musí být výsledek vždy opět stejný;
- **hromadnost neboli obecnost** – vstupní data nejsou definována konkrétními hodnotami, ale jsou označována symbolickými jmény (proměnnými značícími množiny, ze kterých lze data vybrat), díky čemuž lze algoritmus aplikovat na celou skupinu podobných úloh, protože v závislosti na zadaných vstupních hodnotách jsou zohledněny možné alternativy a odchylky v postupu řešení; (Virius 1995, s. 9; Ďuráková et al. 2002, s. 27).
- **srozumitelnost neboli přehlednost** – doc. Ing. Arnošt Motyčka, CSc. ještě dále uvádí jako jednu z nutných vlastností algoritmu i jeho srozumitelnost či přehlednost, která je logickým předpokladem modifikovatelnosti umožňující další rozšiřování či vylepšování stávajícího algoritmu (Motyčka 1999, s. 6).

Vytvořený algoritmus lze vyjádřit velkým množstvím různých prostředků či zápisů, přičemž neexistuje jednotná norma, která by byla jednoznačně „správná“ a používaná všemi. Nejjednodušší a nejméně formalizované je vyjádření algoritmu obyčejnými popisnými větami v daném jazyce (čeština, angličtina, atd.). Jelikož algoritmus může být postup jakékoliv opakující se, a tedy generalizovatelné činnosti, a zároveň jeho neformální verbální popis nevyžaduje nutnou znalost prakticky žádných formálních pravidel, bylo v rámci dotazníkového výzkumu využito právě tohoto přístupu ke zjišťování, mají-li žáci alespoň minimální schopnost algoritmizace a logického uvažování. Žákům byla zadána úloha, ve které měli vlastními slovy popsat, jak se přechází ulice. Více informací k tomuto příkladu je obsahem podkapitoly 5.2.3.

Mezi grafická znázornění algoritmů patří například obdélníkové strukturogramy či pravděpodobně nejznámější vývojové diagramy. Ty jsou založeny na několika základních tvarech (jako je elipsa, kosočtverec, lichoběžník, obdélník, apod.) z nichž každý má svůj jednoduchý význam a definovaný požadovaný obsah. Přechod od jedné části ke druhé je značen šipkami a postup diagramem je zpravidla ve směru svrchu dolů, zleva doprava, či kombinací těchto dvou možností. Výsledný algoritmus (ilustrovaný například v levé polovině obrázku 1) je v tomto zápisu relativně srozumitelný i běžnému člověku ne-programátorovi.

Nejsložitějším a pro ne-programátora nejméně srozumitelným zápisem algoritmu je jeho zpracování ve formě programu v konkrétním programovacím jazyce. Typickým příkladem může být programovací jazyk ALGOL 60 (velice podobný Pascalu), který přestože nebyl masově rozšířený v praxi, byl po několik desetiletí považován za standard pro publikace algoritmů ve vědeckých časopisech (Musílek 2011). V pravé polovině obrázku 1 je pro ilustraci zápis algoritmu pro výpočet kvadratické rovnice v programovacím jazyce Pascal, který je přesným ekvivalentem vývojového diagramu v části levé.




```
unit Unit1;

interface
uses SysUtils;
type matice=Array[1..2] of String;
function vypocet(a,b,c:Real):matice;

implementation
function vypocet(a,b,c:Real):matice;
  var D,P,Im,Re,soucet,rozdil:Real;
      ImS,ReS:String;
  begin
    if a=0 then
              begin
                vypocet[1]:=('linearni rovnice');
                vypocet[2]:=('kde x= '+FloatToStrF((-c/b),ffGeneral,8,3));
              end
            else
             begin
               D:=b*b-4*a*c;
               P:=sqrt(abs(D));
               Im:=P/2/a;
               Re:=-b/2/a;
               ImS:=FloatToStrF(Im,ffGeneral,8,3);
               ReS:=FloatToStrF(Re,ffGeneral,8,3);
               if D<0 then
                         begin
                           vypocet[1]:=('x1= '+ReS+'+i*'+ImS);
                           vypocet[2]:=('x2= '+ReS+'-i*'+ImS);
                         end
                       else
                         begin
                           if D=0 then
                                     begin
                                       vypocet[1]:=('x1= '+ReS);
                                       vypocet[2]:=('x2= '+ReS);
                                     end
                                   else
                                     begin
                                       soucet:=Re+Im;
                                       rozdil:=Re-Im;
                                       vypocet[1]:=('x1= '+FloatToStrF(soucet,ffGeneral,8,3));
                                       vypocet[2]:=('x2= '+FloatToStrF(rozdil,ffGeneral,8,3));
                                     end;
                         end;
              end;
  end;
end.
```

Obrázek 1 – Vývojový diagram a program v PASCALu pro výpočet kvadratické rovnice

1.1.1 Příkazy, cykly a větvení algoritmu

K samotnému vytvoření algoritmu je nehledě na použitý druh zápisu či programovací jazyk možno využít jen velice omezený počet základních příkazů a datových struktur. Většina programovacích jazyků sice umožňuje vytváření vlastních komplikovanějších příkazů, ty se ale stejně opět skládají z těchto příkazů základních.

Základní příkazy lze dělit na jednoduché a strukturované. Jednoduchým příkazem se určuje jeden konkrétní úkon, který se má provést. Strukturovaných příkazů je více, jmenovitě se jedná o podmíněné rozhodování úplné a neúplné, cyklus s podmínkou na začátku, cyklus s podmínkou na konci a cyklus s pevným počtem opakování. Je-li potřeba v rámci cyklu zopakovat celou sekvenci jednoduchých příkazů či celé další cykly, uzavírají se tyto části do takzvaných příkazových bloků.

Jak již bylo řečeno v kapitole 1.1, algoritmus lze vyjádřit více způsoby a v každém z nich je možné využít všechny výše zmiňované příkazy. Tabulka 1 znázorňuje možné použití těchto příkazů v rámci verbálního popisu postupu, v grafickém znázornění pomocí vývojových diagramů a v programovacím zápisu v jazyce Pascal, který byl pro svou perfektní i když značně rigidní strukturovanost po mnoho let využíván jako základní učební pomůcka v rámci výuky programování na středních a vysokých školách, a jednalo se tedy o jakýsi standard principiálně podobný právě výše zmiňovanému jazyku ALGOL 60.

1.1.2 Elementární datové typy a struktury

Je-li potřeba uložit nějaká data, zpravidla je nejprve nutné vytvořit neboli nadeklarovat proměnnou, jejíž datový typ odpovídá našim potřebám. Tyto proměnné jsou poté buď globální, nebo lokální. Globální proměnné se vytváří při zapnutí programu a jejich životnost končí opět spolu s programem, kdežto lokální proměnné lze chápat jako proměnné patřící funkcím a procedurám. Specifickým druhem proměnné je konstanta, neboli nadeklarovaná proměnná s předem a napevno určenou hodnotou.

Každý programovací jazyk má nadefinované základní datové typy, které se však jazyk od jazyka odlišují jak celkovým množstvím, tak konkrétním způsobem práce s nimi. Obecně je však možno rozlišovat takzvané jednoduché a složené datové typy. Pro potřeby této práce jsou zde zmíněny základní datové typy programovacího jazyka Pascal, který byl za tímto účelem vybrán z důvodu jeho přizpůsobení výchovně

vzdělávacímu procesu (tento jazyk obsahuje malé množství striktně vyžadovaných pravidel, což mělo vést k jejich jednoduššímu osvojení žáky).

Jednoduché datové typy v Pascalu jsou *boolean, integer, real* a *char*. Úplně nejjednodušší je typ *boolean*, jenž nabývá jen dvou hodnot a to True/False, pro jejichž uložení stačí jediný bit (1/0). Proměnná typu *integer* poté slouží k uložení celých čísel, přičemž v Pascalu defaultně zabírá 2B a může tedy nabývat hodnot od -32768 do +32767. Reálná čísla ukládaná pomocí typu *real* jsou podstatně složitější, zabírají 4B a využívá se takzvaného semilogaritmického tvaru čísla, který se skládá z Mantisy M, základu číselné soustavy Z a exponentu P. Číslo $X = M \times Z^P$, kde základ číselné soustavy je v binární soustavě samozřejmě 2 (Dušek 2009, s. 6). Posledním jednoduchým datovým typem je *char* s velikostí 1B, což v podstatě nabízí 256 možných kombinací. Takto nízký počet kombinací původně pokrýval takzvané ASCII kódování, které bylo později nahrazeno podstatně univerzálnějším a zpětně kompatibilním UTF-8.

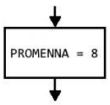
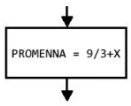
Složené datové typy lze chápat jako kombinaci datových typů jednoduchých. *String* neboli *řetězec znaků* je název všeříkající, jedná se totiž o zřetězení jednoduchých proměnných typu *char* do delšího jednotného celku jako je slovo nebo věta. Složitějším datovým typem je pole *array*, které v podstatě vytváří matici adresovatelnou pomocí indexů, do které tedy můžeme ukládat větší množství jednotlivých dat. Práci s externím textovým souborem lze chápat jako zvláštní datový typ soubor.

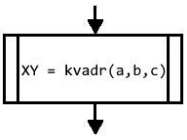
Přestože se tato práce zabývá Scratchem, pochopení Pascalovských datových typů je velice výhodné, protože přílišné zjednodušení Scratche může být pro učitele informatiky poněkud matoucí. Scratch totiž operuje jen se dvěma datovými typy, a to *proměnnou* a *seznamem*. Paradoxní však je, že jednoduchá proměnná ve Scratchi je univerzální a dokáže pojmout většinu výše vypsaných Pascalovských datových typů. Nadeklarovanou Scratchovskou *proměnnou* lze totiž naplnit jak celým číslem, tak reálným číslem, ale i znakem a potažmo i řetězcem znaků.

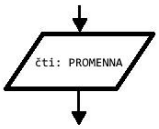
Seznam ve Scratchi je ekvivalentem jednorozměrného pole Array popsaného výše, včetně možnosti indexace jako v Pascalu. Scratchovský seznam je však obdobně jako Scratchovská proměnná univerzální, může tedy obsahovat jakékoliv datové typy bez jejich přesné definice. Naprosto stejným způsobem s proměnnými pracují i interpretované skriptovací jazyky, jako je například Python nebo JavaScript. Toto zjednodušení sice usnadňuje výuku na ZŠ, ale učitelé by si ho měli být vědomi při případném přechodu na komplexnější programovací jazyky typu Java nebo C#.

Scratch dále umožňuje exportovat výše zmiňované seznamy do externího textového *.txt* souboru a vytvářet tak například savy ve hrách (Scratch Wiki 2016). Jedná se však již o velice pokročilou techniku.

Nejmarkantnějším rozdílem mezi Pascalem a Scratchem z hlediska práce s daty je vysoká strukturovanost Pascalovských programů. Veškeré proměnné použité v daném programu musí být nejprve nadeklarovány a případně inicializovány v hlavičce programu, jinak s nimi nejde pracovat. Tento přístup vynucuje psaní čistého a přehledného kódu a vytváří vhodné programátorské návyky. Moderní vyšší programovací jazyky, jako je například Java nebo JavaScript, od vynucování této strukturovanosti upustily (i když ji vysoce doporučují) a ve stejném duchu je vytvořen i Scratch. Seznámit tedy žáky s vytvářením správně strukturovaných programů je prací učitele, a to prací, která by opět neměla být opomíjena.

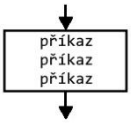
		Verbální popis	Vývojové diagramy	Pascal
vybrané příklady jednoduchých příkazů	**přiřazovací příkaz konkrétní hodnoty**	Ulož do proměnné jménem PROMENNA číslo 8.		`PROMENNA:=8;`
	přiřazovací příkaz výpočtem	Ulož do proměnné jménem PROMENNA výsledek výpočtu 9/3+X.		`PROMENNA:=9/3+X;`

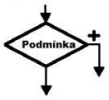
	volání podprogramu / podprocesu	Vyskoč z aktuálního algoritmu, zapni a proveď podprogram pro výpočet kvadratické rovnice pro hodnoty A,B,C a výsledek ulož do XY.		`Var XY: real;`
				`function kvadr (A,B,C: integer): real;`
				
				`XY:=kvadr(A,B,C);`

	načtení hodnoty od uživatele	Načti do proměnné jménem PROMENNA hodnotu zadanou uživatelem.		`readln(PROMENNA);`

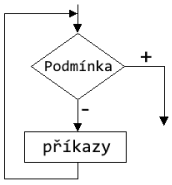
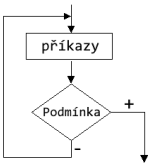
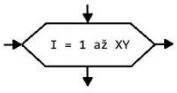
	otevření souboru	Otevři textový soubor XY.		`Var f : text;`
				`Assign(f,'XY.txt');`
				`Append(f);`

	zavření souboru	Zavři textový soubor XY.		`CloseFile(XY.txt);`

	blok příkazů	Proveď všechny příkazy v bloku jeden po druhém.		`begin`
				`  příkaz;`
				`  příkaz;`
				`end;`

základní strukturované příkazy	**podmíněné rozhodování neúplné**	Jestliže je podmínka splněná, proveď odpovídající příkaz či blok příkazů. Jestliže není, pokračuj v algoritmu dál jakoby nic.		`if podmínka` `then příkaz;`

	podmíněné rozhodování úplné	Jestliže je podmínka splněná, proveď odpovídající příkaz či blok příkazů, v opačném případě proveď jiný příkaz nebo blok příkazů.		`if podmínka`
				`then příkaz`
				`else příkaz;`

	cyklus s podmínkou na začátku	Zkontroluj, jestli je podmínka splněná a pokud ano, proveď odpovídající příkaz nebo blok příkazů a poté se vrať na začátek a znovu zkontroluj podmínku. Jestliže splněná není, příkaz nebo blok příkazů přeskoč a pokračuj v algoritmu dále.		`while podmínka` `do příkaz;`
	cyklus s podmínkou na konci	Proveď příkaz nebo blok příkazů a poté zkontroluj, jestli je podmínka splněná. Jestliže podmínka splněná není, vrať se na začátek cyklu a proveď příkazy znovu, jestliže splněná je, pokračuj dále.		`repeat příkaz` `until podmínka;`
	cyklus s pevným počtem opakování	XYkrát proveď příkaz nebo blok příkazů.		`for I:=A to B` `do příkaz;`

Tabulka 1 – Možnosti vyjádření instrukcí v různých typech zápisu diagramu

1.2 Programování a programovací jazyky

Program lze definovat jako *„algoritmus zapsaný v některém programovacím jazyce“* (Ďurákova 2002, s. 28). Programování je tudíž přepis daného algoritmu do daného programovacího jazyka. V tomto kontextu je také možné pohlížet na algoritmus jako na prostředek k transformaci zadaných vstupních dat na požadovaná data výstupní. Každý programovací jazyk je ale jiný a ideální pouze pro řešení určité množiny úloh, nelze tedy obecně tvrdit, že nějaký konkrétní programovací jazyk je „nejlepší“. Právě zde se celá situace ještě více komplikuje, protože přestože lze tvrdit, že některé algoritmy jsou efektivnější než jiné (stejný problém vyřeší v menším počtu kroků), různé programovací jazyky potřebují pro vykonání stejné instrukce různě dlouhou dobu.

Žáci by tedy měli být již od začátku učitelem vedeni ke tvorbě vysoce efektivních a co možná nejjednodušších algoritmů odpovídajících vybranému programovacímu jazyku, které neobsahují zbytečné kroky zpomalující průběh programu.

Z historického hlediska došlo ke značnému vývoji v programovacích jazycích a zejména metodice programování obecně. První počítače byly pomalé a strojový čas drahý, proto byla nutnost co možná nejrychlejších a nejúspornějších kódů, které nemusely být nijak přehledné ani rozšiřitelné. Počítače se ale rychle vyvinuly, objevil se assembler čili jazyk symbolických adres a další vyšší programovací jazyky, ale čím dál složitější programy byly dokonale nepřehledné. Proto v 60. letech vznikla metodika zvaná modulární programování, která velké projekty rozdělovala do menších samostatných projektů. Na to v 70. letech navázala tehdy kontroverzní metodika známá jako strukturované programování. Přesto stále existovala takzvaná sémantická mezera, neboli rozdíl mezi tím, jak by se dal problém vyjádřit a jak je toto řešení potřeba přeformulovat, aby ho počítač pochopil. To v 90. letech vedlo ke vzniku vysoce abstraktního OOP neboli objektově orientovaného programování, které je hlavním proudem dodnes (Pecinovský 2009).

1.3 Objektově orientované programování

Přestože jsou vyšší programovací jazyky abstraktnější než assembler, stále vyžadují myšlení v termínech a strukturách počítače. Z pohledu programování tedy existují dvě oblasti, a to tzv. oblast řešení a oblast problému, přičemž oblast řešení je pro programátora počítač a oblast problému je reálný svět. Řešení reálných problémů v jazyce počítačů byl problém sám o sobě, protože se nejprve musel vyřešit problém v reálném světě a toto řešení se poté muselo naprosto předělat a přeformulovat tak, aby mu rozuměly počítače (což je podstatou výše zmiňované sémantické mezery). Toto je jeden z hlavních důvodů vzniku objektově orientovaného programování, v němž má programátor nástroje, díky kterým si může vytvořit z vlastních nových objektů jakýsi model reality, odpovídající řešenému problému. Tento princip dovoluje řešit problém v pojmech daného problému a ne v termínech konkrétního počítače, na kterém bude program implementován. Objekt lze tedy chápat jako element z oblasti problému neboli reálného světa a jeho reprezentaci v oblasti řešení neboli na počítači.

Základní myšlenkou OOP je, že i když jsou všechny objekty unikátní, zároveň ale patří do nějaké třídy objektů, které mají stejné charakteristiky a chování. V programování je tato takzvaná *třída* de facto námi definovaný abstraktní datový typ, u kterého můžeme vytvářet libovolný počet proměnných zvaných *objekty* a manipulovat s nimi pomocí takzvaného *zasílání zpráv*.

Výše zmiňované *zasílání zpráv* je takřka doslova implementováno i ve Scratchi. Objekty zde mohou rozesílat zprávy a reagovat na ně. Zprávy však nejsou nijak cílené a rozeslanou zprávu přijmou všechny ostatní objekty v celém projektu. Reagovat na ni však mohou jen objekty, které mají nadefinované co dělat po obdržení zprávy daného typu. Jedná se o hrubé zjednodušení celého konceptu, které je uzpůsobeno co nejjednodušší výuce programování. Opět nelze říci, že by toto zjednodušení bylo chybné, je však potřeba mít ho na paměti při případném přechodu na jiný, ať už dětský nebo skutečný vyšší, programovací jazyk.

Jelikož třída popisuje soubor objektů se stejnými charakteristikami definovanými jednotlivými datovými elementy a chováním neboli funkcionalitou, extrémně zjednodušený příklad přenesený do termínu procedurálního programování je, že třídě odpovídá datový typ INTEGER a objekt, který vzniká takzvaným definováním reference, by byla jedna jeho nadeklarovaná proměnná, jako třeba INTEGER X.

Konkrétní objekt může uspokojit jen určité žádosti definované v jeho interface, které je určeno datovým typem daného objektu. Každý datový typ neboli třída má svůj název třídy, interface a implementaci, kde implementace je zpravidla skrytý kód s daty, který definuje jak danou metodu přesně provést. V praxi má tedy třída hlavičku metody asociovanou s každou možnou žádostí definovanou v interface, a když objektu zašleme zprávu s nějakou žádostí, objekt se podívá do implementace, ve které se nachází její celý kód a odkud je daná metoda zavolána. Unified Modeling Language (UML) definuje přesný formát diagramu tříd. Každá třída je reprezentována obdélníkem rozděleným na tři políčka, kde v horním políčku je jméno třídy, v prostředním jsou datové proměnné a v dolním políčku metody, přičemž prostřední políčko může být vynecháno a v dolním se zobrazují jen public methods neboli takzvané veřejné metody.

Objekty je dále možné chápat jako poskytovatele služeb a cílem programátora je dát dohromady sadu objektů, které poskytnou ideální služby k vyřešení daného problému. Problém je tedy potřeba rozložit na sadu objektů, přičemž je třeba dbát o vysokou soudržnost či provázanost jednotlivých objektů v rámci výsledného programu, která je fundamentální kvalitou softwarového designu. Vybrané objekty se musí navzájem dobře doplňovat a zároveň nesmí být jednotlivé objekty příliš komplikované. Každý objekt musí dělat nějakou jednu věc dobře, ale nesmí se snažit toho dělat příliš najednou, čímž se zároveň stává univerzálnějším.

Programátory z hlediska objektově orientovaného programování lze rozdělit do dvou skupin, kterými jsou takzvaní tvůrci tříd a *„client-programmers“* neboli uživatelé tříd, kteří tyto již hotové třídy využívají ve svých aplikacích. Cílem uživatelů tříd je sehnat si hromadu tříd z různých knihoven pro jejich využití a cílem tvůrců tříd je udělat je „blbuvzdorné“ a to tak, že ze třídy zobrazí jen potřebné interface daných metod, a zbytek ponechají v rámci implementace skrytý a nepřístupný za pomoci přístupových práv nabízených daným programovacím jazykem (Eckel 2006). Drtivou většinou dnešních programátorů jsou právě uživatelé tříd, čemuž by měla odpovídat i samotná výuka (Pecinovský 2009).

Jedním z projektů okrajově zmíněných v této diplomové práce je CodeCombat (viz kapitola 3.4). Ten je v podstatě na principu klientského programování celý založen. Hráči v něm využívají již hotové třídy s nadeklarovanými metodami. Pro učitele informatiky, který by tento projekt chtěl ve své výuce využít je tedy příhodné chápat výše popsanou podstatu tohoto způsobu programování.

2 Aktuální stav výuky algoritmizace a programování na ZŠ

Nejaktuálnější a zároveň prakticky jediný výzkum zkoumající rozšířenost výuky algoritmizace byl proveden v roce 2012 v rámci obhájené diplomové práce na katedře informatiky pedagogické fakulty Jihočeské univerzity v Českých Budějovicích. Cílem této práce není znovu zmapovat rozšířenost výuky algoritmizace na základních školách, nýbrž otestovat možnosti praktického nasazení projektu Hour of Code a dalších podobných, či přidružených projektů (viz kapitola 3), na vybrané základní škole, na které se algoritmizace nevyučuje, a žáci jsou tedy tabula rasa. Z tohoto důvodu jsou následující údaje založeny na zjištění výše jmenovaného výzkumu.

Dle online portálu AtlasŠkolství.cz je na celém území České republiky celkem 4138 základních škol (nepočítaje základní školy speciální a umělecké), přičemž výzkum rozvoje algoritmizace v rámci své kvantitativní části udává, že v celé ČR se programováním zabývá jen 50 základních škol, a to ještě *„ve většině případů pouze v zájmových kroužcích, které jsou na školách organizovány, velký podíl také mají povinně volitelné předměty.“* (Bromová 2012, s. 39) Algoritmizaci jako samostatný předmět potom učí jediná základní škola v celé republice. Těchto celkem 50 škol tedy tvoří pouhé 1,2% všech základních škol, s čímž nejspíše souvisí i naprostá absence odpovídajících výukových materiálů, reprezentovaná jedinou učebnicí pro základní školy věnující se alespoň okrajově problematice programování. Touto učebnicí je *Informatika pro základní školy a víceletá gymnázia*, 3. díl autorů Vaníček a Mikeš (Bromová 2012). Prozkoumání akreditovaných učebnic informatiky pro ZŠ ve schvalovací doložce MŠMT (jsou tam celé čtyři učebnice) jen potvrzuje nedostatek kvalitních oficiálních materiálů dostupných pro učitele informatiky (MŠMT 2016).

Nová aktualizovaná verze rámcového vzdělávacího programu pro základní vzdělávání platná od 1.9.2016 formuluje jako jedno z cílových zaměření vzdělávací oblasti IKT vedení žáka ke schopnosti *„využívat při interakci s počítačem algoritmické myšlení“* (Metodický portál RVP 2016), čímž zároveň veškerá spojitost dané vzdělávací oblasti s problematikou programování a algoritmizace končí. Situace se tedy od minulé verze z roku 2010 nezlepšila a to nejen ve vztahu k algoritmizaci a programování, ale i z hlediska formulace ostatních cílů dané vzdělávací oblasti, které jsou stále přinejlepším vágní (v podstatě v oblasti IKT totiž došlo jen k doplnění o tzv. *„Minimální doporučenou úroveň pro úpravy očekávaných výstupů v rámci podpůrných opatření“*).

Takto vágní formulace cílů je dvojsečná zbraň, protože ponechává učitelům informatiky prakticky neomezenou volnost. Dobrý učitel toho dokáže využít a učit kvalitně aktuální problematiku přizpůsobovanou podle neustále se měnící situace, ovlivňované raketovým vývojem celé oblasti informačních technologií. Špatný učitel (nebo líný, a tedy jinými slovy opět špatný učitel) však má možnost zakonzervovat své kurikulum a nereflektovat měnící se vzdělávací potřeby. Není nucen rozšiřovat či upravovat probíranou látku a z důvodu prosté pohodlnosti ani nevyhledává nové přístupy a postupy práce, které by mu jeho práci dokonce mohly ještě usnadnit.

2.1 Mezinárodní diskuze o vhodnosti výuky programování

V úvodní kapitole byly načrtnuty základní pojmy, jako je algoritmus a jeho vlastnosti, programování, programovací jazyky a datové struktury. Tato problematika se může zdát beznadějně komplikovaná a z úvodu této kapitoly je zřejmé, že učitelé ZŠ se jí raději vyhýbají úplně. Jedním z hlavních argumentů učitelů je, že algoritmy vyžadují extrémně vysokou úroveň abstrakce, které děti na ZŠ zpravidla nejsou schopny, přičemž ověření tohoto prohlášení je jedním z cílů praktické části této práce. Na otázku, zdali je možné naučit programovat každého, dokonce existují celé studie (Atwood 2006; Ars Technica 2012; Bricklin 2002).

S překvapivým závěrem přišla studie Saeeda Dehnadio spolupracujícího s prof. Richardem Bornatem, kteří tvrdili, že se žáci dělí do dvou skupin, a to na žáky, kteří jsou a nejsou schopni se naučit programovat. Na základě svého výzkumu přišli tito dva výzkumníci na jednoduchý test, kterým jde oddělit žáky programování neschopné a žáky s vysokou pravděpodobností bezproblémového chápání programovacích konceptů. Přestože sám jeho tvůrce poukazuje na fakt, že po dvou letech ověřování reliability daného testu se nepovedlo prokázat jeho účinnost, žáci, kteří testem byli schopni projít, měli statisticky vyšší pravděpodobnost úspěšného zvládnutí daného kurzu (Dehnadi 2009). Tato studie byla zakončena částečnou akademickou retrakcí *Camels and Humps* přímo od prof. Bornata, který zde uvedl, že jeho definitivní a bojovné tvrzení o správnosti testu a faktické neschopnosti některých lidí naučit se programovat bylo zcestné, avšak výše zmiňované zvýšené předpoklady tento test skutečně odhaluje (2014).

Programátor Jeff Atwood shrnuje v článku *Prosím, neučte se kódovat* na příkladu starosty New Yorku, Mikea Bloomberga, který se v roce 2012 zavázal naučit se

kódovat, jak absurdně se rozšířilo celé hnutí mající za cíl naučit co nejvíce lidí programovat. Starosta či politik, stejně jako téměř kdokoliv jiný, podle programátora Jeffa Atwooda potřebuje umět číst, psát a počítat, ale schopnost psát počítačový kód mu v jeho profesi nijak nepomůže. Naopak nesrovnatelně důležitější je podle něj schopnost orientovat se na internetu (Atwood 2012).

Velice validním je jeho bod: *„[snaha naučit všechny kódovat] staví metodu před problém. Předtím, než se poženete učit se jak psát počítačový kód, přijďte na to, co váš problém skutečně je. Máte vůbec nějaký problém? Dokážete ho vysvětlit ostatním takovým způsobem, aby tomu porozuměli? Prováděli jste dostatečně hluboký výzkum vašeho problému a jeho možných řešení? Vyřeší kódování váš problém? A jste si tím jistí?“* (tamtéž) Tento bod se totiž nejvíce dotýká podstaty samotné algoritmizace. Nejprve je nutné přesně definovat daný problém, velice často je nutné tento problém vysvětlit dalším lidem a zanalyzovat ho. Ničeho z toho žáci ZŠ zpravidla nejsou schopni, přičemž se nejedná o schopnost, která by byla vrozená a samovolně se s časem rozvinula. Cílem výuky programování na ZŠ by mělo být naučit žáky právě postupy pro práci s jakýmkoliv problémem, metody aplikovatelné v libovolném odvětví lidské činnosti a stejně tak v každodenním životě.

Bývalý UI designer Applu, Bret Victor, jde ve svém článku *Learnable Programming – Designing a programming system for understanding programs* ještě dále, když říká, že výuka programování obecně a všechny standardně používané a zažité postupy jsou fundamentálně špatně. Velice tvrdá kritika Breta Victora jde však směrem k využívání zastaralých způsobů výuky programování pomocí programování v konkrétním programovacím jazyce, jakým je například v České republice tak dlouho využívaný Pascal. Podstatou první části jeho argumentů je, že správně napsaný kód zobrazí správný výsledek, ale už ne rozfázovaný postup, jak se vlastně k takovému výsledku dopracovalo, což žákům a studentům velice znesnadňuje pochopení daného řešení. Druhá část jeho argumentů se týká samotných programovacích prostředí, která dnes stále ještě spoléhají na doprovodnou dokumentaci, přičemž implementovat význam každé části kódu přímo do daného prostředí by nebyl problém (Pavlus 2012; Bret 2012).

V závěru tohoto článku Bret Victor shrnuje svá doporučení, zahrnující mimo jiné srozumitelnost samotného zápisu programu, smysluplnost sekvencí, nabídku možností, okamžité zobrazení výsledku kódu a mnoho dalšího. Všechny tyto návrhy jsou v rámci projektů Hour of Code a Scratch zahrnuty v plné míře a lze tedy tvrdit, že minimálně z hlediska designu programovacího prostředí a učebních postupů se jedná o moderní

výukové metody vyhovující nejvyšším standardům odborníka firmy Apple, jejíž celá obchodní image je založená na perfekcionismu.

Jedním z argumentů proč se učit programování je, že znalosti toho, CO se dá s počítačem dělat a JAK to počítač realizuje, se navzájem nevylučují, ba naopak. Pochopení práce počítače zvyšuje schopnost studentů a žáků řešit problémy (Heggart 2014). Dle Briana Heese se spolu s výukou programování žáci zároveň učí „*kontrolovat i detaily své práce, jak aplikovat logiku a jak u úkolu vytrvat.*“ (2014) Toto vše jsou univerzální schopnosti, které jsou aplikovatelné ve všech oblastech lidské činnosti, včetně běžného každodenního života.

2.2 Názory českých odborníků na výuku programování

V předchozí kapitole byly úmyslně vynechány příspěvky z konferencí našich českých předních odborníků na výuku programování, kterými jsou Ing. Rudolf Pecinovský, CSc. a doc. PaedDr. Jiří Vaníček, Ph.D.

První jmenovaný ve svém příspěvku *Jak efektivně učit OOP* (Pecinovský 2005) kritizuje zakonzervovanost českých učitelů, kteří z velké většiny ještě ani nepřišli na to, že nějaké objektově orientované programování existuje a učí stále jen procedurální programování. Celý problém je dále podpořen špatně koncipovanými učebnicemi, které slouží v podstatě jen k výuce syntaxe daného jazyka a samotná podstata objektově orientovaného programování je buď upozaděna nebo úplně opominuta. V témže příspěvku Ing. Pecinovský dále zmiňuje principy, na kterých by správná metodika měla být postavena, jmenovitě lze zmínit například že *„Chceme-li s žáky řešit rozsáhlejší problém kvůli nějakému malému, ale zajímavému jádru, je výhodné jim 'omáčku' předem naprogramovat.“* Jedná se o principy, které jsou v hojné míře využívány také v projektu CodeCombat (viz kapitola 3.4).

Ke kritice existujících učebnice programování se Ing. Pecinovský vrací v příspěvku *Tvorba učebnic a kurzů programování* (2011), kde své výhrady detailněji specifikuje. Největší výtkou je stále upozadění principů architektury objektově orientovaných programů, kterou by se ve skutečnosti naopak mělo úplně začínat. Obecně platí, že to, co je považováno za důležité, by mělo být probráno na začátku nebo alespoň co nejdříve, aby se látka hluboce zafixovala. V opačném případě totiž dochází k jevu v psychologii známému jako funkční fixace (opak kreativity, neboli kognitivní tendence člověka využívat jen známé a neznámé upravovat tak, aby jedinci vyhovovalo).

Další velkou výtkou Ing. Pecinovského je orientace na tzv. *„AHA“* příklady. V těch ukázka daného problému sice ilustruje jak přesně vybraný kód funguje, ale nezasazuje ho do kontextu celého programu, čehož poté učící se nováček-programátor také není schopen. Pro učebnice je dle Pecinovského nejlepší pracovat s menším množstvím programů, které žáci a studenti průběžně modifikují a jsou tak s nimi důkladně seznámeni. V takovém případě žáci mohou pracovat s komplexními kódy, ale stále se mohou soustředit na aktuální problém a vynechat již probraný kód okolo.

Přímo výuce programování na školách se Ing. Pecinovský věnoval ve svém konferenčním příspěvku z roku 2004 *Proč a jak učit OOP žáky základních a středních škol*, ve kterém tvrdí, že:

> *„objektově orientované programování není nic tak komplikovaného, aby je nebylo možno učit na základních a středních školách. Naopak si troufám tvrdit, že při tomto přístupu se žáci naučí řešit složité úlohy daleko dříve, než se nám to dařilo v dobách, kdy jsme je učili programovat 'jenom strukturovaně'“*

V tomto příspěvku je dále zmiňována otázka nejvhodnějšího věku žáků, kdy sám Pecinovský považuje za ideální věk pro začátek výuky programování pátou až šestou třídu, přičemž poznamenává, že osmá a devátá třída již je skoro pozdě.

Co se týče volby jazyka a vývojového prostředí, dle Pecinovského podmínky pro efektivní výuku splňuje nejlépe programovací jazyk Java a vývojové prostředí BlueJ, které bylo pro výuku Javy přímo vytvořeno. V témže příspěvku je dále nastíněn i možný harmonogram kroužku programování pro děti, ve kterém je jasně vidět, že je skutečně přikládán největší důraz na samotnou architekturu OOP, protože k procedurálním algoritmickým prvkům jako jsou podmínky a cykly se žáci dostanou až v úplném závěru celého kroužku (Pecinovský 2004).

Druhým jmenovaným odborníkem byl doc. PaedDr. Jiří Vaníček, Ph.D, který je vyučujícím didaktiky informatiky a dokonce i přímo didaktiky programování na pedagogické fakultě Jihočeské univerzity v Českých Budějovicích. V jeho nejaktuálnějším konferenčním příspěvku z roku 2016, *Výuka algoritmizace patří především do informatiky*, se docent Vaníček zabývá schopností algoritmizace.

V hodinách informatiky se rozvíjejí takřka výhradně uživatelské návyky při používání počítačů, které však skutečně informatické myšlení (ve smyslu computational thinking) nijak nezlepšují a mají velice krátkou životnost (konkrétní software se mění). Přínosnější by bylo vyčlenit si v rámci hodin informatiky čas na výuku algoritmizace,

kde by se žáci naučili *„následovat algoritmus, vytvářet a objevovat jej, porovnat, který z algoritmů je podle různých kritérií lepší, nacházet v algoritmech chyby a schopnost vyjádřit jej v nějakém jazyce tak, aby byl bezesporný.“* (Vaníček 2016). Algoritmizace je dle Vaníčka komplexní problematika, jejíž místo je vyhrazeno přímo v informatice a zároveň není možné ji dostatečně kvalitně učit v jiných předmětech.

2.3 Popis problematiky z pohledu autora práce

Jak již bylo výše zmíněno, vytyčit jako cíl „naučit všechny žáky programovat“, by bylo nesmyslné, až absurdní. Programování a algoritmizace jsou schopnosti nejlépe připodobnitelné k výuce matematiky. Jen málokdo se v praktickém životě setká s něčím jiným než s naprostými matematickými základy. Sčítání, odčítání, násobení, dělení, obvody, obsahy, povrchy a objemy jsou výpočty, které v běžném životě využíváme takřka na denní bázi. Posuneme-li se ale k jen lehce složitější tematice, jako jsou například kvadratické rovnice, situace je okamžitě naprosto odlišná. Stále ještě relativně jednoduchá problematika logaritmů či derivací je již v drtivé většině případů záležitostí čistě teoretickou, zdánlivě odtrženou od reality. Stejně je tomu i v případě programování. Z celkového počtu všech žáků základních škol se jen mizivé procento skutečně bude živit programováním, přesné znalosti nějakého konkrétního vyššího programovacího jazyka jako je například Java nebo C++ jsou tedy takřka zbytečné. Všichni žáci ale mohou po celý život čerpat ze znalosti alespoň základní algoritmizace.

Pointa, kterou většina výše zmiňovaných studií a článků poněkud minula, je, že výuka programování a algoritmizace by (minimálně na základní škole) měla mít za účel objasnění principů a postupů právě zejména algoritmizace, přičemž programování je jen pomůcka k dosažení tohoto cíle. Nejde tedy o schopnost programovat, ve smyslu psát kód, ale o schopnost algoritmizovat, neboli vymýšlet řešení a postupy řešení daného problému. Mnoho lidí má velké problémy, narazí-li na nějakou komplexní situaci, kterou musí vyřešit. Ať už se jedná o nějaký projekt na základní nebo střední škole, závěrečnou práci na škole vysoké, či třeba neobvyklou situaci v zaměstnání, tito lidé jsou ochromeni složitostí dané situace a mnohdy se zablokují již na otázce „kde začít?“ Tuto otázku, spolu s mnoha dalšími, řeší právě jedna složka algoritmického myšlení zvaná dekompozice. Už jen to, když se žáci na ZŠ naučí rozdrobit problém na sekvenci na sebe navazujících kroků, lze považovat za naprosto uspokojivě splněný cíl.

3 Možná východiska výuky programování na ZŠ

V posledních letech se objevuje stále více možností pro výuku programování, které jsou vhodné i pro nasazení na české ZŠ. Mezi jedny z nejvýznamnějších patří právě projekty Hour of Code a Scratch, o nichž detailněji pojednávají následující kapitoly. Kromě těchto dvou projektů bude zvýšená pozornost věnována ještě projektu Khan Academy a projektu Code Combat. Nelze se však omezovat jen na tyto čtyři vyjmenované projekty, protože existuje široké spektrum dalších alternativ a aktivní učitelé si mohou najít i alternativy, které jim budou vyhovovat více.

Největší překážkou je nutnost velice dobré znalosti anglického jazyka. Všechny velké projekty spojené s výukou programování jsou vždy v angličtině, i když většinou nabízejí širokou variantu překladů do různých světových jazyků (včetně češtiny), jenže tyto překlady nebývají aktuální, nebývají kompletní, jsou velice volné a určité informace se tak mohou úplně ztratit, a konkrétně čeština není nejfrekventovanějším jazykem. Dokumentace standardních knihoven programovacích jazyků, jako je například Java, je také vždy v angličtině. Lze tedy tvrdit, že lingua franca informatiky je anglický jazyk a lidé, kteří se chtějí informatikou hlouběji zabývat (mezi které by učitelé informatiky rozhodně měli spadat), by měli být schopni zpracovávat i složité informace podané v angličtině. Stejný názor zastává i Rudolf Pecinovský, který s oblibou říká: *„Pracujete-li jako softwaroví vývojáři, máte jen dvě možnosti: buď se naučíte anglicky, nebo změníte zaměstnání.“* (2012, s. 26)

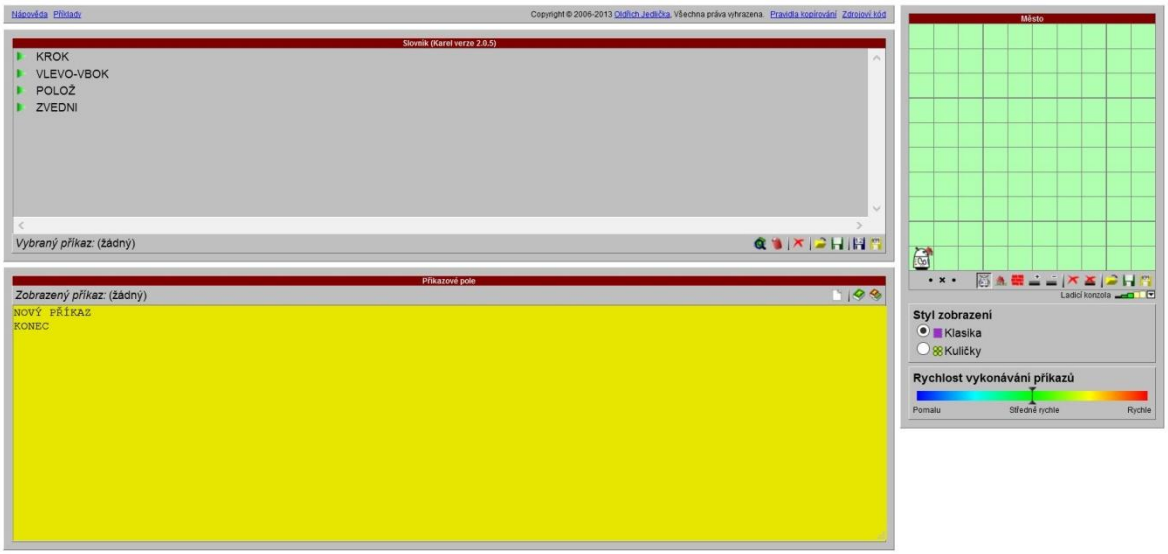
Z technicky nejméně náročných možností pro výuku programování lze pro ilustraci jmenovat anglicky psanou publikaci *Computer Science Unplugged* (Bell 2015), ve které je široké spektrum aktivit, k nimž není potřeba počítač. Nejenže je pro učitele příhodné mít několik takových aktivit připravených pro případ nouze, ale jejich využití navíc umožňuje žákům soustředit se na danou problematiku bez jakéhokoliv rozptylování způsobeného počítačem. Zajímavou a tematicky související aktivitou je například *Activity 5 Twenty Guesses* (s. 43), která ukazuje jak efektivně vytvářet ano/ne otázky, tedy v podstatě vytvářet rozhodovací bloky ve vývojových diagramech. Ve druhé části této knihy lze nalézt problematiku algoritmizace jako takové (konkrétně jsou předvedeny vyhledávací a seřazovací algoritmy) a třetí část obsahuje úvod do programovacích jazyků a tvorby procedur.

Je-li řeč o výuce programování bez počítače, nelze opomenout zmínit, že tradiční výuka programování na středních a vysokých školách zpravidla také začínala bez

počítače, protože pro výuku algoritmizace a vymýšlení vývojových diagramů stačí tužka a papír. Tento přístup však neumožňuje okamžitou automatickou zpětnou vazbu zpravující žáka či studenta o postupu a správnosti jeho práce a nijak nerealizuje takto vytvořený algoritmus. Takováto výuka má na vysokých školách své opodstatnění, ale na ZŠ rozhodně nepatří.

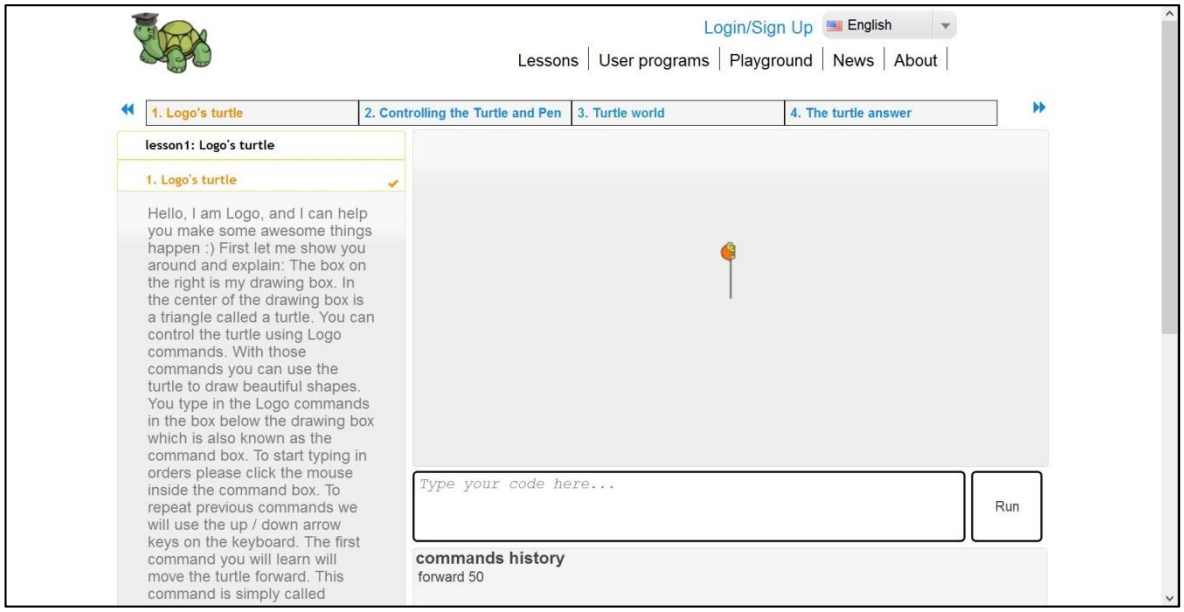
Podstatně vhodnější jsou tři dnes již „tradiční“ programovací jazyky pro výuku programování, kterými jsou KAREL, LOGO a Baltík se všemi jejich mutacemi.

KAREL je jednoduchý programovací jazyk z 80. let, ve kterém žáci ovládají malého robotka pohybujícího se po čtvercové hrací desce složené ze sta jednotlivých polí (jedná se tedy o matici 10x10). V základu umí KAREL jen minimum příkazů určujících jeho pohyb a snímajících jeho prostředí. Kombinací těchto příkazů však lze vytvářet příkazy složitější a implementovat je ve formě procedur (Musílek 2012). I přes své stáří má Karel stále nové a nové implementace (Česká škola 2009).



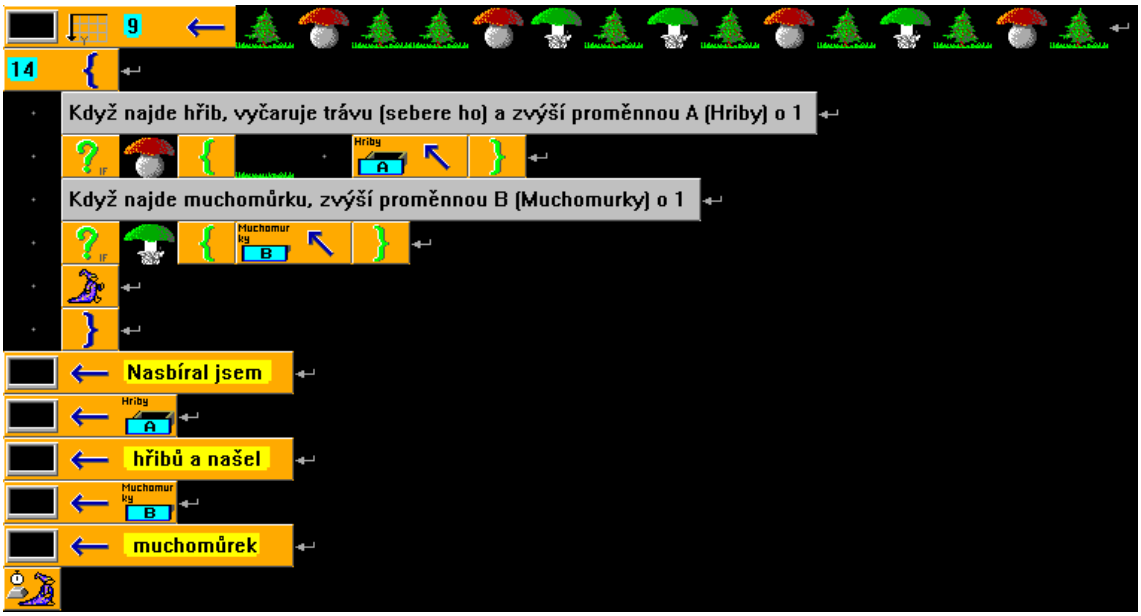
Obrázek 2 – Online implementace jazyka KAREL od Oldřicha Jedličky

Programovací jazyk LOGO, obdobně jako dále zmíněný Scratch, vznikl na MIT (Messachusetts Institute of Technology) a přestože se jeho vznik datuje již do 70. let, dodnes pro něj vznikají nové implementace a rozšíření. Základním rozdílem oproti jazyku KAREL je zrušení omezení pohybu na jednotlivá pole. Robot, zpravidla zobrazovaný jako želvička, se může pohybovat do 360° a o libovolnou uživatelem zadanou vzdálenost a jelikož za sebou vykresluje čáru, lze tak vytvářet zajímavé tvary. Kvalitní online kurz včetně online kompileru vznikl v roce 2011 pod názvem TurtleAcademy, avšak opět není dostupný v českém jazyce (TurtleAcademy 2016).



Obrázek 3 – Online výukový kurz TurtleAcademy pro programovací jazyk LOGO

Posledním zmiňovaným programovacím jazykem je Baltík. Jedná se o komerčně šířený programovací jazyk a vývojové prostředí vytvořené firmou SGP Systems. Licence se zakupují buď roční, anebo časově neomezené a spolu s programovým vybavením je k dispozici i učebnice *Baltík 3 učebnice programování nejen pro děti* od Rudolfa Pecinovského a Jiřího Váchy. Relativně vysokou pořizovací cenu (časově neomezené licence pro školu stojí 25 000 Kč) vynahrazuje velké množství oficiálních a většinou mezinárodních soutěží o hodnotné ceny. Přesto je Baltík v porovnání s ostatními projekty díky těmto nákladům jakousi černou ovcí.



Obrázek 4 – Ukázka programu v programovacím jazyce Baltík

3.1 Projekt Hour of Code

Cílem této diplomové práce bylo nalézt a otestovat projekt, který by umožnil učitelům na českých základních školách představit žákům problematiku programování a to s absolutně nejmenším možným úsílím stran učitele. V současné době lze tvrdit, že čeští učitelé se stále začlenění programování a algoritmizace do běžných hodin informatiky vyhýbají (viz kapitoly 2 a 6). Vybrat v takové situaci projekt, který by na učitele kladl vysoké nároky, ať už z hlediska času nebo nutných schopností a znalostí, by bylo naprosto nereálné. Projekt Hour of Code (dále jen HoC) této podmínce naprosto vyhovuje, protože učitel zde figuruje jen jako koordinátor práce a případný poradce při potížích.

Přestože HoC má už v samotném názvu slovo *hodina,* a původní myšlenkou bylo skutečně v rámci jediné hodiny žákům základních a středních škol ukázat, že programování není něco, čeho by se měli bát, celý projekt se od svého založení v roce 2013 neziskovou organizací Code.org velice rozrostl. Jejich hlavní kampaň je přesto stále zaměřena na tzv. *„týden kódu,“* v jehož průběhu se na celém světě masově realizují ukázkové hodiny v rámci hodin informatiky, zájmových kroužků, apod.

Zakladatelem celého projektu je Hadi Partovi, který v příspěvku *The „Secret Agenda“ of Code.org* (2013) obhajuje myšlenky, na kterých je HoC založen. Obdobně jako je popsáno v kapitole 2.3, není cílem ze všech žáků vytvořit softwarové inženýry, ale samotné pochopení, čeho všeho jsou počítače schopné, může změnit svět k lepšímu. Hadi Partovi zde tvrdí, že:

> *„Většina dnešních právníků a politiků nemá ani tušení jak webová stránka funguje, ale přesto řídí internet. Většina dnešních nemocnic stále používá papírové záznamy, které způsobují enormní náklady. Nesčetné množství světových problémů lze vyřešit technologiemi, ale ne v případě, kdy 90 % škol ani neučí, jak vlastně technologie fungují.“*

Podstatou projektu je tedy přiblížit žákům principy programování a počítačů obecně, motivovat žáky a upoutat pozornost těch, kteří mají potenciál zabývat se problematikou dále. Všichni žáci by dle Partovi měli mít možnost studovat jak počítače fungují, bez ohledu na jejich pohlaví, věk, národnost či ekonomické a sociální pozadí. Právě to je problém, který HoC adresuje nejvíce, protože v roce 2013 platilo, že programování a informační gramotnost vyučovalo jen 10 % škol v USA, které navštěvovali jen privilegovaní jedinci z bohatých rodin (Partovi 2013).

Projekt HoC formuluje svůj ústřední cíl jako:

> *„Cílem Hodiny kódu není někoho učit tak, aby se stal expertem v informatice za hodinu. Hodina stačí pouze na to, abychom ukázali, že informatika je zábavná a kreativní, že je přístupná všem věkovým kategoriím, všem studentům, bez rozdílu odkud jsou. Měřítkem úspěchu této kampaně není kolik informatiky se studenti naučí - úspěchem je široká účast napříč pohlavími, etnickými a sociálně-ekonomickými skupinami a výsledný nárůst zapojení, které sledujeme v kurzech informatiky na všech úrovních.“*
> (https://hourofcode.com/cz)

Míra zájmu o další hodiny programování, zhodnocení proběhlé hodiny HoC a zájem o kroužky programování byly otázky zahrnuté do výzkumných dotazníků právě na základě tohoto hlavního cíle projektu.

Mezinárodní dopad Hour of Code není zanedbatelný. Dle údajů uváděných společností Code.org bylo navázáno 70 mezinárodních partnerství, tutoriály HoC byly přeloženy do 46 světových jazyků a celý kurz je přeložen do 16 jazyků (Code.org 2015). Šíření projektu je dále usnadněno optimalizací pro tablety jak s operačním systémem Google Android, tak Apple iOS. Se vzestupem českého projektu *Tablety do škol* tak ani není potřeba počítač pro každého žák, postačí slušné WiFi připojení pokrývající příslušnou učebnu. V roce 2015 dle odhadů Code.org (přesná čísla ztěžuje absence nutnosti registrace) proběhlo 195 milionů hodin HoC, výuku využilo 250 tisíc tříd a celkem 8 milionů žáků, z nichž 49 % bylo ženského pohlaví (zdroj tamtéž).

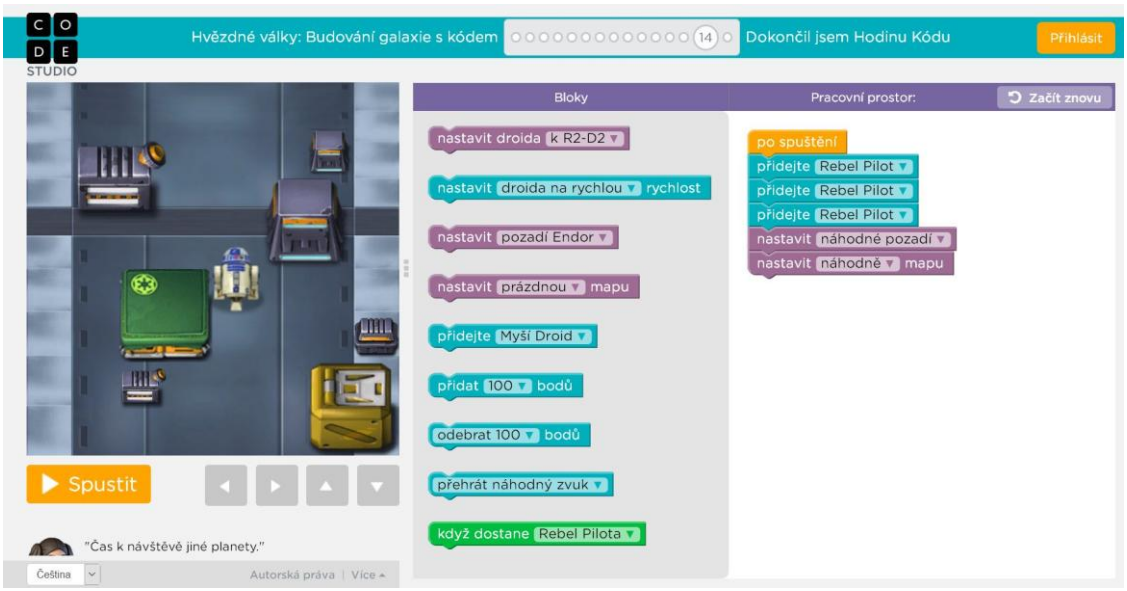
Hour of Code sice neobsahuje achievementy ve stylu například Khan Academy, učitelům je ale navrhnuto odměnit žáky, kteří se zúčastnili Hour of Code vytištěnými certifikáty, které jsou k dispozici na stránkách projektu. Přestože se jedná o drobnost, žáci tak mají možnost pochlubit se svým kamarádům a rodičům a šířit tak projekt dál.

3.1.1 Hlubší analýza projektu Hour of Code

Celý projekt je založen na vizuálním programování pomocí sestavování programu z grafických bloků ve stylu Blockly (tedy prostředí vyvinutého Googlem v roce 2011). Žáci jednotlivé bloky skládají jako kostičky LEGO a mají možnost zobrazit si ekvivalent svého výtvoru v JavaScriptu. Dílčí úkoly jsou kratičké, ale postupně se při průchodu jednotlivými úrovněmi stávájí komplikovanějšími.

Nejstarší verze HoC z roku 2013 (nazývaná autory jako *„klasické bludiště“*) byla založena na ryze procedurálním programování. Žáci se seznamovali s posloupností

příkazů, cykly a podmínkami. Tato podoba je přímo v rozporu s nutností vyučovat objektové paradigma a zaměřovat se nejprve na to nejdůležitější (viz shrnutí vybraných příspěvků Ing. Pecinovského v podkapitole 2.2). Tvůrci HoC však ve vývoji nestagnovali a každý rok ve svých ukázkových hodinách něco změní a vylepší. Na obrázku 5 lze vidět možnost vytvářet nové instance objektů v podobě přidávání nových postaviček a nově i možnosti manipulace s celým prostředím. Restriktivnost jasně specifikovaných úkolů však ze vzdělávacích důvodů zůstává zachována.



Obrázek 5 – Ukázka Hour of Code ve verzi Star Wars z roku 2015

Zájemci o další výuku ať už z řad učitelů nebo žáků mohou pokračovat na stránce https://studio.code.org/, obsahující celkem šest kompletních kurzů. První z kurzů je určen pro děti od čtyř let, které ještě ani neumějí číst. Na základní škole je tedy vhodné začínat až kurzem druhým, na který navazuje třetí a čtvrtý. V případě nedostatku času je k dispozici kurz pátý, který je zrychlený a shrnuje obsah předchozích tří kurzů. Posledním kurzem jsou takzvané *Unplugged lessons*, neboli hodiny s aktivitami, na které není potřeba počítač. Tyto aktivity jsou rovnoměrně zařazeny i do všech předchozích kurzů, zde jsou ale systematicky seřazeny a teoreticky vzato se lze obejít úplně bez počítače.

Z hlediska jazykové dostupnosti, zmiňované na samém počátku celé kapitoly 3, je nutné podotknout, že Hour of Code je dostupný v češtině. Jedná se však o neprofesionální překlad vytvořený českými nadšenými podpůrci projektu. Nově přidané lekce tedy nejsou hned přeloženy a jejich kvalita nejde předem odhadnout.

V rámci překladu je možné narazit na menší nedodělky, podivné formulace očividně způsobené doslovným překladem, či až úsměvné nesmysly jako je překlad Batman = Netopýří chlap (Code.org 2014). Tyto nepřesnosti ale zpravidla dokončení hodiny nebrání a žáci si jich nejspíše stejně ani nevšimnou.

Registrace učitelů ani žáků není nutná a při testování možnosti nasazení tohoto projektu ani nebyla využita. Registrace a vytvoření vlastní třídy, ve které má učitel přehled o postupu svých žáků, je však samozřejmě k dispozici. V rámci této práce ale byla tato možnost považována za práci navíc pro případné učitele, jež by chtěli tento projekt v praxi použít. Jakákoliv práce navíc ale byla považována za jeden z důvodů, který by mohl zájemce z řad učitelů odradit. Průchod Hour of Code byl tedy právě z tohoto důvodu otestován se všemi experimentálními skupinami bez využití registrace a shledán jako naprosto bezproblémový. Učitel však nemá možnost zaznamenávat úspěchy a neúspěchy jednotlivých žáků a žákům se neukládá jejich postup. V případě „pohodlného“ učitele či učitelky je tedy možné si vše jen tak vyzkoušet a teprve je-li projekt shledán zajímavým, je možné se začít zabývat registracemi a vytvářením tříd (což ve výsledku velice zefektivní celý vyučovací proces).

3.1.2 Vybraní partneři projektu a další přidružené projekty

Mezi hlavní partnery projektu patří giganti jako je Amazon, Apple, Disney, Google, Khan Academy (viz samostatná podkapitola dále), Microsoft, Rovio, a mnohé další (Code.org 2016b). Na propagaci a šíření Hour of Code se dále podílejí společnosti typu Bing, msn, Skype, Xbox, Scratch a nesčetné množství dalších vzdělávacích, vědeckých a technických organizací, včetně širokého spektra známých a významných osobností.

Celý projekt odstartoval svou propagaci v roce 2013 videem *What Most Schools Don't Teach* (Code.org 2013), ve kterém svou podporu celé myšlence výuky programování vyjádřily takové osobnosti, jako například zakladatel Microsoftu Bill Gates, zakladatel Facebooku Mark Zuckenberg, zakladatel firmy Valve a tvůrce herní série Halo Gabe Newell, tvůrce Twitteru Jack Dorsey a mnoho dalších. Ve spojitosti s těmito osobnostmi se dokonce vyskytly domněnky, že celý projekt byl ve skutečnosti založen proto, aby bylo pro velké firmy možné levně zaplnit pracovní pozice softwarových vývojářů. Vůči těmto teoriím se ale tvůrce Hour of Code, Hadi Partovi, ve článku *The "Secret Agenda" of Code.org* z roku 2014 vyjádřil následovně:

„Když zdůrazňujeme 'celebrity světa techniky,' není to proto, že by řídili Code.org; je to proto, že jsou vzorem pro miliony [lidí] a velkoryse nabídli svůj čas v reakci na mou prosbu. Naši přispěvatelé neřídí naši činnost. Nemají přístup k našim datům.“

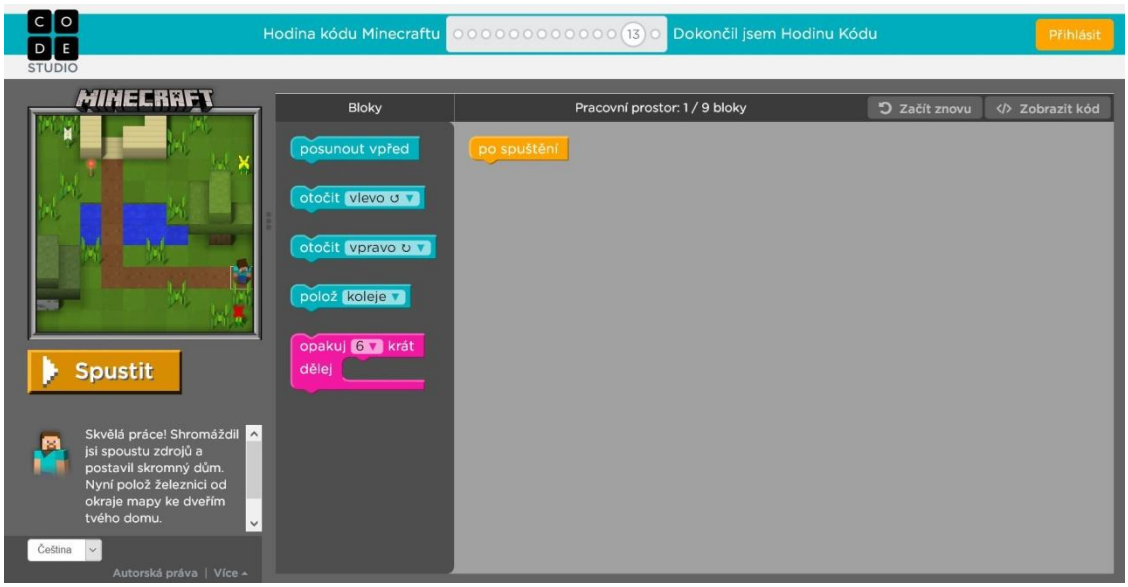
Projekt HoC brzy po jeho začátku podpořil i sám prezident Obama, který natočil video nabádající mladé lidi, aby se učili kódovat. Jedním z jeho argumentů je, že USA je na technologické špici světa, ale aby se tam udržela, je nutné zapojení mladých lidí. Video sice obsahuje apelování na typicky americký patriotismus, ale argumenty prezidenta Baracka Obamy jsou celosvětově univerzální.

Obrázek 6 – Screenshot z proslovu prezidenta Obamy podporujícího Hour of Code

Právě díky podpoře všech těchto lidí a společností je HoC tak populární a oblíbený. První verze Hour of Code byla vizuálně založena na enormně populárních hrách *Angry Birds* a *Plants vs Zombies*, což tvůrcům HoC povolili majitelé zmíněných licencovaných her, firmy Rovio a PopCap. Tato verze byla ještě v témže roce modifikována a několik posledních úrovní změnilo vizuál na *Ice Age*. Po úspěchu oblíbeného animovaného filmu *Frozen* (česky: Ledové království) přibyla v roce 2014

verze HoC *Kóduj s Annou a Elsou*. Spolu s premiérou sedmé episody *Star Wars: The Force Awakens* byla v roce 2015 vytvořena verze HoC zasazená do tohoto univerza (ukázka vybrané úrovně na obrázku 5 výše). V této variantě žáci ovládají oblíbené roboty BB-8, R2-D2 a C-3PO, jimž pomáhají plnit dílčí úkoly. Nejnovější verzí je *Minecraft*, který přibyl v roce 2016 (viz obrázek 7).

Poslední verze je v každém případě jistým zklamáním, protože výuku neposouvá dále k objektově orientovanému programování. Není zde vidět žádná evoluce ve způsobu výuky a stejně tak zde nelze nalézt ani sebemenší náznaky OOP. Naopak se jedná o drastický návrat k samotným základům procedurálního psaní kódu, které bylo k vidění již před třemi lety.



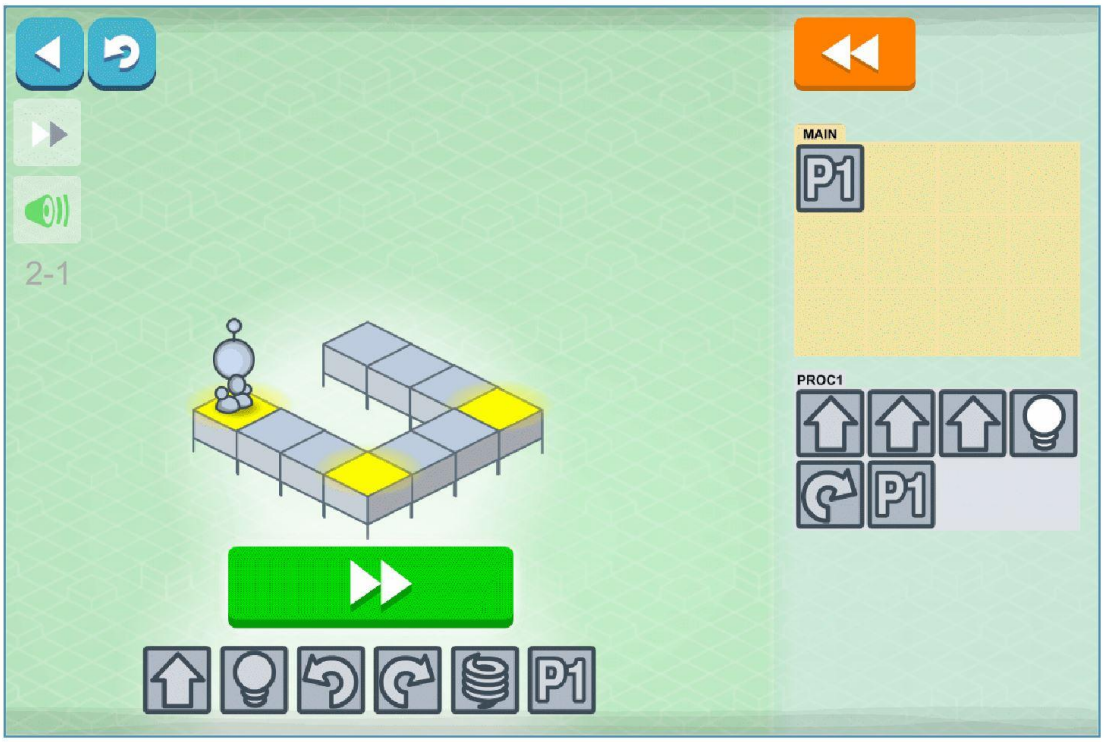
Obrázek 7 – Ukázka Hour of Code v nejnovější verzi Minecraft z roku 2016

Přestože základní myšlenka všech HoC je stále stejná, každá nová verze tutoriálu přináší nové funkce a obměny dílčích úkolů. Pro učitele na školách je tedy možné zapojit do *týdne kódu* stejné třídy každý rok znovu. To, že jedna třída absolvovala HoC například v roce 2016 neznamená, že by se tatáž třída za rok nudila, protože by bylo vše stejné jen s jiným vizuálem. Toto reflektování aktuálních trendů v oblasti oblíbených filmů a her a zároveň ochota velkých firem propůjčit projektu HoC svou licencovanou vizuální podobu je jedním z důvodů vysoce pozitivního přijetí projektu HoC mezi žáky.

Co se týče další spolupráce s jinými vývojáři a společnostmi, v současné době existuje také několik variant Hour of Code nevytvořených přímo společností Code.org. Spadá sem například hra Lightbot vyvinutá v roce 2008, tedy pět let před samotným

HoC. Tato logická hra má všude velmi vysoká hodnocení a výborné recenze. Cílem hráče je pomoci modrému robotkovi (který je přepnutelný na růžovou robotku) pomocí vytvoření sady instrukcí rozsvítit všechna modrá pole.

Původně měla hra dvě placené verze, a to pro děti ve věku 4 až 8 let a pro děti starší osmi let. Obě tyto hry jsou standardizovány pro Apple iOS, Google Android a webové prohlížeče podporující Flash. Po nástupu Hour of Code byla vytvořena ještě třetí verze Lightbota, která funguje v podstatě jako demo, protože zahrnuje úvodní úrovně verze pro děti starší osmi let a je zakončená klasickým certifikátem HoC (Lightbot Inc. 2016).

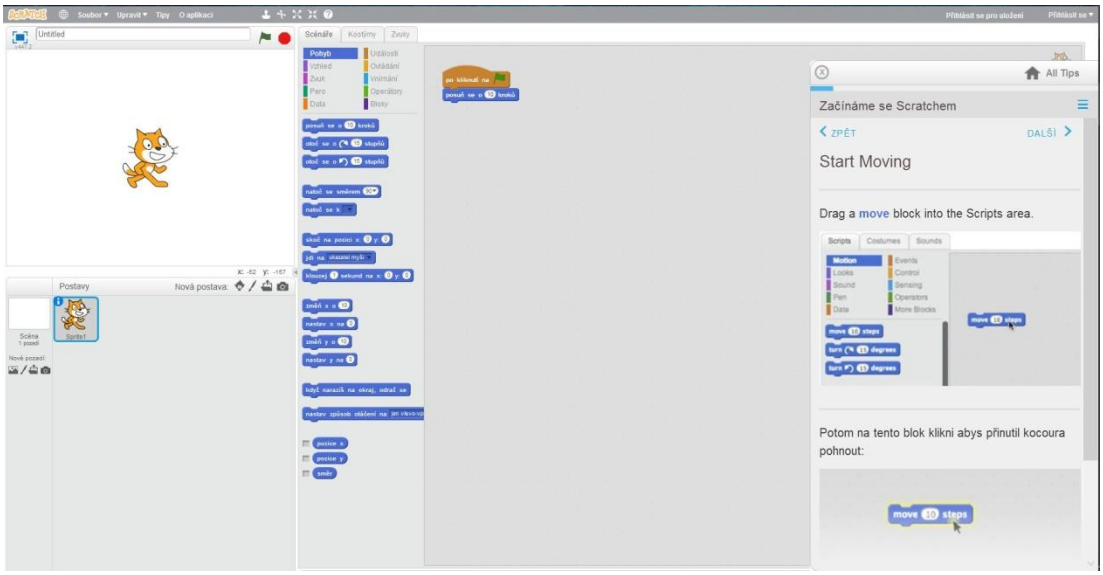


Obrázek 8 – Ukázka vyřešené úrovně 2-1 v HoC verzi logické hry Lightbot

Lighbot byl do výzkumu zařazen pro nejrychlejší žáky a pro demonstraci odlišnosti programovacích jazyků. Žáci sami viděli, že přestože vše *vypadalo* jinak než HoC, podstata jejich práce byla takřka totožná. Této zkušenosti poté bylo využito k velice zjednodušenému vysvětlení rozdílů mezi například Javou a C# a dalšími jazyky.

3.2 Projekt Scratch

Druhým centrálním projektem této práce je Scratch, vyvinutý Massachusetts Institute of Technology (dále jen MIT). Jedná se o elitní soukromou výzkumnou univerzitu, což jen potvrzuje fakt, že její absolventi nasbírali dohromady již 85 Nobelových cen (MIT 2015). V roce 2007 skupina Lifelong Kindergarten Group v MIT Media Lab vytvořila grafické programovací prostředí primárně pro výuku dětí ve věku 8 až 16 let, ačkoliv ve skutečnosti je využíván lidmi všech věkových kategorií (Lifelong Kindergarten Research Group 2016a). Obdobně jako v případě Hour of Code se jedná o projekt s celosvětovým dopadem. Je využíván ve více jak 150 zemích světa a přeložen do 40 světových jazyků, včetně češtiny. Čeština je zde však opět nedokonalá. V pravé části obrázku 9 níže lze vidět úvodní ukázkovou úlohu obsahující část česky a část anglicky. Některé věty zde tedy vůbec nejsou přeloženy.



Obrázek 9 – Základní okno programovacího prostředí Scratch

Ve své původní verzi Scratch neumožňoval vytváření vlastních bloků a tedy práci s podprogramy a funkcemi (Musílek 2012). Tuto funkci přidala nadstavba *Snap!* (dříve BYOB) vytvořená společností MioSoft ve spolupráci s univerzitou Berkeley v Kalifornii (Mönig 2016). Scratch však nezůstal pozadu a tutéž funkci do svého základního programu obratem přidal.

Scratch umožňuje žákům vytvářet vlastní jednoduché hry a/nebo skriptované scénky, které si žáci mohou ukládat přímo do svého počítače nebo nahrát na internet. Kromě online verze, fungující přímo v prohlížeči, je možné si stáhnout i offline variantu pro

Mac, Windows nebo Linux (a to vždy buď ve verzi 1.4 nebo 2.0) a nainstalovat si ji spolu s nutným Adobe AIR přímo do počítače. Internetové připojení poté není potřeba a jedná se tak o vhodnou variantu pro využití ve výuce v hodinách informatiky.

Jelikož je celý projekt primárně zaměřen na děti, tvůrci vytvořili také přímo na stránkách Scratche diskuzní fórum, kde mohou děti i učitelé prezentovat své projekty. Kromě fóra projekt doprovází i oficiální online komunita ScratchEd, která je neustále kontrolovaná administrátory, a jedná se tak o bezpečné místo pro děti i učitele k získávání nových zkušeností, případně se zde lze s nějakým problémem obrátit na ostatní členy komunity. Scratch tak úspěšně kombinuje učení programování s určitými omezenými prvky sociálních sítí.

Aktuálním tématem jsou v současné době také zásady ochrany osobních údajů. Jelikož je Scratch určen zejména pro děti, jejich soukromí je chráněno minimálním možným množstvím informací nutných k registraci (konkrétně se jedná o přihlašovací jméno, pohlaví, stát, rok narození a kontaktní email). Stejně jako v případě Hour of Code však registrace vůbec není nutná a to ani ke stažení vytvořených projektů do PC.

Projekt Scratch už byl adresován v mnoha diplomových pracích, nebude mu zde tedy ponechán takový prostor jako předchozímu projektu Hour of Code. Nebude zde detailně rozebíráno interface a veškeré programovací možnosti Scratche. V následující podkapitole je tento projekt porovnáván s projektem HoC a jsou zde zmíněny jen zajímavosti, opět zpravidla v souvislostech s Hour of Code.

3.2.1 Hlubší analýza projektu Scratch

Dětské programovací jazyky a jejich vývojová prostředí lze v podstatě rozdělit do dvou skupin. Na vývojová prostředí restriktivní (tedy ta, která jsou řízená jednotlivými krátkými a vysoce specifickými úlohami zaměřenými na konkrétní problémy z oblasti programování) a jazyky volné (tedy naopak ty, které umožňují uživatelům vytvářet cokoliv dle libosti, ale zároveň zpravidla *„nevedou uživatele za ručičku“*). Doc. PaedDr. Jiří Vaníček, Ph.D. tyto dvě kategorie programovacích prostředí označuje jako *„uzavřená“* a *„otevřená“* (Vaníček 2016). Mezi první jmenované patří právě Hour of Code, zatímco Scratch je prostředí volné. V případě tohoto způsobu dělení se nelze odkazovat na programovací jazyk jako takový, rozhodující je totiž vždy konkrétní programovací prostředí. Například jazyk LOGO je volný, ale TurtleAcademy (zmiňovaná v úvodu celé této kapitoly) je již prostředí restriktivní.

Rozdělit si programovací jazyky tímto způsobem je pro učitele výhodné z toho důvodu, že pro prvotní seznámení s novým programovacím jazykem je nejvhodnější využití restriktivního vývojového prostředí. Žáci si tak mohou všechny funkce systematicky a řízeně vyzkoušet a naučí se také základní programovací konstrukce.

Tvůrci Scratche sice vytvořili určité návody a postupy pro seznámení s jejich programovacím prostředím, ale jak bylo ukázáno na obrázku 9, česká lokalizace je velice špatná. Kromě jmenovaného průvodce zabudovaného přímo v samotném prostředí Scratche byl vydán ještě *Getting Started Guide* (Lifelong Kindergarten Research Group 2013), což je 16stránkový PDF soubor, který v podstatě replikuje ukázkový příklad z průvodce prostředí Scratche. Pro učitele, jejichž anglický jazyk je na slušné úrovni, je k dispozici mnohem zajímavější materiál, původem také z MIT. Jedná se o *Creative Computing*, což je principiálně metodická příručka pro učitele vyučující programování za pomoci Scratche (Brennan et al. 2014).

V obou výše jmenovaných příkladech je nutné počítat s jazykovou bariérou a většími nároky na učitele (z hlediska času, znalostí, příprav, apod.), které jdou eliminovat seznámením žáků za pomoci restriktivního programovacího prostředí. Hour of Code a Scratch sice nejsou úplně totožná prostředí, ale pro úvod výuky programování je HoC ideálním materiálem. Po průchodu alespoň jedné úvodní HoC jsou již žáci seznámeni s principem programování ve vizuálním programovacím prostředí, chápou základní cykly a podmínky a přidá-li se i Lightbot, znají i princip podprogramů. Učiteli tak stačí jen vysvětlit, jak se pracuje se sprity a záložkami příkazů a později ukázat realizaci podprogramů pomocí vytváření nových bloků.

Při podrobnějším zkoumáním statistik zveřejněných tvůrci Scratche si lze povšimnout určité zajímavosti v souvislosti s rozšířeností Scratche. Popularita a rozšířenost Scratche sice neustále stoupala, avšak od svého vzniku v roce 2007 do objevení Hour of Code v roce 2013 (tedy za šest let své existence) získal Scratch celkem zhruba 37 tisíc aktivních uživatelů. Od chvíle, kdy vznikl Hour of Code se však počet uživatelů Scratche jen do konce roku 2014 téměř ztrojnásobil a dosáhl hodnoty celkem 109 tisíc aktivních uživatelů. S rostoucí popularitou Hour of Code tedy skutečně roste popularita dětského programování obecně. Nejaktuálnější hodnoty Scratche jsou z května 2016, kdy bylo nahlášeno 242 tisíc aktivních uživatelů a křivka neustále roste (Lifelong Kindergarten Research Group 2016a). Od vzniku HoC se tedy počet uživatelů Scratche v průběhu tří let zšestinásobil.

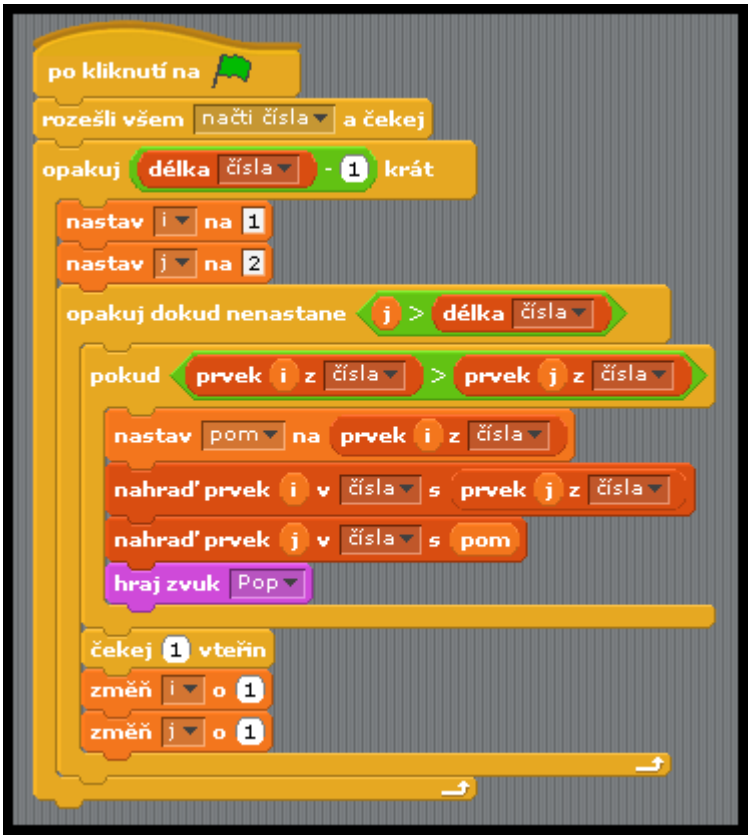
Další zajímavostí je takzvaný Scratch Day, neboli Den Scratche, který oficiálně vznikl v roce 2009 (Lifelong Kindergarten Research Group 2016b). Ve své podstatě se jedná o předchůdce Týdne Kódu, se kterým přišel Hour of Code. Původně se jednalo o uzavřené setkání odborníků na Scratch, tedy konferenci prezentující nápady a návrhy na další rozvoj Scratche, včetně betatestu nových verzí. Tato myšlenka s postupem času zůstala relativně zachována, avšak Scratch Day přestal být lokální událostí, konference o Scratchi se v tento den konají po celém světě a jedná se o oslavu úspěchů, kterých Scratch dosáhl. Přestože je tedy myšlenka celosvětové události konající se v určitý pevně daný termín a šířící daný programovací jazyk u obou projektů obdobná, Scratch opět nikdy nedosáhl slávy jaké dosáhl Hour of Code. Zatímco HoC se pohybuje ve statisícových cifrách (viz předchozí podkapitola), Scratch Day se v roce 2016 konal jen na 659 místech v 74 státech světa.

3.2.2 Dostupné materiály k výuce programování ve Scratchi

Jelikož je Scratch zajímavým a zároveň již také devět let starým projektem, existuje již mnoho akademických prací na toto téma, včetně celé řady učebnic. Pro české učitele však opět nastává tradiční problém jazykové bariéry, protože takřka veškeré tyto materiály jsou v angličtině. Odhlédneme-li od výše zmiňované učebnice *Creative Computing* (která je tvůrci z MIT k poskytnutí zdarma) a od možnosti nechat si zaslat anglické knihy ze zahraničních serverů, příliš možností nezbývá. Pro LOGO existuje česká učebnice *Imagine Logo - učebnice programování pro děti* od autorů Andreje Blaha a Ivana Kalaše (2006) a učebnice *Informatika pro 1. stupeň základní školy* (2012) od Jiřího Vaníčka, která ve své čtvrté kapitole adresuje problematiku programování za pomoci EasyLogo; pro Baltíka existuje učebnice *SGP Baltík* od autorů Jiřího Váchy a Rudolfa Pecinovského, čímž výčet českých učebnic programování pro děti takřka končí. Pro Scratch je v češtině kniha *Programování pro děti* (2013a), kterou z anglického překladu čínského originálu do češtiny přeložila Alena Halousková.

U poslední jmenované ještě zůstaneme, protože právě Alena Halousková v rámci své obhájené diplomové práce vytvořila *Učebnici jazyka Scratch* (2013b). Tato učebnice bohužel nikdy nevyšla tiskem, avšak je volně dostupná na webu. Autorka zde navrhuje velké množství příkladů, včetně postupů jejich řešení, komentářů k probírané problematice a doplňujících otázek pro žáky. Zároveň je zde výborná ukázka toho, že i grafické programování se může zabývat plnohodnotnými programovacími problémy,

jako je například řadící algoritmus probublávání neboli bublinkové řazení, zobrazený na obrázku 10 (Halousková 2013b). Autorka této online učebnice v rámci ukázkových úloh nabízí i další řešené příklady ilustrující i vybrané problémy z matematiky a fyziky.



Obrázek 10 – Algoritmus bublinkového řazení vytvořený ve Scratchi

3.3 Výuka robotiky a programovatelné LEGO Mindstorms

Jedním ze základních principů výuky programování na ZŠ je, že žáci musí vidět, co jimi vytvořený program nebo algoritmus skutečně dělá. Tomuto pravidlu sice vyhovují všechny výše zmiňované projekty, avšak pro žáky je mnohem atraktivnější vidět činnost nějakého jimi ovládaného reálného a hmotného objektu. Zde přichází ke slovu výuka robotiky a mechatroniky, která je ale zároveň nejhůře realizovatelná a to zejména z důvodu vysoké vstupní finanční náročnosti při pořizování vybavení celé učebny.

Tradiční česká stavebnice Merkur v současné době přichází s novinkou, kterou jsou robotické programovatelné sety. Jejich cena je však zatím vysoká (cca. 4000-8000 Kč) a programování je založeno na mikroprocesorech Atmel a jejich IDE Atmel Studio. Takováto varianta je vhodná pro technicky zaměřené střední a vysoké školy, avšak pro plošné nasazení na školách základních je vše příliš složité.

Stejným neduhem naštěstí netrpí stavebnice LEGO Mindstorms, jejíž tvůrci pro své programovatelné centrální řídící jednotky (které jsou také založeny na mikroprocesorech Atmel) vytvořili vlastní ryze grafické a interaktivní IDE (Integrated Development Environment neboli vývojové prostředí) Mindstorms NXT 2.0.

LEGO Mindstorms je založeno na stavebnici LEGO Technic, která je doplněná o řídící jednotku, různé senzory pro snímání okolí a motorky pro pohyb (viz obr. 11).

Obrázek 11 – Řídící jednotka LEGO Mindstorms NXT s motorky a senzory

Mezi základní senzory patří ultrazvukový senzor, dotykové, optické a zvukové čidlo. Ultrazvukový senzor snímá vzdálenost od nejbližší překážky před senzorem, je však relativně nepřesný. Dotykové čidlo funguje na principu tlačítka a snímá se jeho zmáčknutí. Zvukové čidlo reaguje na určitou intenzitu zvuku, nerozlišuje však žádné detaily. Tento senzor nelze naprogramovat na hlasové příkazy, nastavuje se jen hraniční hodnota hluku, při které se něco stane. Tímto „hlukem“ může být tlesknutí, křiknutí, bouchnutí, či jakýkoliv jiný zvuk. Optické čidlo snímá intenzitu světla, které na něj dopadá. Hodnota je udávána v procentech, přičemž 0 je tma a 100 je jasné světlo. Toto čidlo si může samo přisvěcovat a lze tak například snímat barvu povrchu (čidlo rozezná bílý podklad od černého a lze tak robotovi nastavit například pohyb po čáře nebo v určité vymezené oblasti). Motorky zároveň fungují jako rotační senzory.

Výše zmíněné senzory lze zkombinovat, vytvořit dle návodu například inteligentní autíčko (viz obrázek 12), a naučit ho samo podélně parkovat. Tato naše modifikace byla předvedena na akci Hrajme si i Hlavou 2014.

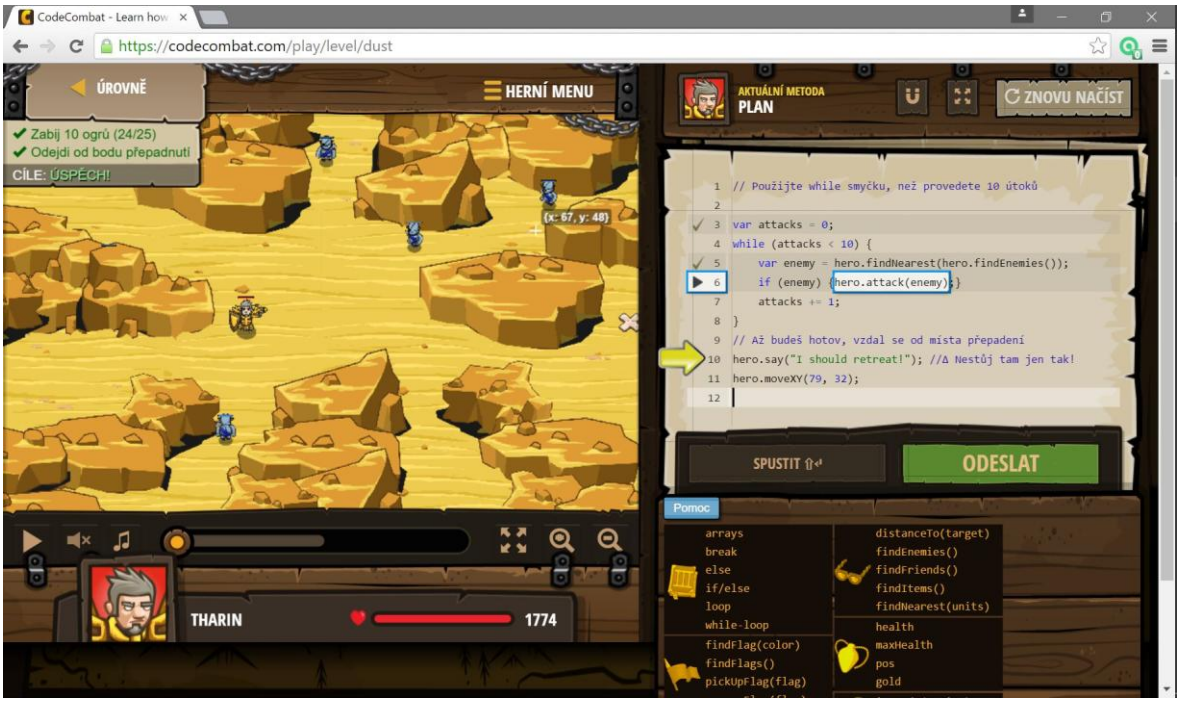
Obrázek 12 – LEGO Mindstorms Intelligent Car

LEGO Mindstorms je vhodné i pro projektovou výuku a propojení s dalšími vzdělávacími oblastmi. Mezipředmětové propojení lze ilustrovat na možnosti vytvářet roboty demonstrující například různé fyzikální principy (Coufal 2010 a 2014). Stejně jako v případě Baltíka i LEGO Mindstorms nabízí široké možnosti z hlediska soutěží, které jsou opět i na mezinárodní úrovni. Ukázka LEGO robotů byla v rámci výzkumu použita jako motivační prvek, který u žáků vzbudil velký ohlas a zájem.

3.4 Projekt CodeCombat

Zatímco předchozí tři projekty byly založeny na možnosti programovat pomocí skládání grafických bloků, projekt CodeCombat zastává diametrálně odlišný přístup. Tvůrci tohoto projektu v rámci příspěvku *3 důvody proč „Počítačová gramotnost“ kazí výuku kódování*, zveřejněném na dedikovaném blogu uznávají, že vizuální programování skutečně je vhodné pro malé děti, avšak tvrdí, že starší a dospělí už z něj nic nezískají. Ještě horší je podle nich fakt, že vydává-li se takováto výuka za programování, skutečné programování potom vypadá nudně a komplikovaně a žáci nemají dostatečně osvojeny skutečně důležité programovací návyky (Saines 2014).

CodeCombat se zaměřuje na výuku skutečného programování za využití skutečných programovacích jazyků Python a JavaScript, včetně všech formálních pravidel syntaxe. Vše stále probíhá formou hry. Na obrázku 13 je vidět rozložení herního okna, ve kterém hráč pomocí zápisu kódu na pravé straně obrazovky ovládá svého vybraného avatara (a potažmo jeho vojáky) na straně levé. Hra je rozdělena do pěti velkých oblastí a každá oblast do dílčích úkolů, jejichž splnění může trvat pouhou minutu, ale také hodinu i déle. Za splnění úkolu je hráč odměněn krystaly, za které si pořizuje nové vybavení pro svého avatara.



Obrázek 13 – Herní okno projektu CodeCombat

Vybavení určuje klasické množství životů, obranu a útok, ale také co všechno avatar může provést, co všechno umí. V podstatě se tedy jedná o knihovny s předdefinovanými metodami. Na obrázku 13 jsou vpravo dole vidět dostupné metody, u kterých se po najetí myši zobrazí plnohodnotná programová dokumentace. S novým vybavením se objevují nové metody a hráč se kromě klasického základu spočívajícího v cyklech, podmínkách atp. postupně učí například i plnohodnotnou tvorbu funkcí a jejich volání. Motivace hráčů je dále podněcována systémem achievementů a levelováním postavy.

Pro hraní je nutná registrace, která je sice zdarma, ale zároveň neobsahuje všechny úrovně a nabízí jen dva hrdiny (respektive jednoho, ale v obou pohlavích). Pro odemčení dalších úrovní je vyžadována měsíční platba. Ta ale není úplně nutná, protože v podstatě celá hra se dá „dohrát“ i zdarma. Placené úrovně jsou totiž namíchány s úrovněmi zdarma a lze na ně tak pohlížet jako na určité rozšíření či prohlubování učiva, jehož vynechání nijak nebrání dalšímu postupu hrou.

Co se týče jazykové verze, základ je samozřejmě anglicky. Český překlad je implementován, avšak nové úrovně neustále přibývají a jejich překlady se objevují nepravidelně. Nelze tedy počítat s tím, že žáci nenarazí na mnohdy velice složité úrovně celé pouze anglicky. Zadání je zpravidla jasné a velice jednoduché, objeví-li se ale i úplně nová konstrukce, jejíž vysvětlení je pouze anglicky, žáci budou nejspíše potřebovat pomoc učitele.

Hra je logicky strukturována od nejjednodušších částečně předpřipravených konstrukcí, do kterých stačí jen doplnit chybějící příkazy, až po naprosto volné úrovně, kdy hráč sám vytváří celý, zpravidla značně složitý zápis všech příkazů pro svého avatara. Z didaktického hlediska lze vytknout chybějící přehledné shrnutí toho, co bylo kde probráno a případně i ucelené teoretické shrnutí nové problematiky, které by se mohlo nacházet například na konci každé z pěti velkých oblastí hry. I přes tyto značně citelné nedostatky je hra koncipována velice dobře. Po průchodu všech volně dostupných úrovní měl autor této diplomové práce pocit, že si vlastně jen hrál a nic moc se nenaučil, avšak při následném pročítání interaktivní učebnice *JavaScript Tutorial* (w3schools.com 2016) si opakovaně uvědomoval: *„…to už umím, to taky, tohle taky...“* Bylo-li by doplněno výše navržené shrnutí, nebylo by této hře téměř co vytknout.

3.5 Projekt Khan Academy

Pro učitele, kteří programování sami nerozumí, ale chtěli by svým žákům nabídnout možnost kvalitního samostudia, se nabízí projekt Khan Academy. Jedná se o projekt založený Salmanem Khanem v roce 2008 (Khan 2016a), který je postaven na myšlence, že učení by mělo být individualizované a není nic inidividualizovanějšího než nabídnout lidem možnost vybrat si co přesně se sami chtějí učit a nechat je určit si i vlastní tempo. Za tímto účelem Khan natáčí instruktážní videa týkající se širokého spektra předmětů, zejména pak matematiky a fyziky. Registrace je zadarmo a celým projektem lze procházet i bez přihlášení, ale v takovém případě se neukládají achievementy (viz dále).

Velkým problémem je zde opět možná jazyková bariéra, protože veškerá videa Khan Academy jsou v angličtině. Sice existuje česká jazyková mutace známá jako Khanova Škola dostupná na adrese *https://khanovaskola.cz/* , která obsahuje překlady anglických videí ve formě velice kvalitních českých titulků, počet takto přeložených lekcí je však velice omezený. Z oblasti informatiky jsou nabídnuty zatím jen dvě série a to *Kryptografie* a *Teorie informace*. Po přihlášení na originální anglickou stránku je ale k dispozici výuka HTML, JavaScriptu a SQL (Khan 2016b). Pro běžné základní školy v České republice je tedy projekt buď úplně nepoužitelný, anebo přínosný v omezené míře. Velký potenciál má ale na základních školách specializujících se na výuku anglického jazyka.

Celý projekt je stále poněkud kontroverzní, protože stran odborníků i veřejnosti sklízí jak velké množství chvály, tak kritiky. Původně telefonní doučování 13leté Nadii, příbuzné žijící na druhé straně státu, se postupně změnilo v zasílání videí, která si Nadia mohla sama přetáčet a opakovat dle potřeby. Brzy Khan doučoval mnoho dalších příbuzných a odtud se jeho YouTube videa lavinovitě rozšířila do celé Ameriky a dále. Khan si uvědomil jak mocným nástrojem takovéto vyučování je a po dotaci od Billa Gatese najmul nové spolupracovníky, se kterými vytvořil systém pro podporu výuky na školách. Systém, který od roku 2011 mnozí učitelé v USA využívají při svých hodinách. Samotné vyučovací hodiny mohou být díky Khanovým videím úplně „převráceny“ – děti se učí doma a vlastním tempem a ve škole zpracovávají úkoly, respektive procvičují si naučenou látku (Thompson 2011).

Zvládnutí lekcí v Khan Academy je odměněno klasickými achievementy, tedy získáváním virtuálních odznáčků. Tato motivace je dle slov učitelů až ohromujícím způsobem efektivní (tamtéž). Někteří kritici však poukazují na fakt, že mnozí žáci se

nesnaží skutečně zvládnout probíranou látku, ale pouze získat dané achievementy. Lekcí tedy jen proskáčou a její obsah prakticky ignorují (Watters 2011).

Mezi hlavní argumenty proti Khan Academy patří „odlidštění“ celého vyučovacího procesu. Látka je zprostředkována skrze přednášky s ukázkami a interakce mezi vyučujícím a učícím se je naprosto nulová. Zvládnutí dané lekce je kontrolováno sérií otázek (výpočtů v matematice), které jsou strojově ověřeny, a po jejich správném zodpovězení se studentovi otevře další lekce. V podstatě se tedy jedná o „tradiční“ učení, kde učitel mluví a student poslouchá, načež prochází řadou drilů. (Watters 2011).

Bez ohledu na to, zda mají pravdu zastánci či kritici, jedná se o projekt, se kterým je i do budoucna třeba počítat, protože v době psaní této práce byla Khan Academy překládána do 36 světových jazyků, od jejího založení proběhlo přes 580 milionů lekcí a studenti zpracovali téměř čtyři miliardy doprovodných příkladů (Khan 2016a).

V této kapitole byly shrnuty všechny v současné době nejvýznamnější možnosti pro výuku programování. Ne všechny jsou v našich podmínkách plně realizovatelné (zejména kvůli jazykové bariéře), avšak všechny mohou být učiteli informatiky využity přinejmenším jako inspirace do hodin. Ze všech výše popsaných projektů byly vybrány dva, jmenovitě Hour of Code a Scratch. Tento výběr je odůvodněn minimálními nároky na vybavenost tříd a přípravu učitelů a má tedy předpoklady pro plošnou realizaci v rámci standardního kurikula informatiky na základních školách.

4 Specifika výzkumu na vybrané základní škole

Celý vlastní výzkum probíhal na ZŠ Chlumec nad Cidlinou na konci školního roku 2014/2015. Mezi základní specifika školy v tomto školním roce patřila naprostá absence učitele s aprobací informatiky a vysoký počet žáků ve všech ročnících. Informatika je zde dále standardně vyučována ve dvouhodinových blocích (dále označovaných jako *dvouhodinovky*) a protože maximální kapacita učebny je 17 žáků, třídy jsou většinou rozděleny na polovinu. Toto rozdělení je genderové a žáci a žákyně mají informatiku střídavě, z jejich pohledu tedy jednou za čtrnáct dní. Všechny tyto aspekty základní školy Chlumec nad Cidlinou v kombinaci s velkou vstřícností učitelů i vedení školy udělaly ze školy ideální cíl pro výzkum zaměřený na možnosti programování na ZŠ.

Dalším z důležitých specifik bylo také období, ve kterém výzkum probíhal. Červen, jakožto konec školního roku a téměř začátek letních prázdnin, hrozil značně sníženou pracovní morálkou a zvýšeným výskytem problémů s chováním mezi žáky, na což jmenovitě upozorňovali i někteří učitelé. Pro posouzení této skutečnosti nejsou žádné skutečně objektivní metriky, ale na základě subjektivního pocitu autora této diplomové práce se až na dvě výjimky prakticky žádné takové problémy nevyskytly. Těmito výjimkami byly jedna šestá a jedna devátá třída, ve kterých bylo nutné vytvořit pět až šest dvojic, protože třídy byly spojené a žáci se po jednom k počítačům nevešli. Žáci dále naprosto nebyli zvyklí pracovat ve skupině a místo produktivní spolupráce se většinou jednalo o obyčejné malicherné hašteření. Právě kombinace těchto dvou silně negativních faktorů byla pravděpodobně příčinou vzniklých obtíží, protože ostatní šesté a deváté třídy problémy obdobného charakteru nevykazovaly.

Výzkum byl dále rozdělen do dvou základních částí. Kvalitativní část výzkumu měla opět dvě fáze. Nejprve byl za pomoci kvalitativního rozhovoru zjišťován názor učitele/učitelky na problematiku programování na ZŠ, a to předtím, než měli možnost vidět ukázku dané výuky v rámci praktické části výzkumu. Druhou fází byl výstupní rozhovor, ke kterému došlo poté, co byl vyučující alespoň v jedné ze svých tříd přítomen po celou dvouhodinovku Hour of Code a viděl tedy, jak by taková hodina programování mohla vypadat. Rozhovory (vstupní i výstupní) proběhly celkem s jedním učitelem a s jednou učitelkou a jejich analýza je obsahem kapitoly 6.

Druhou částí výzkumu bylo kvantitativní šetření dotazníkovou metodou spojené s praktickou ukázkou hodiny programování. Zde byli žáci velice jednoduše uvedeni do problematiky programování a v rámci projektu Hour of Code si vyzkoušeli, na jakém

principu algoritmizace funguje. Na začátku první výzkumné dvouhodinovky žáci vyplnili vstupní dotazník a na konci jim byl předložen dotazník výstupní. Tato praktická část výuky proběhla v rámci devíti různých tříd o již zmiňovaném maximálním počtu 17 žáků (až na ony dvě výjimky, vzniklé na základě nutnosti spojení tříd). Pro bližší informace k dotazníkům a jejich vyhodnocení viz kapitoly 5.1.1 a 5.2, celý projekt Hour of Code je podrobně analyzován v rámci kapitoly 3.1.

Původně měly všechny skupiny po této úvodní dvouhodinovce pokračovat dvouhodinovkou méně restriktivního a kreativnějšího Scratche. To se z logistických důvodů povedlo jen celkem ve třech třídách, a to v jedné šesté, jedné osmé a jedné deváté třídě. Hlavním z těchto důvodů bylo časové vymezení výzkumu, protože červen je obdobím školních výletů a mnoho hodin tudíž žákům odpadlo. Navazující dvouhodina nezatěžovala žáky žádným dotazníkem a umožnila jim dobrovolně si vyzkoušet možnosti Scratche i po svém. Celý projekt Scratch je po teoretické stránce rozebrán v kapitole 3 a analýzu průběhu jeho vlastního praktického využití v těchto třech výzkumných dvouhodinovkách lze nalézt v kapitole 5.4.

Pro zvýšení validity a reliability celého výzkumu byly dále vyčleněny dvě pilotní a dvě kontrolní skupiny (výzkum tedy proběhl celkem ve třinácti skupinách). Pilotní skupiny byly zařazeny na samý začátek a sloužily k přesnému načasování rozvržení hodiny, ke zjištění nejčastějších problémů a k ověření srozumitelnosti dotazníku. Kontrolní skupiny dostaly jak vstupní, tak výstupní dotazník, ale vlastní tělo hodiny bylo na naprosto odlišné téma. Pilotní a kontrolní skupiny jsou blíže rozebrány v podkapitolách 6.1.1 a 6.1.2.

5 Kvantitativní část šetření mezi žáky

Kvantitativní část výzkumu je založena na dotazníkovém šetření mezi žáky, a to ve formě dvou dotazníků. První z nich žáci dostali hned na začátku hodiny bez znalosti obsahu této hodiny a druhý dotazník jim poté byl předložen patnáct minut před koncem dvouhodinovky. Úkolem druhého dotazníku bylo zjistit, jestli si žáci z absolvování ukázkové hodiny projektu Hour of Code něco odnesli a jestli jsou schopni tyto nově nabyté znalosti aplikovat v praxi. Za účelem snížení strachu z neúspěchu a zvýšení návratnosti dotazníků byly oba dotazníky plně anonymní. I přesto dvě žačky a tři žáci výstupní dotazník neodeslali (vstupních dotazníků v rámci experimentálního vzorku bylo získáno celkem 104, kdežto výstupních jen 99) a použitelný vzorek byl dále snížen úmyslným sabotováním jednoho žáka (tento žák ze sedmé třídy na vše odpovídal *„bagr“*). Zpracovaných dotazníků z experimentálního vzorku tedy bylo celkem 98. Pro detailní analýzu výsledků experimentální skupiny a kompletní deskriptivní statistiku zahrnující věkové a genderové složení respondentů viz kapitolu 5.1.4.

Za účelem zvýšení reliability a validity výzkumu byly vyčleněny dvě pilotní skupiny o celkové velikosti 19 žáků a dvě kontrolní skupiny o celkové velikosti 23 žáků (viz kapitoly 5.1.2 a 5.1.3). Výzkum tedy proběhl na vzorku o celkové velikosti 146 žáků. Díky pilotním skupinám bylo možné ověřit časové rozložení dvouhodinovky a srozumitelnost zadaných otázek ve vstupním dotazníku. Pilotním skupinám nebyl předložen dotazník výstupní a posledních patnáct minut bylo využito právě pro ověřování srozumitelnosti dotazníku vstupního a řešení dílčích problémů, které se objevily v průběhu dvouhodinovky.

V rámci kontrolních skupin neproběhla standardní hodina projektu Hour of Code, ale v jedné skupině její učitel informatiky pokračoval ve výuce (tématem byl úvod do prezentací v Power Pointu) a ve druhé skupině probíhalo třídní focení. Výsledky dotazníků pilotních a kontrolních skupin jsou obsahem příloh F, I a J.

Veškeré konkrétní citované odpovědi žáků jsou citovány doslovně, tedy včetně všech chyb typu špatných nebo chybějících interpunkčních znamének, gramatických chyb či nesprávného slovosledu. Z důvodu extrémně vysokého počtu chyb tyto chyby v textu nebudou adresovány klasickým *„[sic]“* běžně využívaným na upozornění, že autor práce si je nedostatků v citaci vědom.

5.1 První výzkumná dvouhodinovka a Hour of Code

Hlavním cílem celého výzkumu bylo zjistit, jestli je vůbec možné nasadit výuku programování na základní školy a jestli je vhodné použít projekt Hour of Code. Tento projekt byl vybrán z důvodu usnadnění práce učitelům informatiky, kteří ani mnohdy programovat sami neumí. Nesmírnou výhodou projektu je dále umožnění všem žákům pracovat plně individuálním tempem. To samozřejmě muselo vést k tomu, že někteří žáci skončili dříve než ostatní a někteří naopak celý kurz nestihli. Rychlejší žáci měli za úkol po skončení ukázkové hodiny Hour of Code pokračovat alternativní ukázkovou lekcí Hour of Code v prostředí Lightbot. V rámci výstupního dotazníku se s obojím počítalo a žáci byli dotazováni, jak daleko se v obou projektech dostali.

Součástí přípravy na výzkumnou dvouhodinovku byl výběr programovacího prostředí, příprava vstupních a výstupních dotazníků a sepsání rozsáhlé písemné přípravy na dvouhodinovku. Zmiňovaná písemná příprava (viz příloha A) je takto rozsáhlá a detailní úmyslně, a to jednak z důvodu zjednodušení případného opakování šetření, jednak protože vyučující výzkumných tříd byli nápadem zaujati a měli zájem právě i o poskytnutí této přípravy. Jelikož ani jeden z těchto vyučujících nemá informatiku vystudovánu v rámci své aprobace a ani jeden z nich programování neovládá, bylo považováno za smysluplné vložit jako součást přípravy i vysvětlivky a postup práce krok za krokem.

Přestože hodinu svým způsobem vedl sám Hour of Code, počítalo se s nutností pomoci vyučujícího. Problematická místa a další komplikace byly zjišťovány v rámci pilotních skupin. Konkrétně se narazilo na naprostou nechuť žáků číst v podstatě cokoliv. Někteří žáci přeskakovali instruktážní a motivační videa (která si v případě, že by nestíhali číst titulky, mohli kdykoliv pozastavit nebo přehrát znovu) a titíž žáci poté samozřejmě měli problémy splnit zadaný úkol. Přesně tento problém se nejvíce projevil ve videích u úrovní 6 a 9 v Hour of Code. Zásah učitele, který musel jít a sám znovu přeříkat obsah těchto videí velké části třídy, byl bohužel nutný.

Hour of Code navrhuje práci ve skupině, respektive aby si žáci v případě problémů navzájem radili a na řešení spolupracovali vrstevníci sami bez pomoci učitele. Na stránce s radami k výuce Hour of Code je navrhováno, aby se žáci vždy nejprve zeptali tří dalších žáků a teprve poté učitele (Computer Science Education Week 2015). Tímto způsobem se učí i radící žáci, kteří se musí soustředit na správnou formulaci svého vysvětlení.

Třídy toho však v současné chvíli naprosto nebyly schopny. Byli-li žáci upozornění na to, že si mohou pomáhat a radit, jakákoliv známka civilizovanosti zpravidla okamžitě zmizela. Děti na sebe křičely přes celou třídu, dokonce se i místy urážely a zesměšňovaly. V několika třídách (jmenovitě se jednalo o jednu šestou, obě sedmé a jednu devátou třídu) ale bylo vidět, že jde jen o zvyk a nebyl by problém žáky na takovou formu práce navyknout. Chyba v tomto případě tedy byla v absenci směřování žáků ke kvalitní skupinové práci, což je schopnost, která se vytváří dlouhodobě a byla pravděpodobně zanedbána a nahrazena takřka výhradně „tradiční“ frontální výukou.

Drobnou odchylkou v průběhu výzkumu bylo, že si skupiny žaček více stěžovaly, „že už nemůžou.“ Ve dvou třídách jim proto bylo dovoleno po dokončení Hour of Code na Lightbotovi pracovat ve dvojicích. Dvojice byly důrazně upozorněny, že se mají radit potichu, aby nerušily ostatní a další práce byla až do konce hodiny bezproblémová.

5.1.1 Detailní analýza dotazníků

Dotazníky byly koncipovány pro žáky druhého stupně základní školy a navíc se předpokládalo vyplnění dvou dotazníků v rámci jedné dvouhodinovky. Z těchto důvodů musely být dotazníky velice stručné a pro jejich bezproblémové zvládnutí byla dokonce po pilotních skupinách jedna z otázek zaměřených na ověření algoritmizačních schopností úplně odstraněna (pro změny v dotaznících a samotných hodinách následkem zjištění z pilotních skupin viz kapitola 5.1.2).

Dotazníky byly vytvořeny pomocí Google Forms a jak vstupní, tak výstupní dotazník byly rozděleny na dvě části. V první části se zjišťovaly informace o samotném žákovi a jeho názorech na programování a informatiku obecně, včetně jeho dosavadních zkušeností. Ve druhé části poté byly jen dvě dlouhé otevřené otázky (popis cesty k sýru pro myšku a popis instrukcí pro přecházení silnice).

Pro umožnění spárování vstupního a výstupního dotazníku byla v obou verzích dotazníků jako první otázka požadavek na zadání vylosovaného číselného kódu. Následující otázky na pohlaví, věk a třídu byly ponechány i ve výstupním dotazníku, protože jejich vyplnění je časově nenáročné a tyto otázky dále sloužily k ověření, že žáci ve vylosovaném číselném kódu neudělali chybu.

Součástí prvního dotazníku dále byly dvě otázky, v nichž měli žáci na stupnici od 1 do 5 nejprve určit, jak moc je hodiny informatiky baví, a poté označit, jaký význam sami přikládají informatice jako takové. Tato otázka se vztahuje k hypotéze, že čím větší význam sami žáci přikládají informatice, tím lépe dopadnou v otázkách algoritmizace.

Následující čtyři otázky se týkaly přímo programování. První z nich byla zaměřena na představu žáků o významu skrývajícího se pod pojmem „programování“ a ve výstupním dotazníku byla tato otázka zopakována, přičemž žáci by měli být po absolvování této hodiny schopni odpovědět lépe a přesněji. Další dvě otázky se týkaly dosavadních zkušeností s programováním a případným zdrojem informací (internet, sourozenec, rodiče, ...). Poslední otázka zkoumala zájem žáků o programování, tedy chtěli-li by si vyzkoušet programování v rámci hodin informatiky.

Dnešní žáci zpravidla na počítači tráví dlouhé hodiny a jejich intuitivně nasbírané zkušenosti způsob jejich uvažování v rámci informatiky značně ovlivňují. Tento aspekt odráží otázka o dalších informaticky orientovaných zájmech žáka, nabízející k výběru hry, internet, tvorbu webových stránek, grafiku, HW a další, včetně možnosti „Jiné“, kde žáci mohli sami specifikovat další alternativy, které se v nabídce nenacházely.

Za účelem umožnění porovnání kvality odpovědí žáků ve vstupním a ve výstupním dotazníku byly celkem tři otázky bodově vyhodnocovány. Jednalo se o představu programování, popis cesty k sýru pro myšku a popis přechodu silnice (viz kapitola 5.2).

5.1.2 Pilotní skupiny

Dvě třídy, jedna osmá a jedna devátá, které se časově nacházely úplně na samém počátku praktické části výzkumu, byly určeny jako pilotní skupiny. V obou pilotních skupinách dohromady bylo celkem dvacet žáků (17 žaček a 3 žáci). Hour of Code již svým názvem napovídá koncipování na jednu hodinu práce, bylo ale potřeba vzít v potaz rozdílná pracovní tempa jednotlivých žáků, nutnost vyplnění dotazníků, kontrolu jejich srozumitelnosti a kompletnosti, začlenění demonstračních a motivačních videí LEGO Mindstorms a vlastní teoretickou vsuvku vyučujícího, která žáky uvedla do problematiky programování.

Obě pilotní skupiny dostaly jen dotazník vstupní s tím, že výstupní dotazník se rozsahem shoduje se vstupním a jeho srozumitelnost a doba nutná na jeho vyplnění tak bude obdobná. Takto získaný extra čas na konci dvouhodinovky byl použit k návratu

ke vstupnímu dotazníku a jeho rozebrání se žáky. Z důvodů níže popsaných změn v dotazníku a absence dotazníku výstupního nejsou odpovědi žáků z pilotních skupin nijak zakomponovány do výsledků experimentální části výzkumu, přesto jejich odpovědi můžete nalézt v příloze F.

Přestože vstupní dotazník měl otázky formulované jasně a srozumitelně, žáci se u otázky s myší a sýrem téměř vždy „zasekli“. Vždy bylo potřeba, aby vyučující na projektoru vysvětlil otázku sám. Toto vysvětlení bylo v podstatě jen přečtení otázky nahlas slovo od slova, tak jak to je v zadání, přesto to ale žákům k pochopení naprosto stačilo. Proto bylo původní znění otázky ponecháno beze změny. V této otázce se také poprvé ukázalo, že jedním z největších problémů celého výzkumu nebude pravděpodobně absence algoritmického myšlení u jednotlivých žáků, ale obecně nízká úroveň jejich čtenářské gramotnosti. Tento problém se dále promítl i v průběhu Hour of Code, kdy (jak již bylo řečeno) měla téměř polovina žáků tendenci motivační a instruktážní videa vypínat a přeskakovat, protože je nebavilo je číst. Danou komplikaci neodstranil ani výslovný a důrazný pokyn, že videa jsou důležitá a bez nich nebudou žáci vědět, co mají dělat. V experimentální části výzkumu s tímto bylo potřeba počítat a na krizových místech, kterými bylo vysvětlení cyklů u úlohy 6 a podmínek v úloze 9, žákům pomoci a poradit.

V pilotních skupinách bylo zjištěno, že nechávat žáky ručně přepisovat odkazy na jednotlivé stránky je zbytečně zdlouhavé a matoucí. Ke snížení přechodových časů mezi jednotlivými fázemi dvouhodinovky byly proto pro žáky v hlavní části výzkumu vytvořeny BAT soubory s příslušnými internetovými odkazy. Na sdíleném disku si žáci dle instrukcí našli složku s těmito přehledně seřazenými soubory (viz obrázek 14).

Název	Datum změny	Typ	Velikost
1. Vstupní dotazník (číslo 1)	18.06.2016 23:00	Dávkový soubor systému Windows	1 kB
2. HourOfCode - Bludiště	18.06.2016 23:00	Dávkový soubor systému Windows	1 kB
3. Lightbot	18.06.2016 23:00	Dávkový soubor systému Windows	1 kB
4. Výstupní dotazník (číslo 2)	18.06.2016 23:00	Dávkový soubor systému Windows	1 kB
5. Hour of Code - Celý kurz	18.06.2016 23:00	Textový dokument	3 kB

Obrázek 14 – Screenshot zachycující seřazení dávkových souborů pro žáky

Samotná tvorba dávkových BAT souborů je velmi jednoduchá. V tomto případě se jednalo o textový soubor s koncovkou .bat, jehož obsahem je pouze klíčové slovo `start` následované kompletním URL požadované stránky. Takto jednoduchý dávkový soubor

však neobsahuje určení prohlížeče, ve kterém se má stránka otevřít. Bylo proto nutné provést kontrolu nastavení defaultního prohlížeče. Při této kontrole se zjistilo, že ve dvou případech je jako defaultní nastaven nefunkční prohlížeč (tento prohlížeč nešel ani otevřít), po přenastavení na prohlížeč jiný už nebyly žádné problémy.

Na základě pilotních skupin byly do vstupního dotazníku přidány celkem dvě otázky. První z nich zjišťovala, mají-li žáci již nějaké zkušenosti s programováním a případně v jakém rozsahu. Jestliže žáci odpověděli, že mají, druhá nově přidaná otázka se doptávala na to, kde se daný respondent či respondentka s programováním setkali. Další změnou bylo přidání otázek do výstupního dotazníku. Přestože žáci pilotních skupin výstupní dotazníky nevyplňovali, byla nalezena menší komplikace. Z důvodu v kapitole 3.1.1 popisovaného nevyužití registrovaných tříd a následné nemožnosti sledovat pokrok žáků byly do výstupních dotazníků přidány také dvě otázky. První z nich se žáků dotazovala, jestli stihli projít celý Hour of Code a v případě, že ne, do jaké úrovně se nakonec dostali. Druhá otázka byla pro žáky, kteří HoC stihli a pokračovali Lightbotem. Tato otázka také zjišťovala, jak daleko se v této druhé hře žáci dostali.

Posledním výrazným zásahem bylo ubrání jedné bodované úlohy. Ta byla založena na včeličce hledající květinu, kde žáci měli vybrat jedno z pěti již hotových nabízených řešení, vedoucích k cíli. Úloha byla z dotazníků vyjmuta ze dvou důvodů - možnost vybírat z již hotových řešení mohla z lenosti žáků vést k náhodnému odkliknutí čehokoliv a žáci byli již tak dost vyčerpaní. Ubráním této úlohy se snížila zátěž na žáky a optimalizoval čas potřebný k vyplnění dotazníků.

Co se týče časového rozložení hodiny, nejrychlejší čas vyplnění vstupního dotazníku byl 7 minut, průměrný čas byl 12 minut a z obou pilotních skupin celkem tři žáci potřebovali pro vyplnění dotazníku více než patnáct minut, přičemž nejpomalejšímu z nich to trvalo 19 minut. Na vstupní dotazník bylo tedy pro ostré testování vyhrazeno 15 minut. Průchod všemi dvaceti úrovněmi Hour of Code zabral nejrychlejšímu z žáků 32 minut, průměrný čas byl 44 minut a celkem tři žáci nestihli projít celý Hour of Code pod 60 minut. Značný rozptyl časů potvrzuje vhodnost doplnění Hour of Code pomocí práce v alternativním prostředí Lightbot (viz kapitola 3.1.2), čímž se zajistilo udržení pozornosti nejrychlejších žáků a zároveň se zabránilo, aby tito žáci vyrušovali žáky pomalejší. V Lightbotovi se nejdále dostali čtyři žáci, a to až do úrovně 2-5.

Nečekanou komplikací se ukázala také aktualizace Hour of Code, kdy v době výzkumu Studio CODE aktualizovalo svůj základní hodinový kurz a žáci v pilotní skupině tak zjistili, že úrovně 16 až 20 jsou celé kompletně anglicky. Původně se tento

úvodní kurz skládal z jedenácti úrovní na motivy Angry Birds a devíti úrovní s motivem Plants vs Zombies. V aktualizované verzi bylo posledních pět úrovní Plants vs Zombies nahrazeno motivem Ice Age a celý tento úsek v době výzkumu vůbec nebyl lokalizován do češtiny. Originální verzi ale inženýři ze Studia CODE ponechali jako součást jejich *Kurzu K8 Úvod do počítačové vědy* (k nalezení na webu projektu), v ostré části výzkumu byla tedy použita tato verze.

Kromě několika výše jmenovaných změn ve vstupním dotazníku doznal menší reorganizace i samotný plán hodiny. Původně byla v plánu hned po vstupním dotazníku teoretická vsuvka vyučujícího, kterou by žáci byli uvedeni do problematiky programování. Na tu se mělo navázat motivací žáků třemi krátkými videi LEGO Mindstorms, které demonstrují praktičtější využití základních programovacích principů. Mnohem logičtější se ukázalo být zařazení videí LEGO Mindstorms až po Hour of Code před závěrečný výstupní dotazník. Takto navazovala hodina od nejobecnější teorie přes vysvětlení základů programování pomocí Hour of Code až po ukázku praktické aplikace nově nabytých vědomostí. Po průchodu Hour of Code a vyzkoušení Lightbota byli již žáci schopni pochopit, na jakém principu programování funguje a stručný komentář u videa s LEGO Mindstorms krokodýlem, popisující jeho program, tak měl cílený účinek.

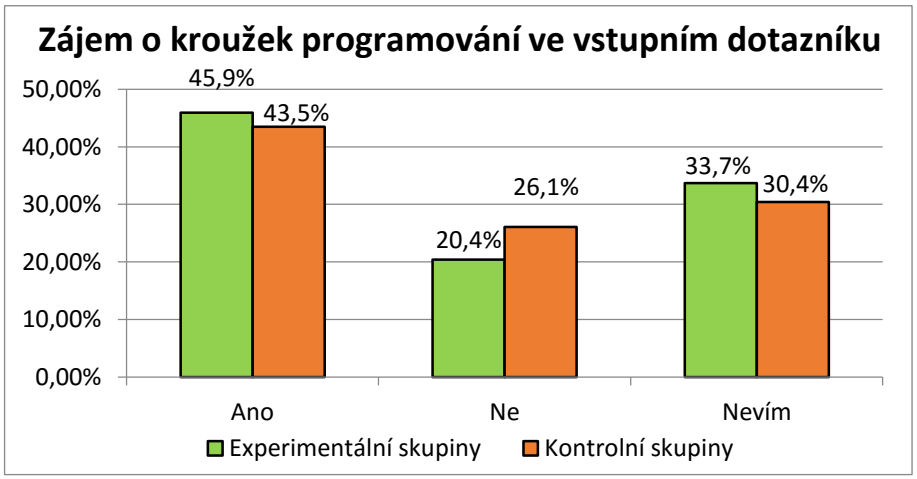
5.1.3 Kontrolní skupiny

Jako kontrolní byly designovány celkem dvě skupiny, konkrétně se jednalo o jednu šestou a jednu devátou třídu. V rámci dané kontrolní skupiny byl třídě umožněn přístup pouze k vstupnímu dotazníku a třída byla instruována ponechat si svá náhodně vylosovaná čísla. Ve sdílené složce s výzkumem se nacházel jen vstupní dotazník a třídě nebylo řečeno ani nijak jinak naznačeno, že budou na konci druhé hodiny vyplňovat dotazník další. Toto opatření bylo učiněno za účelem získání nezkreslených odpovědí. Kdyby žáci tušili, že budou vyplňovat ještě jeden dotazník na stejné téma, mohli by si vedle své zadané práce během hodiny nebo o přestávce hledat možné odpovědi. Tato možnost samozřejmě žákům zůstala, ale tentokrát platilo, že jestli si žáci budou odpovědi vyhledávat, bude to jen a pouze z jejich vlastní iniciativy. Třeba proto, že je otázky na téma programování nějak zaujaly a ne pouze proto, že chtějí vyplnit správně jakýsi dotazník.

Obsah dvouhodinové výuky informatiky u kontrolních skupin byl poté naplněn libovolným tématem, které zrovna mělo být na řadě podle platného tematického plánu. V šesté třídě se jednalo o výuku základů práce s prezentacemi a v deváté třídě bylo náplní hodiny focení na tablo. V obou případech byli žáci po celou dobu plně vytíženi a neměli čas si cokoliv vyhledávat s výjimkou přestávek, při kterých jejich činnost nebyla kontrolována.

Patnáct minut před koncem druhé hodiny dané kontrolní dvouhodinovky byli žáci požádání o vyplnění druhého dotazníku, načež jim bylo v několika větách objasněno *„o co vlastně šlo“* a byla jim alespoň nabídnuta možnost, aby si Hour of Code, Lightbota i Scratch vyzkoušeli sami doma. Jestliže žáky otázky na téma programování jakýmkoliv způsobem zaujaly, ve složce s výzkumem mohli nalézt soubor *5. Hour of Code - Celý kurz.txt* ve kterém byly veškeré odkazy k tématu algoritmizace a programování, které byly použity v rámci standardních dvouhodinovek výzkumu. Celý obsah tohoto dokumentu je k dispozici v příloze E. Jelikož ani výzkum v rámci kontrolních skupin by neměl být samoúčelný a neměl by žáky od tématu programování odradit, bylo jim ještě jednoduše dodatečně vysvětleno, proč jejich hodina vypadala takto divně a nelogicky a jak se liší experimentální a kontrolní skupiny.

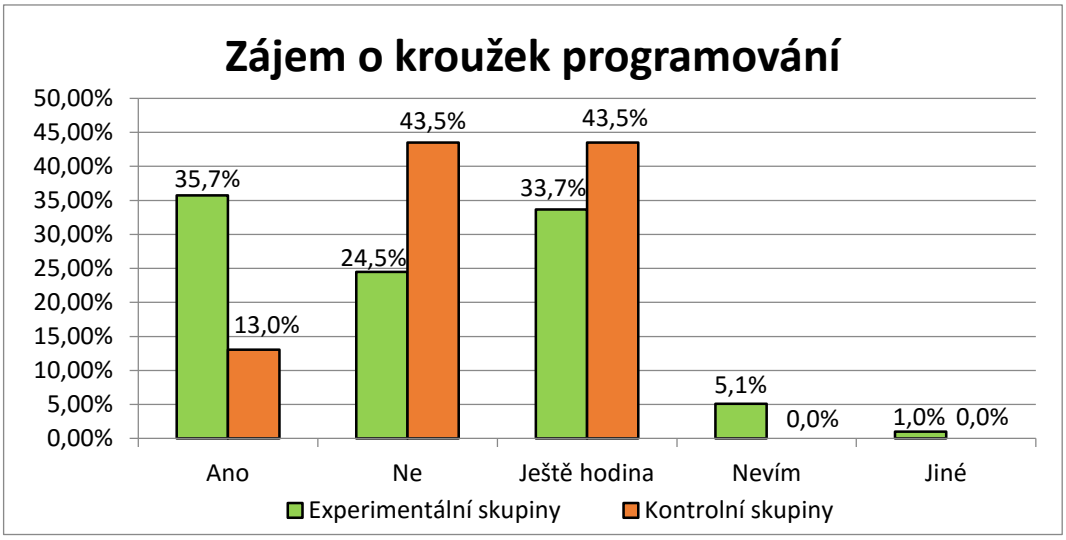
Původním smyslem kontrolních skupin bylo ověřit zlepšení z hlediska bodovaných úloh. Od tohoto cíle ale bylo z důvodů popsaných v kapitole 5.2.2 upuštěno. Namísto toho byla pozornost zaměřena na oficiálně stanovený cíl Hour of Code, kterým je zvýšit zájem žáků o programování. Pro názornost jsou hodnoty v grafech stanoveny v procentech. Tímto způsobem je možné vizuálně porovnat četnosti zastoupení odpovědí ve dvou skupinách diametrálně odlišných velikostí (experimentální skupina měla 98 respondentů, zatímco kontrolní skupina jen 23).



Graf 1 – Zájem o programování kontrolních a experimentálních skupin ve vstupním dotazníku

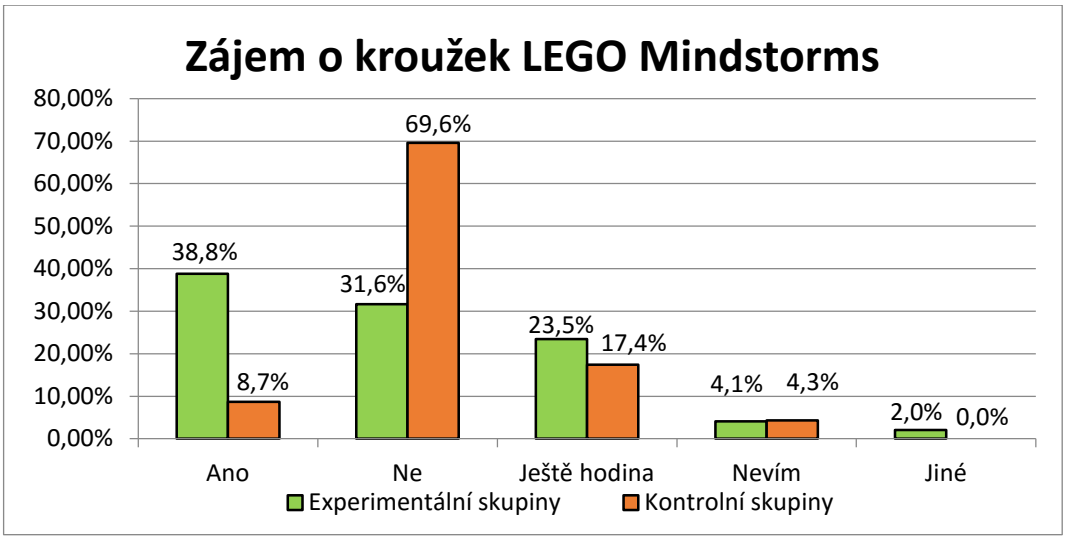
Zájem žáků o vyzkoušení programování, dotazovaný ve vstupním dotazníku, byl v kontrolních skupinách velice podobný skupinám experimentálním, přičemž rozdíly se pohybovaly v rozmezí zhruba 2 až 6 %. Téměř polovina žáků kontrolních skupin (43,5 %) zájem o programování měla, a zbývající žáci byli rovnoměrně rozděleni mezi nezájem a nerozhodnost (viz graf 1).

Situace se výrazně změnila v odpovědích ve výstupním dotazníku, kde byli žáci dotazování na zájem o kroužek programování nebo robotiky LEGO Mindstorms. V případě kroužku programování byl zájem minimální, pouhých 13 % v porovnání s 35 % v experimentálních skupinách. Zbývající respondenti v kontrolních skupinách odpověděli v 43,5 % jednoznačným ne a stejný počet 43,5 % vybralo, že před rozhodnutím by si museli vyzkoušet alespoň jednu hodinu. Tyto výsledky a rozdíly mezi kontrolními a experimentálními skupinami zachycuje graf 2 níže.



Graf 2 – Zájem o kroužek programování v kontrolních a v experimentálních skupinách

Ještě výraznější byl potom rozdíl v dotazu zjišťujícím zájem o kroužek LEGO Mindstorms. Zde téměř 70 % respondentů odpovědělo jednoznačným ne, zatímco ano vybralo pouhých 8,7 % respondentů. Vezmeme-li v potaz, že po shlédnutí ukázkové hodiny HoC s doprovodným projevem, představujícím i další projekty zabývající se programováním, mělo v experimentálních skupinách o kroužek robotiky zájem téměř 40 % respondentů, jedná se již o značný rozdíl.



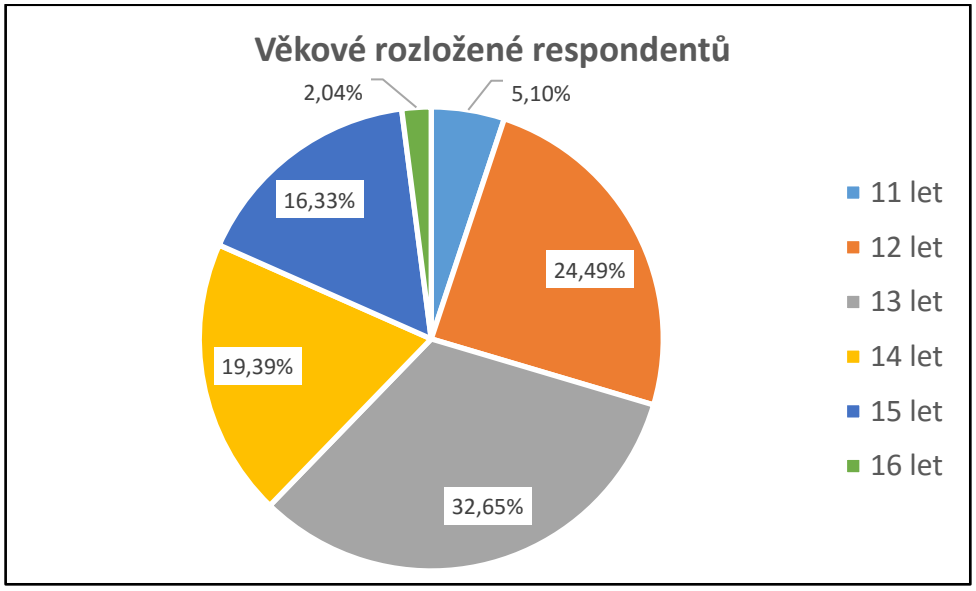
Graf 3 – Zájem o kroužek robotiky v kontrolních a v experimentálních skupinách

Na základě zjištění v kontrolních skupinách lze tvrdit, že ukázková hodina programování v rámci výzkumu v experimentálních skupinách splnila účel, proklamovaný tvůrci Hour of Code. Zájem žáků, kteří se s programováním alespoň v takto zjednodušené formě setkali, byl v případě dotazu na kroužek programování 3násobně vyšší a v případě kroužku robotiky LEGO Mindstorms dokonce více jak 4násobně vyšší. Využití projektu HoC tedy výrazně zvyšuje zájem žáků o programování a je vhodnou ukázkou pro potencionální zájemce o kroužky či jiné zájmové útvary.

5.2 Experimentální skupiny

Vzorek respondentů se skládal ze žáků šestých až devátých tříd, přičemž majoritní skupinou byli žáci tříd sedmých v celkovém zastoupení 41,8%. Vzorek by byl rovnoměrně rozložen po všech ročnících, nebýt vyčlenění kontrolních skupin z 6. a 9. tříd a školního výletu pro 8. třídy. Z hlediska genderového rozložení převažovaly dívky, kterých bylo 55 % oproti 45 % chlapců.

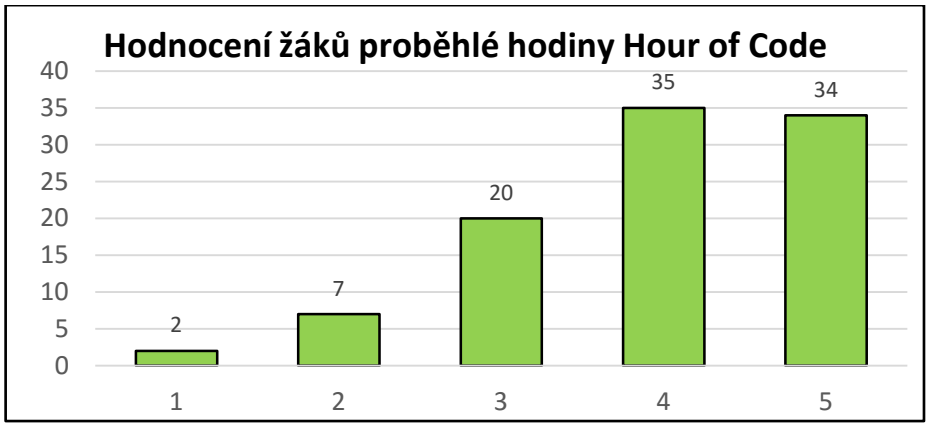
Nejmladším respondentům bylo v době provádění výzkumu 11 let a nejstarším 16. Výše zmíněné rozložení žáků do ročníků také výrazně ovlivnilo rozložení věkové, kde největší skupinou byli se svými 32,6 % žáci ve věku 13 let. Četnosti krajních hodnot byly minimální. Žáků 11letých bylo celkem 5,1 % a žáků 16letých pouhá 2 %.



Graf 4 – Koláčový graf procentuálního věkového rozložení experimentálního vzorku

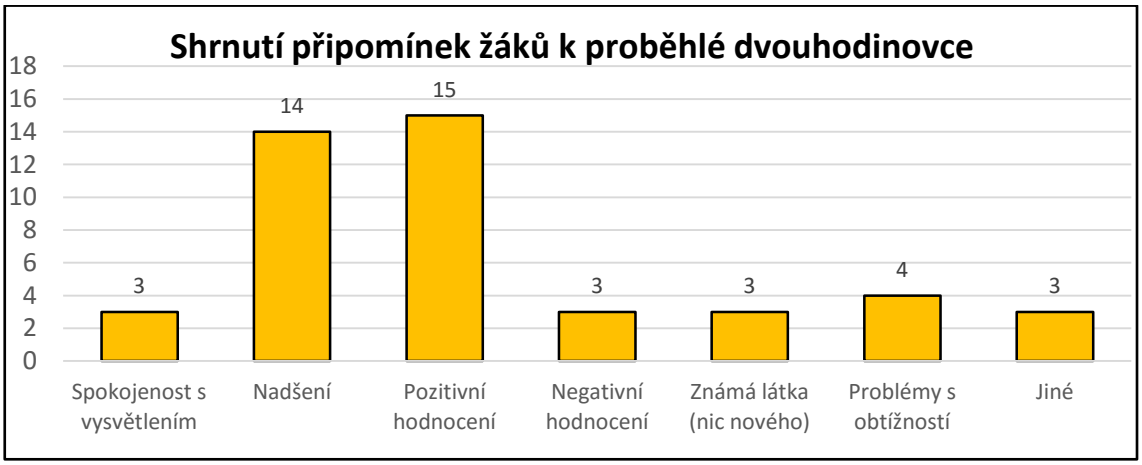
Reakce třídy na výuku byly ve všech skupinách téměř stejné. Jednalo se o novou tematiku pro žáky, se kterou se nikdy dříve v běžných hodinách informatiky nesetkali. Tomu odpovídaly i veskrze nadšené reakce na hodinu a pozitivní přijetí dané problematiky. Vnitřní motivace žáků tedy byla podnícena samotnou atraktivitou projektu i bez většího zapojení učitele.

Žákovské hodnocení proběhlé hodiny Hour of Code bylo ve výstupním dotazníku zjišťováno otázkou, jak moc je hodina bavila. Odpověď byla zaznamenávána na pětistupňové škále, kde **1** znamenala *„ani trochu“* a **5** *„strašně moc.“* Shrnutí této škály zachycuje graf 5.



Graf 5 – Sloupcový graf žákovského hodnocení proběhlé hodiny Hour of Code

Žákům byla dále ve výstupním dotazníku nabídnuta možnost vyjádřit jakékoliv další připomínky k hodině a jejímu obsahu. Otázka nebyla povinná a žáci na to byli v každé skupině upozorněni. Možnosti okomentovat hodinu využilo celkem 45 respondentů a respondentek, jejichž odpovědi bylo možné generalizovat do sedmi obecných kategorií zobrazených v grafu 6.



Graf 6 – Volné připomínky žáků překódované do obecných kategorií

Přestože dotazník obsahoval výše jmenovanou otázku, zjišťující, jak moc se žákům hodina líbila, celkem 32 žáků využilo volné otázky na jakékoliv připomínky k hodině k tomu, aby dále vyjádřili svou spokojenost či nespokojenost. Celkem 29 žáků hodnotilo hodinu pozitivně, přičemž 14 těchto odpovědí bylo klasifikováno jako projevy nadšení. Pouze tři žáci vyjádřili negativní hodnocení. Další tři žáci vyjádřili svou spokojenost s kvalitním vysvětlením, které pro ně bylo dostatečně pochopitelné a srozumitelné. Tři žáci konstatovali, že se nic nového nenaučili, avšak dva z nich

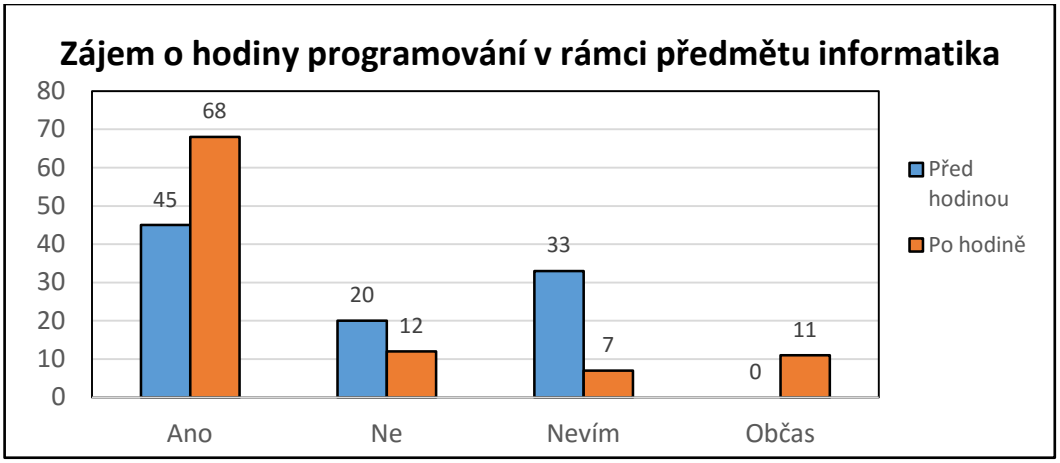
dodali, že pro člověka, který se s programováním nikdy nesetkal, byla úvodní hodina výborná a bavila i je. Čtyři žáci komentovali subjektivně vnímanou vysokou obtížnost zadaných úkolů.

Komentáře posledních tří žáků nešlo generalizovat a byly zařazeny do kategorie *Jiné*. Jedna žačka z šesté třídy byla zaujata ukázkami LEGO Mindstorms a do komentářů napsala *„chtěla bych se v hodině naučit sestavit robota na ovládání (pejska).“* Přestože tento komentář značí zaujetí danou problematikou, nelze ho zařadit do žádné z šesti kategorií. Stejně tak nezařaditelný je komentář dívky ze sedmého ročníku *„bylo to moc zdlouhave ale to musi a musi se hodne přemyšlet.“* Vybrané komentáře, které se autorovi práce zdály něčím zajímavé či přínosné, jsou vypsány v tabulce 2.

Pohlaví a ročník	**Citované komentáře respondentů [*sic*]**	**Kategorizováno jako**
Dívka, 7. třída	*„Hodina byla skvěle vysvětlena“*	Spokojenost s vysvětlením
Dívka, 6. třída	*„Hodina mě velmi bavila a mám velký zájem chodit do jakéhokoli zájmového kroužku s podobnou tématikou!“*	Nadšení
Dívka, 7. třída	*„Tato hodina se mi moc líbila a klidně bych ji vyměnila za normální hodiny.“*	Nadšení
Dívka, 6. třída	*„Hodina se mi velice líbila, nikdy jem neprogramovala, ale určte vím, že se o prázdninách k tomu to tématu vrátím.“*	Nadšení
Chlapec, 9. třída	*„hodina byla pohodova aspon nedelame furt jenom prezentace :D “*	Pozitivní hodnocení
Dívka, 6. třída	*„byla tu sranda,narozdil od jinych hodin ty byl klid coz u nas moc caste neni,bylo to zajimavi a bylo to neco jineho, zase sme se neco dozvedeli a v teto dobe se nam to hodne hodi(podle me)“*	Pozitivní hodnocení
Chlapec, 9. třída	*„S hodinou (2h) nemám žádné problémy, přivítal bych, kdyby se podobné hodiny nebo alpoň tokový styl vyučování se zavedl do běžné výuky informatiky“*	Pozitivní hodnocení
Dívka, 7. třída	*„hodina me moc nebavila a nebaví me ani hodiny informatiky nejsem počítačoví typ“*	Negativní hodnocení
Dívka, 7. třída	*„Ta hodina byla docela znuděná mohlo být míň toho bludiště.... “*	Negativní hodnocení
Chlapec, 7. třída	*„Celková hodina byla fajn. Bylo to příjemné spestření pro ty co programování vůbec nerozumí, ale ten kdo si už programování vyzkoušel tak se toho moc nedozvěděl.“*	Známá látka
Chlapec, 7. třída	*„od 16 levelu už jsem to přestal chápat“*	Problémy s obtížností
Dívka, 6. třída	*„chtěla bych se v hodině naučit sestavit robota na ovládání (pejska)“*	Jiné
Dívka, 7. třída	*„bylo to moc zdlouhave ale to musi a musi se hodne přemyšlet“*	Jiné
Dívka, 7. třída	*„budete příští rok ve škole“*	Jiné

Tabulka 2 – Vybrané komentáře z otevřené otázky na připomínky žáků k hodině

Vstupní a výstupní dotazník dále obsahoval otázku, zdali by žáci měli zájem vyzkoušet si programování v hodinách informatiky (v případě dotazníku vstupního) a jestli by v těchto hodinách chtěli pokračovat (v případě dotazníku výstupního).

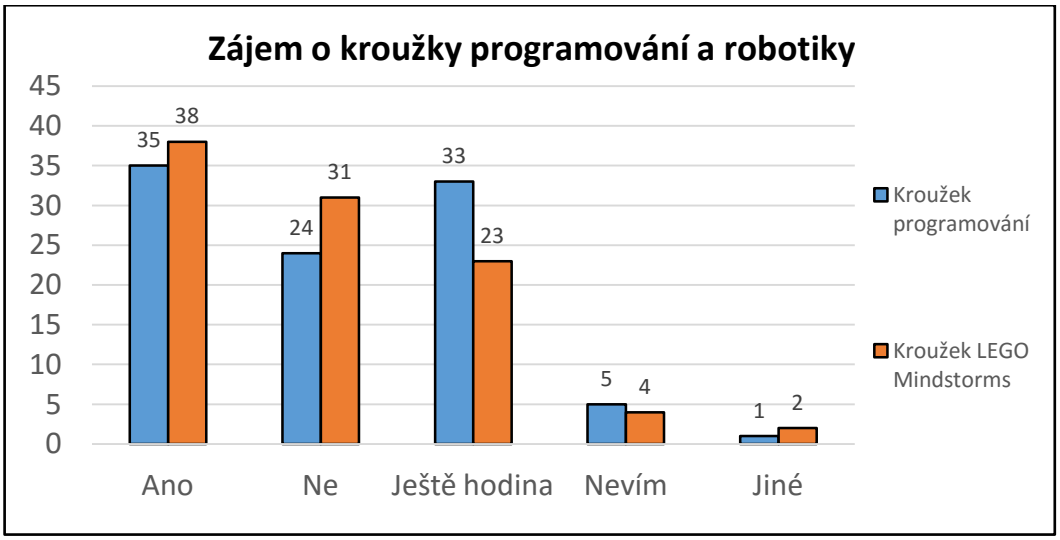


Graf 7 – Sloupcový graf shrnující zájem žáků o programování v hodinách informatiky

Z odpovědí žáků je patrný 46% zájem o programování již před setkáním s Hour of Code, který po proběhlé dvouhodinovce ještě dále vzrostl na 69,4 %. Žáci měli ve vstupním dotazníku na výběr ze tří odpovědí (ano/ne/nevím) a mohli vybrat i odpověď vlastní. Ve výstupním dotazníku byla možnost *nevím* zrušena, protože po absolvování Hour of Code si žáci mohli udělat dostatečný názor na celou problematiku a rozhodnout se ano, nebo ne. Přesto jim byla opět ponechána možnost vlastní odpovědi, které využilo celkem 23 respondentů. Na základě těchto odpovědí byly vytvořeny dvě obecné kategorie, a to *Občas* a opět *Nevím*. Žáci ale volné odpovědi využili i k vyjádření nadšeného souhlasu, protože *Ano* pravděpodobně dostatečně nevystihovalo to, co chtěli říct. Jmenovitě například jeden žák šesté třídy napsal *„určitě každý den,“* či žačka také ze šesté třídy napsala *„Samoydřejmě, bylo to výborné.“*

Obdobně byly konstruovány i otázky na zájem o kroužky programování a/nebo LEGO Mindstorms. Žáci měli opět na výběr ano/ne/nevím (kde poslední možnost byla formulovaná jako *„Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu“*), plus volnou odpověď. Této možnosti využilo celkem dvanáct žáků, jejichž odpovědi byly rozděleny do již buď existujících kategorií, nebo do dvou kategorií nově vytvořených na základě jejich odpovědí. Těmito přidanými kategoriemi jsou obligátní *Nevím* a *Jiné*. Mezi nezařaditelné odpovědi v kategorii *Jiné* patřil například realistický komentář žáka deváté třídy *„Podle počtu volného času.“*

Ze shrnutí odpovědí v grafu 8 je dále patrné, že zájem žáků o kroužky je podstatně nižší než o přidání tematiky programování do hodin informatiky. Je to i logické, změny obsahu hodin informatiky žákům do volného času nijak nezasáhnou, kdežto v případě kroužku musí žáci za tento zájem zaplatit částí svého volna.



Graf 8 – Sloupcový graf zájmu žáků o volnočasové kroužky programování a robotiky

Otázka byla koncipována s tím, že se jedná o víceméně momentální rozhodnutí žáků, které si ani nestihli skutečně nechat projít hlavou, pročež byla mezi nabídnuté možnosti zařazena i nutnost ještě alespoň jedné hodiny. Tu v případě programování ve Scratchi a podobných ryze softwarových prostředích vybralo 33,7 % žáků a v případě robotiky s LEGO Mindstorms celých 23,5 %.

Jednomyslné *Ano* vybralo 35,7 % žáků v případě kroužku programování a 38,8 % v případě kroužku robotiky. Zaujetí více než třetiny všech žáků touto problematikou se dá považovat za velice slušný výsledek. Mgr. Petr Coufal, který již od roku 2006 vede kroužek robotiky s LEGO Mindstorms (Coufal 2014), však upozornil, že mezi žáky je velice častá tendence předpokládat, že si budou hlavně stavět ze stavebnic LEGO. Ve chvíli, kdy jim ale dojde, že kroužek vyžaduje nemalé množství přemýšlení při programování, často raději skončí. Ukázková hodina programování žákům sice předvedla alespoň základy toho, co programování v grafickém prostředí obnáší, přesto by však bylo nutné s touto tendencí žáků počítat.

5.2.1 Vyhodnocení otázky na definici programování

Definice programování měla maximální počet tří bodů. Odpovědi typu *„nevím“* nebo jiné naprosto chybné odpovědi jako *„že se něco aktualizuje,“* případně odpovědi, jejichž podstatou byla uživatelská práce s aplikačním softwarovým vybavením, byly ohodnoceny nula body. Stejně tak byly ohodnoceny i definice založené pouze na slovu *„programování,“* protože vysvětlit termín jen a pouze tímtéž termínem ve stylu *„programování je programování počítačů“* nedokazuje sebemenší pochopení stran daného žáka. Varianty na *„tvorbu internetových stránek“* byly ohodnoceny za jeden bod, protože HTML, CSS jsou velice specifickým druhem „programování“ a je velice sporné, jestli tito žáci měli na mysli JavaScript (který už programovacím jazykem je). Odpovědi, jejichž podstatou byla nějaká variace na dále nespecifikovanou *„tvorbu programů, aplikací“* či dokonce jmenování konkrétního programovacího jazyka bylo za dva body. Nejpřesnější definice obsahující *„tvorbu programů pomocí instrukce počítači“* či jednodušeji řečeno *„naučení počítače něčemu novému“* či *„říkání počítači co přesně má krok za krokem dělat“* byly hodnoceny za maximální tři body.

Po proběhlé hodině se ve výstupních dotaznících objevovaly variace na *„úkol“* či *„rozkaz“* pro počítač. Přestože se svým způsobem jedná o synonyma slova *„instrukce,“* program se nikdy neskládá z jedné jediné instrukce. Takováto odpověď tedy nebyla hodnocena třemi body, ale pouze jedním, protože žáci sami viděli a vyzkoušeli si, že program se skládá z více instrukcí.

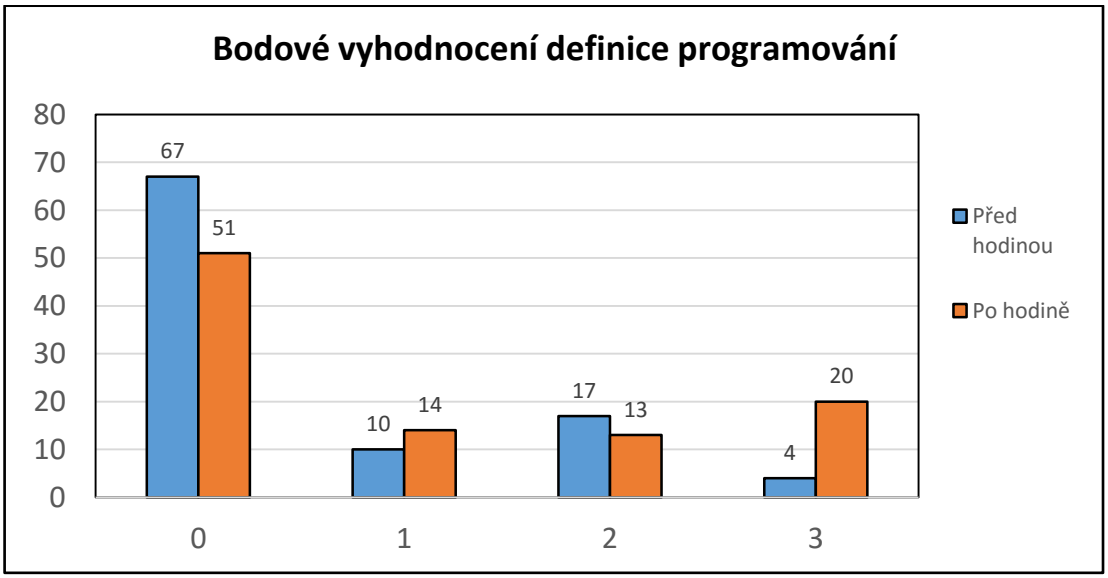
Bodů za odpověď	Typ odpovědi
0	nevím; používání a nastavování aplikací; programování je programování
1	tvorba internetových stránek; ovládání počítače/robotů; neurčité zadání pokynů (bez určení komu a proč)
2	tvorba programů/her/aplikací **nebo** jeden příkaz/instrukce počítači; jmenování programovacího jazyka
3	tvorba programů/her/aplikací pomocí příkazů/instrukcí počítači; příkazy počítači co má dělat; učení počítače něčemu novému

Tabulka 3 – Bodování odpovědí definice programování

Zajímavým faktem je, že žáci se po absolvování Hour of Code a doprovodného výkladu vyučujícího více soustředili na samotné instrukce a příkazy počítači, avšak to, že **kombinací** těchto instrukcí se vytváří nové aplikace, se v odpovědích nevyskytlo. Tento jev je možné vysvětlit nízkým důrazem na samotné definování toho, co to vlastně program je. Hour of Code ve svých videích zdůrazňuje, jak instrukce fungují a jak se dají kombinovat, avšak to, že program je kombinace instrukcí, je zmíněno jen okrajově. Obsahovala-li však odpověď, že se jedná zadávání příkazů počítači, byla ohodnocena třemi body. Pro větší přehlednost je hodnocení shrnuto v tabulce 3.

Jako nejpřesnější odpověď ve vstupních dotaznících lze označit odpověď žáka osmé třídy *„Programovaní her,programů.* ***Všechno co funguje např. na PC je potřeba na programovat.“*** [sic] To, že žák chápe, že cokoliv se na počítači používá, musel někdo nejdříve naprogramovat, je de facto vystižení samotné podstaty programování.

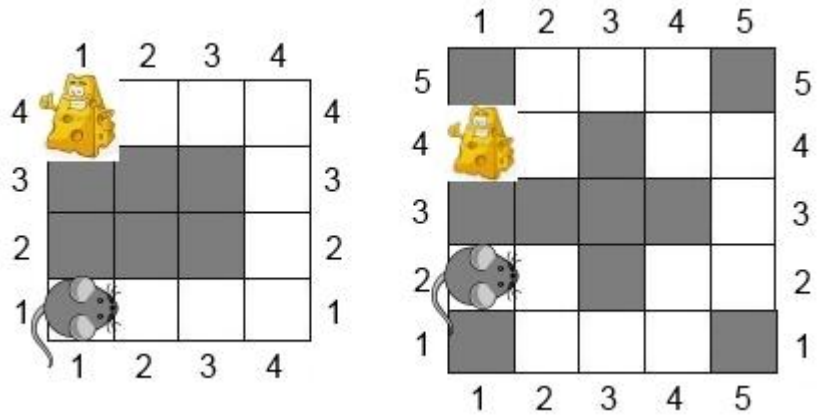
Mezi výstupními dotazníky se objevilo větší množství kvalitních odpovědí. Jednou z nejpřesnějších odpovědí mezi výstupními dotazníky je potom odpověď žáka deváté třídy *„Popis postupu pro pc,“* což je v podstatě definice algoritmu. Obdobně kvalitně odpověděla žačka osmé třídy *„Je to zadání kódu, který řekne danému přístroji co má dělat.“* či žačka sedmé třídy *„Programování je ovládání počítače/robota pomocí příkazů.“* Poněkud zkreslená, ale stále velice dobrá odpověď byla u jiné žákyně sedmé třídy *„je to krok po kroku delate veci ktere zadate do pc a robot zacne delat zadane pokyny."* Shrnutí vyhodnocení všech odpovědí žáků zobrazuje graf 9 níže.



Graf 9 – Bodové vyhodnocení úspěšnosti žáků při definování programování

5.2.2 Vyhodnocení otázky na popis cesty k sýru pro myšku

Otázka na popis cesty k sýru pro myšku kontrolovala schopnost žáků sestavit posloupnost instrukcí, která povede k požadovanému cíli. Žákům nebyly jmenovitě nabídnuty cykly, avšak byli upozorněni, že zápis mohou nějakým způsobem zkrátit, že mohou zapsat menší množství instrukcí a práci si tak ulehčit. Přijít na tento způsob už bylo plně v rukou samotných žáků. Cílem této otázky bylo zachytit, jsou-li žáci schopni zakomponovat nové myšlenkové konstrukce do svých řešení již po první expozici HoC. Plné znění otázky je k nalezení v přílohách B a C. Na obrázku 15 je myška ze vstupního dotazníku vlevo a úkol z výstupního dotazníku vpravo.



Obrázek 15 – Zadání úlohy s myškou ve vstupním a výstupním dotazníku

V případě druhého (výstupního) dotazníku byla cesta pro myšku složitější, protože v rámci Hour of Code se žáci setkali s cykly a podmínkami a stačilo jim jen přijít na to, kterou část kódu nechat kolikrát opakovat. Teoreticky nejkratší možný zápis by byl:

ve vstupním dotazníku `3x(3x(vpřed) vlevo);`
ve výstupním `3x(vpřed vpravo vpřed vlevo vpřed vpřed vlevo) vpřed vpravo vpřed;`

Podstatou tohoto úkolu tedy bylo hledání opakujících se vzorců. Ve druhém případě již také někteří žáci stihli kromě Hour of Code absolvovat i část Lightbota, kde se dozvěděli o procedurách. Mohli tedy také vytvořit například něco jako:

```
zatáčka = (vpřed vpravo vpřed);
rovinka = (vlevo vpřed vpřed vlevo);
Program = 3x(zatáčka rovinka) zatáčka;
```

Této možnosti však ani jeden z žáků nevyužil ani ve vstupním, ani ve výstupním dotazníku. Samotné hodnocení celé úlohy probíhalo podle tabulky 4. Jestliže myš došla k sýru, bylo to za jeden bod. Byla-li použita nějaká složitější konstrukce, připsal se odpovídající počet bodů. Žáci, kteří ve svém postupu udělali nějakou chybu, a myš nedošla do cíle, tak přesto mohli obdržet body za využití nějaké nové konstrukce. V případě, že úloha byla citelně úplně nedokončená a nejednalo se tak o menší chybu ve smyslu špatné formulace (což je například místo vlevo dát vpravo), žák obdržel nula bodů.

Bodů za odpověď	Typ odpovědi
0	myš podle instrukcí vůbec nedošla k sýru
+1	myš podle instrukcí došla k sýru
+1	byl využit jednoduchý cyklus (opakování jednoho příkazu), např. **`3x vpřed;`**
+2	byl využit složitější cyklus (obsahující více než jeden příkaz nebo vnořený cyklus), např. **`3x (vpřed vpřed vpřed vlevo);`** nebo **`3x(3x(vpřed) vlevo);`**
+3	byla využita procedura, kterou se žáci učili v Lightbotovi
+4	byl využit rozhodovací příkaz KDYŽ a vytvořen univerzální algoritmus

Tabulka 4 – Bodování odpovědí otázky s myškou a sýrem

Přestože žáci měli využívat jen zadané instrukce a domyslet si konstrukce, byli zároveň upozorňování na možnost být kreativní. Někteří žáci proto slovo **`vpřed`** nahradili slovem **`dopředu`** (čímž si tedy práci přidali), případně **`vlevo`** na `L` a **`vpravo`** na `P`. Tyto změny nebyly hodnoceny za extra body, s jednou výjimkou, kterou byl žák sedmé třídy. Ten sice využil tutéž úpravu, avšak nadeklaroval ji jako nové proměnné způsobem: „**`Vp=Vpřed Vl=Vlevo Vř=Vpravo řešení:Vp3x,Vl,Vp3x,Vl,Vp3x`**," za což mu byl přidělen extra bod. Na využití dvou do sebe vnořených cyklů přišli ve vstupním dotazníku pouze dva žáci, žádná složitější konstrukce se však neobjevila.

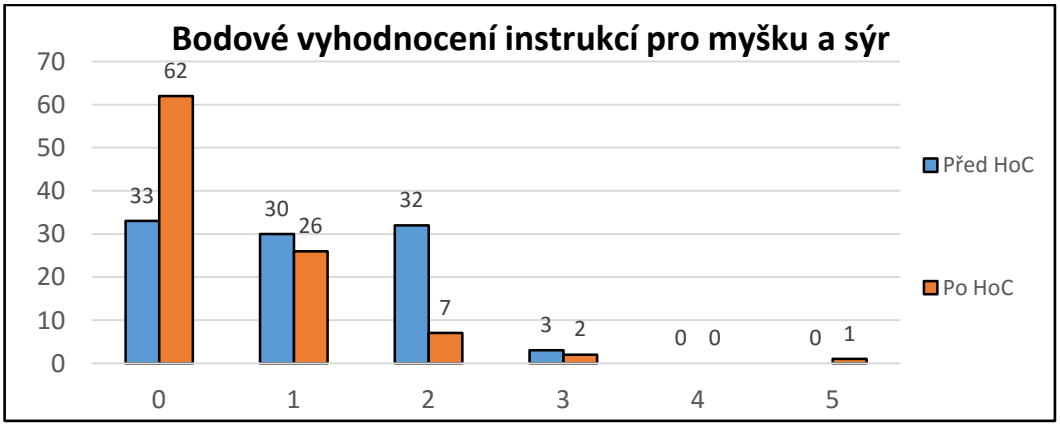
Velkým problémem se ukázala nechuť žáků alespoň trochu přemýšlet nad druhou, delší a složitější variantou myšky. Přestože žáci již byli seznámeni minimálně se dvěma způsoby, jak si práci signifikantně ulehčit (cykly a podmínky, z Lightbota i procedury), žáci se většinou ani o vyplnění tohoto řešení vůbec nesnažili a otázku nechali buď úplně nezodpovězenou, nebo načatou a nedokončenou.

Z vysoce kvalitních úspěšných řešení lze představit například řešení žáka deváté třídy „**`vpřed,vpravo,vpřed,vlevo, 2* vpřed, vlevo, opakovat 4*`**". Téměř dokonalé řešení předvedla žačka deváté třídy, která využila vnořených podmínek KDYŽ

a vytvořila tak plnohodnotný algoritmus aplikovatelný na jakoukoliv úlohu tohoto typu „**opakovat dokud nedosáhne sýru vpřed když cesta v před vpřed jinak když cesta vpravo vpravo jinak vlevo**“. Velice originální bylo řešení žačky sedmé třídy „**VLEVO, PRUŽINA, VPŘED**“. Touto *pružinou* je myšlen speciální krok z Lightbota, který umožňoval skok na vyvýšené políčko. Byla-li by pro myš definovaná odpovídající procedura, řešení by bylo správné. Jelikož žáci byli nabádáni k pružnému myšlení, řešení bylo odměněno alespoň jedním bodem, protože svým způsobem k sýru vedlo. Popsala-li by žačka i co si pod tímto příkazem představit (např.: „pružina = skok na vyvýšené políčko před myškou“), řešení by bylo uznáno jako využití procedury a ohodnoceno připsáním dalších tří bodů.

S odhlédnutím od výše jmenovaných výjimek byl však celkový úspěch žáků ve druhé myší úloze podstatně nižší. Zatímco ve vstupním dotazníku 98 žáků získalo dohromady 103 bodů, ve výstupním dotazníku titíž žáci získali jen 51 bodů. Aplikovat na tento výsledek metodu normalizovaného zisku by tedy bylo nesmyslné.

Pokles v hodnocení může být způsoben signifikantním ztížením druhé úlohy, nedostatkem času pro vstřebání nové látky, únavou žáků po proběhlé dvouhodinovce, anebo jen neochotou snažit se zpracovat složitější úlohu. Výsledek v podstatě potvrzuje známé didaktické pravidlo, konstatující, že po každé expozici novým učivem musí následovat důkladná fixace; na kterou zde bohužel nebyl dostatečný prostor. Na druhou stranu objevení pouhých několika málo excelentních (výše jmenovaných) řešení také může naznačovat, že skutečně ne všichni žáci mají předpoklady pro osvojení algoritmizace a programování a bylo by tedy skutečně vhodnější zařídit spíše volitelný předmět či jiný zájmový útvar, než se snažit vyučovat látku plošně. Na tuto otázku nelze zatím jednoznačně odpovědět, jelikož je množství možných důvodů selhání příliš vysoké a vyžaduje další výzkum v této oblasti.



Graf 10 – Bodové vyhodnocení kvality instrukcí v úloze o myši a sýru

5.2.3 Vyhodnocení otázky na přecházení silnice

Poslední hodnocená otázka se týkala popisu přecházení silnice. Po žácích bylo požadováno, aby popsali, jak přejít ulici někomu, kdo to ještě nikdy nedělal a ani neviděl dělat někoho jiného. Tato otázka zkoumala schopnost zformulovat dostatečně jasné a detailní instrukce na příkladu z reálného života. V tomto případě se jednalo o jednoduchý algoritmus s jednoduchými podmínkami, který si ze své podstaty vyžaduje dostatečně srozumitelnou formulaci řešení a zároveň obsahuje nutnost zakomponování logických podmínek. Žáci se sami museli zamyslet nad možnými alternativami a variantami a definovat algoritmus, zachycující alespoň ty nejzákladnější možné situace odpovídající vybrané variantě (někteří žáci popisovali přechod mimo vyznačený přechod pro chodce, někteří na přechodu pro chodce a někteří na semaforu).

Opět se tato otázka nevyhodnocovala celkově, ale po dílčích částech. Adresuje-li odpověď daný problém, je žákovi připočten odpovídající počet bodů. Detailní zachycení bodování je opět k nalezení v podobě tabulky 5.

Bodů za odpověď	Součást odpovědi
+1	určení výchozí pozice (přijdi k silnici, zastav se před silnicí, apod.)
+1	rozhlédnout se (chybí určení kam, mají se snad podívat na nebe?)
+2	podívat se vlevo a vpravo
+1	podmínka IF (co dělat, když něco nejede, případně jede; zpravidla citelně chybí opačná informace – když nic nejede, přejdi, ale co dělat když jede?)
+2	podmínka IF-ELSE (co dělat když něco ne/jede, jinak dělat co)
+1	cyklus (když nic nejede, jdi, a když něco jede, stůj opakuj vše od začátku)

Tabulka 5 – Bodování odpovědí algoritmu pro přecházení silnice

Naprosto ideální odpověď by tedy mohla vypadat následovně: *„Těsně před přecházením silnice se zastav. Podívej se vpravo a vlevo. Když ani z jedné strany nic nejede, přejdi, jinak nepřecházej a vše zopakuj od začátku.“* Takováto odpověď definuje výchozí pozici, kam se má přecházející podívat, co dělat když a co dělat jinak a v případě jedoucího auta vše zopakovat. V případě zakomponování více IF nebo IF-ELSE podmínek, jako například *„když nic nejede - přejdi, když něco jede, ale je to daleko přejdi,“* se při řešení body nenásobily. V tomto ukázkovém příkladu se jednalo o dvě jednoduché podmínky IF (protože v obou případech chybí instrukce, co se má dělat, jede-li něco) a byl by tedy ohodnocen +1 bodem. Maximální možný počet dosažitelných bodů je tedy 6.

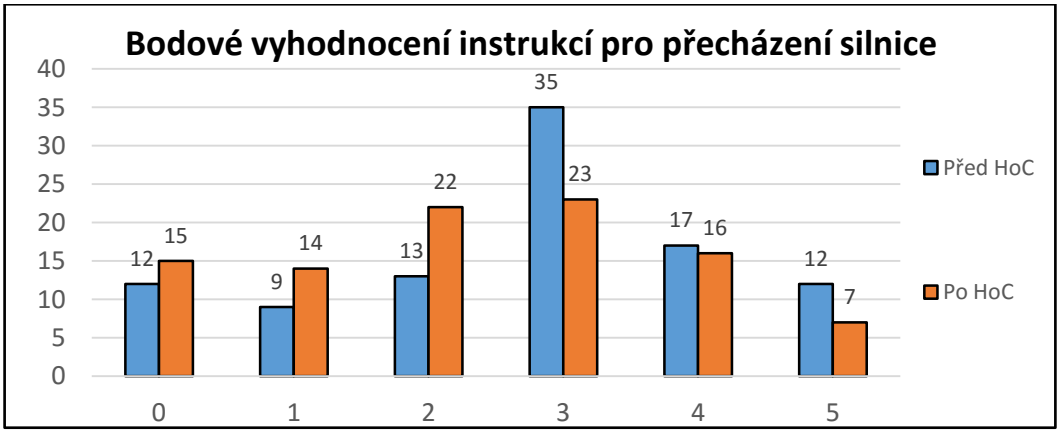
Jelikož tato otázka zůstala do výstupního dotazníku beze změny a žáci byli citelně vyčerpaní, zadání bylo verbálně rozšířeno o instrukci, že jestliže jsou se svojí odpovědí spokojení a nechtějí ji nijak upravit a vylepšit, potom mají nechat pole prázdné. V takovém případě se při vyhodnocování výstupního dotazníku přenesl počet bodů z dotazníku vstupního.

Ani jeden žák nedosáhl maximálního počtu šesti bodů. U nejúspěšnějších odpovědí zpravidla chybělo buď určení výchozí pozice, anebo cyklus v případě, že něco jede. Algoritmicky suverénně nejlepší odpověď vytvořil žák ze sedmé třídy (ohodnoceno pěti body namísto maximálními šesti, protože chybí určení výchozí pozice):

„1) **Podívej se v pravo i vlevo;když nic nepojede;**
rozběhni se přímo na druhou stranu silnice a potom zastav když pojede;
stůj na místě a opakuj příkaz 1“

Stručná odpověď *„jdi na přechod“* by krásně ilustrovala client-programmer přístup, tedy využití již existujícího a ověřeného řešení. Procedura pro přecházení na přechodu pro chodce však nebyla definovaná a žáci by tak v tomto případě stejně museli doplnit, co se na tom přechodu přesně má dělat. Na základě takto elementární a nedoplněné instrukce by přejití opět nebylo bezpečné, takže se v podstatě jen jednalo o definování výchozí pozice. Odpovědi tohoto typu tedy byly hodnoceny za jeden bod.

Snížení množství získaných bodů a tedy zhoršení výsledků výstupního dotazníku oproti vstupnímu (stejně jako v úloze o myši a sýru) je pravděpodobně z důvodu popsaných v předchozí podkapitole. V obou případech však velkou roli hrála vyčerpanost žáků, kteří mnohdy jen načali řešení, které poté nedotáhli do konce.



Graf 11 – Bodové vyhodnocení kvality algoritmu pro přecházení silnice

5.3 Vyhodnocení dílčích hypotéz

Úvodní hypotézu práce tvrdící, že průchodem alespoň jedné Hour of Code se u žáků zvýší schopnost algoritmizace a která byla ověřovaná na výše popsaných příkladech s myší a přecházením silnice, nebylo možné ani potvrdit, ani vyvrátit. Příliš velké množství neovlivnitelných proměnných zkreslilo výsledky výstupního dotazníku žáků. Žáci tak dosahovali horších výsledků než před absolvováním Hour of Code. Nejpravděpodobnějšími důvody tohoto selhání bylo již opakovaně zmiňované vyčerpání žáků a nedostatek času a prostoru pro důkladnou fixaci nové látky.

V optimálním případě by byly výstupní dotazníky nasazeny až na začátku následující dvouhodinovky (nebo ještě lépe až po další dvouhodinovce programování popisované v kapitole 5.4), což však v případě tohoto výzkumu z logistických důvodů nešlo. Výzkumy v hodinách informatiky jsou ztíženy velice častým a běžným rozdělením tříd do více skupin a jejich střídáním každý druhý týden. Vyzkoušet na všech skupinách obě dvouhodinovky by zabralo celý měsíc plus začátky hodin nutné pro vyplnění výstupních dotazníků následujících čtrnáct dní. Narušit až takto výrazně tematický plán školy nebylo možné a vše tedy muselo být provedeno co nejkompaktněji, což se silně negativně projevilo na bodovaných úlohách ve výstupním dotazníku. Původním záměrem této práce bylo zjišťovat, zda došlo ke zlepšení již po absolvování jedné Hour of Code, bohužel ale na základě výše popsaných komplikací muselo být od tohoto záměru úplně ustoupeno.

Jako dílčí hypotézy byly určeny:

- H1: Míra významu, kterou žáci přikládají informatice jakožto oboru, přímo souvisí s jejich úspěšností v úloze na definování programování.
- H2: Čím vyšší je šíře informaticky orientovaných zájmů žáků, tím lepší je jejich schopnost algoritmizace.
- H3: Věk žáků má přímou pozitivní souvislost s kvalitou jejich odpovědí.

Všechny tři hypotézy jsou založeny na proměnných ordinálního typu (s výjimkou věku žáků, což je proměnná kardinální, avšak statistickým testem pro ověření kombinace kardinální a ordinální proměnné je dle Skutila (2011, s. 179) opět Kendallův korelační koeficient – s proměnnou *věk* je tedy nadále nakládáno jako s proměnnou ordinální). Míra významu, šíře zájmů a věk žáků jsou vždy porovnávány s vyhodnocenými výsledky všech tří bodovaných otázek ve vstupním dotazníku.

Hodnoty ve výstupních dotaznících nebyly použity, protože jsou z výše zmiňovaných důvodů zkreslené a vypočtené výsledky by neodpovídaly realitě.

Kendallův korelační koeficient je neparametrický statistický test, který má dvě podoby známé jako Tau α a Tau β. Výpočet obou verzí je založen na určení počtu takzvaných konkordantních a diskordantních párů. Verze Tau β se volí v případě většího počtu vázaných párů (*tied pairs*) a její výpočet je podstatně složitější oproti verzi Tau α. Ani jedna verze Kendallova korelačního koeficientu není součástí standardních statistických funkcí programu Microsoft Excel, a proto byl pro výpočet využit online kalkulátor Wessa.net (Wessa 2012) a pro ověření výsledků těchto výpočtů další online kalkulátor Vhex.net (Calculator.vhex.net 2016). Výsledky se ve všech případech shodovaly a jsou proto považovány za přesné.

Výsledkem výpočtu Kendallova korelačního koeficientu je jedno číslo v rozmezí od -1 do +1, kde hodnoty kolem nuly značí naprostou nezávislost porovnávaných proměnných a hodnoty blížící se 1 znamenají naprostou (funkční) závislost. Skutil (2011, s. 177) interpretuje dílčí kategorie tohoto koeficientu dle Chráska (2007) následovně:

> *„0 = naprostá nezávislost; 0,0 až 0,2 = velmi slabá závislost; 0,2 až 0,4 = nízká závislost; 0,4 až 0,7 = střední závislost; 0,7 až 0,9 = vysoká závislost; 0,9 až 1,0 = velmi vysoká závislost; 1 = naprostá (funkční) závislost.“*

První z dílčích hypotéz zjišťovala souvislosti mezi mírou významu, kterou žáci přikládají informatice jakožto oboru, a jejich schopností definovat, co to je programování. Tato hypotéza byla založena na předpokladu, že čím větší význam žáci tomuto oboru přikládají, tím spíše se sami o informatiku zajímají a vyhledávají si dodatečné informace. Korelace se schopností definovat programování byla vypočtena na hodnotu 0,03004. Jelikož se jedná o hodnotu aproximující nulu, lze tvrdit, že mezi těmito dvěma proměnnými není žádná souvislost. Přikládání vysokého významu informatice jako oboru tedy automaticky nevede k rozšiřování znalostí respondentů v oblasti programování.

Zajímavější výsledky se objevily v případě druhé dílčí hypotézy. V té byla šíře informaticky orientovaných zájmů žáka (vypočtených jako součet žákem vyjmenovaných či vybraných zájmů v odpovídající otázce ve vstupním dotazníku) porovnávána se všemi třemi bodovanými úlohami ze vstupního dotazníku. Korelační

koeficient při porovnávání šíře zájmu s kvalitou definice programování vyšel 0,11583, při porovnávání s vytvořením jednoduché posloupnosti instrukcí pro myš 0,15741 a při vytváření jednoduchého algoritmu pro přecházení silnice 0,10072. Všechny tyto tři případy sice stále spadají do kategorie definované jako *„velmi slabá závislost,“* i to však naznačuje, že čím více se žáci informatikou skutečně zabývají, tím spíše jsou schopni zpracovávat algoritmické úlohy. Jedná se samozřejmě jen o vstupní předpoklady měřené a hodnocené před expozicí samotnému programování, i tak však výsledek naznačuje, že zvyšující se kvantita žákových schopností práce na počítači mu umožňuje lépe uvažovat v termínech počítače.

Třetí a poslední hypotéza se zabývala věkem žáků, přičemž vstupním předpokladem bylo, že čím jsou žáci starší, tím budou spíše schopni využívat abstraktního myšlení a jejich odpovědi budou kvalitnější. Z hlediska věku žáků vyšla korelace se schopností definovat programování -0,03737, což je opět hodnota aproximující nulu. Nelze tedy tvrdit, že starší žáci lépe chápou, co to programování je. Programování je činnost, kterou se žáci sami od sebe na základní škole zabývat nezačnou a znalost definice (potažmo obsahu) programování tedy nezávisí na věku. Lze tedy vyvodit, že starší žáci nemají o nic lepší představu o daném pojmu než žáci mladší.

Korelační koeficient při porovnávání věku s instrukcemi pro myš vyšel pouhých 0,0014 a při porovnávání s algoritmem pro přecházení silnice -0,0652. Stejně jako v předchozím případě se jedná o hodnoty blížící se nule (v případě instrukcí pro myš se dokonce jedná již o naprostou nezávislost). Zdá se tedy, že tvrzení o vhodnosti výuky programování pro jakoukoliv věkovou kategorii je skutečně podložené. Žáci devátých tříd nejsou schopni řešit zadané úlohy o nic lépe, než žáci tříd šestých.

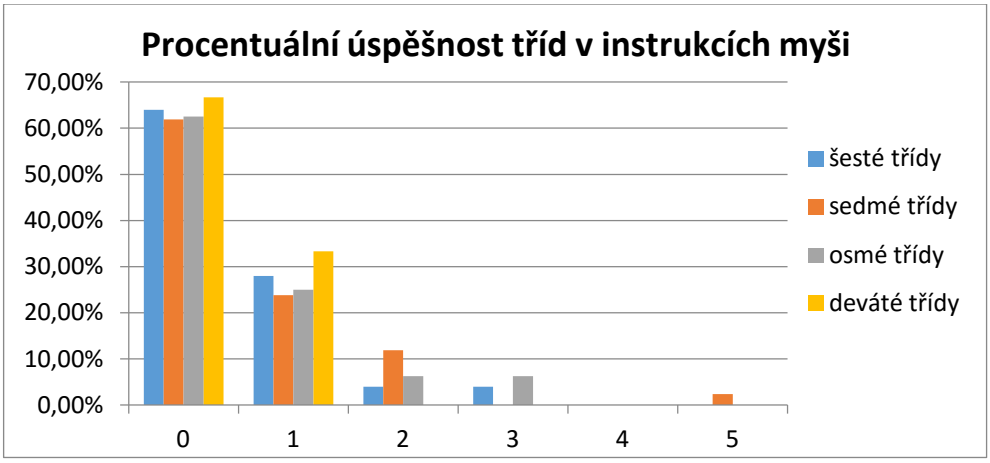
Tři výše popisované dílčí hypotézy byly stanoveny úmyslně konfrontativně vůči propagandisticky znějícím výrokům, kterými se většina projektů z kapitoly 3 prezentuje. Tyto projekty v podstatě tvrdí, že jsou naprosto pro každého bez jakéhokoliv omezení, ale svá tvrzení nedokládají konkrétními objektivními studiemi. Autorova (přestože zatím stále ještě omezená) zkušenost s výukou programování na ZŠ však doposud na základě subjektivního pocitu naznačovala, že k pochopení komplexních konstrukcí typu podmínek a procedur je nutné abstraktní myšlení, které se podle J. Piageta rozvíjí až kolem dvanáctého roku (tedy kolem sedmé třídy).

Ani jedna ze tří výše jmenovaných hypotéz nemůže být potvrzena, což je ale samozřejmě také naprosto plnohodnotný výsledek. Přístup žáků k informatice, šíře jejich zájmu a dokonce ani věk na základě výsledků této studie skutečně nijak

neovlivňují vstupní předpoklady pro rozvoj algoritmického myšlení. Teoreticky je tedy možné nasadit výuku programování pomocí Hour of Code a Scratch do jakéhokoliv ročníku na 2. stupni ZŠ, protože podle zjištění dosud provedených šetření mají všechny třídy shodné vstupní předpoklady.

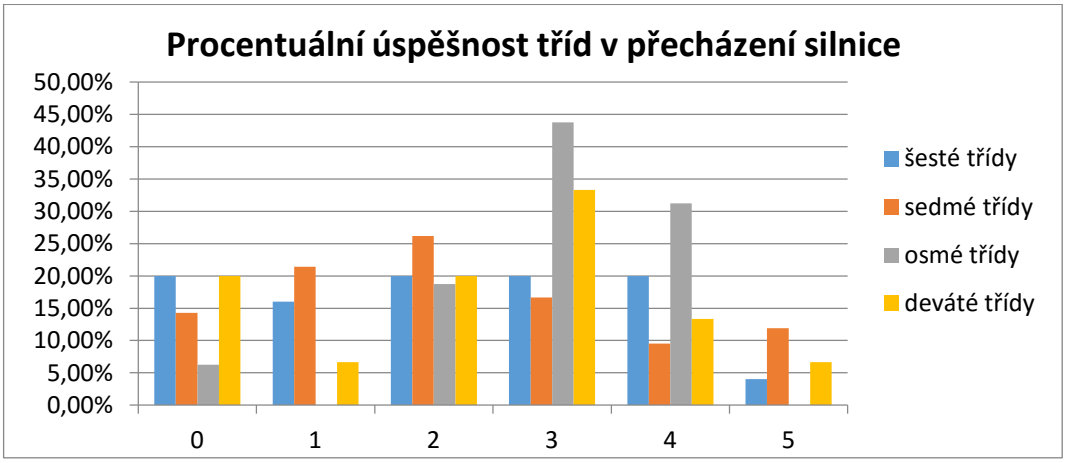
Testování projektu Hour of Code si dalo jako poslední cíl hrubý odhad nejvhodnější věkové kategorie pro začátek výuky programování. Abstraktní myšlení, které je nutné pro tvorbu algoritmů, se dle J. Piageta u žáků rozvíjí až kolem 12. roku, tedy kolem 6. a 7. třídy, ale samotní tvůrci projektu Hour of Code tvrdí, že je jejich projekt *„Určen pro věk od 4 do 104.“* (Code.org 2016a). Jak již bylo v úvodu řečeno, na omezeném vzorku a bez možnosti longitudiální studie není možné nejvhodnější věk stanovit přesně a jednoznačně, jedná se však o odhad, který je zamýšlen jako výchozí bod pro případné další studie. V případě vytváření tohoto odhadu je nutné pracovat s výsledky žáků po absolvování Hour of Code, a mít tedy na paměti nedostatky těchto výsledků.

Pro určení nejúspěšnějších ročníků bylo využito procentuální grafické zobrazení. Výsledky žáků byly nejprve seskupeny podle ročníků, a protože sedmých tříd bylo výrazně nejvíce, výsledky byly dále převedeny na procenta. Každá třída tedy měla dohromady 100%.



Graf 12 – Procentuální úspěšnost tříd v úloze s instrukcemi pro myš

Zatímco v případě instrukcí pro myš byly všechny třídy takřka shodně (ne)úspěšné (viz graf 12), výsledky při tvorbě algoritmu pro přecházení silnice jsou zajímavější. Žáci osmých tříd dopadli výrazně lépe než ostatní třídy, s minimem „špatných“ odpovědí (odpovědí hodnocených 0, 1 nebo 2 body bylo mezi žáky osmých tříd pouhých 25 % oproti ostatním třídám, které se pohybovaly mezi 45 a 60 %).



Graf 13 – Procentuální úspěšnost tříd v algoritmu pro přecházení silnice

5.4 Druhá výzkumná dvouhodinovka a Scratch

Scratch byl autorem této práce vybrán jakožto ideální programovací prostředí, technicky vzato přímo navazující na značně restriktivní Hour of Code a umožňující žákům demonstrovat doslova nekonečné možnosti tvorby libovolného obsahu, ať už jednoduchých skriptovaných scének nebo komplexnějších her a hříček. Z důvodu červnového termínu výzkumu byla tato dvouhodinovka realizována jen v celkem třech skupinách a to ve třídě šesté, osmé a deváté, čímž se zároveň efektivně pokrylo téměř celé věkové spektrum žáků. Toto ale bohužel také znamenalo nutnost absence pilotních skupin a s tím spojenou ztrátu benefitu z předběžného otestování efektivity konkrétních postupů v hodině. Mezi jednotlivými dvouhodinovkami tak docházelo k drobným alteracím v plánu i obsahu hodiny za účelem zvýšení efektivity výuky a poslední dvouhodinovka se od té první tedy značně lišila.

Jelikož Hour of Code žáky seznámil se základními principy práce ve vizuálním programovacím jazyce založeném na programovacím prostředí Blockly, samotné ovládání Scratche pro ně nepředstavovalo větší překážku. Ve velké šíři možností Scratche se žáci sice ztráceli, kdyby však na tuto výuku mohlo být vyčleněno více než dvě dvouhodinovky, látka by se dala rozdělit do menších bloků, které by pro žáky byly stravitelnější. I v této velice nahuštěné verzi by však výuka programování na ZŠ byla reálně proveditelná a pro žáky rozhodně přínosná.

V této části práce budou podrobněji rozebrány dvě předpřipravené úlohy ve Scratchi a reakce tříd. Prvním ze dvou úkolů stanovených pro tuto dvouhodinovku bylo společné vytvoření jednoduché naskriptované scénky, čímž se měli žáci seznámit s novým, nijak neomezeným programovacím prostředím. Zde obecně dělalo žákům nejvíce problém orientovat se v obsahu jednotlivých kategorií na záložce Scénáře a dále vybrat správný příkaz z více podobných (například *opakuj XY krát* vs. *opakuj dokola*).

Úvodní úloha spočívala ve vytvoření scénky, ve které kocour Scratch vypadá, jako když sedí v autě, které mu po kliknutí na zelenou vlaječku ujede. Ve chvíli kdy se auto dotkne okraje obrazovky, Scratch se má za ním rozeběhnout a volat „Počkej!“ Zhruba v polovině obrazovky má kocour své pronásledování auta vzdát. Po kliknutí na mezerník se má celá scénka zresetovat a sprity se mají vrátit na výchozí pozici. Řešení této úvodní úlohy krok po kroku ukazoval vyučující na projektoru a žáci měli za úkol toto řešení replikovat na svých počítačích. Tímto způsobem se žáci naučili orientovat se ve vlastním prostředí, přidávat a ubírat sprity, měnit scénu, manuálně vytvořit restart

(tedy jinými slovy vrátit sprity na výchozí pozici), vytvářet bloky příkazů pro jednotlivé sprity, vybrat odpovídající začátek daného bloku příkazů a dokonce i zasílat zprávy mezi sprity. Kromě toho se žáci také naučili své výtvory ukládat do počítače a opět načítat. Časově byla tato část plánována maximálně na 25 minut, přičemž zbytek dvouhodinovky byl vyhrazen pro druhou úlohu. Kvůli její náročnosti byli žáci vždy rozdělení do dvojic, a to i v případě, že bylo počítačů dostatek. V této situaci si mohli žáci ve dvojici zkoušet zároveň v jednu chvíli více nápadů na řešení daného dílčího problému a postupovat tak podstatně rychleji, čehož také některé skupiny využily.

Žáci na základní škole jsou velice hraví a počítačové hry jsou pro ně přitažlivým tématem. Jako druhá úloha této hodiny byla tedy vybrána tvorba „jednoduché“ hry, která byla ve skutečnosti velice komplexní. V této hře je kocour Scratch ovládán hráčem šipkami na klávesnici a snaží se chytit náhodně utíkající myš. Při tom ho po celou dobu pronásleduje počítačem ovládaný pes. Jestliže hráč chytí myš, jeho skóre se zvýší o jeden bod, ale jestliže pes chytí hráče, skóre je o bod sníženo. Předem vytvořená celá hra byla žákům ještě před vlastním začátkem jejich tvorby ukázána (se skrytým řešením), aby na vlastní oči viděli, co přesně se po nich chce a jak by zhruba výsledek měl vypadat. V takovéto úloze a obecně v celé problematice algoritmizace je pro žáky největší problém přesná formulace na sebe navazujících kroků, tedy co přesně se má kdy stát a jak toho docílit. Formulaci těchto dílčích kroků zde opět obstarával vyučující a žáci měli za úkol vymyslet jejich přesnou realizaci přímo ve Scratchi.

První ze tří skupin, která si programování ve Scratchi vyzkoušela, byla šestá třída, která byla minule i v tomto případě spojená a ve třídě bylo celkem 22 žáků. Přestože tito žáci zvládli Hour of Code relativně bez problémů, ve Scratchi narazili na mnoho problémů. Plánované tempo hodiny zde bylo naprosto nerealizovatelné a pracovní tempo jednotlivých žáků extrémně odlišné. Za celou dvouhodinovku se stihl jen první ukázkový úkol s naskriptováním jednoduché scénky. Žáci si při komplikacích nebyli schopni ani ochotni navzájem pomáhat, ti nejrychlejší se nudili a ti nejpomalejší byli celkem zoufalí, protože vůbec nevěděli, co mají dělat. Protože se druhý úkol vůbec nestihl, vyučující alespoň v posledních pěti minutách hodiny třídě ukázal své předpřipravené řešení, na kterém žáci mohli vidět, že skutečně lze vytvořit jednoduchou hru za pomoci toho, co se již naučili. Na základě předchozí výzkumné dvouhodinovky a této dvouhodinovky Scratche je možné usoudit, že šestá třída je na takto komplexní téma ještě příliš mladá a je tedy vhodné problematiku algoritmizace zařadit až do vyšších ročníků, nejoptimálněji poprvé do první poloviny sedmého ročníku.

Případné nasazení do nižších tříd by nejspíše bylo realizovatelné, avšak vyžadovalo by podstatně pomalejší pracovní tempo a mnohem důkladnější vysvětlování a zejména fixaci a zkoušení jednotlivých možností Scratche.

Jako druzí měli dvouhodinovku Scratche žáci deváté třídy. Opět se jednalo o spojenou skupinu, ve které bylo tentokrát 19 žáků. Zde již nebyl problém stihnout první společnou úlohu přibližně v plánovaném čase (celkem 28 minut), ve druhé úloze se ale naplno projevila absence pilotní skupiny. V plánu totiž bylo, že vyučující bude zadávat dílčí instrukce všem najednou slovně, načež počká než nějaká skupina či více skupin vymyslí jejich realizaci ve Scratchi, tu demonstrují ostatním žákům na projektoru a teprve poté se přejde k další dílčí instrukci. Toto řešení nebylo z nejlepších, protože replikace řešení rychlejších žáků žáky pomalejšími byla příliš zdlouhavá a rychlejší žáci tak rychle ztráceli pozornost a nudili se. I přesto se téměř všichni dostali alespoň do poloviny celého úkolu a tři skupiny dokonce splnili úkol naprosto celý, a to bez dílčích instrukcí jen na základě vstupní formulace problému a ukázky požadovaného výsledku, kterou viděli na začátku této části dvouhodinovky.

Těmto žákům bylo poté umožněno buď pomáhat slabším skupinám, sami si dále zkoušet ve Scratchi vytvořit cokoliv je napadne (přičemž se mohli zeptat na jakýkoliv problém, na který by při tomto svém zkoušení narazili), či se v tichosti věnovat libovolné vlastní činnosti, ať už do jiných předmětů či něčeho na internetu. První ani třetí možnosti ani jedna ze skupin nevyužila, všichni tito žáci dále na základě vlastního rozhodnutí pokračovali ve zkoumání Scratche. Jedna z těchto skupin zkoumala možnosti přidání zvukových efektů do Scratche, druhá skupina zkoušela vytvoření vlastního skriptovaného scénáře s lodí na moři a třetí skupina vytvářela variaci na zadanou hru. Takto vytvořili hru, ve které ovládali žraloka pod mořem, který chytal kolem plovoucí rybičky.

Ani v poslední skupině, kterou byla osmá třída o patnácti žácích, nebyla úvodní úloha vůbec problém, a tentokrát se stihla přesně za naplánovaných 25 minut. Pro druhou úlohu se dále počítalo s extrémními rozdíly mezi tempy žáků a žáci dostali předem vytvořený seznam s přesnou formulací jednotlivých dílčích kroků, obsahující také rady s řešením nejčastějších problémů. Celý tento seznam s detailním zadáním druhé úlohy je obsahem přílohy K.

S pomocí takovéhoto zadání bylo opět možné, aby žáci pracovali vlastním tempem, stejně jako v rámci Hour of Code. Veškeré vyrušování a nepozornost spojená s nudou byly takto efektivně eliminovány. Žáci ve dvojicích pracovali dle svých možností

a všichni neustále přesně věděli, co mají dělat. Pět z celkových sedmi skupin dokončilo celou úlohu včas a ve zbývajícím čase mělo stejné možnosti jako devátá třída, tedy pomoci pomalejším, pokračovat dle vlastního zájmu ve Scratchi, anebo si v tichosti dělat, co chtějí. Poslední jmenované možnosti využily jen dvě z těchto pěti skupin, přičemž jedna z nich skončila na Facebooku a jedna u nějakých online her. Zbývající tři skupiny opět pokračovaly ve Scratchi. Jedna z nich opět vytvářela vlastní jednoduchý scénář, jedna variaci na hru (tentokrát s motýlky na louce) a jedna ze skupin se rozhodla vylepšit svou vytvořenou hru. Tato skupina provedla několik drobných úprav v instrukcích a vylepšila poněkud jednotvárný pohyb myši a psa ve hře. Poté tito žáci do hry přidali i zvukové efekty.

Shrnout na závěr tuto část výzkumu je složité, protože vlastní průběh hodiny se bez možnosti využití pilotních skupin měnil a bez konzistentního obsahu jednotlivých dvouhodinovek není možné výsledky generalizovat. Každopádně je na místě doporučení zařadit výuku algoritmizace a programování nejdříve do sedmého ročníku a upozornit na nutnost počítat s extrémními rozdíly v tempu práce žáků, což se dá vyřešit předem vytvořeným přesným zadáním, podle kterého mohou žáci postupovat víceméně sami. V každém případě lze ale konstatovat, že zájem o Scratch byl veliký, protože z osmi skupin, které druhou úlohu vyřešily s předstihem a mohly si samy vybrat, co budou dělat, celých šest skupin z vlastní vůle pokračovalo ve zkoumání možností Scratche a vytváření vlastních skriptů i her. Při dalších návštěvách na této základní škole jsem navíc byl vždy doslova obsypán žáky, kteří se mě ptali, kdy bude další hodina *„s tím oranžovým kocourem.“* Naplnění kapacit případného kroužku programování by tedy rozhodně nebyl nejmenší problém.

5.5 Návrh třetí navazující dvouhodinovky Scratche

V případě zařazení tematiky programování do standardního kurikula předmětu informatiky by bylo vhodné vyčlenit na toto téma podstatně více času než dvě dvouhodinovky. Každopádně nejzákladnější myšlenky v extrémně zhuštěné podobě jdou předvést v rámci dvou dvouhodinovek, na které by bylo optimální navázat ještě minimálně jednou další. V té by žáci mohli dostat problém zadán jen obecně formulovaný (například *„Vytvořte hru, ve které…“*) a vyzkoušeli by si tak i poslední a nejnáročnější fázi algoritmizace, tedy vymýšlení jednotlivých dílčích kroků komplexního řešení. Samozřejmě i v rámci tří dvouhodinovek by bylo téma algoritmizace a programování stále extrémně zhuštěné, poskytlo by ale již dostatečný vhled do dané problematiky a možný odrazový můstek pro případné zájemce o hlubší studium v rámci volitelných hodin či nějakého zájmového útvaru.

Naprosto nejideálnějším řešením by byla možnost mít kromě standardní informatiky ještě samostatný předmět programování. Ten by byl žákům nabídnut buď jako povinně volitelný předmět, nebo zájmový kroužek, přičemž tři dvouhodinovky, které by byly součástí standardní výuky informatiky, by žáci absolvovali například na konci školního roku a jejich volba programování by tak byla založená na informovaném rozhodnutí a skutečném zájmu ze strany žáka.

Na základě proběhlého výzkumu lze tvrdit, že zájem by skutečně byl, avšak dotazníky byly rozdány již na konci první dvouhodinovky. Hour of Code je podstatně restriktivnější, než naprosto volný Scratch, takže žáci, které bavilo vymýšlet relativně jednoduché řešení konkrétního problému, by mohli být odrazeni nutností komplexnějšího uvažování a vytváření rozsáhlých vlastních řešení a postupů. Žáci totiž mohli výuku považovat spíše za naprosté hraní namísto řešení problémů. Nejpřesnější zjištění míry zájmu by bylo až po proběhlé dvouhodinovce Scratche (či dokonce dvou dvuhodinovkách), kdy by již žáci podstatu celé problematiky chápali více a přesněji, a počáteční nadšení z něčeho nového by poněkud opadlo. V této oblasti tedy zůstává prostor pro další výzkumy.

6 Kvalitativní část šetření mezi učiteli

Druhou částí praktického výzkumu bylo kvalitativní šetření založené na metodě polostrukturovaného rozhovoru s učiteli informatiky. Respondenty jsou jeden učitel a jedna učitelka, přičemž oba učí třídy, ve kterých probíhala kvantitativní část výzkumu. S oběma respondenty byly vedeny celkem dva výzkumné rozhovory. První z nich proběhl předtím, než učitelé viděli ukázku výuky programování za pomoci Hour of Code a Scratche, a druhý rozhovor proběhl v závěru celého výzkumu po zhlédnutí ukázkových hodin.

Všechny předem připravené otázky (viz tabulka 6) byly oběma respondentům vždy položeny a v případě nutnosti bylo požádáno o další klarifikaci konkrétního výroku respondenta. Oba respondenti byli nabádáni, aby se v rámci svých odpovědí nestrachovali zmínit i jakékoliv další asociace, které je napadnou buď v reakci na konkrétní otázku či jako následek vlastní odpovědi.

Otázky/body ke vstupním rozhovorům	Otázky/body k výstupním rozhovorům
● Celková délka učitelské praxe	● Co si o celém výstupu a jeho obsahu myslí?
● Má vyučující jako jednu z aprobací informatiku?	● Změnil/a by v hodině něco? (přidat, ubrat, apod.)
● Jaká je druhá aprobace, případně obě?	● Jestli v tom vidí nějaký přínos žákům
● Délka výuky informatiky v letech	● Myslí si, že byl o hodinách výraznější genderový rozdíl?
● Týdenní hodinová dotace vyučované informatiky	● Myslí si, že je obecně v informatice nějaký genderový rozdíl?
● Obsah hodin obecně (co za témata probírají)	● Bylo to nějak přínosné i pro něj/pro ni?
● Jestli vyučující umí programovat	● Chtěl/a by si toto v budoucnu i sám/sama vyzkoušet?
● Když ano, v jakých jazycích	● Bude takovéto hodiny standardně zařazovat do své výuky?
● Má o problematiku zájem ve svém volném čase?	● Změnil se jeho/její názor na výuku algoritmizace na ZŠ?
● Je programování součást výuky?	● Myslí si, že jsou Hour of Code a Scratch vhodné i na střední?
● Jestliže ne, tak proč není?	● Má nějaké libovolné poznámky k tomuto tématu?
● Kdyby učil/a na střední a ne na ZŠ, vyučoval/a by ho?	● Jestli chce zaslat podklady k této výuce (přípravy a odkazy)
● Měla by se vyučovat algoritmizace na školách?	
● Bylo by to pro žáky přínosné nebo zbytečná zátěž navíc?	
● Co si obecně myslí o žácích a jejich schopnostech?	
● Zajímají se žáci o technologie a jejich využití?	
● Učí se sami žáci něco informatického? (web, programování, grafika,…)	
● Setkal/a se už někdy s Hour of Code nebo Scratchem? (když ne, kratičké vysvětlení a popis principu práce)	
● Co si o těchto projektech myslí?	
● Odhad, jak žáci přijmou mé ukázkové hodiny	
● Bude-li to bavit, chtěl/a by s tím v budoucnu pokračovat?	

Tabulka 6 – Připravené podklady k polostrukturovaným rozhovorům

Oba vstupní a oba výstupní rozhovory byly vždy nahrávány ve formě jednoduchého audio záznamu, který poté prošel procesem kompletní transkripce. V rámci dodržování etických pravidel výzkumu byli oba učitelé nejprve požádáni o možnost zaznamenávat své rozhovory, s čímž sice souhlasili, ale nepřáli si zveřejnění rozhovorů ani v podobě nahrávek, ani jako plné transkripce. Z tohoto důvodu tedy třináctistránková transkripce rozhovorů není součástí příloh a citovány budou v případě nutnosti jen konkrétní pasáže (přičemž tyto citace budou obdobně jako v kvantitativní části výzkumu přepsány doslovně včetně všech chyb). Oba učitelé byli předem seznámeni s přibližným obsahem rozhovoru a tématem magisterské práce. Bylo jim vysvětleno, co se s rozhovorem bude dít a jak bude do práce zakomponován, dále byli ujištěni o samozřejmosti zachování anonymity. Na základě všech těchto informací byl v rámci domlouvání konkrétního data a času konání vlastního rozhovoru získán informovaný souhlas.

Cílem rozhovorů bylo zjistit názory učitelů na problematiku programování obecně a identifikovat, jaké faktory ze subjektivního hlediska učitelů ovlivňují možnost či nemožnost nasazení programování do běžné výuky. V následujících dvou podkapitolách se nachází detailní analýza a interpretace rozhovorů obou vyučujících. Popis skutečného obsahu rozhovoru (tedy toho, co učitelé přímo nahlas řekli) je však vždy striktně oddělen od samotné interpretace. Následující podkapitoly vždy začínají soupisem faktů, týkajících se daného učitele či učitelky, poté popisem pozorovaných extralingvistických jevů, načež následuje skutečný obsah rozhovoru a nakonec interpretace.

Při zpracovávání kvalitativního výzkumu bylo postupováno dle Mgr. Kateřiny Juklové, Ph.D. (in Skutil 2011). Po úplně transkripci byl rozhovor rozdělen na segmenty (části textu obsahující určitou informaci). Segmenty byly poté v procesu kódování pojmenovány klíčovými slovy. Takto vzniklé kódy byly kategorizovány, přičemž výsledný souhrn je k nalezení v tabulce 7 v kapitole 6.3.

Následující text může vyznívat jako silná kritika, není tak však zamýšlen. Jedná se o cílenou snahu nalézt příčiny absence výuky programování na vybrané ZŠ a analyzovat konkrétní postoje učitelů, kteří souhlasili s poskytnutím pomoci s výzkumem. Oba učitelé dělají mnohé správně a tato práce se jen zaměřuje na specifické detaily, které teoreticky mohou přímo souviset s možností začlenění tematiky programování do běžných hodin informatiky. Analýza a interpretace se zabývá problematickou oblastí a výsledky nezní lichotivě, nejde však o snahu tyto učitele nijak urážet ani znemožňovat.

6.1 Rozhovory s učitelem

Učitel informatiky není aprobovaným informatikem. Ačkoliv určitá snaha doplnit si aprobaci byla, nepovedlo se ji dovést do zdárného konce. Učitel je aprobovaný tělocvikář. Délka jeho učitelské praxe byla v době pořizování rozhovoru celkem sedm let, z toho šest let vyučoval informatiku. Celkem dvacet let před vstupem do školství se tento učitel živil jako voják se specializací v oblasti radistiky. Programovat učitel neumí a učit se nechce. Respondentův projev byl stručný a volné asociace minimální.

Pozorované extralingvistické jevy

Při vstupním rozhovoru nebyl respondent unavený, ale ani nijak energický. Jeho projev byl celkem plynulý, ale silně monotónní. Respondent se dále zdál být silně nervózní a měl obavu z nahrávání, a to i přes získání informovaného souhlasu při předchozí schůzce, na které byly domluveny podrobnosti ohledně rozhovoru. Po opětovném ujištění, že nahrávka nebude nikde zveřejněna, bude přepsána, smazána a přepis bude sloužit pouze pro účely této diplomové práce, se však poněkud uklidnil.

Jako podklad k rozhovoru byl využit textový soubor v mobilu s připravenými otázkami nebo body (viz tabulka 6 výše). Respondent si na vlastní žádost vše nejprve přečetl, a když viděl, že otázky skutečně nejsou nic záludného ani nepříjemného, uklidnil se a mohlo se přejít k samotnému rozhovoru. V podstatě stejně probíhal také rozhovor výstupní.

Faktický obsah vstupního rozhovoru

V průběhu rozhovoru učitel opakovaně degradoval význam nových moderních technologií, jako například *„To je ta mobilní doba... mobil všechno – papírky všude polepený! To je to... No kluci, tady musí bejt nástěnka, to je nejlepší věc! Tam to jenom napícháš a je to.“* Učitel dále sám zmiňoval, že se zajímal o *„interaktivní výuku,“* kterou definoval jako práci s multimédii (konkrétně videem). Přes určité faktické nedostatky však učitel vykazoval značnou snahu praktické orientace výuky, jeho cílem je naučit žáky chápat základní principy práce s počítačem (*„…jak si ukládají soubor, kam si ukládají soubor, co to je internet,“* později dále specifikoval *„snažím se aby se naučili nějakou tu instalaci, nainstalovat si systém, nebo zálohu systému udělat...“*), což je samozřejmě dobře. Tyto principy jsou však většinou opět směrovány na výuku konkrétních postupů práce s konkrétním aplikačním vybavením.

Jako dva největší problémy učitel jmenoval nedostatečnou časovou dotaci hodin informatiky: *„když máte jednou za čtrnáct dní vlastně dvě hodiny, tak s nima neprobereš nic“* a nezájem stran většiny žáků *„některé nebaví vůbec nic.“* Problémy s časem a to jak z hlediska nedostatku času pro vyhledávání nových věcí stran učitele, tak z hlediska nedostatku času v hodinách informatiky učitel zdůrazňoval několikrát.

Co se týče žáků, učitel v rozhovoru kladl důraz na tradiční přístup k morálce (tedy ticho ve třídě a samostatná, frontálně řízená práce), avšak podporuje samostatné řešení problémů. Žáci by si měli být schopni s konkrétní komplikací poradit a vyřešit ji pokud možno vlastními silami za pomoci nápověd či internetu.

Žáci mají dle učitele problém s logickým myšlením, kombinací různých kroků nutných pro vyřešení a rozhodně mají problémy se čtením čehokoliv. Samozřejmě až na výjimky. *„Voni sou zvyklí, že všechno jim člověk řekne.* [...] *Hrozně málo lidí si dokáže poradit, třeba na internetu. Má připojení, má nápovědu, poraď si. Hrozně málo lidí je schopných si poradit.“*

Učitel nejprve sám řekl, že schopnost programování by se hodila, ale později specifikoval, že *„programovat se už učit nebudu.“* Kvůli specializaci spojaře na vojně je učitelův zájem spíše směřován na hardware a sítě. Manipulovat s hardwarem a operačními systémy podle něj však v současných školních podmínkách nejde, protože je to složitá záležitost, na kterou chybí ukázkové pracovní stanice (kde žáky zkažená systémová instalace ničemu nevadí, počítač se dočasně odstaví a pracuje se dál na normálních počítačích) a i kdyby byly, chyběla by časová dotace.

Zájem o informatiku má podle učitele jen malé procento žáků a vše je dále ztíženo údajnou nemožností doma si naučenou látku procvičit. Žáci nemají počítače, a když je mají, nemají sadu Office. Vyžadovat úkoly také nelze, protože žáci se vymlouvají právě na to, že nemají potřebný aplikační software. OpenOffice jsou zde jen na prvním stupni, ale rozdílnost ovládání od MS Office znamená, že se na druhém stupni nepoužívají. Pro děti je prý nové ovládání příjemnější.

Programování se v současné době na škole neučí a nikdo to učit za posledních šest let nezkoušel. Zájem o takovou výuku u žáků by stejně nebyl, navíc *„[grafické programování] to je určitě bude bavit, to jo... ale to je takový... pro děti, no...“* Na základní školy programování nepatří a na všeobecné střední školy typu gymnázií by grafické programování bylo maximálně pro zpestření. Programování jako takové patří na specializované střední školy, kde je i údajně naprosto odlišný přístup žáků k učení.

Tato domněnka byla opakována vícekrát. ZŠ má nevýhodu v tom, že žáci nejsou nijak třídění, kdežto na střední škole již ano a to na základě svého vlastního zájmu.

Učitel byl před zhlédnutím ukázkových hodin zvědavý, jak bude vše probíhat, protože sice jde svým způsobem o hraní a mohlo by je to bavit, ale číst si postupy a řídit vlastní práci je pro žáky problém, práci odflakují a nesoustředí se na ni.

Interpretace vstupního rozhovoru

Jak již bylo ve faktickém obsahu řečeno, reference na moderní technologie obecně degradovaly jejich význam. To poukazovalo na nízký rozhled učitele v oblasti technologických novinek, určitým způsobem se nejspíše jednalo o nechuť snažit se udržovat si přehled a zjišťovat si neustále nové informace (základní předpoklad dobrého učitele informatiky), což si respondent nejspíše podvědomě kompenzoval připisováním minimálního významu technologickému pokroku. V souvislosti s výše vypsanou citací týkající se nástěnky lze konstatovat, že bylo naprosto očividné, že respondent nemá ani tušení o existenci například elektronických nástěnek typu Lino (http://en.linoit.com/). Tento příklad je zde vypíchnut jen pro ilustraci absence zájmu o nové technologie. Přestože učitel samozřejmě nemusí (a ani nemůže) znát veškeré aplikace a novinky, nějaká forma práce se sdílenými daty je základem. Nedostatek technologického přehledu pravděpodobně souvisí s absencí aprobace informatiky a je tak poněkud pochopitelný. Na druhou stranu chápat *„interaktivní výuku“* jako obyčejná multimédia je u učitele informatiky poněkud problém.

Neustále zdůrazňování nedostatku času v kombinaci se zdůrazňováním neschopnosti většiny žáků poté znělo jako výmluva. *Proč se vůbec snažit, když to žáky nebaví a stejně na to v hodinách není čas?* Přestože učitel opakoval, že někteří žáci hlubší zájem o informatiku mají, opakovaně zdůrazňoval, že to jsou vzácné výjimky (jednou dokonce přímo specifikoval, že 2 %). Jinými slovy to znamená, že drtivá většina zájem nemá.

Tento nezájem je dost možná spojený s běžným chápáním předmětu informatiky jako takového povinného kurzu MS Office. Je-li většina informatiky chápána jako výuka práce s Excelem a Wordem (a prezentace v PowerPointu jsou skoro za odměnu), zájem mezi žáky vysoký být ani nemůže. V průběhu rozhovoru bylo opakovaně řečeno, že se žáky se pracuje v širším rozsahu činností, než jen MS Office, ale na základě četnosti takřka neustálého odkazování právě na MS Office lze vyvodit, že buď jsou všechny tyto činnosti chápány jen jako doplňkové, anebo je MS Office považováno za to nejdůležitější, čemu by se žáci měli naučit.

Faktický obsah výstupního rozhovoru

Učitel byl nejprve dotázán na cokoliv, co si myslí o proběhlé hodině. Hodina byla považována za přínosnou a za její největší klad byla považována netradičnosti výuky, při které byli žáci doslova nuceni logicky přemýšlet. Žáci, které hodina zaujala, tak měli možnost vidět novou oblast informatiky a dostali úvodní materiály pro případné další samostudium (viz příloha E).

Připomínkou k hodině bylo snížit tempo druhé dvouhodinovky Scratche. Tato připomínka však byla založena na první ze tří proběhlých dvouhodin, kdy autor této práce sám dospěl ke stejnému závěru a pro další skupiny připravil návod v písemné podobě. Žáci tak měli možnost pracovat vlastním tempem a následující dvě skupiny tak byly neporovnatelné se skupinou první.

Ve vstupním rozhovoru učitel zmiňoval, že si myslí, že hodina více zaujme chlapce, a tento svůj názor potvrdil i po proběhlé hodině. Odůvodněním je, že logické postupy a logické přemýšlení jsou spíše schopnostmi chlapců. Informatika obecně je viděna spíše jako *„klučičí doména“* a zájem děvčat je spíše výjimkou.

Učitel hodinu považoval za přínosnou i pro sebe, protože se jednalo o *„zpestření hodin,“* které by bylo teoreticky použitelné i v praxi. Nasazení do praxe však neplánuje dříve, *„než se to pořádně otestuje.“* Dětské programovací jazyky jsou i nadále viděny víceméně jako hraní a pouhá zajímavost. Algoritmizace a programování dle učitele stále patří na střední a specializované školy, kde si ale projekt dovede určitě představit.

Interpretace výstupního rozhovoru

Z výstupního rozhovoru bylo patrné, že tento projekt učitele zaujal, ale jen ve velice omezené míře. Názory stran programování se ani po zhlédnutí ukázkových hodin nijak výrazně nezměnily. Opakované označování projektu jako *„hraní“* a *„pro zpestření“* naznačuje, že takováto výuka není brána příliš vážně a uvažoval-li by vyučující o zařazení do hodin, jednalo by se opět o téma přinejmenším okrajové.

6.2 Rozhovory s učitelkou

Ani tato učitelka není aprobovanou informatičkou. Aprobace nebyla nikdy doplňována, ale vyučující je držitelkou ECDL certifikátu, který si sama individuálně dodělávala v době mateřské dovolené. Její aprobace jsou matematika a chemie. Délka její učitelské praxe je 22 let, z toho asi 15 let označovaných jako *„skoro od začátku co se začala informatika učit“* vyučovala také informatiku. S problematikou programování se ještě nesetkala, ale byla této myšlence od samého počátku nakloněna. Vůči novým technologiím není uzavřená a sama pracovala jako školitelka informačních a komunikačních technologií na prvních stupních základních škol.

Pozorované extralingvistické jevy

Respondentka nebyla ani při jednom z těchto dvou rozhovorů unavená, její projev byl jasný a velice energický. Byla naprosto klidná, přirozená a podklad k rozhovoru předem ani vidět nechtěla. Své myšlenky učitelka formulovala srozumitelně a po velkých blocích obsahujících mnoho tematicky spjatých asociací. Z otázek učitelka nervózní nebyla, a přestože jí bylo nabídnuto prohlédnutí otázek (kvůli zajištění stejných podmínek rozhovorů pro oba učitele), tato nabídka byla odmítnuta.

Faktický obsah vstupního rozhovoru

Učivo sedmých tříd, které učitelka vyučuje, je zaměřeno zejména na MS Word. Žáci jsou ale vnímáni jako velice šikovní a ze strany učitelky je tak silná snaha zpestřit hodiny širokou škálou různých aktuálních témat a mezipředmětových vztahů. Žáci například vypracovávají prezentace na témata z oblasti matematiky. *„už se tak nedržím striktně těch osnov nalajnovaných* [...] *ale že prostě spíš to beru po praktické stránce, co se jim hodí do budoucnosti.“*

Učitelka sama programovat neumí, s odůvodněním, že informatiku nestudovala a programovat doposud nikdy nepotřebovala. Vzápětí ale dodává, že si myslí, že programování na základních a středních školách citelně chybí, protože mnoho vysokých technických škol bere znalost základů programování za samozřejmost. Tento fakt zdůrazňuje na vlastní zkušenosti, kdy syn studující vysokou školu musel dohánět tyto základy sám, protože na většině škol se programování skutečně neučí.

Nezařazení programování do osnov je odůvodněno tím, že osnovy si učitelé vypracovávali sami a RVP je velice volné (zjištění konstatované již v kapitole 2):

„Takže to programování se tady neučilo, řekla bych proto, že jsme to nikdo neuměl, nebylo to potřeba, vyloženě to není v osnovách že v informatice má být to to to, takže jsme se přizpůsobili tomu co umíme a co známe."

Na střední školu by však programování zařadila určitě a to bez ohledu na druh školy:

„na základku možná na nějaký ty základy potom v devítce třeba ke konci, ale na střední školu bych to programování dala určitě, protože dneska to potřebujou na ty technický obory. A kor třeba ty všeobecný gymply, ty by vlastně vůbec to měly zvládat. [...] Na tý základce nevím jestli by to bylo to pravý vořechový, protože pro ty děti na ty základce ta úroveň jde níž. Spíš bych to dala na tu střední školu. Jo, na tu základku spíš možná ne."

V přístupu k žákům z hlediska morálky je opět využita striktnější direkce výuky. Protože se v případě jejího hlavního předmětu jedná o matematiku, je prý nutné, aby bylo ve třídě ticho a klid. Třídy jsou na tento režim navyklé a pracují tak dobře. Přísnější režim se osvědčil natolik, že prý někteří žáci po nástupu na střední školu přicházejí poděkovat, že je učitelčin přístup na přechod na SŠ skutečně dobře připravil.

Žáci jsou považování za šikovné, ale z hlediska učitelky žáci sami své síly a schopnosti přeceňují omezením na své typické činnosti, tedy hry a Facebook. Je-li potřeba umět něco do života přínosného, nejsou zpravidla samostatně schopni ani základů. Na druhou stranu *„všeobecně zase už to je jiný, než když jsem začínala, kdy třeba dvě děti měly z celé třídy počítač doma a dneska vlastně zase už má víc dětí, každej, už má doma počítač."*

Zájem žáků o technologie a novinky v informatice jsou opět omezeny na trendy a zajímavosti. Tvořivé informatické činnosti (blogy, weby,...) se však opět vyhýbají, protože by to pro ně byla zátěž a děti jsou pohodlné. Genderově se žáci se liší jen přístupem k informatice. Chlapci jsou intuitivní, kdežto dívky podstatně pečlivější.

Zájem učitelky o Hour of Code byl zřejmý, přestože se s projektem nikdy nesetkala, popisovala ho slovy: *„na jednu stranu to je kouzlo. Protože to vypadá jako hry a na druhou stranu to není to klasický učení."* Po seznámení s projektem v případě shledání jeho užitečnosti a použitelnosti má učitelka v plánu projekt využívat i v dalších letech.

Jako obecným problém při zavádění nových projektů je určeno získávání informací pro učitele. Na škole se například zavádí projekt *Tablety do škol*, který je ale považován za uspěchaně a nekvalitně pojatý. Instruktoři, kteří učí, jak tablety ve výuce použít, jsou neschopní a zjišťovat jak co funguje, je příliš časově i energeticky náročné.

Interpretace vstupního rozhovoru

Důraz na mezipředmětové vztahy by měl být základem hodin informatiky. Procvičovat učivo na vykonstruovaných příkladech je zpravidla zbytečné, protože na základě právě mezipředmětových vztahů lze lépe demonstrovat, že to, co se žáci učí, jim skutečně k něčemu je. I přes chvályhodný tematický přístup se ale obsahem hodin opět z hlediska informatiky jedná víceméně o kurz MS Office.

K začlenění hodin programování do výuky se ale učitelka staví výrazně pozitivněji, což je s největší pravděpodobností ovlivněno zejména osobní zkušeností (syn, který programování na VŠ potřebuje, ale na nižších stupních se s ním nikdy nesetkal). Je otázkou, jak by se učitelka v tematice stavěla, kdyby tato situace nenastala.

Z hlediska času učitele je popisován stejný problém. Učitelé jsou přetíženi množstvím různých úkolů a pro hledání dalších informací do svých předmětů takřka není čas. Tento problém je v případě učitelky částečně adresován snahou zapojovat do hodin aktuální témata.

Rozhled učitelky v oblasti informatiky byl rozhodně lepší a otevřenost novým technologiím ještě o hodně více. Opět však z důvodu časové vytíženosti nebyla znatelná příliš velká snaha o rozšiřování stávajících znalostí. Až na případ, že by se jednalo o kvalitní školení, ve kterém by učitelka obdržela již zpracované konkrétní informace

Každý projekt musí být podpořen kvalitním základem, tedy kurzem pro učitele, kteří projekt mají používat. Jestliže je projekt prezentován špatně, postoj učitelů je odpovídající. Optimální by tedy mohlo být při snaze propagovat Hour of Code nebo Scratch vytvořit tým školitelů, kteří by objížděli republiku a učitelům nejprve předvedli ukázkové hodiny a poté poskytli dostatečně hluboké školení, na jehož základě by již učitelé neměli problém dále rozvíjet své znalosti a schopnosti v dané oblasti.

Faktický obsah výstupního rozhovoru

Názory prezentované ve výstupním rozhovoru, týkající se shlédnuté ukázkové hodiny byly velice podobné s názory učitele:

> *„určitě to je pro... ty žáky zpestření a... zas uplně něco jiného, než klasika, co se tady učí. Tím, že samozřejmě je to formou her, tak samozřejmě k tomu mají blíže, na druhou stranu zase vidím i přínos v tom, že přece jenom ... trošku musí přemýšlet,...* [...] *někdo se o tom potom může třeba i víc zajímat a to, ten přínos tam taky určitě je.“*

Na dotaz, zdali se hodlá výukou programování dále zabývat, učitelka reagovala:

> *„Určitě by mě to zajímalo, protože jak jsem říkala, že se nebráním vždycky něčemu novému, něčemu zajímavému, že mě to někam ponese výš, že nezabřednu jenom u těch klasickejch starejch věcí, a furt dokola.* [...] *jako já bych nikde abych hledala na internetu, sháněla, četla literaturu nebo něco, prostě na to čas nemám. Že jo. To prostě už ne, to člověk spíš musí se s tím nějakým způsobem setkat, třeba jako já – mě to zajímá teď – se o to zajímat víc.* [...] *když už jsem to viděla, že je takováhle možnost, tak jo, tak mě to zajímá, ale kdybych měla někde hozený skritpa a... nauč se to... tak...jo? Tak to ne už...“*

Původní názor na nevhodnost problematiky programování na základní škole (viz vstupní rozhovor) však učitelka sama přehodnotila:

> *„protože teď jsem vlastně viděla vlastně, že to jde i udělat i touhletou jednodušší formou.... než klasickejma jenom příkazama,* [...] *proto jsem řikala, že... se to nehodí protože ty žáci vlastně sotva pojmou některé vědomosti, takže když to je touto formou, kdy vlastně ty příkazy se NEMUSÍ učit, ale touto formou tam vlastně jenom vyskakují, tak si myslím, že touto formou to lze vyučovat i na základní škole.“*

Na střední školu již však tento druh výuky byl považován za vhodný jen jako vstupní seznámení, úvod do problematiky.

Interpretace výstupního rozhovoru

Shlédnutá hodina byla hodnocena velice pozitivně. Přestože o problematice programování bylo referováno i nadále spíše jako o *„zajímavosti,“* demonstrace výuky učitelku přesvědčila, že zařazení programování na ZŠ je vhodné. Na druhou stranu po zhlédnutí ukázky byla stylizace Hour of Code vyhodnocena jako příliš dětská pro žáky středních škol, kde by byla HoC vhodná jen jako vysloveně vstup do programování.

Jako největší problém byl opět zopakován nedostatek času, ale zájem o problematiku stran učitelky byl očividný. Pravděpodobně preferovaná forma by bylo stále odborné školení, ukázková hodina však možná byla impulzem pro individuální vyhledávání informací stran programování a prezentovaných projektů.

6.3 Porovnání názorů a vyvozené závěry

Na základě proběhlých rozhovorů lze konstatovat, že názory těchto dvou učitelů a zejména jejich přístup k informatice a novým technologiím se diametrálně odlišují. Přesto však bylo možné najít určité shodné body. Z obou rozhovorů je například velice citelná fixace na výuku práce s konkrétním aplikačním vybavením a zejména pak se sadou Microsoft Office. Oba učitelé se však snaží vést žáky k samostatnosti, ke schopnosti efektivně řešit problémy.

Oba respondenti dále zmiňovali programování jako zpestření a zajímavost vhodnou spíše pro střední školy. Zatímco se však názor učitele po zhlédnutí ukázkové hodiny téměř nezměnil, názor učitelky byl silně modifikován. Programování je stále viděno jako zajímavost a zpestření, avšak v této podobě je realizace na základních školách dle učitelky reálná. Učitel sice tento postoj také potvrdil, jednalo se však o zdráhavé potvrzení pravděpodobně nepodložené hlubším osobním přesvědčením.

Velice podobný názor měli oba učitelé i na samotné žáky. Nejprve se jejich komentáře shodují v nutnosti klidu ve třídě a shodují se i v popisu práce samotných žáků. Žáci jsou viděni jako relativně šikovní, ale velice pohodlní / líní. Predikce učitele o problémech spojených s nutností čtení instrukcí a vysvětlení se nakonec skutečně potvrdila, někteří žáci měli problémy s průchodem Hour of Code právě proto, že instruktážní videa vypínali a tabulky s instrukcemi si nečetli.

Stran žáků se však učitelé neshodli na genderových rozdílech. Učitel tvrdil, že informatika je (až na výjimky) doménou chlapců, naproti tomu učitelčino tvrzení je založené na rozdílnosti přístupu žáků k práci. Chlapci jsou intuitivnější, dívky pečlivější, ale předpoklady pro studium mají rovnocenné.

Dalším významným bodem, ve kterém se tvrzení obou učitelů rozcházela, byla časová dotace předmětu informatiky. Jeden z původních návrhů hypotéz spočíval na učitelově tvrzení o nedostatku času v hodinách informatiky, tedy že programování se neučí, protože už stejně není kam ho do kurikula předmětu přidat. Tento předpoklad byl nakonec z pracovních hypotéz vyškrtnut a označen za subjektivní, protože učitelka naopak viděla časovou dotaci předmětu informatiky jako nadprůměrnou v porovnání s jinými školami. Je-li tedy k dispozici více hodin, část tohoto *přebytku* teoreticky může být věnována výuce programování a jedná se tedy jen o záležitost otevřenosti tématu při formulaci kurikula předmětu.

Učitelův odhad, že tematika by se týkala jen minima žáků, byl z procesu formulace hypotéz vyřazen na základě výsledků sesbíraných v kvantitativní části šetření. Zájem žáků byl naopak velký, ačkoliv pokrok žáků nebyl z výše jmenovaných důvodů spojených s bodovanými otázkami ve výstupních dotaznících reálně měřitelný.

Kategorie	Interpretace odpovědí učitele	Interpretace odpovědí učitelky
Otevřenost studiu programování	Učit se programovat už nechce a **nebude**. Po předvedení hodin má však v úmyslu se na projekty podívat dále.	Sama problematiku **nevyhledává, ale** je-li jí představen nový systém, **zájem má**.
Názory na programování	Programování je **chápáno jako syntax konkrétního jazyka**, vizuální programování je hraní pro děti a má minimální přínos.	Programování je chápáno abstraktněji, jako **něco důležitého** (silné ovlivnění osobní životní zkušeností), náznaky chápání opět jako pouhé **použití syntaxe jazyka**.
Programování na ZŠ	Nasazení klasického programování na ZŠ je **blbost**, programování jako takové by se týkalo 2 % lidí, určitě by to ale přínos byl.	**Možná na konci devítky**, ale spíše na střední škole. Po ukázkové hodině je **výuka programování na ZŠ realizovatelná**.
Programování na SŠ	Na všeobecná gymnázia pro zajímavost, jinak **jen na specializované střední školy.**	Rozhodně by mělo být **všude bez výjimky**.
Fixace na MS Office	**Velice silná**, základ výuky je MS Excel, Word a PowerPoint (ale později je přidána i práce s fotkami, videem, internetem, apod.)	**Slabší, ale stále silná**. Velká snaha o **začlenění mezipředmětových vztahů a aktuálních témat**.
Problematika žáků	Většina žáků **není schopna sama řešit problémy**, **nemají schopnosti logického uvažování**, jsou **líní**, **nečtou**. Zhruba před pěti lety se chování žáků ohromně zhoršilo.	Žáci jsou **pohodlní, ale šikovní**.
Rozdíly chlapci vs. dívky	**Informatika je doménou chlapců**, protože mají technické a logické myšlení.	Rozdíl je jen v přístupu žáků, **chlapci jsou intuitivnější a dívky pečlivější**.
Nedostatek hodin informatiky/času v hodinách	**Hodin je strašně málo** a nestihne se probrat nic.	**Hodin je více než na jiných základních školách**, prostor pro programování by byl.
Časová vytíženost učitelů	Naprostý nedostatek času.	Nedostatek času, škola se řeší neustále. Administrativa neustále přibývá.
Pasivita při vyhledávání novinek	Není čas, není zájem. Novinek je tolik, že se v nich nedá orientovat.	Není chuť. Vyhledávat novinky je namáhavé, ale jsou-li představeny někým jiným, zájem a otevřenost je velká.

Tabulka 7 – Finální kategorizace kódovaných segmentů odpovědí respondentů

Na základě metody konstantní komparace byly vypracovány dvě pracovní hypotézy, jejichž ověřování je možným podkladem pro kvantitativní výzkum na větším vzorku učitelů základních škol v České republice:

H1: Učitelé, kteří nemají aprobaci informatika, si jako výuku programování představí tradiční psaní kódu, spíše než vizuální a hravé programování za pomoci dětských programovacích jazyků.

H2: Zvýšení osvěty učitelů informatiky pomocí ukázkových hodin a školení o dětských programovacích jazycích zvýší rozšířenost výuky programování na základních školách.

První hypotéza je založena na faktu, že oba učitelé se nikdy dříve s dětskými programovacími jazyky nesetkali a jejich představa programování skutečně bylo ruční psaní procedurálního kódu v nějakém programovacím jazyce. Programování, skrývající se pod jejich původní představou, považovali za nerealizovatelné v předmětu informatika na základní škole. Po zhlédnutí ukázky Hour of Code a zjištění o existenci dětských programovacích jazyků byli oba tematikou zaujati a výhledově plánují využití projektu Hour of Code ve výuce.

Druhá hypotéza přímo vychází z hypotézy předcházející. Oba učitelé se shodli na naprostém nedostatku času pro hledání takovýchto vzdělávacích projektů. Přestože oba tvrdí, že se snaží aktualizovat své znalosti a následkem toho i obsah hodin, cíleně tuto činnost neprovádějí a sami preferují, seznámí-li je s novou problematikou někdo jiný.

Jelikož obsah předmětu nejvíce ovlivňují právě učitelé, kteří ho vyučují (a RVP nejsou dostatečně restriktivní), absence problematiky v hodinách je logicky způsobená samotnými učiteli. Cílem výzkumných rozhovorů bylo nalézt důvod či důvody bránící zapojení výuky programování do hodin informatiky. Jako tento důvod byla po analýze a interpretaci rozhovorů označena neznalost celé problematiky, která je dále komplikovaná přetěžováním učitelů. Ti nemají čas ani síly se sami učit novou a relativně vysoce komplexní problematiku.

Možným řešením obou problémů, navrhovaným již v průběhu interpretace rozhovorů, je tedy vytvoření školení učitelů informatiky, která by ve srozumitelné formě prezentovala dětské programovací jazyky a vybrané konkrétní projekty. I nadále by zvládnutí této problematiky vyžadovalo aktivní zapojení stran učitelů a samotné školení by nebylo automaticky zaručeným způsobem pro masové rozšíření výuky programování na ZŠ, ale učitelé by tak měli alespoň představu, že něco takového existuje, a že to rozhodně není takový problém, jak se může zdát.

ZÁVĚR

První kapitola teoretické části práce se zabývala základními pojmy a definicemi z oblasti algoritmizace a programování. Nejprve byly vysvětleny konstrukce z tradičního procedurálního programování, jako jsou například cykly, podmínky, proměnné, apod. Poté byl stručně nastíněn historický vývoj programovacích jazyků, který byl zakončen přechodem k objektovému paradigmatu. Podkapitola o objektově orientovaném programování dále rozebírá základní myšlenky a principy OOP.

Druhá kapitola se skládala z popisu současné situace z hlediska stavu výuky algoritmizace a programování na ZŠ a byla zde nastíněna problematika z pohledu RVP. V odpověď na otázku, zda by se programování na základních školách vůbec mělo vyučovat, byly shrnuty vybrané názory světových i českých odborníků z řad učitelů a vývojářů. Tyto názory se neomezovaly pouze na příznivce programování, prostor byl dán i argumentům odpůrců. V závěru druhé kapitoly byl popsán přístup autora této práce a jeho pohled na podstatu a smysl celé výuky programování na ZŠ.

Hraniční kapitolou mezi teoretickou a praktickou částí byla kapitola třetí, ve které byla navržena a stručně představena možná východiska pro výuku programování na základních školách. Ve stručnosti zde byly zmíněny tradiční programovací jazyky jako je Karel, LOGO nebo Baltík. Větší prostor byl ponechán představení novějších projektů, jmenovitě Khan Academy, CodeCombat a LEGO Mindstorms. Tyto tři projekty byly zmíněny jako vhodný doplněk k centrálním dvěma projektům této práce, které byly analyzovány nejdetailněji. Představení Hour of Code obsahovalo popis celého projektu, jeho historii a vývoj, spolupráci s velkými společnostmi a podporu známých a významných světových osobností. Projekt Scratch je relativně známým nástrojem pro výuku programování na školách a jeho popisem se již zabývalo více prací. Zde byl proto Scratch analyzován jen z hlediska přímých souvislostí a možností návaznosti s projektem Hour of Code.

Praktická část této diplomové práce byla rozdělena do tří kapitol. První, a nejkratší z nich, popisovala specifika vybrané základní školy, na které celým výzkum probíhal. Druhá praktická, a v celkovém pořadí pátá kapitola, obsahovala celou kvantitativní část výzkumu, který proběhl mezi žáky v hodinách informatiky. Poslední kapitola celé práce poté obsahovala kvalitativní část výzkumu, jenž je založena na polostrukturovaných rozhovorech s jednou učitelkou a jedním učitelem informatiky. Cílem této části bylo zjistit možné příčiny tak malé rozšířenosti programování v hodinách informatiky.

Kvantitativní část výzkumu byla založena na ukázkové hodině Hour of Code s Lightbotem na závěr pro nejrychlejší žáky. Žáci experimentální skupiny nejprve na začátku dvouhodinovky vyplnili vstupní dotazník, poté absolvovali Hour of Code a dvouhodinovka byla zakončena dotazníkem výstupním. Vstupní i výstupní dotazník obsahovaly mimo jiné i tři bodově vyhodnocované otázky. První z nich zjišťovala, co si žáci představí pod pojmem *programování* před a po ukázkové hodině, druhá zjišťovala schopnost žáků vytvořit jednoduchou posloupnost instrukcí a nalézt mezi nimi opakující se vzory a poslední otázka studovala schopnost žáků vytvořit vlastními slovy bez jakéhokoliv syntaktického omezení jednoduchý algoritmus popisující relativně jednoduchý problém přecházení silnice.

Kromě experimentální skupiny byly pro otestování vstupního dotazníku využity dvě skupiny pilotní a pro validaci výsledků byly dále vyčleněny dvě skupiny kontrolní. Celkem se celého kvantitativního výzkumu zúčastnilo 146 žáků ve 14 skupinách.

Projekt Hour of Code si neklade za cíl naučit žáky co nejvíce, ale zaujmout pozornost co největšího počtu žáků. Tento cíl byl ověřován v rámci výstupního dotazníku v experimentálních skupinách otázkami zjišťujícími: jak moc žáky hodina bavila (na pětistupňové škále vybralo 70 % žáků hodnoty 4 nebo 5 značící, že hodně), jestli by chtěli v programování pokračovat v hodinách informatiky (69 % žáků ano) a jestli by měli zájem o kroužek programování a robotiky (39 % žáků ano). Hour of Code tedy stanovený cíl naplňuje v míře více než uspokojivé.

Projekt Hour of Code dále tvrdí, že je určen pro všechny žáky bez rozdílu věku či sociokulturního pozadí (Code.org 2016a). Toto tvrzení ověřovaly dílčí hypotézy, které porovnávaly výsledky v bodovaných úlohách vstupního dotazníku s mírou významu přikládaného žáky informatice, s šíří jejich zájmu a s jejich věkem. Hypotézy byly ověřovány Kendallovým korelačním koeficientem Tau β. Ani v jednom případě nebylo možné potvrdit nulové hypotézy, které tvrdily, že mezi ověřovanými kategoriemi existuje souvislost. Toto zjištění potvrzuje tvrzení projektu Hour of Code, protože z hlediska vstupních předpokladů nebyl nalezen žádný faktor, který by je ovlivňoval.

Přestože, jak již bylo řečeno, si Hour of Code neklade za cíl naučit žáky co nejvíce, tato práce se snažila zjistit, je-li už po absolvování jedné ukázkové hodiny znatelný nějaký pokrok v uvažování žáků. Tento pokrok měl být porovnáván na výše zmiňovaných třech bodovaných otázkách. Z logistických důvodů ale musely být vstupní i výstupní dotazníky součástí jedné dvouhodinovky, což způsobilo nečekané problémy.

Žáci byli ve výstupních dotaznících již velice unavení a navíc neměli vůbec čas na fixaci nové látky. Jejich odpovědi ve výstupních dotaznících tak byly velice často zejména kvůli zmiňovanému vyčerpání nedokončené, což samozřejmě zničilo jejich vypovídající hodnotu. Ověření skutečného přínosu Hour of Code tak zůstává otázkou pro další výzkum.

Kvalitativní část výzkumu byla založena na metodě polostrukturovaného rozhovoru s učiteli informatiky. Respondenty byli jedna učitelka a jeden učitel, přičemž s oběma byly vedeny celkem dva výzkumné rozhovory – jeden před a druhý po zhlédnutí ukázkové hodiny Hour of Code. Oba respondenti byli vybídnuti k volným asociacím a v případě nutnosti byli požádáni o další klarifikaci svých výroků.

Všechny čtyři rozhovory byly zpracovány dle Mgr. Kateřiny Juklové, Ph.D. (in Skutil 2011). Rozhovory tedy prošly plnou transkripcí, segmentací, kódováním a kategorizováním. V analýze jsou rozbory rozhovorů vždy striktně odděleny na popis faktického obsahu rozhovoru a interpretaci ze strany autora této práce.

Cílem rozhovorů bylo zjištění faktorů ovlivňujících nasazení (či spíše absenci) programování v hodinách informatiky na ZŠ. Metodou konstantní komparace tvrzení obou učitelů byly vytvořeny dvě hypotézy, které jsou možným podkladem pro další výzkum kvantitativního charakteru mezi učiteli základních škol v České republice.

> H1: Učitelé, kteří nemají aprobaci informatika, si jako výuku programování představí tradiční psaní kódu, spíše než vizuální a hravé programování za pomoci dětských programovacích jazyků.
> H2: Zvýšení osvěty učitelů informatiky pomocí ukázkových hodin a školení o dětských programovacích jazycích zvýší rozšířenost výuky programování na základních školách.

Zjišťovány byly dále názory a postoje učitelů na programování před a po ukázkové hodině. Pozitivním faktem je, že po shlédnutí Hour of Code byli oba učitelé projektem zaujati a měli v úmyslu se jím dále sami zabývat.

Závěrem lze tedy tvrdit, že využití Hour of Code v podmínkách vybrané školy bylo nejen bez problémů realizovatelné, ale i přínosné jak pro žáky, tak pro učitele. Testování experimentálního nasazení projektu do výuky v rámci běžných hodin informatiky dopadlo nanejvýš uspokojivě. I přes určité výhrady stran procedurálního přístupu celého projektu Hour of Code a přehlížení aktuálního objektového paradigmatu je projekt Hour of Code doporučen k nasazení do výuky v rámci hodin informatiky za účelem seznámení žáků základních škol s problematikou programování, která je v ČR v současné době přinejmenším opomíjená, či spíše až naprosto neexistující.

ZDROJE

Ars Technica. (2012). *Is it true that „not everyone can be a programmer"?* [online]. New York City: Condé Nast Digital [cit. 2016-06-12]. Dostupné z: <http://arstechnica.com/information-technology/2012/09/is-it-true-that-not-everyone-can-be-a-programmer/>

ATWOOD, Jeff. (2006). *Separating Programming Sheep from Non-Programming Goats*. [online]. Coding Horror: programming and human factor. Berkeley [cit. 2016-06-12]. Dostupné z: <https://blog.codinghorror.com/separating-programming-sheep-from-non-programming-goats/>

ATWOOD, Jeff. (2012). *Please, Don't Learn to Code*. [online]. Coding Horror: programming and human factor. Berkeley [cit. 2016-06-12]. Dostupné z: <https://blog.codinghorror.com/please-dont-learn-to-code/>

BELL, Tim, Ian WITTEN, Mike FELLOWS a Matt POWELL, JARMAN, Sam (ed.). (2015). *Computer Science Unplugged: An enrichment and extension programme for primary-aged students* [online]. 3.1 edition. Raleigh, North Carolina: Lulu Press, 245 s. [cit. 2016-06-14]. Dostupné z: <http://csunplugged.org/books/>

BLAHO, Andrej a Ivan KALAŠ. (2006). *Imagine Logo: učebnice programování pro děti*. Brno: Computer Press. Česká škola (Computer Press). ISBN 80-251-1015-X.

BORNAT, Richard. (2014). *Camels and humps: a retraction* [online]. Middlesex University, 8 s. [cit. 2016-06-13]. Dostupné z: <http://www.eis.mdx.ac.uk/staffpages/r_bornat/papers/camel_hump_retraction.pdf>

BRENNAN, Karen, Christian BALCH a Michelle CHUNG. (2014). *Creative Computing* [online]. Harvard University, Cambridge (Massachussetts) [cit. 2016-06-17]. Dostupné z: <http://scratched.gse.harvard.edu/guide>

BRET, Victor. (2012). *Learnable Programming: Designing a programming system for understanding programs.* [online]. Worry Dream. Berkeley [cit. 2016-06-13]. Dostupné z: <http://worrydream.com/LearnableProgramming/>

BRICKLIN, Dan. (2002). *Why Johnny can't program*. [online]. Software Garden. Dan Bricklin's Web Site [cit. 2016-06-13]. Dostupné z: <http://www.bricklin.com/wontprogram.htm>

BROMOVÁ, Jana. (2012). *Výuka algoritmizace na ZŠ: aktuální stav*. Bakalářská práce. Jihočeská univerzita, Pedagogická fakulta, Katedra informatiky, 97 s. Vedoucí práce Doc. PaedDr. Jiří Vaníček, Ph.D.

Calculator.vhex.net. (2016). *Kendall tau correlation calculator* [online]. [cit. 2016-06-02]. Dostupné z: <http://calculator.vhex.net/calculator/statistics/kendall-tau-correlation>

CODE.ORG. (2013). *What Most Schools Don't Teach* [online video]. YouTube, 2013-02-26. [cit. 2016-05-07]. Dostupné z: <https://www.youtube.com/watch?v=nKIu9yen5nc>

CODE.ORG. (2014). *Vytvoř si vlastní Flappy hru, úroveň 10* [online]. Seattle (Washington) [cit. 2016-06-23]. Dostupné z: <https://studio.code.org/flappy/10>

CODE.ORG. (2015). *Code.org 2015 Annual Report* [online]. Seattle (Washington) [cit. 2016-05-21]. Dostupné z: <https://code.org/about/2015>

CODE.ORG. (2016a). *Hour of Code* [online]. Seattle (Washington) [cit. 2016-05-23]. Dostupné z: <https://hourofcode.com/cz>

CODE.ORG. (2016b). *Major Partners and Corporate Supporters* [online]. Seattle (Washington) [cit. 2016-07-03]. Dostupné z: <https://code.org/about/partners>

Computer Science Education Week. (2015). *Teach the Hour of Code in your classroom* [online]. Seattle (Washington) [cit. 2016-06-21]. Dostupné z: <https://csedweek.org/educate/hoc>

COUFAL, Petr. (2010). *Využití stavebnice LEGO ve výuce fyziky*. Bakalářská práce. Univerzita Palackého, Přírodovědecká fakulta, Katedra experimentální fyziky, 54 s. Vedoucí práce RNDr. Pavel Krchňák, Ph.D.

COUFAL, Petr. (2014). *Využití stavebnice LEGO Mindstorms ve výuce*. Bakalářská práce. Univerzita Hradec Králové, Přírodovědecká fakulta, Katedra informatiky, 54 s. Vedoucí práce Doc. RNDr. Štěpán Hubálovský, Ph.D.

Česká škola. (2009). *Dobrý Karel ještě žije* [online]. [cit. 2016-07-02]. Dostupné z <http://wiki.ceskaskola.cz/Home/vybrane-clanky-z-let-2001-2009/ict/dobry-karel-jeste-zije>

DEHNADI, Saeed. (2009). *A cognitive study of early learning of programming* [online]. Saeed's Homepage. Spolupracoval Richard Bornat. London: School of Engineering and Information Sciences Hendon [cit. 2016-06-13]. Dostupné z: <http://www.eis.mdx.ac.uk/research/PhDArea/saeed/>

ĎURÁKOVÁ, Daniela, Jiří DVORSKÝ a Eliška OCHODKOVÁ. (2002). *Základy algoritmizace*. Ostrava, 289 s. Scriptum. VŠB Technická univerzita Ostrava, Katedra informatiky.

DUŠEK, František. (2009). *Principy počítačů (PRINPOC): Podklady k seminářům* [pdf]. Univerzita Hradec Králové, Pedagogická fakulta, Katedra Informatiky, 22 s. Scriptum.

ECKEL, Bruce. (2006). *Thinking in Java*. 4th edition. Upper Saddle River, NJ: Prentice Hall, 1057 s. ISBN 978-0-13-187248-6.

HALOUSKOVÁ, Alena. (2013a). *Programování pro děti: naučte se programovat při tvorbě skvělých her*. Brno: Computer Press. ISBN 978-80-251-3809-0.

HALOUSKOVÁ, Alena. (2013b). *Učebnice jazyka Scratch*. Brno: Masarykova univerzita, Fakulta informatiky. Dostupné z: <http://perun.ms.mff.cuni.cz/ucebnicejazykascratch/>

HARPER, Douglas. (2016). *Algorithm* [online]. Online Etymology Dictionary. Lancaster [cit. 2016-03-10]. Dostupné z: <http://www.etymonline.com/index.php?term=algorithm>

HEESE, Brian. (2014). *Why Teach Computer Science in High Schools* [online]. Copyright © 2016 The Huffington Post: US Edition. [cit. 2016-04-15]. Dostupné z: <http://www.huffingtonpost.com/brian-heese/why-teach-computer-science_b_4350315.html>

HEGGART, Keith. (2014). *Coded for Success: The Benefits of Learning to Program*. [online]. Edutopia. George Lucas Educational Foundation. Marin County (California), 16-06-2014 [cit. 2016-06-18]. Dostupné z: < http://www.edutopia.org/discussion/coded-success-benefits-learning-program>

HEROUT, Pavel. (2001). *Učebnice jazyka JAVA*. České Budějovice: KOPP, 349 s. ISBN 80-7232-115-3.

HLADÍKOVÁ, Marie. (2016). *Baltík* [obrázek]. Kroužek programování [online]. Holice: Gymnázium Dr. Emila Holuba. [cit. 2016-06-15]. Dostupné z: <http://www.krouzek-programovani.8u.cz/baltik.php>

JEDLIČKA, Oldřich. (2013). *Robot Karel: vývojové prostředí* [online]. Copyright © 2006-2013. [cit. 2016-06-15]. Dostupné z: <http://karel.oldium.net/>

KHAN, Salman. (2016a). *Press room*. [online]. Khan Academy. Mountain View (California), 2015-06-10 [cit. 2016-06-15]. Dostupné z: <https://khanacademy.zendesk.com/hc/en-us/articles/202483630-PressRoom>

KHAN, Salman. (2016b). *Computer Programming*. [online]. Khan Academy. Mountain View (California), c2016, [cit. 2016-06-15]. Dostupné z: <https://www.khanacademy.org/computing/computer-programming>

LIFELONG KINDERGARTEN RESEARCH GROUP. (2013). *Gettings Started Guide* [online PDF]. MIT Media Laboratory, Cambridge, Massachussetts Institute of Technology [cit. 2016-04-17]. Dostupné z: <https://cdn.scratch.mit.edu/scratchr2/static/__c0f55e53b5d38e8912c777ed5b9c0030__//pdfs/help/Getting-Started-GuideScratch2.pdf>

LIFELONG KINDERGARTEN RESEARCH GROUP. (2016a). *Statistics* [online]. MIT Media Laboratory, Cambridge, Massachussetts Institute of Technology [cit. 2016-06-26]. Dostupné z: <https://scratch.mit.edu/statistics/>

LIFELONG KINDERGARTEN RESEARCH GROUP. (2016b). *Scratch Day* [online]. Scratch Wiki. MIT Media Laboratory, Cambridge, Massachussetts Institute of Technology [cit. 2016-06-27]. Dostupné z: < https://wiki.scratch.mit.edu/wiki/Scratch_Day>

LIGHTBOT INC. (c2016). *Lightbot* [online]. Ontario (Kanada) [cit. 2016-04-03]. Dostupné z: <https://lightbot.com/hocflash.html>

Metodický portál RVP. (2016). *Rámcový vzdělávací program pro základní vzdělávání*. [online]. Praha: Výzkumný ústav pedagogický v Praze, 164 s. [cit. 2016-06-12]. Dostupné z: <http://www.nuv.cz/uploads/RVP_ZV_2016.pdf>

MIHAI, Dan. (2014). *Sensors and Accessories for the Mindstorms NXT*. [obrázek]. Smashing Robotics. c2012-2016 [online]. Dostupné z: <http://www.smashingrobotics.com/sensors-and-accessories-for-the-mindstorms-nxt/>

MIT. (2015). *MIT Nobel Prize* [online]. Cambridge, Massachussetts Institute of Technology [cit. 2016-07-02]. Dostupné z: <http://web.mit.edu/ir/pop/awards/nobel.html>

MÖNIG, Jens a Brian HARVEY. (c2016). *Snap!* [online]. Berkeley, University of California [cit. 2016-06-10]. Dostupné z: <http://snap.berkeley.edu/>

MOTYČKA, Arnošt. (1999). *Algoritmizace*. Brno: Konvoj, 75 s. Scriptum. ISBN 80-85615-80-0.

MŠMT: Ministerstvo Školství, Mládeže a Tělovýchovy. (2016). *Schvalovací doložky učebnic*. [online]. c2013-2016 [cit. 2016-06-12]. Dostupné z: <http://www.msmt.cz/vzdelavani/skolstvi-v-cr/schvalovaci-dolozky-ucebnic-2013>

MUSÍLEK, Michal. (2011). *Kapitoly z dějin informatiky*. Vyd. 1. Hradec Králové: Gaudeamus, 193 s. Scriptum. ISBN 978-80-7435-129-7.

MUSÍLEK, Michal. (2012). *Dětské programovací jazyky*. Univerzita Hradec Králové, 97 s. Scriptum. Dostupné z: <http://www.musilek.eu/michal/pdf/deproja0.pdf>

MUSÍLEK, Michal. (2013). *Projekt Scratch*. Journal of Technology and Information Education [online]. Olomouc: Univerzita Palackého, 2013(1) [cit. 2016-07-06]. ISSN 1803-537X. Dostupné z: <http://jtie.upol.cz/pdfs/jti/2013/01/15.pdf>

PARTOVI, Hadi. (2014). *The "Secret Agenda" of Code.org* [online]. Seattle (Washington) [cit. 2016-03-12]. Dostupné z: <http://blog.code.org/post/73963049605/the-secret-agenda-of-codeorg>

PAVLUS, John. (2012). *Dear Everyone Teaching Programming: You're Doing It Wrong.* [online]. MIT Technology Review. Massachusetts [cit. 2016-06-13]. Dostupné z: <http://www.technologyreview.com/view/429438/dear-everyone-teaching-programming-youre-doing-it-wrong>

PECINOVSKÝ, Rudolf. (2003). *Výuka objektově orientovaného programování žáků základní a středních škol* [online]. Ostrava, 12 s. [cit. 2016-06-15]. Příspěvek na konferenci Objekty 2003. Dostupné z: <http://www.vyuka.pecinovsky.cz/prispevky/2003_OB_Vyuka_OOP_zaku_ZS_a_SS.pdf>

PECINOVSKÝ, Rudolf. (2004). *Proč a jak učit OOP žáky základních a středních škol* [online]. Žilina, 10 s. [cit. 2016-06-14]. Příspěvek na konferenci Žilinská didaktická konferencia 2004. Dostupné z: <http://www.vyuka.pecinovsky.cz/prispevky/2004_ZDK_Proc_a_jak_ucit_OOP_zaky_ZS_a_SS.pdf>

PECINOVSKÝ, Rudolf. (2005). *Jak efektivně učit OOP* [online]. Ostrava, 8 s. [cit. 2016-06-14]. Příspěvek na konferenci Tvorba softwaru 2005. Dostupné z: <http://www.vyuka.pecinovsky.cz/prispevky/2005-SW_Jak_efektivne_ucit_OOP.pdf>

PECINOVSKÝ, Rudolf. (2009). *Myslíme objektově v jazyku Java: kompletní učebnice pro začátečníky*. 2. aktualizované a rozšířené vyd. Praha: Grada, 570 s. Knihovna programátora (Grada). ISBN 978-80-247-2653-3.

PECINOVSKÝ, Rudolf. (2010). *Metodika Design Patterns First* [online]. Nové město na Moravě, 6 s. [cit. 2016-06-13]. Příspěvek na konferenci Počítač ve škole 2010. Dostupné z: <http://www.vyuka.pecinovsky.cz/prispevky/2010_PC_Metodika_Design_Patterns_First.pdf>

PECINOVSKÝ, Rudolf. (2011). *Tvorba učebnic a kurzů programování* [online]. Žilina, 7 s. [cit. 2016-06-13]. Příspěvek na konferenci Objekty 2011. Dostupné z: <http://www.vyuka.pecinovsky.cz/prispevky/2011_OB_Tvorba%20u%C4%8Debnic%20programov%C3%A1n%C3%AD.pdf>

PECINOVSKÝ, Rudolf. (2012). *Java 7: učebnice objektové architektury pro začátečníky*. Praha: Grada, 496 s. Knihovna programátora (Grada). ISBN 978-80-247-3665-5.

PECINOVSKÝ, Rudolf. (2013). *OOP - learn object oriented thinking and programming*. Řepín: Tomáš Bruckner, 527 s. Academic series. ISBN 978-80-904661-8-0.

SAINES, George. 2014. *3 Reasons Why „Computational Literacy" Is Ruining Coding Education*. [online]. CodeCombat Blog. San Francisco (California) [cit. 2016-06-15]. Dostupné z: <http://blog.codecombat.com/3-reasons-why-computational-literacy-is-ruining-coding-education>

Scratch Wiki. (2016). *Saving Data* [online]. Massachusetts: Massachusetts Institute of Technology, 2016-06-13 [cit. 2016-03-15]. Dostupné z: <https://wiki.scratch.mit.edu/wiki/Saving_Data>

SKUTIL, Martin. (2011). *Základy pedagogicko-psychologického výzkumu pro studenty učitelství*. Praha: Portál, 256 s. ISBN 978-80-7367-778-7.

THOMPSON, Clive. (2011). *How Khan Academy Is Changing the Rules of Education*. [online]. Wired. California: Condé Nast Digital [cit. 2016-06-12]. Dostupné z: <http://www.wired.com/2011/07/ff_khan/>

Turtle Academy. (2016). *LOGO Lessons* [online]. Copyright © 2011-2016. [cit. 2016-03-26]. Dostupné z: <https://turtleacademy.com/lessons/en>

VANÍČEK, Jiří. (2012). *Informatika pro 1. stupeň základní školy: informační a komunikační technologie*. V Brně: Computer Press. ISBN 978-80-251-3749-9.

VANÍČEK, Jiří. (2016). *Výuka algoritmizace patří především do informatiky*. České Budějovice: Jihočeská univerzita, 5 s. [cit. 2016-06-27]. Příspěvek na konferenci Počítač ve škole 2016 v Novém Městě na Moravě. Dostupné z: <http://www.pocitacveskole.cz/system/files/soubory/sbornik/2016/vanicek1.pdf>

VIRIUS, Miroslav. (1995). *Základy algoritmizace*. Praha: ČVUT, 179 s. Scriptum. ISBN 80-01-01346-4.

w3schools.com. (2016). *JavaScript Tutorial*. [online]. Refsnes Data, c1999-2016 [cit. 2016-06-15]. Dostupné z: <http://www.w3schools.com/js/default.asp>

WATTERS, Audrey. (2011). *The Wrath Against Khan: Why Some Educators Are Questioning Khan Academy*. [online]. Hack Education: The History of the Future of Education Technology. California [cit. 2016-06-13]. Dostupné z: <http://hackeducation.com/2011/07/19/the-wrath-against-khan-why-some-educators-are-questioning-khan-academy>

WESSA, Patrick. (2012). *Kendall tau Rank Correlation (v1.0.11) in Free Statistics Software (v1.1.23-r7)* [online]. Office for Research Development and Education. [cit. 2016-06-02]. Dostupné z: <http://www.wessa.net/rwasp_kendall.wasp/>

SEZNAM POUŽITÝCH OBRÁZKŮ

SEZNAM POUŽITÝCH TABULEK A GRAFŮ

P Ř Í L O H Y

A. Příprava na experimentální dvouhodinovku HoC

Příprava – Hour of Code a Lightbot

Škola - výchovné zařízení: ----------			**Třídy:** šesté až deváté
Školní rok: 2014/2015	**Předmět:** INF	**Vyuč.hodina:** ----------	**Datum:** červen 2015
Učivo: Projekt Hour of Code a Lightbot (1. část výzkumu)			**Poznámky:**
Univerzita: **Hradec Králové** **Fakulta:** **Pedagogická**			
Údaje o vyučujícím praktikantovi: Bc. Tomáš Hornik		**Ročník:** 1. (navazující)	**Aprobace:** AJ, INF
Pedagogický vedoucí praxe: PhDr. Michal Musílek, Ph.D.			
Cíle hodiny: Žáci definují co to je programování. Žáci využívají cykly a podmínky. Žáci vytvářejí vlastní procesy (ti kteří stihnou Lightbota)			
Pomůcky a podklady: soubor: *1. Vstupní dotazník (číslo 1).bat* (+ dotazník Google Forms) soubor: *2. Hour of Code - Bludiště.bat* soubor: *3. Lightbot.bat* soubor: *4. Výstupní dotazník (číslo 2).bat* (+ dotazník Google Forms) soubor: *5. Hour of Code – Celý kurz.txt*			

Čas	Fáze	Činnost učitele	Činnost žáka
3	Motivace	Vyučující se představí a povrchně nastíní žákům obsah této dvouhodiny, která má čtyři části: vstupní dotazník, Hour of Code, Lightbota a výstupní dotazník (o tomto druhém dotazníku se ale žákům nezmíní)	Žáci poslouchají.
15	Dotazník	Vyučující nechá žáky vylosovat si ze sáčku nastříhaná náhodná čísla, podá žákům stručné vysvětlení vyplňování dotazníku a teprve poté nasdílí *.bat* soubor s odkazen na vstupní dotazník. Poté kontroluje postup práce žáků ze zadní řady a pouze v případě potřeby a na žádost žáků vysvětlí případné nejasnosti.	Žáci si vylosují čísla, zapnou si pomocí nasdíleného *.bat* souboru vstupní dotazník a v klidu si ho vyplní. V případě jakéhokoliv dotazu se přihlásí a zeptají. V případě dřívějšího dokončení dotazníku si mohou v tichosti dělat co chtějí, než dokončí i ostatní.
5	Motivace Expozice	Vyučující se zeptá žáků „Jsou počítače chytré?“ (odpověď je, že NE, chytří musejí být programátoři!) Pro ilustraci podá příklad s robotem. Poté vysvětlí k čemu je programování a co to vlastně je, přičemž zdůrazní, že se technicky vzato jedná **tvorbu nových programů pomocí sledu instrukcí** od programátora pro počítač, který má podle nich něco přesně nějakým způsobem provést (poznámka 1 a 2)	Žáci poslouchají a případně odpovídají na položené dotazy.
2	Motivace	Vyučující vysvětlí, že robotika a programování nejsou zase tak složité a oznámí, že základy si dnes žáci vyzkouší hrou.	Žáci poslouchají, případně se ptají na cokoliv by jim bylo nejasného.
45	Expozice	Vyučující rozešle žákům .bat soubor s odkazem na Hour of Code, upozorní žáky na nutnost dávat při videích pozor a číst si instrukce a nechá žáky pracovat vlastním tempem až téměř do konce dvouhodinovky. Ze zadní řady kontroluje průběh činnosti žáků a pomáhá při problémech či zásecích (poznámka 3 a 4)	Žáci vlastním tempem procházejí Hour of Code, mohou v tichosti spolupracovat a radit se (ale nesmí tím nerušit ostatní!!!)

--	Expozice / Fixace	Vyučující zkontroluje práci hotových žáků, pochválí je a zadá jim pokračování ve formě dalšího tutoriálu, tentokrát ve formě hry Lightbot (opět skrze nasdílený *.bat* soubor).	Ve chvíli, kdy některý ze žáků dokončí celý kurz Hour of Code, přihlásí se a čeká na další instrukce. Poté až do konce pracuje na postupu v Lightbotovi.
5	Motivace	Vyučující ukáže žákům všechny tři předpřipravená videa z YouTube s modely LEGO Mindstorms a vysvětlí jim, že se s nimi klidně mohou setkat třeba na střední škole nebo v nějakém kroužku. Poté vyučující nasdílí soubor Hour of Code – Celý kurz.txt, ve kterém mohou zájemci najít veškeré dokazy použité v této hodině a sami v tomto programování pokračovat. Na závěr vyučující pozve žáky na akci Hrajme si i hlavou, kde mohou LEGO roboty vidět na živo.	Žáci sledují videa a mohou pokládat doplňující dotazy. Zájemci si stáhnou nasdílený txt soubor a uloží si ho buď na email nebo na flashku.
15	Dotazník	Vyučující rozešle žákům .bat soubor s odkazem na druhý dotazník. Nechá žáky až do konce hodiny dotazník vyplňovat a sám opět asistuje ze zadní řady pouze v případě problému.	Žáci vyplňují druhý dotazník, do kterého zadají místo jména to samé číslo, které zadávali do dotazníku prvního. Po skončení dotazníku mají žáci možnost pracovat na čemkoliv chtějí, či si jen tak hrát, ale nesmí rušit ostatní!

Poznámky k realizaci:

1.) Pro ilustraci lze využít příkladu, kdy si mají žáci představit robota a třeba štěňátko někde na vysoké skále. Oba dva dostanou instrukci „*Jdi*“, co se stane? Žáci většinou správně uhodnou, že robot se rozbije a štěňátko na ně bude koukat jak na hlupáky. Vyučující musí dodat vysvětlení – roboti dělají jen a pouze co jim programátor řekne, a to včetně naprosté pitomosti. Je tedy úkolem programátora chování robota naprogramovat. Potom lze (ale nemusí se) pokračovat příkladem, kdy vyučující vyvolá jednoho žáka či žákyni někde z prostředku třídy a řekne jí/jemu „*Představ si, že jsi robot a já programátor a dávám ti jedinou jednoduchou instrukci – dojdi k tabuli.“* Žák či žákyně tak zpravidla učiní, na čemž jde demonstrovat jak by to vypadalo ve skutečnosti. Jestliže by robot byl humanoidní a seděl, nejprve by musel dostat instrukci vstát, potom otočit se směrem do uličky a tak dál až by se dostal k tabuli. Kdyby to byl velký silný robot a dostal jen instrukci jdi k tabuli, prostě by se rozjel a skrz počítače a lavice se k té tabuli proboural. Zde žáci zpravidla pochopí nutnost přesných instrukcí a jejich rozdrobení do dílčích kroků.

2.) Vyučující v rámci vysvětlování co to vlastně to programování je a k čemu je to dobré řekne, že *„informatiku použijete v případech, o které se zajímají chlapci i dívky, jako je zachraňovat životy, pomáhat lidem, spojovat lidi atd. Přemýšlejte o věcech každodenního života, které používají informatiku: mobil, mikrovlnná trouba, počítač, semafory ... všechna tato zařízení potřebují informatiky, aby je pomohli vymyslet a sestrojit. Informatika je umění smíchat lidské myšlenky s digitálními nástroji pro zlepšení našich schopností a sil. Informatici pracují v mnoha rozdílných oblastech: píšou aplikace pro smartphony, léčí nemocné, vytvářejí animované filmy, pracují v sociálních médiích, konstruují roboty, kteří zkoumají jiné planety a ještě daleko více.“* (převzato z: https://hourofcode.com/cz/cs/resources/how-to) Vyučující by ale měl zdůraznit, že podstatou programování jako takového je vytváření nových programů pomocí sledu instrukcí od programátora pro počítač, podle kterých dané zařízení něco přesně nějakým způsobem provede.

3.) Na začátku Hour of Code je nutné zkontrolovat, že všichni žáci mají kurz v češtině! Jestliže ne, je třeba jazyk přepnout dole uprostřed. Dále je potřeba sledovat průběh hodiny a počítat s nechutí žáků číst cokoliv a tedy i odklikávat videa, apod. Je tedy nutné žákům, kteří se zaseknou v určitých částech kurzu vysvětlit, o co vlastně jde. Těmito kritickými částmi se ukázál být princip fungování bloků opakování a podmínky. Ukázalo se být dostačujícím zopakovat žákům jen vysvětlení, které bylo i ve videích, s ukázkou na aktuální úloze.

4.) Žáci si mohou v rámci Hour of Code a Lightbota navzájem radit a spolupracovat spolu, jen musí pracovat relativně potichu, ay nerušili ostatní. Pravidlem ale je, že si navzájem nesmějí sahat na myš, mohou si tedy radit jen slovně. Situace, kdy jeden žák řekne druhému *„Hele já ti ukážu jak se to dělá“*, vezme mu myš a na jeho počítači danou úroveň vyřeší je naprosto nepřínosná, proto toto pravidlo. Jeden žák může druhého maximálně slovně instruovat co má dělat, ale je dobré žákům vysvětlit, že když už si budou takto pomáhat, měli by v rámci těchto instrukcí také podávat vysvětlení proč tomu tak je a jak to vlastně funguje.

B. Zadání vstupního dotazníku pro žáky

Výzkum programování na ZŠ z pohledu žáků

Toto NENÍ test! A díky náhodně vybranému číslu je dotazník anonymní, přesto ho nepodceňuj a skutečně se snaž ho vyplnit co nejlépe.

*Povinné pole

Tvůj vylosovaný číselný kód *

Pohlaví *

- Chlapec
- Dívka

Věk *

Třída *

- 6. třída
- 7. třída
- 8. třída
- 9. třída

Vyber na stupnici od 1 do 5 jak moc tě baví hodiny infomatiky *

	1	2	3	4	5	
Ani trochu	●	●	●	●	●	Strašně moc

Vyber na stupnici od 1 do 5 za jak důležitou považuješ informatiku obecně (jak moc je podle tebe užitečná?) *

	1	2	3	4	5	
Nedůležitá, je to k ničemu	●	●	●	●	●	Velice důležitá a užitečná

Co si představíš, když se řekne "programování" ? *

Dosavadní zkušenosti *

Vyber, jaké jsou tvé zkušenosti s programováním

- Nikdy jsem to nezkoušel/a a ani nevím, co to je
- Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a
- Zkoušel/a jsem si jednoduché grafické programování, ale nebavilo mě to
- Zkoušel/a jsem si jednoduché grafické programování a bavilo mě to
- Zkoušel/a jsem klasické programování v nějakém programovacím jazyce
- Ovládám základy nějakého programovacího jazyka
- Je to můj koníček, programuji běžně a rád/a
- Jiné: [redacted]

Jestliže už jsi se setkal/a s programováním, odkud?

Popiš mi prosím stručně kdy a jak jsi se s programováním setkal/a (od sourozence, od kamaráda, sám/sama na internetu, apod.)

[redacted]

Chtěl/a by sis vyzkoušet programování o hodinách informatiky? *

- Ano
- Ne
- Nevím, co to programování je, takže nevím
- Jiné: [redacted]

Zajímáš se ještě o něco ze světa informatiky a počítačů?

Ze seznamu níže vyber vše, co tě baví, o co se zajímáš nebo alespoň co často používáš. Jestliže v seznamu něco chybí, dopiš to do posledního volného políčka.

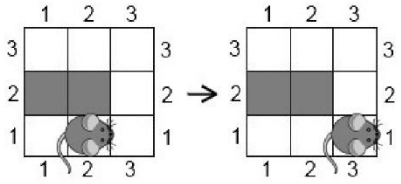
- ■ Počítačové hry
- ■ Facebook
- ■ Blogy
- ■ Nějaké konkrétní internetové stránky
- ■ Tvorba webových stránek
- ■ 2D Grafika (fotky, a další ploché obrázky)
- ■ 3D Grafika
- ■ Hardware (počítače po technické stránce)
- ■ Multimédia (muzika a filmy)
- ■ Počítačové sítě
- ■ Tabulkový procesor (MS Excel)
- ■ Textový editor (MS Word)
- ■ Prezentace (PowerPoint)
- ■ Linux
- ■ Android
- ■ Jiné: [redacted]

Pokračovat »

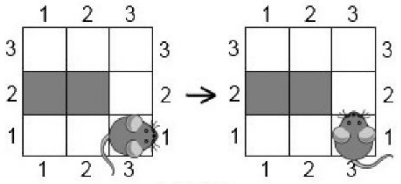
Výzkum programování na ZŠ z pohledu žáků

*Povinné pole

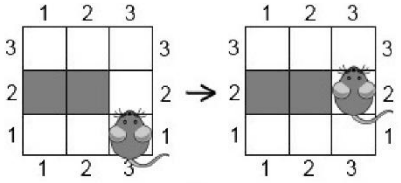
Pohyb myšky (vysvětlení pro následující otázku)



VPŘED



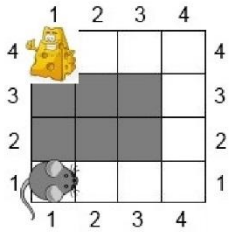
VLEVO



VPŘED

Popiš myšce cestu k sýru *

Myška chápe příkazy VPŘED (popojde o jedno pole dopředu, tedy tam, kam ji zrovna míří čumáček), VLEVO (myška se na místě otočí o 90° vlevo) a VPRAVO (myška se na místě otočí o 90° vpravo). Dej myšce na obrázku dole instrukce, podle kterých se dostat k sýru. Šedivá pole jsou beton, kterým se myška neprokouše. Pokus se vymyslet nějaký fígl, jak zápis cesty zkrátit. Jestliže na žádnou fintu nepřijdeš, tak nevadí, ale napiš alespoň nejzákladnější řešení.



Jak přejít silnici a nenechat se zajet?

Představ si, že máš někomu vysvětlit jak se přechází silnice a ten někdo vůbec netuší, jak se to dělá. Co přesně bys takovému člověku řekl/a? Zkus být co nejstručnější, ale nenech ho zajet.

« Zpět Odeslat

Nikdy přes Formuláře Google neposílejte hesla.

Používá technologii Google Forms

Obsah není vytvořen ani schválen Googlem.

Nahlásit zneužití - Smluvní podmínky služby - Další smluvní podmínky

C. Zadání výstupního dotazníku pro žáky

Výzkum programování na ZŠ z pohledu žáků

Opět se nejedná o TEST, ale tentokrát o výstupní závěrečný dotazník. A díky náhodně vybranému číslu je dotazník stále anonymní, přesto ho nepodceňuj a skutečně se snaž ho vyplnit co nejlépe.

*Povinné pole

Tvůj vylosovaný číselný kód *

Pohlaví *

- Chlapec
- Dívka

Věk *

Třída *

- 6. třída
- 7. třída
- 8. třída
- 9. třída

Vyber na stupnici od 1 do 5 jak moc tě bavila tato hodina *

(nepočítej do toho ale dotazníky, pouze Hour of Code a případně Lightbota)

	1	2	3	4	5	
Ani trochu	●	●	●	●	●	Strašně moc

Jak bys někomu vysvětlil/a co to je "programování" teď, po této hodině? *

Chtěl/a by jsi pokračovat v takovýchto hodinách o informatice i nadále? *

- Ano
- Ne
- Jiné: [redacted]

Stihnul/stihla jsi projít celý Hour of Code? *

Hour of Code má na konci certifikát a skládá se ze 20 příkladů. Jestliže ne, vyber u jakého příkladu jsi skončil/a.

[redacted]

Kolik jsi toho stihl/a z Lightbota? *

Vyber ze seznamu u jaké části jsou skončil/a.

[redacted]

Kroužek programování *

Kdyby byla možnost vybrat si kroužek programování, ve kterém by se pracovalo například ve Scratchi (podobné ovládání jako Hour of Code, akorát se neřeší se hádanky, ale lze vytvořit téměř cokoliv podle sebe, například hry, scénky, apod.), vybral/a by sis tento kroužek? Měl/a by jsi zájem ho navštěvovat?

- Ano
- Ne
- Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu
- Jiné: [redacted]

Kroužek robotického programování LEGO Mindstorms *

Kdyby byla možnost vybrat si kroužek robotického programování, kde by se pracovalo s LEGO Mindstorms roboty (jako byl třeba ten krokodýl na videu), vybral/a by sis tento kroužek? Měl/a by jsi zájem ho navštěvovat?

- Ano
- Ne
- Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu
- Jiné: [redacted]

Připomínky

Sem prosím napiš jakékoliv připomínky k proběhlé hodině. Pamatuj, že dotazník je anonymní, můžeš tedy napsat téměř cokoliv. Nezapomínej ale na to, že vše se vždycky dá říct slušně!

[redacted]

Pokračovat »

Používá technologii Google Forms

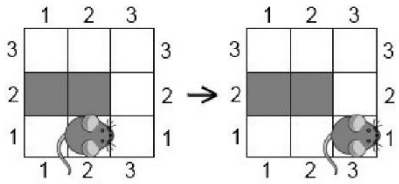
Obsah není vytvořen ani schválen Googlem.

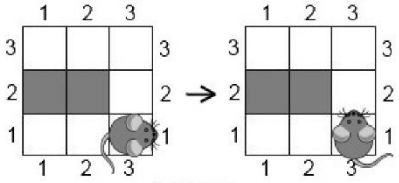
Nahlásit zneužití - Smluvní podmínky služby - Další smluvní podmínky

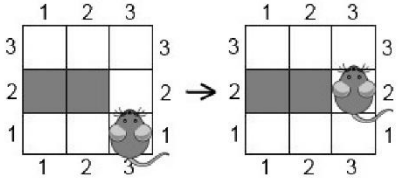
Výzkum programování na ZŠ z pohledu žáků

*Povinné pole

Pohyb myšky (vysvětlení pro následující otázku)

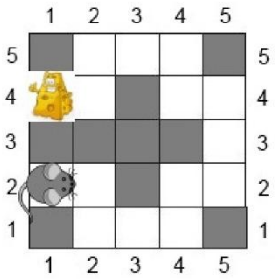






Popiš myšce cestu k sýru *

Myška chápe příkazy VPŘED (popojde o jedno pole dopředu, tedy tam, kam ji zrovna míří čumáček), VLEVO (myška se na místě otočí o 90° vlevo) a VPRAVO (myška se na místě otočí o 90° vpravo). Dej myšce na obrázku dole instrukce, podle kterých se dostat k sýru. Šedivá pole jsou beton, kterým se myška neprokouše. Pokus se vymyslet nějaký fígl, jak zápis cesty zkrátit. Sice tady nejsou bloky jako v Hour of Code, ale můžeš využít závorek a násobení nebo dokonce procesů jako v Lightbotovi. Jestliže na žádnou fintu nepřijdeš, tak nevadí, ale napiš alespoň nejzákladnější řešení.



Jak přejít silnici a nenechat se zajet?

Otázka je stejná jako v minulém dotazníku. Jestliže jsi se svojí minulou odpovědí naprosto spokojen/á, napiš sem znovu to samé. Pokus se ale využít toho, co jsi dnes o této hodině viděl/a. Představ si, že ten, komu cestu vysvětluješ vezme tvou odpověď skutečně doslova. Co přesně bys takovému člověku řekl/a? Zkus být co nejstručnější, ale nenech ho zajet.

« Zpět Odeslat

Nikdy přes Formuláře Google neposílejte hesla.

Používá technologii Google Forms

Obsah není vytvořen ani schválen Googlem.

Nahlásit zneužití - Smluvní podmínky služby - Další smluvní podmínky

D. Náhodně generovaná čísla pro žáky

Pozn.: Čísla byla generována pomocí online nástroje, který je k nalezení na adrese https://www.random.org/sequences/ . Zde byla jako nejnižší hodnota nastaveno 1000 a jako nejvyšší 9999 (takže všechna čísla budou mít čtyři cifry). Poté bylo prvních 480 čísel zkopírováno do MS Word a naformátováno následovně: font Comic Sans MS, velikost fontu 24, tučně, řádkování jednoduché. Tabulka níže je přesnou kopií tabulky s hodnotami použitými při tomto výzkumu (jen s menším fontem) a poskytnuta je za účelem zjednodušení případné snahy o reprodukci celého výzkumu.

3588	2673	8304	5663	3792
7225	6947	8044	7274	9914
7602	4005	7129	1541	4542
1661	9409	7914	6527	5968
6896	9508	1249	6993	8114
5945	2435	2498	8567	9458
4049	9954	4852	6641	8818
1985	5371	9795	9814	8865
2892	3924	4914	5414	7670
2769	1991	7117	1208	6945
9532	7793	1029	5517	7603
9655	6600	2487	9119	7614
8377	7523	4680	1712	1223
4554	3250	5791	2651	9424
7328	3673	4873	9461	6521
5521	1369	7011	7987	5283
9101	4864	5479	8738	1597
2860	1618	1538	8805	4300
3652	8952	7446	8118	6827
9219	7997	7752	6475	9113
1500	6585	6465	9579	2561
5299	3811	9626	5169	2396
4441	5243	7380	3685	2850
1978	4095	8639	8326	6870
5246	4971	7324	2194	4373
5331	9578	9227	9390	2148

7415	3683	5684	7968	7539
9330	8542	1120	5301	2772
3637	1874	2228	1819	9302
4434	7000	3762	9634	9917
6959	5260	1912	3297	3330
1949	2300	8107	8747	8577
3902	9538	2301	5007	7834
8390	4843	9322	9575	5324
3120	5505	8543	1403	2706
6347	6986	4825	2551	4872
8656	7448	3966	7570	7374
4432	2578	7721	7412	9259
5294	3642	5485	7610	8733
3520	6192	5764	3522	3845
6078	2025	5668	4629	2515
8545	7243	6084	6049	7537
1629	1728	5874	1101	8288
6957	9190	7474	4205	6586
2108	3200	1967	4653	3725
8364	1641	5093	9321	6991
8982	4116	2609	2032	8643
7724	2486	4415	8735	3497
7250	2493	9005	8555	2385
9830	1301	4377	2304	1802
7420	1025	8816	7920	1176
7259	3770	4772	2464	6845
9885	6300	7312	2747	5855
3290	7830	4624	3369	1241
7480	4397	1507	9881	9454
1975	7093	4497	1273	7782
8684	1575	1639	1451	8561
6415	8436	4895	9929	3015
3453	8009	2469	5756	7469
6485	8370	1043	3448	1987
2454	3755	8331	7564	5991

6908	7251	8312	6596	2682
9351	3230	4029	4582	2817
8689	9300	2722	2599	1623
7852	8242	8238	2176	3166
9452	6431	2306	3805	3884
4280	2659	2314	6802	6925
2776	6667	9537	4036	6685
2737	8000	2820	4576	3392
9200	9250	9555	2758	3209
2736	3700	7885	5368	8184
2671	5195	3132	6789	5248
1145	2998	4315	7099	9333
8177	5632	4399	8759	1415
2041	3001	6427	4673	8225
2811	2816	8953	1512	4464
6978	9500	6948	3760	8959
2896	5784	6412	2321	2102
4075	3207	1884	4442	3556
2297	7671	8560	5841	5878
2585	1924	9536	2111	3293
7860	3437	5527	4584	2075
8769	2169	5038	8101	4734
1532	9503	3019	1702	6218
2825	6684	8894	4190	8591
6228	4572	5145	1751	6028
8872	9666	3616	9843	4394
2647	4925	8566	2794	5635
4105	5962	2201	2317	4012
8202	7821	9244	7723	4353
6151	3714	5703	2404	1588
9769	8742	4009	5114	3784
6653	5554	4015	4933	7012
1945	8248	6104	6568	9501
4702	2495	8503	6216	3912
5619	2236	6187	2988	5869

E. Soubor „Hour of Code – Celý kurz“

Tento soubor v elementárním formátu .txt s Unicode kódováním byl žákům nasdílen v závěru první dvouhodinovky poté co si žáci prošli celý Hour of Code a někteří i část Lightbota. Jestliže žáky obsah hodiny bavil, jednalo se takřka o jedinou možnost jak jim dovolit pokračovat v rozvíjení tohoto zájmu samostatně a bez vedení učitele. Kopírování sdíleného souboru nebylo nijak registrováno (aby si žáci nemysleli, že by tím mohli cokoliv získat ze strany učitele a byl tak zajištěn skutečný zájem žáka), dle odhadu si tento soubor stáhlo zhruba 40% všech žáků.

Video LEGO Mindstorms **RoboGator**
https://www.youtube.com/watch?v=Pc9XSSfmjsQ

Video LEGO Mindstorms **Rubikova kostka**
https://youtu.be/staapsj3eRQ?t=158

Video LEGO Mindstorms **3D Tiskárna**
https://youtu.be/oFOpMILT7_Y?t=35

Hour of Code - Bludiště
http://studio.code.org/s/20-hour/stage/2/puzzle/1

Celý základní kurz Hour of Code
(další úrovně hry jako bylo bludiště jsou všechny ty části, které mají vpravo kolečka s čísly. I když budete přeskakovat „Aktivity bez počítače“, úrovně procházejte postupně, tak jak tam jsou za sebou, a nevynechávejte videa!)
http://studio.code.org/s/20-hour

Lightbot
(na češtinu si jazyk změníte na úvodní obrazovce vlaječkou vpravo nahoře, můžete si také na stejné úvodní obrazovce změnit šedého robota na růžovou robotku)
http://lightbot.com/hocflash.html
Pozn.: zájemci si mohou na GooglePlay koupit plnou verzi s větším počtem úrovní

Akce **Hrajme si i Hlavou**
(koná se v Hradci Králové na nábřeží u Náměstí Svobody 18. a 19.6.2015 zde můžete vidět ukázku LEGO Mindstorms robotů na živo, všichni jsou vítáni)
http://www.hrajme-si-i-hlavou.cz/

Scratch
(zde si můžete vytvořit cokoliv chcete a internet je plný návodů)
https://scratch.mit.edu/projects/editor/?tip_bar=getStarted

F. Výsledky vstupního dotazníku pilotních skupin

Číselný kód	Pohlaví	Věk	Třída	Oblíbá INF	Význam INF	Představa "programování"	Dosavadní zkušenosti	Zájem o vyzkoušení programování v hodinách IVT	Další zájmy z oblasti IVT a počítačů	Popiš myšlce cestu k sýru	Jak prohodit dva bonbony?	Jak přejít silnici a nenechat se zajet?
7259	Dívka	14	8. třída	2	2	nevím	Nikdy jsem to nezkoušel/a a ani nevím, co to je	Nevím, co to programování je, takže nevím	Facebook, Multimédia (muzika a filmy), Android	o 3 pole vpřed, vlevo, o 3 pole vpřed, vlevo a zase o 3 pole vpřed	jeden bonbon si položím druhý dám do první ruky a ten první bonbon vezmu druhou rukou	musí počkat až nepojedou žádná auta a pak přejít
2314	Dívka	14	8. třída	2	2	nevím	Nikdy jsem to nezkoušel/a a ani nevím, co to je	Ano	Nějaké konkrétní internetové stránky, Multimédia (muzika a filmy), Android	o 3 pole vpřed,před o 3 pole,vlevo o 3 pole.	Položila bych jeden na zem	musíš počkat a rozhlídnout se do prava a do leva,počkat si a přejít.
2722	Dívka	14	8. třída	4	3	nějakou techniku	Nikdy jsem to nezkoušel/a a ani nevím, co to je	Ano	Počítačové hry, Facebook, Blogy, 2D Grafika (fotky, a další plochè obrázky), Multimédia (muzika a filmy), Tabulkový procesor (MS Excel), Textový editor (MS Word), Prezentace (PowerPoint), Android	vpřed, vpřed, vlevo, vpřed, vpřed, vpřed, vlevo, vpřed, vpřed, vpřed	vyhodila je do vzduchu a chytla opačně	počkat až nepojedou žádná auta a pak přejít po přechodu
9578	Dívka	14	8. třída	3	2	Nevím co to je.	Nikdy jsem to nezkoušel/a a ani nevím, co to je	Nevím, co to programování je, takže nevím	Blogy, Nějaké konkrétní internetové stránky	Vpřed na 4, políčko, otočit vlevo, vpřed na 4, políčko, otočit vlevo, rovně až k sýru.	Výhodím je do vzduchu a chytnu je, ale tak, abych e měla barvou prohozené.	Pujde na přechod, tam počká až všechna auta zastaví a přejde silnici.
3924	Dívka	15	8. třída	3	3	donutit počítač, aby dělal to, co chci aby dělal	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	je mi to jedno	Facebook, Nějaké konkrétní internetové stránky, Multimédia (muzika a filmy), Prezentace (PowerPoint)	3xvpřed 3x nalevo 3xdo leva	Položím je na stůl nebo jinou plochu , a pak si je vezmu, tak jak jsem chtěla .	Podívaš se do leva a pak do prava ,když nic nejede tak můžeš jít .
8894	Dívka	15	9. třída	3	2	co já vim :DDDDD	Zkoušel/a jsem si jednoduché grafické programování, ale nebavilo mě to	Ne	Facebook	vpřed vlevo vpřed vlevo vpřed	přehodím si je ve stejnou chvíly :DDDDD	stojím a rozhlížím se na levou a pravou stranu a pak teprve jdu :D (LOL)
1538	Dívka	15	9. třída	2	5	Nastavení počítačů.	Nikdy jsem to nezkoušel/a a ani nevím, co to je	Ano	Nějaké konkrétní internetové stránky	3 vpřed, vlevo, 3 vpřed, vlevo, 3 vpřed	Hodila oba dva najednou a chytila je.	Přez přechod, když nejede auto.
5527	Dívka	15	9. třída	1	3	Nevím, co to je	Nikdy jsem to nezkoušel/a a ani nevím, co to je	Nevím, co to programování je, takže nevím	Facebook, Blogy, Nějaké konkrétní internetové stránky, Tvorba webových stránek, Hardware (počítače po technické stránce), Android	Vpřed, vlevo , vpřed,vlevo , vpřed	Ve stejné chvíli přehodit	Zastavit se před silnicí, podívat se vlevo,vpravo a znovu vlevo, pokud jede nějaké auto, počkáme až přejede. Když jsme si jistí ,že nic nejede, rychle přejdeme silnici Pokud jsou tam semafory, počkame na zeleného panáčka
3616	Dívka	15	9. třída	3	3	práce s programem	Zkoušel/a jsem si jednoduché grafické programování, ale nebavilo mě to	Ano	Facebook, Prezentace (PowerPoint), Android	Vpřed,vpřed,vpřed,vpravo,vpřed,vpřed,vpřed,vlevo,vpřed,vpřed,vpřed-cil	Jeden bych si položila	Řekla bych mu, že se přechází pouze na zeleno a že se musí rozhlídnout
9503	Chlapec	15	9. třída	3	3	Vytváření počítačových programů	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	Ne	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, Multimédia (muzika a filmy), Textový editor (MS Word), Prezentace (PowerPoint), Android	vpřed, vlevo, vpřed, vpravo, vpřed	žonglování	stůj, podívej se doleva a doprava. Rychle jdi
2609	Chlapec	15	9. třída	4	5	Práce s programama	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	Ne	Počítačové hry, Facebook, Multimédia (muzika a filmy), Počítačové sítě, Android	vpřed,vlevo,vpřed	Hodil je vzduchem	Chodit na přechodu na zelenou a rozhlídnout se
8566	Dívka	16	9. třída	3	3	práce s programem	Zkoušel/a jsem si jednoduché grafické programování, ale nebavilo mě to	Ano	Facebook, Nějaké konkrétní internetové stránky	Vlevo, vyaskočit, vpřed, vpřed, seskočit dolu	přendám to opatrně 2 pstama	Koukni na obě strany, jestli nic nejede a a budemit volnou cestu na druhou stranu, můžes rychle přejít.
5145	Dívka	15	9. třída	3	3	informace o programu	Zkoušel/a jsem klasické programování v nějakém programovacím jazyce	Nevím, co to programování je, takže nevím	Facebook, Blogy, Nějaké konkrétní internetové stránky, Tvorba webových stránek, 2D Grafika (fotky, a další plochè obrázky), Multimédia (muzika a filmy), Počítačové sítě, Android	vpřed,vpřed,vlevo,vpřed,vpred,vlevo,vpřed	v každý ruce bych měla 2bonbony jeden by byl černy a druhý modrý	šla bych za nim,vzala bych si ho před silnici a vysvětlila mu,jak fungují semafory a použila ho o bezpečnosti...
2201	Dívka	14	9. třída	3	3	nevím	Nikdy jsem to nezkoušel/a a ani nevím, co to je	Nevím, co to programování je, takže nevím	Facebook, Multimédia (muzika a filmy)	vpřed vpřed vpřed vlevo vpřed vpřed vlevo vpřed vpřed vpřed	přehodím si je ve stejný čas	sledujte semafory a až bude zelená barva můžete přejít
1991	Dívka	15	9. třída	4	2	"naučení" počítačů,mobilů atd. jak fungovat	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	Ne	Počítačové hry, Facebook, Blogy, Nějaké konkrétní internetové stránky, 2D Grafika (fotky, a další plochè obrázky), 3D Grafika, Multimédia (muzika a filmy), Počítačové sítě, Prezentace (PowerPoint), Android	vpřed vpřed vpřed vlevo vpřed vpřed vpřed vlevo vpřed vpřed vpřed	oba je sním	ať se podhrabe
4105	Dívka	14	9. třída	3	2	kancelář plnou lidí za počítačem a na brazovkách jsou nějaké kódy (číselné i písmenkové)	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	Ano	Počítačové hry, Facebook, Blogy, Nějaké konkrétní internetové stránky, 2D Grafika (fotky, a další plochè obrázky), Multimédia (muzika a filmy), Tabulkový procesor (MS Excel), Textový editor (MS Word), Prezentace (PowerPoint), Android, Twitter	VPŘED VPŘED VLEVO VPŘED VPŘED VPŘED VLEVO VPŘED VPŘED VPŘED	žonglování	u silnice na chodníku se zastav a rozhlédni se nejdřív na levo pak na pravo, jestli nic nejede můžeš jít. jestli jsi v británii nebo austrálii... auta jezdí naopak a proto se podívej nejdřív na pravo a potom až na levo
6151	Dívka	15	9. třída	3	3	Že, se programují nějaké věci nebo přeprogramovají různé aplikace.	Nikdy jsem to nezkoušel/a a ani nevím, co to je	Nevím, co to programování je, takže nevím	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, 2D Grafika (fotky, a další plochè obrázky), Prezentace (PowerPoint)	Vpřed Vpřed Vlevo Vpřed Vpřed Vpřed Vlevo Vpřed Vpřed Vpřed	Žonglovat	Nejdřív se podívej do leva a pak do prava, když nic nejede jdi. V prostředku se z nova podívej do prava a potom přejdi. A nebo choď na zelenou barvu pro chodce.
4415	Chlapec	15	9. třída	3	2	že se na programuje nějaký disk nebo program	Nikdy jsem to nezkoušel/a a ani nevím, co to je	Ne	Počítačové hry, Facebook, 3D Grafika	takže nechápu myš popojít v předzase vpřed a zýase v před pak do prava a vpřed vpřed vpřed a do leva leva leva	dal bych si ruce zazádá a prohodil bych je tak aby to nikdo nepoznal	No já bych mu řek aise zastaví na oraj chodníku a několikrát se rozhlídne jestli náké auto nejede jestli je přechod takstejně musípočkataž nákè autpo zastaví
8370	Dívka	14	9. třída	4	5	nevím	Nikdy jsem to nezkoušel/a a ani nevím, co to je	Nevím, co to programování je, takže nevím	youtube	VPŘED VPŘED VPŘED	prohodím si je ve stejny cas	ja to liez neznam jak se to dela

G. Výsledky vstupního dotazníku

Číselný kód	Pohlaví	Věk	Třída	Obliba INF	Význam INF	Představa "programování"	Dosavadní zkušenosti	Jestliže už jsi se setkal/a s programováním, odkud?	Zájem o vyzkoušení programování v hodinách IVT	Další zájmy z oblasti IVT a počítačů	Popiš myšce cestu k synu	Jak přejít silnici a nenechat se zajet?
2998	Chlapec	11	6. třída	5	3	Že třeba něco upravíme uděláme nějakou hru	Zkoušel/a jsem si jednoduché grafické programování a bavilo mě to		Ano	Počítačové hry, Facebook, 3D Grafika, Android	vpřed, vpřed, vlevo, vpřed, vpřed, vlevo, vpřed, vpřed	Mul se ozlídnou do prava do leva a ještě 1 do prava a když nic nejede může jít
1945	Dívka	12	6. třída	4	3	nevím, co to je	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Nevím, co to programování je, takže nevím	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, 2D Grafika (fotky, a další plošné obrázky), 3D Grafika, Multimédia (muzika a filmy), Android	vpřed, vlevo,vpřed	vpřed, vpřed, vpřed, vlevo, vpřed, vpřed, vlevo, vpřed, vpřed, vpřed
4095	Dívka	13	6. třída	3	4	programování	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Nevím, co to programování je, takže nevím	Facebook	pjde nahoru otočí se do leva a pjde čumáčkem do předu a bde u sýru	řekla bych aby šel furt po silnici a nerozičel se
5619	Chlapec	12	6. třída	2	4	nevím co to je	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Nevím, co to programování je, takže nevím	Počítačové hry	dopředu o dvě pole pote nahoru o tri pole a pak do leva o tri pole	rozhlednì se do leva a pak do prava a pak zase doleva a predi
5243	Dívka	11	6. třída	2	3	instalování různých programů.	Něco málo jsem si o tom zjišťovál/a, ale nikdy jsem programovat nezkoušel/a		Ne	Facebook	dopředu,dopředu,dopředu,vpravo,vpřed,vpřed,vpřed,vlevo,vpřed	Říct ať se pořádně rozhlídne , alepsoň 2krát za sebou a když nic nejede tak může přejít .
4572	Chlapec	12	6. třída	4	4	programace her	Něco málo jsem si o tom zjišťovál/a, ale nikdy jsem programovat nezkoušel/a		Ne	Počítačové hry, Nějaké konkrétní internetové stránky	vpřed,vpřed,vpřet,vlevo,vpřet,vpřet,vlevo,vpřet,vpřed,vpřed	nejdříve se koukni vlevo vpravo a zase vlevo když nic nejede tak můžeš jit ale když jede auto takpočkáme a pak až přeiede můžeš jít
1912	Chlapec	12	6. třída	4	3	neco se programuje stahuje jine verze?	Něco málo jsem si o tom zjišťovál/a, ale nikdy jsem programovat nezkoušel/a		Nevím, co to programování je, takže nevím	Počítačové hry, Facebook, Počítačové sítě, filmy	rovne doleva dopredu rovne doleva a rovne	no musis se koukout na levou a pravou stranu kdzy jede auto nechod az odjede musis přejid
3683	Chlapec	11	6. třída	4	2	naprogramovat robota nebo něco jiného	Něco málo jsem si o tom zjišťovál/a, ale nikdy jsem programovat nezkoušel/a		Ano	natáčet na youtube	VPŘED VLEVO VPŘED	vpřed vpřed vpřed vlevo vpřed vpřed vpřed vpřed vpřed vpřed
7380	Chlapec	12	6. třída	5	5	lidi co dělají stránku a přidávají odkazy	Ovládám základy nějakého programovacího jazyka		Ano	Počítačové hry, Facebook, Blogy, 2D Grafika (fotky, a další plošné obrázky), Hardware (počítače po technické stránce), Multimédia (muzika a filmy), Textový editor (MS Word), Prezentace (PowerPoint)	3x vpřed,otočit v levo,celé znovu,3x vpřed	přechází se přez přechod prvně se rozhlídnout (v levo v pravo a znovu)
3811	Dívka	12	6. třída	3	4	vytváření nove slozky nebo delat nějakou prezentaci	Něco málo jsem si o tom zjišťovál/a, ale nikdy jsem programovat nezkoušel/a		Ano	Facebook, Nějaké konkrétní internetové stránky, Android, youtube	vpred o 3 policka vlevo a opet o 3 policka vpred vpravo a opet o 3 policka vpřed myška je u sýru	prechazej silnici pouze na vyznacenych mistech pro prechod silnice jako je treba prechod po chodce,zmacni tlacitko na sloupu semaforu a cekej az i na druhem konci prechodu kde je druhy semafor objevi zeleny panacek, to i dava znamení ze muzes jit ale nezapomen se nadji tlacat rozhlednout doprava,doleva,doprava a potom svižne přejdi prechod
6986	Chlapec	12	6. třída	4	3	nastavování programu	Zkoušel/a jsem si jednoduché grafické programování a bavilo mě to		Ano	Počítačové hry, vytváření prezentace	levo levo vpred pravo vpred vpred vpred pravo	podivat do prava doleva do prava doleva a kdyz nic nejede muzes jit
4971	Dívka	12	6. třída	3	2	počítače stahování programu písniček filmu	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Nevím, co to programování je, takže nevím	Počítačové hry, Facebook, Multimédia (muzika a filmy)	pujde nahoru otočí se do leva čumáčkem na horu a pujde k syru	aby se rozhlédl do prava a do leva potom přešel svižnou chůzí
8542	Chlapec	12	6. třída	4	4	vytváření aplikací a her	Zkoušel/a jsem si jednoduché grafické programování a bavilo mě to		Ano	Počítačové hry, Facebook, Blogy, Nějaké konkrétní internetové stránky, Tvorba webových stránek, 2D Grafika (fotky, a další plošné obrázky), 3D Grafika, Multimédia (muzika a filmy), Počítačové sítě, Prezentace (PowerPoint), Android	vpřed vpřed vpřed vlevo vpřed vpřed vlevo vpřed vpřed vpřed	přijdeš k přechodu (bílé čáry na přechodu) rozhlédneš se nejdřív doleva pak doprava a pak zase do leva když nic nejede půjdeš rychlým krokem přes přechod
2228	Chlapec	12	6. třída	5	3	vytváření her a aplikací	Zkoušel/a jsem si jednoduché grafické programování a bavilo mě to		Ano	Facebook	Vpřed-Vpřed-Vpřed-Vlevo-Vpřed-Vpřed-Vlevo-Vpřed-Vpřed-Vpřed	Nejdříve se kouknete do prava a pak do leva a pak muzete přejít
4397	Dívka	12	6. třída	3	3	Dělání něco s programem	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Nevím, co to programování je, takže nevím	Facebook, Blogy, 2D Grafika (fotky, a další plošné obrázky), 3D Grafika, Multimédia (muzika a filmy)	vpřed,vpřed,vpřed,vlevo,vpřed,vpřed,vpřed,vlevo,vpřed,vpřed	Porozhlídni se do leva a o prava, jestli nic nejede tak přejdi a když je tam auto tak počkej až přejede nebo ti auto zastaví
8747	Dívka	11	6. třída	5	5	Předávat nové info. mě spíše zajímají jiné věci.	Něco málo jsem si o tom zjišťovál/a, ale nikdy jsem programovat nezkoušel/a		Ano	Blogy, 2D Grafika (fotky, a další plošné obrázky), Hardware (počítače po technické stránce), Prezentace (PowerPoint), Funkce počítače	Aby cesta byla kratší měla by se myš (na 3. řádek) myška jít vpravo a pak už jenom vpřed.	Tenčlověk musíynát základní pravidlo, podívat se vpravo a pak vlevo a když nic nopojede přejdeme.
3966	Chlapec	14	7. třída	3	3	viry	Zkoušel/a jsem si jednoduché grafické programování, ale nebavilo mě to		Ne	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, 2D Grafika (fotky, a další plošné obrázky), 3D Grafika, Hardware (počítače po technické stránce), Multimédia (muzika a filmy), Počítačové sítě, Tabulkový procesor (MS Excel), Textový editor (MS Word), Prezentace (PowerPoint), Linux, Android	VPŘED VPŘED VPŘED VLEVO VPŘED VPŘED VPŘED VLEVO VPŘED VPŘED VPŘED	čekat až nejede vozidlo a pak VPŘED!!!
5791	Chlapec	13	7. třída	3	2	Dělání z počítačem	Zkoušel/a jsem si jednoduché grafické programování a bavilo mě to		Ano	Počítačové hry, Facebook, Tvorba webových stránek, Android	3x vpred, 1xDoleva,3xRovne,1xDoleva,3xVpred	Nejdrive se rozlidni vpravo a vlevo pote jeste jednou a pak muzes prechazet.
3200	Chlapec	13	7. třída	2	3	Nevím	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Nevím, co to programování je, takže nevím	Počítačové hry, Facebook, 2D Grafika (fotky, a další plošné obrázky)	3x vpřed , vlevo , 3x vpřed , vlevo , 3x vpřed	Nejprve se před přechodem (bilima čárkama na silnici) zastavíš a pokud tam jsou semafory zmáčkneš tlačítko čekejte. Až se na semaforu rozsvítí zelená barva , tak se ještě ujistíš že nic nejede a můžeš pelit na druhou stranu
6600	Chlapec	13	7. třída	2	3	vytvoření nějakého programu	Něco málo jsem si o tom zjišťovál/a, ale nikdy jsem programovat nezkoušel/a		Nevím, co to programování je, takže nevím	Počítačové hry, Facebook	3x vpřed vlevo 3x vpřed vlevo vpřed	Musí se najít přechod a pořádně se rozhlídout a obě strany a když si bude jistej tak prostě přejde

Číselný kód	Pohlaví	Věk	Třída	Obliba INF	Význam INF	Představa "programování"	Dosavadní zkušenosti	Jestliže už jsi se setkal/a s programováním, odkud?	Zájem o vyzkoušení programování v hodinách IVT	Další zájmy z oblasti IVT a počítačů	Popiš myšce cestu k sýru	Jak přejít silnici a nenechat se zajet?
5764	Chlapec	13	7. třída	3	3	instalace nějakého programu do počítače.(např:spuštění)	Zkoušel/a jsem si jednoduché grafické programování a bavilo mě to		Ano	Počítačové hry, Facebook, 2D Grafika (fotky, a další ploché obrázky), 3D Grafika, Multimédia (muzika a filmy), Počítačové sítě, Android	vpřed,vpřed,vpřed,vlevo,vpřed,vpřed,vpřed,vlevo,vpřed,vpřed,vpřed	Měl by ses rozhlídnout a ta ky jít na přechod tam počkat až ti tam skočí zelenej panáček.
5668	Chlapec	13	7. třída	4	3	Instalace nějakého programu (např: Spouštění)	Zkoušel/a jsem si jednoduché grafické programování, ale nebavilo mě to		Ne	Počítačové hry, Facebook, 3D Grafika, Android	vpřed-vpřed-vpřed-vlevo-vpřed-vpřed-vpřed-vlevo-vpřed-vpřed-vpřed	Stačí se jen rozhlédnout VLEVO A VPRAVO a můžeš jít
4914	Chlapec	14	7. třída	1	4	naprogramování	Zkoušel/a jsem si jednoduché grafické programování, ale nebavilo mě to		Ne	Počítačové hry, Facebook	3x vpřed 1x vlevo 3x vpřed 1x vlevo 3x vpřed	rohlédmout se do leva doprava a když nejede žádné auto tak přejdeme.
3250	Chlapec	14	7. třída	4	3	nějakej program. vytváření programu.	Zkoušel/a jsem si jednoduché grafické programování a bavilo mě to		Ano	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, 2D Grafika (fotky, a další ploché obrázky), 3D Grafika, Multimédia (muzika a filmy), Prezentace (PowerPoint), Android	3x VPŘED , VLEVO , 2X VPŘED , VLEVO , 2x VPŘED pořád vpřed	aby se rozhlidnul a pak sel
8747	Chlapec	14	7. třída	3	3	Vytváření rúných programů, stránek	Zkoušel/a jsem si jednoduché grafické programování a bavilo mě to		Ano	Počítačové hry, Nějaké konkrétní internetové stránky, Tvorba webových stránek, Hardware (počítače po technické stránce), Multimédia (muzika a filmy), Linux, Android	VPŘED 3x VLEVO 1x VPŘED 3x VLEVO 1X VPŘED 3X	Rozhlédnout se jestli nejede auto z obou stran a když nic nejede tak muže přejít.
7752	Chlapec	13	7. třída	3	2	vytváření webových stránek, vytváření počítačových her	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Ne	Počítačové hry, Facebook, Multimédia (muzika a filmy)	3 VPŘED VLEVO 3* VPŘED VLEVO 3* VPŘED	nejlépe najít přechod, když stojíš před přechodem, tak se koukni doleva potom doprava a poté znovu doleva, když nic nejede můžeš přejít silnici. POZOR!!, sice na přechodu máš přednost před autem, ale vždy si počkej než zastaví
7721	Chlapec	14	7. třída	3	4	Zadání programovacího jazyka do určitého souboru.	Zkoušel/a jsem klasické programování v nějakém programovacím jazyce		Ano	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, Tvorba webových stránek, Hardware (počítače po technické stránce), Multimédia (muzika a filmy), Počítačové sítě, Linux, Android	o 3 pole v před,otočit v levo,o 3 pole v před,otočit v levo a o 3 pole v před nebo 3* o tři pole v před a dvakrát vlevo	dojdi na kraj silnice a rozhlídni se doleva a doprava,pokud auto nepojede přejdi,pokud pojede počkej
9769	Chlapec	13	7. třída	5	3	Že se něco naprogramuje nebo-li stáhne.	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a		Ano	Facebook, něco na tom schánět	1+1+1=vpřed,vlevo, 3.vpřed,vlevo,3.vpřed	přijdeš k přechodu, podíváš se do leva a do prava a když nic nepojede tak můžeš přejít nebo když tě někdo pustí
2647	Chlapec	12	7. třída	2	4	C++, Java, Dos Většinou si vybavím Linux, protože na něm sem se to trošku naučil, ale žádný master nejsem	Zkoušel/a jsem klasické programování v nějakém programovacím jazyce		Ne	Počítačové hry, Tvorba webových stránek, 3D Grafika, Hardware (počítače po technické stránce), Počítačové sítě, Linux	VPŘED /n 3 VLEVO VPŘED /n 3 VLEVO VPŘED /n 3	1)Podívej se v pravo i vlevo když nic nepojede; rozběhni se přímo na druhou stranu silnice a potom zastav když pojede; stůj na místě a opakuj příkaz 1
7117	Dívka	16	9. třída	4	3	Že něco se unstaluje	Zkoušel/a jsem si jednoduché grafické programování a bavilo mě to		Asi né je to zbytečné	Facebook, 3D Grafika, Android, youtube	Furt rovně Zahni směr nahoru a jsi u sýru	Rozhlídnout se a pak projít
1641	Chlapec	15	9. třída	3	5	programy	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a		Ne	Facebook	vlevo vlevo vlevo vpred vped nahoru dolu dopredu dozadu	
7968	Dívka	15	9. třída	5	5	Instalace nějakého programu. :-)	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Ano	Facebook, Blogy, Nějaké konkrétní internetové stránky, Multimédia (muzika a filmy), Počítačové sítě, Tabulkový procesor (MS Excel), Prezentace (PowerPoint), Android	Vlevo, vlevo, vlevo, vpřed, vpřed, před doprava doprava, doprava	Poivt se na levo a napravo a po té kdyby nic nejelo tak přejít nebo hednudue čekat na zelenou.
5301	Dívka	15	9. třída	4	4	nic	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a		Ne	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, Multimédia (muzika a filmy)	Rovně Pak dvakrát doleva	
2486	Chlapec	15	9. třída	3	3	nastavovaní určitého počítačového sistému	Zkoušel/a jsem si jednoduché grafické programování, ale nebavilo mě to		Ne	Počítačové hry, Facebook, Multimédia (muzika a filmy), Počítačové sítě, Android	vlevo,vpřed,vpřed	rozhlídnuse do leva a do prava a když nic nejede tak přejdu silnici
5784	Chlapec	15	9. třída	4	4	vytváření programů	Zkoušel/a jsem si jednoduché grafické programování, ale nebavilo mě to		Ano	Počítačové hry, Nějaké konkrétní internetové stránky, Hardware (počítače po technické stránce), Multimédia (muzika a filmy), Počítačové sítě, Android, windows	VLEVO, VLEVO, VPŘED, VPRAVO, VPŘED, VPŘED, VPŘED, VPRAVO, VPŘED.	Pokud tam nejedou žádná auta tak jednoduše přejdi. Pokud tam jezdí, tak jdi k přechodu, počkej až auta před přechodem zastaví, pak přejdi.
2495	Chlapec	14	9. třída	3	3	vytvořit plán něčeho důležitého	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Nevím, co to programování je, takže nevím	Facebook	vpře,vlevo,vpřed	
9536	Chlapec	15	9. třída	3	4	na programu jí se nake programy k tomuaby fungovali	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a		Nevím, co to programování je, takže nevím	Počítačové hry, Facebook	vpřed v levo v před v levo před	
8560	Chlapec	15	9. třída	4	5	Používání příkazových řádků .	Zkoušel/a jsem klasické programování v nějakém programovacím jazyce		Ano	Počítačové hry, Hardware (počítače po technické stránce), Multimédia (muzika a filmy), Prezentace (PowerPoint), Android	3* VPŘED VLEVO 3*VPŘED VLEVO 3*VPŘED	Rozhlédni se utíkej
3770	Chlapec	15	9. třída	4	5	Udávání příkazů pomocí HTML atd.	Zkoušel/a jsem si jednoduché grafické programování a bavilo mě to		Ano	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, Hardware (počítače po technické stránce), Prezentace (PowerPoint), Linux, Android	3xVPŘED 1xVLEVO 3xVPŘED 1xVLEVO 3xVPŘED	Musíš vždy počkat do tédoby,dokud ve všech jízdních pruzích nepojede žádné auto.
3437	Dívka	15	9. třída	4	5	složitý program který múže vytvářet programy	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a		Ano	Počítačové hry, Multimédia (muzika a filmy), Tabulkový procesor (MS Excel), Textový editor (MS Word), Prezentace (PowerPoint)	3vpřed,vlevo,3vpřed,vlevo,3vpřed vpřed, vpřed ,vpřed, vlevo, vpřed, vpřed, vpřed, vlevo, vpřed, vpřed, vpřed 3vpřed vlevo	stoupneš na kraj rozhlédneš se když vidíš auto sttojíš když ne přejdeš rovně na druhou stranu

Číselný kód	Pohlaví	Věk	Třída	Obliba INF	Význam INF	Představa "programování"	Dosavadní zkušenosti	Jestliže už jsi se setkal/a s programováním, odkud?	Zájem o vyzkoušení programování v hodinách IVT	Další zájmy z oblasti IVT a počítačů	Popiš myšce cestu k sýru	Jak přejít silnici a nenechat se zajet?
5632	Chlapec	15	9. třída	2	3	vubec nevím	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a		Nevím, co to programování je, takže nevím	Počítačové hry, Facebook	vpřed vlevo vpřed	aby se zozhlednul do prava a doleva a když nic nepojede tak může přejít
4009	Chlapec	15	9. třída	5	3	Že programu ju nějaký program.	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a		Ano	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, 2D Grafika (fotky, a další ploché obrázky), Multimédia (muzika a filmy), Počítačové sítě, Textový editor (MS Word), Prezentace (PowerPoint), Android	Tři políčka to předu. Potom se otočí do leva a popojde zas o tři políčka do přeedu. Pak zas se otočí do leva a popojde o ři řádky do předu a bude v cíli a dostane sýr. :)	Přijdeš na před přechod (to jsou bíle čáry na silnici) a počkáš než ti skočí zelený panáček na semaforu. (semafor je ta velká tič která ukazuje česvené světlo a zelené) Když se ti rosvítí zelený panáček tam budeš přejít silnici.
4015	Chlapec	16	9. třída	1	4	Práce se soubory.	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a		Ne	Počítačové hry, Facebook	3x VPŘED VLEVO 3x VPŘED VLEVO 3xVPŘED	Před přecodem slinice se musí rozhlédnout, nedřív do leva a pak do prava a znovu do leva.Pokud nic přez silnici nejede, můžete přeit.
4825	Chlapec	15	9. třída	3	2	Neco udělat něakou stránku.	Zkoušel/a jsem si jednoduché grafické programování a bavilo mě to		Nevím, co to programování je, takže nevím	Počítačové hry, Facebook, Android	vlevo volevo vlevo vpřed vpřed vpřed vpravo vpravo vpravo vpravo	Jít na přechot
5703	Dívka	14	7. třída	4	3	Že můžeme v tom programu něco vyrobit.	Zkoušel/a jsem si jednoduché grafické programování a bavilo mě to	Ano setkala.Když dělám nějakou prezentaci do školy,doma nebo tady na hodině informatiky.	Možná	Facebook, YouTube	V levo,před,před a před	Zastavit se na přechod a kouknou se do leva a prava,když nic nejede můžeme přejít silnici a když něco jede,tak stojíme na přechodu a čekáme až auto přejede nebo vyčkáme jestli nám zastaví a my budeme zase moc přejít.
3001	Chlapec	14	7. třída	4	4	nevím	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Ano	Počítačové hry, Facebook, 3D Grafika		vpred
6585	Dívka	12	7. třída	4	3	nic moc	Zkoušel/a jsem si jednoduché grafické programování, ale nebavilo mě to	ne asi nikdy	Ano	Počítačové hry, Facebook, Blogy, Počítačové sítě, Android	vpřed vpřed vpřed levo vpřed vpřed vpřed pravo vpřed vpřed vpřed	
6667	Chlapec	13	7. třída	2	4	Prostě jakoby programy	Zkoušel/a jsem si jednoduché grafické programování, ale nebavilo mě to	Ve škole v informatice	Ano	Počítačové hry, Facebook, 2D Grafika (fotky, a další ploché obrázky), 3D Grafika, Počítačové sítě, Prezentace (PowerPoint), Android	vpřed vlevo vpřed vlevo vpřed	nevim
6104	Chlapec	17	7. třída	3	2	bagr	bagr	bagr	bagr	bagr	3x vpřed vlevo 3x vpřed vlevo 3x vpřed	bagr -,-
7129	Chlapec	13	7. třída	4	4	změna programových chyb (nedodělávek)	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	od kamaráda pomohl nahrát placený antivir zadarmo :)	Ano	Počítačové hry, Facebook, Počítačové sítě, Prezentace (PowerPoint)	Vp=Vpřed Vl=Vlevo Vř=Vpravo řešení:VP3x,Vl,Vp3x,Vl,Vp3x	musíš se podívat do stran jestli nic nejede přejdi (nejlepší je jít po přechodu)
9244	Chlapec	12	7. třída	3	3	třeba programovat hry atd....	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	u kamarada	možná	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, Multimédia (muzika a filmy)	vpřed,vpřed,vřed,vlevo,vpřed,vpřed	zastavit se a rozhlídnout se jestli něco nejede.když nic nejede mužeme rychle přejít
9250	Chlapec	12	7. třída	2	4	vytváření prezentací, psaní na wordu	Zkoušel/a jsem si jednoduché grafické programování a bavilo mě to	sám na internetu	možná	Počítačové hry, Nějaké konkrétní internetové stránky, Prezentace (PowerPoint)	vpřed vpřed vpřed vlevo vpřed vpřed vpřed vlevo vpřed vpřed vpřed	Podívat se nejdřív doleva , potom doprava, jestli nejede auto. Pak přejít.
8503	Dívka	13	7. třída	4	4	Navrhování nebo výroba nějákých stránek.	Zkoušel/a jsem si jednoduché grafické programování a bavilo mě to		Nevím, co to programování je, takže nevím	Blogy, Prezentace (PowerPoint), Android	vpřed - vpřed - vpřed - vlevo -vpřed- vpřed -vpřed - vlevo -vpřed -vpřed -vpřed	zastavíš se před silnicí ,podíváš se vpravo i vlevo a když žádné auto nejede tak jí můžeš přejít
5962	Dívka	13	7. třída	3	4	Programovat počítač nainstalovat něco do počítače	Nikdy jsem to nezkoušel/a a ani nevím, co to je	Asi nesetkala	Ano	Facebook, Multimédia (muzika a filmy), Tabulkový procesor (MS Excel), Textový editor (MS Word), Prezentace (PowerPoint), Android, Youtube	3krát vpřed, vlevo, 3krát vpřed,vlevo, 3krát vpřed Sýr :D	Podíváš se do leva pak do prava pak zase do leva a pak jdeš ale během cesty se zase podíváš do leva a do prava a jseš na místě :D
5195	Dívka	13	7. třída	1	2	nevím	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Ne	Facebook	vpřed, vpřed,vpřed,vlevo, vpřed,vpřed,vpřed,vlevo,vpřed,vpřed,vpřed	Rozhlednu se jestli nejede auto a kdyz nejeda tak jdu přes silnici. a nohama delam: levá vpřed, pravá vpřed, leva vpřed, pravá vpřed,...
3673	Dívka	13	7. třída	3	3	To bohužel nevím :D Ale asi to bude něco s programy	Nevím, asi jsem to nezkoušela a nebo si na to nepamatuji		Nevím, co to programování je, takže nevím	Počítačové hry, Facebook, Blogy, Nějaké konkrétní internetové stránky, 2D Grafika (fotky, a další ploché obrázky), Multimédia (muzika a filmy), Počítačové sítě, Pracování s Wordem	Myšička může preskočit beton, takže se jím nemusí prokousávat... Ale jinak: VPŘED, VPŘED, VPŘED, VLEVO, VPŘED, VPŘED, VPŘED, VLEVO, VPŘED,VPŘED, VPŘED,	Musíš se rozhlédnout napravo, nalevo, napravo, nalevo a když nic nejede tak můžeš jít.....
3700	Dívka	14	7. třída	3	2	Prostě že se neco instaluje	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a		Ne	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, Multimédia (muzika a filmy), Počítačové sítě, Prezentace (PowerPoint), YouTube	Vpřed vpřed vpřed vlevo vpřed vpřed vpřed v pravo vpřed vpřed vpřed	Stoupni si k silnici ale né uplně potom se koukni doleva doprava a zase do leva a když nic nejede tak přeběhni silnici.A když jede auto tak ho nechte přejet a až přejede tak přeběhněte silnici.
2578	Chlapec	15	8. třída	4	3	je to ze se tam programuje nejaky program	Zkoušel/a jsem si jednoduché grafické programování, ale nebavilo mě to	naucil jsem se to sam	Ano	Počítačové hry, Facebook, Hardware (počítače po technické stránce), Android	vlevo. vpred,vlevo	rozhlidnout se a pak to prejit
1985	Chlapec	11	8. třída	2	2	c++	Zkoušel/a jsem si jednoduché grafické programování, ale nebavilo mě to		Ne	Hardware (počítače po technické stránce)	přejdi z 1 na 3 z 1 na 3 z 3 na 1	prejdi z 1 na 4 z 4 na 4 z 4 na 1
9461	Chlapec	15	8. třída	2	3	Vytváření nových programů nebo systému.. Či se dají programovat i hry.	Zkoušel/a jsem si jednoduché grafické programování, ale nebavilo mě to	Na internetu jsem narazil na programování. Nějak jsem to zkoušel ale moc mě to nebralo..	Ano	Počítačové hry, Facebook, Tvorba webových stránek, Multimédia (muzika a filmy), Počítačové sítě, Android	Vpřed, vpřed, vpřed, vlevo, vpřed, vpřed, vpřed, vlevo, vpřed, vpřed, vpřed	Nejdřív bych mu vysvětlil že se má rozhlédnou na obě strany.. Potom bych mu řekl aby co nejrychleji přešel cestu .. Pokud je tam přechod tak vyčkat na zelenou barvu. A až pak může jít po cestě .. Pokud je tam červená tak by musel stát .. A prakticky čekat na tu zelenou barvu.

Číselný kód	Pohlaví	Věk	Třída	Obliba INF	Význam INF	Představa "programování"	Dosavadní zkušenosti	Jestliže už jsi se setkal/a s programováním, odkud?	Zájem o vyzkoušení programování v hodinách IVT	Další zájmy z oblasti IVT a počítačů	Popiš myšce cestu k sýru	Jak přejít silnici a nenechat se zajet?
9814	Chlapec	14	8. třída	2	2	Něco jako nastavení.	Nikdy jsem to nezkoušel/a a ani nevím, co to je	Internet.	Ano	Počítačové hry, Facebook, Blogy, Tvorba webových stránek, Multimédia (muzika a filmy), Prezentace (PowerPoint)	vpřed o 4 vpřed o 4 doleva o 4	Jít tou nejkratší cestou.
8952	Dívka	13	8. třída	4	3	Naprogramovat třeba nějaké progamy do počítače nebo editování?	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Ano	Facebook, Blogy, Nějaké konkrétní internetové stránky, Multimédia (muzika a filmy), Prezentace (PowerPoint)	2x skoč vpřed,otoč se za levou rukou,2x vpřed otoč se za levou rukou a 2x vpřed.	Dej pozor na auta.až žádne nepojede tak přejdi.
3588	Dívka	15	8. třída	3	3	Že se aktualizují nějaké programy....	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a		Ne	Facebook, Blogy, Prezentace (PowerPoint)	vpřed (o 4 polička) , vlevo ¨, vpřed (o 3 polička) , vlevo, vpřed (o 3 polička)	Ta určitá osoba by měla jít na přechod, rozhlédnout se. Pokud nic nejede ani z pravé ani z levé strany. může přejít přechod.
7251	Chlapec	14	8. třída	2	2	Programovaní her,programů. Všechno co funguje např. na PC je potřeba na programovat.	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	z Internetu.	Ano	Počítačové hry, Tvorba webových stránek, 3D Grafika, Prezentace (PowerPoint), Android	3x Vpřed,Vlevo,3x Vpřed,Vlevo,3x Vpřed.	Rozhlédnout se a pak pokud nic nejede,rychle přeběhnout silnici(lépe po přechodě).
4864	Chlapec	14	8. třída	3	4	programování programu,nějakých složek atd.	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	od kamaráda	Ne	Počítačové hry, 2D Grafika (fotky, a další ploché obrázky), Tabulkový procesor (MS Excel), Textový editor (MS Word), Prezentace (PowerPoint), Android	3x vpřed , 1x vlevo,3x vpřed, 1x vlevo, 3x vpřed	jít na přechod, rozhlédnout se: doleva doprava doleva a může přejít pokaď nic nejde, a pokaď jede počkat až projede
7328	Chlapec	15	8. třída	3	4	nevim	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a		Ano	Počítačové hry, Nějaké konkrétní internetové stránky, Multimédia (muzika a filmy)	VPŘED,VPŘED,VPŘED,VLEVO,VPŘED,VPŘED, VPŘED,VLEVO.VPŘED,VPŘED,VPŘED,	Rozhlednout se vlevo,vpravo.Pokud nejede vozidlo můžeš přejít.
2769	Chlapec	14	8. třída	2	2	Že měco programujeme	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	Od sourozence	Ne	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky	Vpřed,vpřed,vpřed,vlevo,vpřed,vpřed,vpřed,vlevo,vpřed,vpřed,vpřed	Nejdříve se musíte rozhlédnou jestli nejede auto na pravou stranu a na levou stranu.Když nic nepojede opatrně přejdete silnici aby jste neupadli a můžete pokračovat dál v cestě.
2892	Dívka	14	8. třída	3	3	programujem programy	Nikdy jsem to nezkoušel/a a ani nevím, co to je	nikdy	Ano	Počítačové hry, Nějaké konkrétní internetové stránky	vlevo,vpřed,vpřed,vpravo	jdi na přechod rozhlídni se a když nic nejede tak jdi
3652	Dívka	14	8. třída	4	5	instalování, zkoušení programů	Zkoušel/a jsem klasické programování v nějakém programovacím jazyce		Ano	Facebook, Nějaké konkrétní internetové stránky, Tvorba webových stránek, 2D Grafika (fotky, a další ploché obrázky), Multimédia (muzika a filmy), Prezentace (PowerPoint), Android	VPŘED,VPŘED,VPŘED,VLEVO,VPŘED,VPŘED, VPŘED,VPRAVO,VPŘED,VPŘED, VPŘED	POSTAVÍM SE PŘED TU SILNICI A NEJDŘÍVE SE PODÍVÁM VLEVO, PROTOŽE TAM PO TÉ STRANĚ JEZDÍ AUTA BLÍŽ K NÁM A PAK SE PODÍVÁM VPRAVO. POKUD NIC NEJEDE TAK PŘEJDU A POKUD ANO TAK POČKÁM DOKUD NEBUDE SILNICE VOLNÁ.
9452	Dívka	13	8. třída	4	4	Programování je nastavení nějakému programu aby se nám v něm lépe pracovalo a aby se třeba trochu rozšířil.	nikdy sem t nezkoušela, ale ráda by jsem se s tím seznámila a vyskoušela si to	Můj táta umí programovat takže od něj, ale kdy to jsi už nepamatuji.	Ano	Nějaké konkrétní internetové stránky, 2D Grafika (fotky, a další ploché obrázky), 3D Grafika, Multimédia (muzika a filmy), Tabulkový procesor (MS Excel), Textový editor (MS Word), Prezentace (PowerPoint)	Myško 3x vpřed, potom vlevo a takhle ještě 2x	Najdi přechod pokud tam žádný není tak to nevádí. Pořádně se rozhlédni na pravo i levo, a když nic nejede tak rychle přejdi, když něco jede tak počkej až nic nepojede.
1500	Dívka	14	8. třída	3	5	nevím co to je	Nikdy jsem to nezkoušel/a a ani nevím, co to je	nevim co to je	Nevím, co to programování je, takže nevím	Počítačové hry, Facebook, Prezentace (PowerPoint)	půjde třikrát vpřed , otočí se vlevo, půjde zase třikrát vpřed a otočí se vlevo a půjde třikrát vpřed.	Rozhlédni se ,jestli něco nejede nejdřiv nalevo a pak na pravo potom, když nic nepojede tak přejdi rovně po silnici
9219	Dívka	14	8. třída	3	4	Nikdy jsem to neslyšela.	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Ano	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, Multimédia (muzika a filmy), Tabulkový procesor (MS Excel), Textový editor (MS Word), Prezentace (PowerPoint)	Musí jít 3x vpřed potom vlevo a třikrát vpřed a potom vlevo pak 3x vpřed.	Pokud je čelem k přechodu musím jít vpřed.
7250	Dívka	14	8. třída	3	3	vytváření	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Nevím, co to programování je, takže nevím	Facebook	vpřed, vpřed,vpřed, vlevo, vpřed, vpřed,vpřed, vpravo, vpřed, vpřed	zastavit u přechodu pro chodce, rozhlédnout se vlevo, vpravo a vlevo a když nic nejede tak přejít po přechodu
5299	Dívka	14	8. třída	3	4	Vyrábění a tvoření svých vlastních programů	Nikdy jsem to nezkoušel/a a ani nevím, co to je	Nejsem si jistá co programování znamená,tak nevim jestli jsem se s ním setkala	Nevím, co to programování je, takže nevím	Počítačové hry, Facebook, Prezentace (PowerPoint)	Myška se posune o 3 políčka do předu a otočí se do leva o 90° poté pojede o 3 pole vpřed a zahna do leva.Půjde o 3 pole vpřed a je u sýra.	Jdu na přechod a rozhlédnu v prov poté v levo a znovu v pravo a poté rychle přejdu silnici bez jakéhokoliv rozmýšlení.
1884	Dívka	13	7. třída	2	3	něco s počítači	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a		Ano	Facebook, Blogy, Nějaké konkrétní internetové stránky, 2D Grafika (fotky, a další ploché obrázky), Multimédia (muzika a filmy)	Vpřed,vpřed,vpřed,vlevo,vpřed,vpřed,vpřed,vlevo,vpřed,vpřed,vpřed	Podívat se doleva , pak doprava a pokud nic nejede , může přejít.
4925	Dívka	13	7. třída	2	2	programy	Nikdy jsem to nezkoušel/a a ani nevím, co to je	nevim co znamena programovani	Nevím, co to programování je, takže nevím	Facebook, Textový editor (MS Word), Prezentace (PowerPoint), Android	dopředu dopředu dopředu do leva do předu dopředu do předu do leva dopředu dopředu a dopředu	rozhlídnout se v levo v pravo a v levo jestli nic nejede tak se muže přejit
7885	Dívka	13	7. třída	4	4	Tak budeme pracovat s nějakým programem, s kterým jsme ještě nepracovali.	Nikdy jsem to nezkoušel/a a ani nevím, co to je	Ještě jsem se stím nesetkala	Nevím, co to programování je, takže nevím	Facebook	dopředu, dopředu, dopředu ,nahoru, dopředu, dopředu, do leva ,dopředu, dopředu.	Když přijdete k přechodu, tak se pořádně rozhlídni jestli nejede auto(většiou jso někde zapnutý semafory, když svítí červený panáček znamenáto STOP, když svítí zelený panáček můžete JÍT) A když není tak počkám až zastaví auto a můžu přejít na druhou stranu
1639	Dívka	14	7. třída	4	4	vůbec nevím	Nikdy jsem to nezkoušel/a a ani nevím, co to je	asi nikdy	Nevím, co to programování je, takže nevím	Počítačové hry, Facebook	v před,v před,v před a vlevo v před,v před,v před a v pravo	Tak jít na přechod,a rozhlídnout se jestli něco nejede,a když jo tak počkáme až to projede a pak znovu rozhlídnout a když nic nejede tak můžeme jít.
3019	Dívka	13	7. třída	3	5	grafy	Zkoušel/a jsem si jednoduché grafické programování, ale nebavilo mě to	od tety	Nevím, co to programování je, takže nevím	Facebook, Nějaké konkrétní internetové stránky, Prezentace (PowerPoint)	vpřed , vpřed , vpřed, doleva , vpřed , vpřed , vpřed , doleva , vpřed , vpřed , vpřed	Díváš se doprava doleva , pokud nepojede auto můžeš přejít , zda pojede auto stojíš stále na chodníku a znovu se rozhlížiš .
2736	Dívka	13	7. třída	3	4	nějaká složka kde se dělají různé programy	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a		Nevím, co to programování je, takže nevím	Facebook	vpřed,vpřed,vlevo vpřed,vpřed,vpřed,vlevo vpřed,vpřed,vpřed,SÝR	Jdi na přechod.Pokud není na silnici přechod,tak se rozhlédni do leva a až budeš v prostředku silnice tak do prava.Pokud nic nejede,tak můžeš jít.
4434	Dívka	12	7. třída	2	3	práce na počítači	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	Na internetu	Nevím, co to programování je, takže nevím	Počítačové hry, Facebook, Tvorba webových stránek, Prezentace (PowerPoint)	vpřed, vpřed,vpřed, vlevo, vpřed, před, vlevo, vpřed, vpřed	Musíš dojít k přechodu, když tam není simafor i policajt tak se musíte rozhlídnout doprava doleva a asi dvakrát projistotu když vás pustí jedno auto musíte si dávat pozor i na to vprotisměru až vás pustí dvě tak mužete přejít.

Číselný kód	Pohlaví	Věk	Třída	Obliba INF	Význam INF	Představa "programování"	Dosavadní zkušenosti	Jestliže už jsi se setkal/a s programováním, odkud?	Zájem o vyzkoušení programování v hodinách IVT	Další zájmy z oblasti IVT a počítačů	Popiš myšce cestu k sýru	Jak přejít silnici a nenechat se zajet?
7602	Dívka	13	7. třída	3	4	Asi nějaké obnovování programů, nevím, nikdy jsem se o to moc nezajímala.	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Nevím, co to programování je, takže nevím	Facebook, Multimédia (muzika a filmy), Prezentace (PowerPoint), Android	3x vpřed; 1x vlevo; 3x vpřed; 1x vlevo; 3x vpřed a je u sýra	1. Podívat se ještě na chodníku doleva 2. Podívat se ještě na chodníku doprava 3. Podívat se naposledy na chodníku ještě doleva 4. Jít na druhou stranu chodníku rychlým krokem
7793	Dívka	13	7. třída	1	3	že se neco naprogramuje(nevím co to je)	Nikdy jsem to nezkoušel/a a ani nevím, co to je	ne	Nevím, co to programování je, takže nevím		vpřed,vpřed,vpřed,vlevo,vpřed,vpřed,vpřed,vlevo,vpřed,vpřed,vpřed(3x vpřed,1xvlevo,3x vpřed1,x vlevo3x vpřed	rozhlídneš se vlevo,vpravo,vlevo když nic nejede rychle přejdeš
9538	Dívka	13	7. třída	1	1	Mého spolužáka - chce být programátor. (nevím co to je, ani mě to nezajímá - asi něco s počítačem)	nevím co to je	-	asi ne	používám počítač jen, když musím - třeba na domácí úkoly atd. vadí mi, že všichni lidi jsou na něm závislí - vím, že každý má právo mít svůj názor, ale podle mě je počítač „DROGA"	vpřed, vpřed, vpřed, vlevo, vpřed, vpřed, vpřed, vlevo, vpřed, vpřed, vpřed	rozhlídni se a když nic nepojede po silnici tak přejdi
9626	Dívka	13	7. třída	4	3	nevím	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Ano	Počítačové hry, Facebook, 2D Grafika (fotky, a další ploché obrázky), 3D Grafika, Prezentace (PowerPoint)	vpřed vpřed vpřed vlevo vpřed vpřed vpřed vlevo vpřed vpřed vpřed	jít na přechod
6465	Dívka	13	7. třída	3	3	vůbec nevím	Nikdy jsem to nezkoušel/a a ani nevím, co to je	ještě jsem se s ním nesetkala	Ano	Facebook, Prezentace (PowerPoint)	vpřed,vpřed,vlevo,vpřed,vpřed,vpravo,vpřed,vpřed	nejdřív se rozhlédnout vlevo pak vpravo pak zase v levo a když nic nejede může jít
1575	Dívka	13	7. třída	4	2	Programování počítačů	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a		Ano	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, Multimédia (muzika a filmy), Počítačové sítě, Prezentace (PowerPoint), Android	3x vpřed 1x vlevo 3x vpřed 1x vlevo 3x vpřed	Nejdřív se přijde nakraj silnice před předchod rozhlídne se doleva doprava a ještě jednou když nic nejede může přejít.
7997	Dívka	13	7. třída	3	5	Různé programy..	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	Z bratrem když jsme přeinstalovávaly notebook..	Ne	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, Tvorba webových stránek, Multimédia (muzika a filmy), Počítačové sítě, Prezentace (PowerPoint)	Vpřed , vpřed , vlevo , vpřed , vpřed , vlevo , vpřed , vpřed ... Sýr ...	Když přijdeš před silnici tak se rozhlédni v pravo a vlevo.. když by něco jelo tak počkej než to auto přejede.. a pak přejdeš přechod..v případě že stojíš na přechodu a ňáké auto zastaví znamená to že ti dává přednost takže můžeš přejít přechod - musíš se podívat jestly nejede něco na druhé straně..v tom případě přejdeš přechod a nic tě nezajede..
8639	Dívka	13	7. třída	4	4	Instalování různých programů.	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	Ukazovala mi to jednou moje sestřenka.	Ano	Facebook, Blogy	vpřed,vpřed ,do leva, vpřed ,vpřed, do leva .	Nejdříve musíš najít nějaký přechod a rozhlédnou se do prava a do leva a pak přejít. A když se přechází po nějaké silnici kde není přechod tak se rozhlednout do prava a do leva a pak rychle přejít .
2776	Dívka	13	7. třída	3	3	Nwm ...	Nikdy jsem to nezkoušel/a a ani nevím, co to je	...	...	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, Multimédia (muzika a filmy), Počítačové sítě, Tabulkový procesor (MS Excel), Textový editor (MS Word), Prezentace (PowerPoint), Android, ...	Myška musí ít třikrát vpřed na číslo 1,4 potom se otočit do leva a zase jít třikrát vpřed na číslo 4,4 zase se otočit v levo a jít třikrát vpřed až se dostane k lahodnému sýru.	Postaviš se skoro na okraj chodníku a rozhlédneš se do levé strany , do pravé strany a potom zase do levé strany . Když nic nepojede nebo auta z levé i pravé strany zastaví a pustího tak teprve může přejít na druhou stranu .
7474	Dívka	13	7. třída	2	3	Nastavování různých aplikací.	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Nevím, co to programování je, takže nevím	Facebook	3 políčka vpřed,3 políčka do leva a 3 políčka do leva	Musí se správně rozhlédnout do leva,do prava a když nic nejede tak teprve může přejít silnici.
4873	Dívka	13	7. třída	3	2	Instalace nových aplikací.	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a		Nevím, co to programování je, takže nevím	Počítačové hry, Facebook, Multimédia (muzika a filmy)	Myška pude 3 vpřed potom otočka vlevo , 3 vpřed potom otočka vlevo , 3 vpřed a je u sýra.	Nejdřív najdi přechod pro chodce a dojdi k němu. Počkej až se na semaforu rosvítí zelený panáček. Pokud se rozsvítí tak můžeš přacházet a nemělo by tě nic zajet. POZNÁMKA : vždycky se radši rozhlédni nejdřív nalevo a potom napravo.
8312	Dívka	12	7. třída	4	3	Myslím si, že programování je potřebné k tvorbě her a webových stránek.	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Ano	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, Multimédia (muzika a filmy), Textový editor (MS Word), Prezentace (PowerPoint), Android	vpřed, vpřed, vpřed, vlevo, vpřed, vpřed, vpřed, vlevo, vpřed, vpřed, vpřed	Nejprve se musíš kouknout doleva, doprava a znovu doleva. Když vidíš, že nejede žádné auto, nebo je auto daleko, abys to stihl, přejdi, ale pro jistotu se během přecházení alespoň jednou koukni doleva a doprava.
3637	Dívka	12	6. třída	4	5	Vytváření programů.	Zkoušel/a jsem si jednoduché grafické programování a bavilo mě to	od kamarada	Ano	Počítačové hry, Facebook, Blogy, Nějaké konkrétní internetové stránky, 2D Grafika (fotky, a další ploché obrázky), Multimédia (muzika a filmy), Počítačové sítě, Textový editor (MS Word), Android	VPŘED, VPŘED, VPŘED, VLEVO, VPŘED, VPŘED, VPŘED, VLEVO, VPŘED, VPŘED, VPŘED.	Takže, rozhledni se nejdřív v pravo a pak vlevo, pokud nic neuvidíš můžeš jít, ale pokud zahledneš nějaký dopravní prostředek neho projet a až pak přejdi na druhou stranu. pokud zahledneš anutomobil až v dalce můžeš přejí ještě ně automobil projede, ale pokud je velmi blízko nech ho projet.
1629	Dívka	12	6. třída	4	4	Vytváření nějakého programu.	Nikdy jsem to nezkoušel/a a ani nevím, co to je	Nevím	Nevím, co to programování je, takže nevím	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, Multimédia (muzika a filmy), Textový editor (MS Word), Prezentace (PowerPoint)	3x vpřed, vpravo, 3x vpřed, vpravo a 3x vpřed	Na kraji silnice se rozhlédnout do leva i do prava. A pak kolmě na silnici přejít na druhou stranu silnice. Doporučuji chodit po přechodu.
9330	Dívka	12	6. třída	4	3	program	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	nevím	Ano	Počítačové hry	3 vpřed, vlevo, 3 vpřed, vlevo, 3 vpřed	zastavit rozhlédnout se vpravo vlevo a když nic nejede může přejít
6957	Dívka	12	6. třída	3	4	nevím co to je...	Nikdy jsem to nezkoušel/a a ani nevím, co to je	nevím....	Nevím, co to programování je, takže nevím	Facebook, 2D Grafika (fotky, a další ploché obrázky), Hardware (počítače po technické stránce)	Před o tři pole,Vlevo o tři pole, Azase vpřed o tři pole....	No řekla bych mu ať jse rozhlédne do leva do prava a zase a ať de.....
6959	Dívka	13	6. třída	3	4	přesouvaní dat	Nikdy jsem to nezkoušel/a a ani nevím, co to je	od kmarada	Ne	Počítačové hry, Facebook, 2D Grafika (fotky, a další ploché obrázky), Multimédia (muzika a filmy), Android	v před o 1 políčko vlevo o 2 políčka a 2 políčka v před	o 3 políčka vpřed vlevo o 3 políčka vpřed o 3 políčka v levo
2469	Dívka	12	6. třída	3	3	spouštění něakých programů	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	z internetu	Ano	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky	rovně rovně rovně do leva rovně rovně rovně do leva rovně rovn sír.	Rozhlédnout se do leva pak do prava a pak zase do leva a dyž nic neede tak můžeme jít.

Číselný kód	Pohlaví	Věk	Třída	Obliba INF	Význam INF	Představa "programování"	Dosavadní zkušenosti	Jestliže už jsi se setkal/a s programováním, odkud?	Zájem o vyzkoušení programování v hodinách IVT	Další zájmy z oblasti IVT a počítačů	Popiš myšce cestu k sýru	Jak přejít silnici a nenechat se zajet?
4432	Dívka	12	6. třída	5	4	Něco málo jsem si o tom zjišťovala ale moc nevím co to je	Je to můj koníček, programuji běžně a rád/a	sama na internetu	Ano	Facebook, Blogy, Prezentace (PowerPoint)	vpřed,vpřed,vpřed,vlevo,vpřed,vpřed,vpřed,vlevo,vpřed,vpřed,vpřed	Představ si silnici na které přechod není. Než budeš chtít přejít tak si dobře rozmysly co uděláš. Pokud víš tak si velmy zkušený ale pokud nevíš pomůžu ti jak na to. Nejdříve se musíš rozhlédnout a to uděláš tak že se nejdříve koukneš do prava ujistíš se že tam nic nejede pak se koukneš do leva jestly tam nic nejede a pak tohle můžeš z opakovat několikrát aby si měl/a větší jistotu že tam nic opravdu nejede. Poté co ses rozhlédnul můžeš bezpečně přejít a pamatuj vždycky se musíš rozhlédnout at je to na přechodě nebo ne.
8545	Dívka	12	6. třída	4	3	že je to nastavujeme nějaký program	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	Doma od rodičů.	Ano	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, Prezentace (PowerPoint)	Vpřed o tři pole,vlevo,vpřed o tři pole,vlevo a vpřed o tři pole.	Dojít k silnici rozlídnout se vpravo i vlevo.Když nic nejede můžete přejít. Nebo u přechodu se semaforem můžete přejí když je zelená,ale raději se rozhlédněte.
7011	Dívka	12	6. třída	4	4	vyrábění na počítači nějakých programů	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Nevím, co to programování je, takže nevím	Počítačové hry, Nějaké konkrétní internetové stránky, Textový editor (MS Word), Prezentace (PowerPoint)	1- 3krát vpřed 2-, 1krát vlevo 3-, 3krát vpřed 4- , 1krát vlevo 5-, a 3krát vpřed	osoba se na kraji chodníku rozhlédne na silnici kde není přechod na obě strany , pokud uvidí v blízkosti auto počká až auto přejede a rychle přejít, přeběhnout silnici a v nejkratší vzdálenosti - rovně
9885	Dívka	13	6. třída	4	4	program	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	nevím	Ano	Počítačové hry, Facebook	3 vpřed , vlevo , 3 vpřed , vlevo , 3 vpřed	přejíd když nejedou auta nebo počkat u přechodu až nějaké auto zastaví

H. Výsledky výstupního dotazníku

Číselný kód	Pohlaví	Věk	Třída	Jak tě bavila tato hodina	Nové vysvětlení co to je "programování"	Chtěl/a by jsi pokračovat v takovýchto hodinách IVT?	Stihnul/stihla jsi projít celý Hour of Code?	Konečná úroveň Lightbota?	Zájem o kroužek programování	Zájem o kroužek LEGO Mindstorms	Připomínky	Popiš myšce cestu k sýru	Jak přejít silnici a nenechat se zajet?
2228	Dívka	12	6. třída	5	nevim----:D	Ano	Ano, včetně úspěšně vyřešené dvacítky	1-3	Ano	Ano	já nevim	vpravo vlevo vpřed nevim	
6986	Chlapec	12	6. třída	5	programovani programu	Ano	Ano, včetně úspěšně vyřešené dvacítky	1-5	Ano	Ano	celkem dobry	nttwewetwertwatra	ezremmezezzezteztezt
5619	Chlapec	12	6. třída	4	je to paráda	jenom nekdy	Ano, včetně úspěšně vyřešené dvacítky	1-8	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ano	bavilo me to	vpred dolu vpred vpred nahoru vpred nahoru nahoru nahoriu do strany nahoru vpred vprd dolu do strany	rozhlednout se a prejit
5243	Dívka	11	6. třída	1	pořád to samé , že se instalují nějaké programy , td.	Ne	Ano, včetně úspěšně vyřešené dvacítky	1-8	Ne	Ne		vpřed,vpravo,vpřed,vpřed,vpřed,vlevo,vpřed,vpřed,vpřed,vlevo,vpřed,vlevo,vpřed,vpřed,vpřed,vlevo,vpřed.	rozhlídnout se ,alespoň 2krát a když nic nepojede ta přejdu ...
2998	Chlapec	11	6. třída	5	Dou naprogramovat i lego hry tiskárna 3D,	Ano	20 (ale nedokončil)	Vůbec jsem se k Lightbotovi nedostal/a	Ano	Ano		Vpřed, Vpravo, vpřed,Vlevo,vpřed,vpřed,vlevo,vpřed,vpravo,vpřed,vlevo,vpřed,vpřed,vlevo,vpřed,vpravo,vpřed,vlevo,vpřed,vpřed,vlevo,vpřed,vpravo,vpřed	Musíme se rozlinout a když nic nejede muzeme jit
1912	Chlapec	12	6. třída	4	hry	Ano	Ano, včetně úspěšně vyřešené dvacítky	1-5	Ano	Ne		rovne doprava rovne doleva rovne rovne doleva rovne doprava rovne rovne rovne dolena rovne doleva rovne rovne rovne doleva rovne	
4095	Dívka	13	6. třída	3	programování je že člověk dává počítači příkazy	Ano	14	Vůbec jsem se k Lightbotovi nedostal/a	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ne	tato hodina byla dobrá, docela i namáhavá	pujde na hor a čumákem do předu	pjde o jednu kotku do předu pak dolu a dopbřrd na horu do prava nahoru do prav anahoru rovně a dolk e sýru
1945	Dívka	12	6. třída	3	nerozumím tomu	Ano	Ano, včetně úspěšně vyřešené dvacítky	1-7	Ne	Ne		vpřed, vlevo, vpřed, vpřed	vpřed, vpravo, vpřed, vlevo, vpřed, vpřed, vlevo, vpřed, vpravo, vpřed, vlevo, vpřed, vpřed, vlevo, vpřed, vpravo, vpřed, vlevo, vpřed, vpřed, vlevo, vpřed, vpravo, vpřed
8542	Chlapec	12	6. třída	5	baví me to mnohem víc pridu domu a zacnu programovat nakou hru nebo aplikaci hru v aplikaci game maker	urcite	20 (ale nedokončil)	Vůbec jsem se k Lightbotovi nedostal/a	Ano	Ano		vpred vpravo vpred vpravo vpred vpred vlevo vpred vpravo vpred vlevo vpred vpred vlevo vpred vpravo vpred vlevo vpred vpred vlevo vpred vlevo vpred	rozhledeme se na prechodu
4397	Dívka	12	6. třída	3	Něaké ovládání	Nevim	17	Vůbec jsem se k Lightbotovi nedostal/a	Ne	Ne		vpřed,vpravo,vpřed,vpřed,nahoru,vpřed,vpravo,vpřed,vpřed,vlevo,vpřed,vpřed,vpřed,vpravo	
4572	Chlapec	12	6. třída	5	docelatomu rozumím	jasne že jo	3	Vůbec jsem se k Lightbotovi nedostal/a	Ne	Ano		vpřet,vpravo,vpřet,vlevo,vpřet,vpřet,vlevo,vpřet,vpravo,vpřet,vlevo,vpřet,vpřet,vlevo,vpřet,vpravo,vpřet.vlevo,vpřet,vpřet,vlevo,vpřet,vpravo,vpřet	nejdřív se rozhedni vpravo vlevo a zase vpravo když nic nejede muzeme jit a kdyz jo tak chvilku pockame a pak muzeme jit
4971	Dívka	12	6. třída	3	ovládíní počítače a robotů	jen někdy	14	Vůbec jsem se k Lightbotovi nedostal/a	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu	chtěla bych se v hodině naučit sestavit robota na ovládání (pejska)	pujde rovně pak nahoru do leva a k sýru	řekla byh musíš být opatrný proto se nejdříve podívej do leva a do prava a pak ještě do prava a mužeš přejít svižným tempem hlavně neutíkej mohl by si spadnout a srazilo by tě auto !! nevidělo by tě a loudat se taky není dobrý nápad !!!!!
7380	Chlapec	12	6. třída	5	stránky,roboty,vědce	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-5	Ano	Ano	vše bylo fan až na to že to bylo moc lehké a vše jsem už dělal doma	vpřed,vpravo,vpřed,vlevo, (2xopakovat 2xvpřed,vlevo,vpřed,vpravo,vpřed,vlevo)	po přechodu a rozhlídnu se 2xvlevo,vparo
3811	Dívka	12	6. třída	3	rideni pocitace,animace robotu atd. i hrani her	Ano	20 (ale nedokončil)	Vůbec jsem se k Lightbotovi nedostal/a	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	spis ne	byla tu sranda,narozdil od jinych hodin ty byl klid coz u nas moc caste neni,bylo to zajimavi a bylo to neco jineho, zase sme se neco dozvedeli a v teto dobe se nam to hodne hodi(podle me)	jedno policko vpred vpravo jedno policko vpred vpravo dve policka vpred vpravo jedno policko vpred vpravo jedno policko vpred vpravo dve policka vpred vlevo jedno policko vpred vpravo jedno policko vpred vlevo dve policka vpred vlevo jedno policko vpred vlevo mys je u syru	uz sem to predtim psala..bylo to dlouhe ale dala sem si stim praci takze bych to normalne nekomu na ulici kde by slo o zivot
3683	Chlapec	11	6. třída	5	lego mindstorms	určitě každý den	Ano, včetně úspěšně vyřešené dvacítky	1-7	Ano	Ano	bylo to jen na dvě hodiny	vpřed vpravo vpřed vlevo vpřed vpřed vlevo vpřed vlevo vpřed vlevo vpřed vpřed vlevo vpřed vlevo vpřed vlevo vpřed vpřed vpravo vpřed vlevo vpřed	vpřed vpravo vpřed vlevo vpřed vpřed vlevo vpřed vlevo vpřed vlevo vpřed vpřed vlevo vpřed vlevo vpřed vlevo vpřed vpřed vpravo vpřed vlevo vpřed
8747	Dívka	11	6. třída	5	Programování elektroniky.(příkazy, které jim dáme)	Samoydřejmě, bylo to výborné	11	Vůbec jsem se k Lightbotovi nedostal/a	Ano	Všechny kroužky, které se programování týkají bych okamžitě přijala.	Hodina se mi velice líbila, nikdy jem neprogramovala, ale určte vím, že se o prázdninách k tomu to tématu vrátím.	Vpřed, vlevo,Vpřed, vpřed ...... napsala bych i fígl, ale už nemám čas.	Neztihla.
6600	Chlapec	13	7. třída	3	nevim	Ne	16	Vůbec jsem se k Lightbotovi nedostal/a	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu	žádné	3x vpřed vlevo 3x vpřed vlevo před	rozhlídnout se na přeechodu
3250	Chlapec	14	7. třída	4	vytvareni programu pomoci kostek	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-4	Ano	Ano	žadný	porad vpřed	rozhlídnout se a pak jit
5668	Chlapec	13	7. třída	5	Instalování programu	Ano	17	2-1	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ano	Docela mě to bavilo	vpřed-vpravo-vpřed-vlevo-vpřed-vpřed-vlevo-vpřed-vpravo-vpřed-vlevo-vpřed-vpřed-vlevo-vpřed-vpravo-vpřed-vlevo-vpřed-vpřed-vlevo-vpřed-vpravo-vpřed	Rozhlídnout se vpravo a vlevo
5764	Chlapec	13	7. třída	5	dáváš mu instrukce	Ano	Ano, včetně úspěšně vyřešené dvacítky	1-1	Ano	Ano	byla velmi dobrá hodina	vpřed,vpravo,vpřed,vpřed,vlevo,vpřed,vpřed,vlevo,vpřed,vpřed,vlevo,vpřed,vlevo,vpřed,vpřed,vpravo,před,pravo,před	měl by ses rozhlídnout a jít na přechod počkat než ti tam skočí zelenej panáček
3966	Chlapec	14	7. třída	5	VPŘED VLEVO VPRAVO	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-4	Ne	Ne	žádný	VPŘED VPRAVO VPŘED VLEVO VPŘED VPŘED VLEVO VPŘED VPRAVO VPŘED VLEVO VPŘEDV PŘED VLEVO VPŘED VPRAVO VPŘED VLEVO VPŘED VPŘED VLEVO VPŘED VPRAVO VPŘED	počkat až nepojede vozidlo a pak VPŘED
5791	Chlapec	13	7. třída	2	ZE z neceho treba z lega muzu udelat robota,Tiskarnu 3D	Ano	Ano, včetně úspěšně vyřešené dvacítky	1-7	Ne	Ne	Bylo to super	1x rovne 1x prava 1x dopredu 1x doleva 2x rovne 1x do¨leva 1x rovne 1x doprava 1x rovne 1x doleva 2x rovne 1x doleva 1x rovne 1xdoprava 1x doleva 2x rovne 1x doleva1 x rovne 1x doprava 1x rovne	rozhlidnou se do leva a doprava a jeste jednou a muzeme jit :)
9769	Chlapec	13	7. třída	4	je to ovládání ruzných věcí	ano i ne	16	Vůbec jsem se k Lightbotovi nedostal/a	Ano	Ano		vpřed prava leva 2.předleva vpřed vpravo vpřed vlevo předpřed vlevo vpravo vlevo předpředvpravo před vlevo	
2647	Chlapec	12	7. třída	4	Zadávání příkazů nějakému zařízení	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-6	Ano	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu	Celková hodina byla fajn. Bylo to příjemné spestření pro ty co programování vůbec nerozumí, ale ten kdo si už programování vyzkoušel tak se toho moc nedozvěděl.		
4914	Chlapec	14	7. třída	3	se ovlada pomoci příkazu	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-4	Ano	Ne		1 vpred 1 vpravo 1vpred 1vlevo 2vpred 1 vlevo 1vpred	wsdfasd

Číselný kód	Pohlaví	Věk	Třída	Jak tě bavila tato hodina	Nově vysvětlení co to je "programování"	Chtěl/a by jsi pokračovat v takovýchto hodinách IVT?	Stihnul/stihla jsi projít celý Hour of Code?	Konečná úroveň Lightbota?	Zájem o kroužek programování	Zájem o kroužek LEGO Mindstorms	Připomínky	Popiš myšce cestu k sýru	Jak přejít silnici a nenechat se zajet?
8747	Chlapec	14	7. třída	4	Tvoření programů, strojů..	Ano	20 (ale nedokončil)	2-5	Ano	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu		VPŘED 1x VPRAVO 1x VPŘED 3x VLEVO 1x VPŘED 1x VPRAVO 1x VLEVO 1x VPŘED 3x VLEVO 1X VPŘED 1X VPRAVO 1X VLEVO 1X VPŘED 3X VLEVO 1X VPŘED 1X VLEVO 1X	Rozhlédnout se na obě strany jestli nejede auto a přejít.
3200	Chlapec	13	7. třída	4	Programování je že elektronická věc dělá co jí řekneme když je na to naprogramovaná	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-1	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu		Vpřed , vlevo , vpřed , vpravo , 2x vpřed , vlev , vpřed , v pravo , 2x vpřed , v pravo , vpřed , v levo , 2x před , v levo , vpřed	Když je na přechodu semafor , tak zmáčkneme čekejte , a až skočí na semafor zelená , tak se rozhlédneme a když je vše v pořádku tak můžeme jít
7721	Chlapec	14	7. třída	4	Psaní do dokumentu programovacím jazykem.	Ano	Ano, včetně úspěšně vyřešené dvacítky	Vůbec jsem se k Lightbotovi nedostal/a	Ano	Ano		1pole dopředu,otočit vpravo,1 pole dopředu,otočit vlevo,2 pole dopředu,otočit v levo,1 pole dopředu,otočit vpravo,1pole dopředu,otočit vlevo,2 pole dopředu,otočit v levo,1 pole dopředu,otočit v pravo,1 pole v před,otočit v levo,dvakrát vpřed,1 vlevo,1 pole dopředu,otočit vlevo,jedno pole dopředu	Běž ke kraji silnice,rozhlédni se na levo a napravo,pokud nepojede žádné auto přejdi.
7752	Chlapec	13	7. třída	5	programování je, že něco napíšete a panáček na kterého jste to napsali to udělá	Ano	Ano, včetně úspěšně vyřešené dvacítky	1-5	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu	bylo to super	VPŘED, VPRAVO, VPŘED, VLEVO, 2x VPŘED, VLEVO, VPŘED, VPŘED, VLEVO, 2x VPŘED, VLEVO, VPŘED, VPRAVO, VPŘED, 2x VPŘED, VLEVO, VPŘED, VPRAVO, VPŘED	přijít k silnici kouknout doleva pak doprava pak doleva a když nic nejede přejít
3770	Chlapec	15	9. třída	5	rozkazování počítači.	Ano	Ano, včetně úspěšně vyřešené dvacítky	3-2	Ano	Ano		VPŘED VPRAVO VPŘED VLEVO 2XVPŘED VLEVO VPŘED VPRAVO VPŘED VLEVO 2X VPŘED VEVO VPŘED VPRAVO VPŘED VLEVO 2X VPŘED VEVO VPŘED VPRAVO VPŘED	
5784	Chlapec	15	9. třída	5	Popis postupu pro pc	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-4	Ano	Ano		Vlevo, Vlevo, Vpřed, Vpravo, Vpřed, Vpřed, Vpravo. Vpřed	Jdi k přechodu pro chodce, počkej až auta před přechodem zastaví. přeii.
7117	Dívka	16	9. třída	2	No jako mylsáím si že nemá cenu to vysvětlovat! Mají poslouchat :P	Ani ne jak kdy	20 (ale nedokončil)	2-5	Ne k životu to nepotřebuji!	Ne	hm	UŽž jsem to dělala názor mám na to stejný	Rozhlídnout se v pravo vlevo a je to !
5301	Dívka	15	9. třída	4	Nevím	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-5	nemluv	HM		nevím už	
2486	Chlapec	15	9. třída	4	asi furt stejně	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-2	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ne	hodina byla docela zábavná teda dokud mi to šlo	vpřed,vpravo,vpřed,vlevo, 2* vpřed, opakovat 4*	rozhlídnu se na pravo ,na levo a když nic nejede tak přejdu
9536	Chlapec	15	9. třída	3	programátoři naprogramovávaí počítače ruzne stroje	Ne	Ano, včetně úspěšně vyřešené dvacítky	2-5	Ne	Ano		v před v levo v před v pravo v před v před v levo v před v pravo v pred v levo v před v levo v pred v levo v před v levo v před	
8560	Chlapec	15	9. třída	4	Zjednodušování	asi ne normální hodiny informatiky mě baví víc	Ano, včetně úspěšně vyřešené dvacítky	2-5	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu	žádné	NASER SI	NASER SI
1967	Chlapec	15	9. třída	4	naprogramovaní stroje a on potom plní příkazy	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-2	Ne	Ano	hodina byla supej	vpřed vpravo vpřed vlevo	
4009	Chlapec	15	9. třída	5	Je to jako hra.	Ano	10	2-2	Ano	Ano	Někdy to bylo težký ale zvládl jsem to a bavilo mě to. :)	Vpřed potom do prava pak vpřed do leva před a todhle budem dělat 3x.	Jako předtím jak jsme to napsal.
7968	Dívka	15	9. třída	5	ovládaní třeba hry	Ano	Ano, včetně úspěšně vyřešené dvacítky	1-2	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu	nemám žádnou připomínku. Líbilo se mi to ;)	vpřed, vpravo, vpřed, vlevo, vpřed vpřed, vlevo, vpred, vpravo, vpřed, vpřed, vlevo, vpřed, vpravo, vpřed, vlevo. vpřed vpřed	ja tak
4825	Chlapec	15	9. třída	3	Programování robota	Ano	6	1-7	Ano	Ne		vpred v pravo vpred v pravo vpred vpred vlevo vpred v pravo vpred v levo pred v pred v levo vpred v lpravo vpred vlevo vpred vpred v levo vpred v levo preed	
3437	Dívka	15	9. třída	4	programování je když nekdo něčemu říká co má dělat	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-4	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ano		opakovat dokud nedosáhne sýru vpřed když cesta v před vpřed jinak když cesta vpravo vpravo jinak vlevo	rozhlédnout se když jede auto čekat jinak přejít
1641	Chlapec	15	9. třída	3	posun vpred	Ano	Ano, včetně úspěšně vyřešené dvacítky	3-2	Ne	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu		vpred vpravo vpred pred vpred vlevo vpred vpravo vpred vpred vpred vlevo vpred vpravo vpred vpred vpred vlevo vpred vpravo vpred	prejit na zelenou
2495	Chlapec	14	9. třída	4	moc ne	jen občas	Ano, včetně úspěšně vyřešené dvacítky	1-8	Ne	Ne	hodina byla pohodova aspon nedelame furt jenom prezentace :D	vpřed,vpravo,vpřed,vlevo,vpřed,vpřed,vlevo,vpřed, vpravo,vpřed,vlevo,vpřed,vpřed,vlevo,vpřed,vpravo ,vpřed,vlevo,vpřed,vpřed,vlevo,vpřed,vpravo,vpřed	
4015	Chlapec	16	9. třída	4	Operace se soubory. / Nastavování a zprovoznění programů.	Ano	Ano, včetně úspěšně vyřešené dvacítky	1-6	Podle počtu volného času	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu	S hodinou (2h) nemám žádné problémy, přivítal bych, kdyby se podobné hodiny nebo alpoň tokový styl vyučování se zavedl do běžné výuky informatiky	VPŘED VPRAVO VPŘED VLEVO 2x VPŘED VLEVO VPŘED VPRAVO VPŘED VLEVO 2xVPŘED VLEVO VPŘED VPRAVO VPŘED VLEVO 2xVPŘED VLEVO VPŘED VLEVO VPŘED	Před přechodem silnice se musí rozhlídnout vbejdřív vlevo pak v pravo a znovu v pravo.
7129	Chlapec	13	7. třída	5	příkazy aby se stroj udělala požadovanou věc	Ano	Ano, včetně úspěšně vyřešené dvacítky	Vůbec jsem se k Lightbotovi nedostal/a	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ne	Tato hodina byla super není co vytknout	Vf=vpřed Vp=vpravo Vl=vlevo řešení:Vf,Vp,Vf,Vl,2xVf,Vl,Vf,Vp,Vf,Vl,2xVf,Vl,Vf,Vo,Vf,Vl,2xVf,Vl,Vf,Vo,Vf	podívej se na obě strany jestli nic neje přejdi (nejlepší je jít po přechodu)
6104	Dívka	13	7. třída	1	baqr	Ne	1	3-2	Ne	Ne	baqr	vpřed	baqr
3001	Chlapec	14	7. třída	4	posonuti konkretního obrasku	Ano	20 (ale nedokončil)	1-7	Ano	Ano	zadne	vpred vlevo vpred	vped vlevo vpred
5703	Dívka	14	7. třída	5	Trošku mě to unavelo,ale bavilo mě to,ale nemohla bych to dělat furt :)	Jako ne furt,ale ano	Ano, včetně úspěšně vyřešené dvacítky	1-8	Ne	Ne		Vlevo,před a před	Zastavit je na přechod rozhlídno se vpravo a vlevo.Když nic nejde můžeme přejít,když jede auto vyčkáme až přejede a když zastaví,tak to znamená,že ná ošlí a můžeme přeiít :)
3673	Dívka	13	7. třída	4	Nastavení robota, aby dělal to co cheme	Možná	Ano, včetně úspěšně vyřešené dvacítky	1-8	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Asi ne	Docela dobrá hodina :)	VLEVO, PRUŽINA, VPŘED	Musíš se rozhlédnout!!!! A když nic nepojede, můžeš jít
9250	Chlapec	12	7. třída	4	pohybování postavičkou pomocí bloků.	Ne	20 (ale nedokončil)	Vůbec jsem se k Lightbotovi nedostal/a	Ne	Ano	od 16 levelu už jsem to přestal chápat	vpřed vlevo vpřed vlevo vpřed vpřed vlevo vpřed vpravo vpřed vlevo vpřed vpřed vlevo vpředvpravo vpřed vlevo vpřed vlevo vpřed vpravo vpřed	nejdřív se podívat doleva potom doprava jestli něco nejede a potom přejít
5962	Dívka	13	7. třída	2	Je to něco hodně těžkého třeba programova draka z lega :D	Možná občas ale rozhodně ne každou hodinu	Ano, včetně úspěšně vyřešené dvacítky	1-8	Ne	Ano ale jednou za čas	Ta hodina byla docela znuděná mohlo být míň toho bludiště....	vpřed vpravo 2krát vpřed vlevo vpřed vpravo 2krát vpřed vlevo vpřed vpravo 2krát vpřed vlevo vpřed vpravo	Z minulou odpovědí jsem na plno zpokojena :D :D :D :D :D :D :D
3700	Dívka	14	7. třída	2	Nevím	Ne	19	Vůbec jsem se k Lightbotovi nedostal/a	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu	Žádné připomínky nemám.	vpřed vpravo vlevo vpřed vpřed vlevo	Prostě se rozhlídni na obě strany a pokuť nic nejede přejdi silnici.
6585	Dívka	12	7. třída	4	je to krok po kroku delate veci ktere zadate do pc a robot zacne delat zadane pokyny	Ano	Ano, včetně úspěšně vyřešené dvacítky	1-1	Ano	Ano	bylo to moc zdlouhave ale to musi a musi se hodne přemyšlet	vpřed pravo vpřed vpřed vpřed levo vpřed pravo vpřed vpřed pravo vpřed vpřed vpřed levo vpřed vpřed pravo vpřed levo vpřed vpřed vpřed levo vpřed vpřed pravo vpřed	nevím

Číselný kód	Pohlaví	Věk	Třída	Jak tě bavila tato hodina	Nově vysvětlení co to je "programování"	Chtěl/a by jsi pokračovat v takovýchto hodinách IVT?	Stihnul/stihla jsi projít celý Hour of Code?	Konečná úroveň Lightbota?	Zájem o kroužek programování	Zájem o kroužek LEGO Mindstorms	Připomínky	Popiš myšce cestu k sýru	Jak přejít silnici a nenechat se zajet?
8503	Dívka	13	7. třída	4	že řekneme co má daná věc dělat a ona to udělá (nejdříve se ,ale musí naprogramovat)	nevím	Ano, včetně úspěšně vyřešené dvacítky	1-1	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu		vpřed - vpravo - vpřed - vlevo - 2x vpřed - vlevo - vpřed - vpravo -vpřed - vlevo - 2x vpřed -vlevo - vpřed -vpravo -vpřed -vlevo- 2x vpřed -vlevo - vpřed -vlevo -vpřed	zastaví pře silnicí koukne se vpravo i vlevo a když nejede žádné auto tak může přejít
5195	Dívka	13	7. třída	2	ja bych to nedokázala vysvetli protoze to sama pořádne nechapu	Ne	2	Vůbec jsem se k Lightbotovi nedostal/a	Ne	Ne	hodina me moc nebavila a nebaví me ani hodiny informatiky nejsem počítačoví typ	vpřed, vpravo, vpřed,vlevo, vpřed, vpřed, vlevo, vpřed, vpravo,vpřed a opakuj do té doby dokud se tam nedostanes	rozhlednu se a pak jdu leva, prava, levá, prava, leva,...
9814	Chlapec	14	8. třída	1	Nastavování	Ano	19	1-5	Ano	Ne	cvxcvxcvxc	c	c
8952	Dívka	13	8. třída	4	Ovládání	Ano	Ano, včetně úspěšně vyřešené dvacítky	1-8	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ano		Vpřed,Vpravo,Vpřed,Vlevo,2x vpřed,vlevovpředvpravo,vpřed,vlevo,2x vpřed,vlevo,vpřed,vpravo,vpřed,vlevo,2x vpřed,vpravo,vpřed,vpravo,vpřed	Rozhlédni se a dej pozor na auta
2578	Chlapec	15	8. třída	3	porgramovani programu	Ano	Ano, včetně úspěšně vyřešené dvacítky	1-7	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ne		vpred,vlevo,vpred,vpred,vlevo,vpred	aby se rozhledl a pak to presel
3588	Dívka	15	8. třída	3	Nevím	Ne	20 (ale nedokončil)	1-3	Ne	Ne	Žádné připomínky, ale prostě mě to nebaví.	vpřed, vlevo, vpravo, vpřed, vlevo, vpravo, vpřed, vlevo,vpřed,vlevo,vlevo,vpravo	Dojít k přechodu, rozhlídnout se pokud nic nejede, přejít přes přechod.
9461	Chlapec	15	8. třída	4	Navedl ho na odkazy co jsme probirali tuto hodinu.	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-1	Ano	Ano	Tato hodina mě celkem bavila.. Jelikož jsem se naučil pár věcí do života.	zadám příkaz vpřed, vpravo vpřed vpravo opakování 2x vpřed vlevo vpřed vpravo vpřed opakování 2x vpřed vlevo vpřed vpravo vpřed vlevo 2x vpřed vlevo vpřed vpravo vpřed	To samé jako v minulém dotazníku.
7251	Chlapec	14	8. třída	5	je to zadání různých pokynů.	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-5	Ano	Ano	Žádné.	Vpřed,Doprava,Vpřed,Doleva,2xVpřed,Doleva,Vpřed,Doprava,Vpřed,Doleva,2xVpřed,Doleva,Vpřed, Doprava,Vpřed,Doleva,2xVpřed,Doleva,Vpřed	Otočit Doleva,Doprava a 20x Vpřed
1985	Chlapec	13	8. třída	3	klikáni do klavesnice	mě je to jedno	Ano, včetně úspěšně vyřešené dvacítky	2-4	nevím	Ano	Uvod dobrej ale tohle vsechno znam	Tady sýr není...	Řekl bych Buď opatrnej a at te neprejede auto.
3652	Dívka	14	8. třída	4	vytváření programů, pohyb v programech	Ano	Ano, včetně úspěšně vyřešené dvacítky	1-8	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ne		VPŘED, VPRAVO, VPŘED, VLEVO, VPŘED 2X, VLEVO, VPŘED, VPRAVO, VPŘED, VLEVO, VPŘED 2X, VLEVO, VPŘED,VPRAVO, VPŘED,VLEVO, VPŘED 2X, VLEVO, VPŘED, VPRAVO, VPŘED	
5299	Dívka	14	8. třída	4	Je to, že zadáme počítači nějakou práci a on nám to vyplní podle toho jak isme to zadalv	Občas	15	Vůbec jsem se k Lightbotovi nedostal/a	Možná	Ne		otočit do leva jít o 2 pole vpřed	Rozhlédnu se a poté rovně a co nejrychleji přejdu silnici
9452	Dívka	13	8. třída	5	Skoro stejně jako na začátku. Je to zadání kódu, který řekne danému přístroji co má dělat.	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-5	Ano	Ano	Bylo to moc krátké.	vpřed doprava vpřed doleva vpřed vpřed doleva vpřed doprava vpřed doleva vpřed vpřed doleva vpřed doprava vpřed doleva vpřed vpřed doleva	přejdi silnici, ale nejdřív se rozhlédni a když jede auto tak nechoď a když nejede tak jdi.
2769	Chlapec	14	8. třída	5	Že dokážeme vyrobit robota který bude dělat věci jak chceme.	Ano	16	Vůbec jsem se k Lightbotovi nedostal/a	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu		Vpřed,vpravo,vpřed,vlevo,vpřed,vpřed,vlevo,vpřed,vpravo,vpřed,vlevo,vpřed,vpřed,vlevo,vpřed,vpravo ,vpřed,vlevo,vpřed,vpřed,vlevo,vpřed,vlevo,vpřed	Rozhlédněte se a mkdyž nic nepojede přejděte silnici
4864	Chlapec	14	8. třída	5	je to zadání pokinu počítači co ma dělat	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-5	Ano	Ano	Žádne	1x vpřed,vpravo, vpřed , doleva 3x vpřed doleva ,vpřed doprava vpřed doleva 3x vpřed doleva vpřed doprava vpřed doleva 3x vpřed doleva vpřed doprava vpřed	stejná jako minule
2892	Dívka	14	8. třída	2	ovládání a říkání rozkazů, naprogramovat si to	Ano	15	Vůbec jsem se k Lightbotovi nedostal/a	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu	bylo to hezký	vpřed,vpravo,vlevo,vpřed,vpřed,vlevo,vpřed,pvravo ,vpřed, vlevo,vpřed,vpřed,vlevo,vpřed,vpravo,před,vlevo,vpřed,vpřed	jít,zastavit,otočit vlevo,vpravo,vlevo jít rovně
7250	Dívka	14	8. třída	4	když zadáme počítači nějaké pokyny a on je vyplní dle zadaného programu	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-2	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ne	tato hodina mě hodně bavila a něco jsem se i přiučila	vpřed, vpravo, vpřed, vpravo, vpřed, vpřed, vlevo, vpřed, vpravo, vpřed, vlevo, vpřed, vpřed, vpravo, vpřed, vlevo, vpřed, vpřed, vlevo, vpřed	
1500	Dívka	14	8. třída	4	Zadání rozkazu počítači	Ano	18	Vůbec jsem se k Lightbotovi nedostal/a	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ano	Vše bylo doré a líbilo se mi to	půjdeš vpřed,otočíš se vpravo půjdeš vpřed , otočíš se vlevo půjdeš 2 krát vpřed otočíš se vlevo půjdeš vpřed , otočíš se vpravo půjdeš 2 krát vpřed a otočíšsevlevo, půjdeš rovně , zahneš do leva půjdeš vpřed otočíš se vlevo půjdeš rovně a otočíš se vpravo	rozhlídnout se a přejít
7328	Chlapec	15	8. třída	5	zadávaní pokynů počítači	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-5	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu		VPŘED,VPRAVO,VPŘED,VPRAVO,VPŘED,VPŘED,VLEVO,VPŘED,VPRAVO,VPŘED, VLEVO,VPŘED,VPŘED,VLEVO,VPŘED,VPRAVO,VPŘED,VLEVO,VPŘED,VPŘED,VLEVO,VPŘED	
4925	Dívka	13	7. třída	3	nevím	Ne	Ano, včetně úspěšně vyřešené dvacítky	1-2	Ne	Ne		dopředu doprava dopředu doleva dopředu dopředu doleva dopředu doleva dopředu doleva dopředu do leva dopředu dopředu doleva dopředu	rozhlídnout se jestli neco nejede
1639	Dívka	14	7. třída	5	nevím	Ano	Ano, včetně úspěšně vyřešené dvacítky	Vůbec jsem se k Lightbotovi nedostal/a	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu	Žádne nejsou	V před,v před pak v levo3x v před	1.rozhlídnout jestli nejede auto 2.když jede tak počkat až projede 3.a když projede rozhlídnout znovu a když nejede tak jít
3019	Dívka	13	7. třída	3	počítač který ovládá roboty	Ne	Ano, včetně úspěšně vyřešené dvacítky	1-2	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ne	Hodina byla dobře vysvětlena	vpřed , vpravo , vpřed , vpravo , vpřed , vpřed , vpřed , vlevo , vpřed , vlevo , vpřed , vpřed , vpřed , vlevo , vpřed vpře , vpřed , vpřed , vlevo , vpřed	Díváme se doprava a doleva , pokud nejede auto tak přejedeme , pokud jede tak stůj kde stojíš.
4434	Dívka	12	7. třída	4	Program který je na logiku.	Možná	20 (ale nedokončil)	Vůbec jsem se k Lightbotovi nedostal/a	Ne	Ano	Dnešní hodina byla zábavná i dobrá.	vpřed, vpravo, vpřed, vpravo, vpřed, vpřed, vpřed, vlevo, vpřed, vlevo , vpřed, vpřed, vpřed, vlevo, vpřed, vpřed, vpřed	Rozhlídnout se pořádně doleva a doprava.
7885	Dívka	13	7. třída	4	počítač,který ovládá roboty	Ano	Ano, včetně úspěšně vyřešené dvacítky	1-2	Ano	Ano	Hodina byla skvěle vysvětlena	vpřed,doprava, vpřed, vpřed, vpřed, doleva,vpřed,doleva,vpřed,vpřed, doleva,vpřed,doleva doleva, doleva vpřed	Díváme se doprava a doleva jestli nic nejede a když nic nejede můžeme přejít silnici (nebo nám zastaví auto
2736	Dívka	13	7. třída	4	že to je ten kdo programuje roboty stroje...	nevím	20 (ale nedokončil)	1-4	Ne	Ne		vpřed,vpravo vpřed,vpřed,vpřed,vlevo vpřed,vpřed,vlevo vpřed,vpřed,vpřed,vlevo vpřed,vpřed,vpřed,vpravo vpřed,vpřed,vlevo vpřed,vpřed,vpřed,vpřed,vlevo vpřed,vpřed,vpřed,vpřed	rozhlednu se a jdu.
1884	Dívka	13	7. třída	3	vúbec nevím jak to mám někomu vysvětlit...	Ne	Ano, včetně úspěšně vyřešené dvacítky	1-1	Ne	Ne		vpřed, vpravo,vpred,vlevo ,3x vpřed, vlevo , vpřed , vpravo , 3x vpřed , vlevo , vpravo - opakovat dokud se nedostanu k sýru.	kouknu se vlevo , vpravo a pokud nic nejede , přejdu.
7602	Dívka	13	7. třída	4	Bez programování by vlastně žádný robot nefungoval. Takže dodávka znalostí.	Ano	Ano, včetně úspěšně vyřešené dvacítky	1-2	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu		1x dopředu; 1x vpravo; 2x dopředu; 1x vlevo; 1x dopředu; 1x vlevo; 2x dopředu; 1x vlevo; 1x dopředu; 1x vlevo; 1x dopředu 1x vlevo 2x dopředu	

Číselný kód	Pohlaví	Věk	Třída	Jak tě bavila tato hodina	Nově vysvětlení co to je "programování"	Chtěl/a by jsi pokračovat v takovýchto hodinách IVT?	Stihnul/stihla jsi projít celý Hour of Code?	Konečná úroveň Lightbota?	Zájem o kroužek programování	Zájem o kroužek LEGO Mindstorms	Připomínky	Popiš myšce cestu k sýru	Jak přejít silnici a nenechat se zajet?
7793	Dívka	13	7. třída	5	je to programování panáčka.Panáček je jako robot.Naprogramujete ho a on dělá co mu nakážete.	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-4	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu	Tato hodina se mi moc líbila a klidně bych ji vyměnila za normální hodiny.	nestihla	nestihla
7997	Dívka	13	7. třída	5	Hejbání s bloky a ovládání panáčka..	Ano	Ano, včetně úspěšně vyřešené dvacítky	Vúbec jsem se k Lightbotovi nedostal/a	Ano	Ano		Vpřed , vpřed , vlevo , vpřed , vpřed , vlevo , vpřed , vpřed	
8639	Dívka	13	7. třída	5	hejbání s bloky a sestavovat ,,cestu,,.	Ano	18	Vúbec jsem se k Lightbotovi nedostal/a	Ano	Ano		rovně,do leva,rovně,do leva,do prava,rovně,do leva,rovně,do leva	
9626	Dívka	13	7. třída	5	nevím	Ano	Ano, včetně úspěšně vyřešené dvacítky	Vúbec jsem se k Lightbotovi nedostal/a	Ano	Ano	budete příští rok ve škole	vpřed vpravo vpřed vlevo vpřed vpřed vlevo vpřed vpravo vpřed vlevo vpřed vpřed vlevo vpřed vpřed vpřed vpřed vpřed vpřed	jít na přechod a rozhlédnout se vlevo a vpravo
6465	Dívka	13	7. třída	5	přidávání bloků které říkají věci co mají dělat	Ano	20 (ale nedokončil)	1-3	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu	žádné nemám	vpřed,vlevo,vpřed,vpravo,vpřed,vpřed,vlevo,vpřed,vpravo,vpřed,vlevo,vpřed,vpřed,vlevo,vpřed,vpravo,vpřed,vpřed	rozhlédnout se vlevo pak vpravo pak zase vlevo a když nic nejede může jít
1575	Dívka	13	7. třída	4	Nevím.	Možná	20 (ale nedokončil)	2-3	Možná	Možná	Hodina byla dobrá.	1x rovně 1x vpravo 1x rovně 1x vlevo 2x rovně 1x vlevo 1x rovně 1x vpravo 1x rovně 1x vlevo 2x rovně 1x vlevo 1x vpravo 1x rovně 1x vlevo 2x rovně 1x vlevo 1x rovně 1x vpravo 1x rovně	
2776	Dívka	13	7. třída	4	Ovládání pohybu (činosti) strojů robotů a prostě elektroniky .	Ne vždy ale občas bych si to zopakovala	Ano, včetně úspěšně vyřešené dvacítky	2-4	Možná	Možná	Žádné připomínky nemám ;)	Vpřed, otočit v pravo ,vpřed ,otočit v levo ,2x vpřed ,otočit v levo , vpřed , otočit v pravo ,vpřed , otočit v levo ,2x vpřed ,otočit se v levo ,vpřed ,otočit se v pravo ,vpřed ,2x vpřed ,otočit se v levo ,vpřed ,otočit se v pravo , vpřed.	
4873	Dívka	13	7. třída	3	Stále nevím.	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-2	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ne	Žádné nemám. =D	vpřed doprava vpřed doleva vpřed vpřed doleva vpřed doprava vpřed doleva vpřed vpred doleva vpřed doprava vpřed doleva vpřed vpřed doleva	Najít přechod pro chodce.
7474	Dívka	13	7. třída	3	Nastavování her.	Ne	Ano, včetně úspěšně vyřešené dvacítky	1-1	Ne	Ano		vpřed,do prava,vpřed,do prava,vpřed,do leva,vpřed,do prava,vpřed,do leva,vpřed,do leva,vpřed,do leva,vpřed	POřádně se rozhlédnout do leva a do prava pak teprve můžeme přejít.
8312	Dívka	12	7. třída	5	Programování je ovládání počítače/robota pomocí příkazů.	Ano	Ano, včetně úspěšně vyřešené dvacítky	2-3	Ano	Ano	Hodina byla skvělá, ráda bych se zúčastnila další.	vpřed, vpravo, vpřed, vpravo, vpřed, vpřed, vlevo, vpřed, vpravo, vpřed, vlevo, vpřed, vpřed, vlevo, vpřed, vpravo, vpřed, vlevo, vpřed, vpřed, vlevo, vpřed	Nejdřív se pozorně rozhlédni doleva, doprava a zase doleva. Když nejede žádné auto, můžeš přejít, ale pro jistotu se znovu rozhlédni doleva, doprava a doleva.
2469	Dívka	12	6. třída	3	zeto bylo dobrý	obcas	14	2-3	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ano	tato hodina byla dobra a zabavna	bylo to tobry	bylo to tobry
9330	Dívka	12	6. třída	4	nevím	jen někdy	12	Vúbec jsem se k Lightbotovi nedostal/a	Ne	Ne		vpřed vlevo 2 vpřed vlevo	zastavím rozhlédnu se rozhlédnu se vpravo vlevo když nic neiede mužu iít
6957	Dívka	12	6. třída	2	Programování je to že vlastně ňákého robota naprogramujete a to co mu řeknete to uděl.. Je to vlastně nařízení něčeho.....	Ano	14	Vúbec jsem se k Lightbotovi nedostal/a	Ne	Ne	Žádné připomínky nemám...	Vpřed,Vpravo,vpřed o tři pole,vlevo ,vpřed,vpravo,vpřed o tři pole,vlevo,vpřed,vlevo odvě pole,vlevo,vpravo,vpřed.	Prostě jsi počkat až pojede auto a vběhnout pod auto...
1629	Dívka	12	6. třída	4	jsou to úkoli pro počítače	Ano	Ano, včetně úspěšně vyřešené dvacítky	1-8	Ano	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu	byla to obrovská zábava	vpřed, vpravo, vpřed, vlevo, 2x vpřed, vlevo, vpřed, vpravo, vpřed, vlevo, 2x vpřed, vlevo, vpřed, vlevo, 2x vpřed, vlevo, vpřed, vlevo, vpřed	na kraji silnice se rozhlédnout do prava i doleva.a přejít silnici kolmo naní.
3637	Dívka	12	6. třída	5	Říkáte počítači co má dělat. ovládáte ho svími příkazi.	Ano	Ano, včetně úspěšně vyřešené dvacítky	3-2	Ano	Ano	Hodina mě velmi bavila a mám velký zájem chodit do jakéhokoli zájmového kroužku s podobnou tématikou!	VPŘED, VPRAVO, VPŘED, VLEVO, VPŘED, VPŘED, VLEVO, VPŘED, VPRAVO, VPŘED, VLEVO, VPŘED,VPŘED, VLEVO,VPŘED, VPRAVO,VPŘED, VLEVO,VPŘED VPŘED,VLEVO, VPŘED, VPRAVO, VPŘED.	Stejna odpověď jako v minulem dotazniku.
7011	Dívka	12	6. třída	4	říkáme mu co má dělat	Ano	18	Vúbec jsem se k Lightbotovi nedostal/a	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu	žádné připomínky nemám líbila se mi	1 - vpřed 2- vlevo vpřed vlevo vpřed opakovat až do cíle	otočit doprava jestli nic nejede dále otočit doleva když nic nejede ta jít vpřed , jestli jede tak zastavit a počkat až auto odjede a jít rychle vpřed
6959	Dívka	13	6. třída	3	ovladani počitace	Ano	13	Vúbec jsem se k Lightbotovi nedostal/a	Ne	Ne	dozvěděla sem se neco noveho	1 plicko vpřed vpravo vpřed 2 polícka vlevo 2 vpředu vlevo 3 plicka vpřed vlevo 1 policko vpřed vlevo 3polícka vpřed	nechavam to jako přetím
8545	Dívka	12	6. třída	5	třeba naprogramovaní robota(aby mohl chodit).	Ano	14	Vúbec jsem se k Lightbotovi nedostal/a	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu		Vpřed,vpravo,vpřed,vlevo,vpřed o dvě pole,vlevo,vpřed,vpravo,vpřed,vlevo,vpřed o dvě pole,vpravo,vpřed,vlevo,vpřed,vpravo,vpřed o dvě pole,vlrvo,vpřed,vlevo,vpřed	nechávam stejně, neměním od předchozího komentáře
4432	Dívka	12	6. třída	5	Říkáte počítači,programu co má dělat	Ano	18	Vúbec jsem se k Lightbotovi nedostal/a	Ano	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu	Bylo to super zase jsem se dozvěděla něco nového. :)	vpřed,vpravo,vpřed,vpravo,vpřed,vpřed,vlevo,vpravo,vpřed,vlevo,vpřed,vpřed,vpřed,vlevo,vpřed,vpravo.vpřed. a tak dál	stejné jako v prvním dotazníku :)
9885	Dívka	13	6. třída	5	nevím	Ano	Ano, včetně úspěšně vyřešené dvacítky	1-5	nevím asi ano	nevím asi ano		vpřed , vpravo , vpřed , vlevo , 2 vpřed , vlevo , vpřed , vpravo , vpřed , a tak dál	

I. Výsledky vstupního dotazníku kontrolních skupin

Číselný kód	Pohlaví	Věk	Třída	Oblíba INF	Význam INF	Představa "programování"	Dosavadní zkušenosti	Jsi líže už jsi se setkal/a s programováním, odkud?	Zájem o vyzkoušení programování, v hodinách IVT	Další zájmy z oblasti IVT a počítačů	Popiš myšce cestu k sýru	Jak přejít silnici a nenechat se zajet?
8000	Dívka	11	6. třída	3	2	nějaké programy	Nikdy jsem to nezkoušel/a a ani nevím, co to je	moc jsem se s programováním nesetkál/a ..ani mě to moc nezajímá	Ne	Facebook, Nějaké konkrétní internetové stránky, 2D Grafika (fotky, a další plošné obrázky), Multimédia (muzika a filmy), Počítačové sítě, Prezentace (PowerPoint)	vpřed vpřed vpřed vlevo vpřed vpřed vpřed vpravo vpřed vpřed	rozhlednout se z každe strany jestli nejede auto ...ale nejlépe jít na přechod
9795	Dívka	12	6. třída	2	3	nevím co to je	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Ne	Počítačové hry, Multimédia (muzika a filmy)	vpřed, vpřed, vpřed, vlevo, vpřed, vpřed, vpřed, vlevo, vpřed, vpřed,vpřed	rozhlídni se jestli něco nejede na obě strany. když nic nejede tak můžeš jít.
3714	Dívka	12	6. třída	3	4	Pod programováním si představím že někdo utvář nějaké programy např. počítačové hry nebo prezentace.	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	Ještě ne.	Ano	Nějaké konkrétní internetové stránky, 2D Grafika (fotky, a další plošné obrázky), Multimédia (muzika a filmy), Prezentace (PowerPoint)	Vpřed, vpřed, vlevo, vpřed, vpřed, vpravo, vpřed, vpřed, vpřed.	Musí nejdřív dojít k silnici pak se rozhlédnout 2 do leva a 1 do prava a když nic nejede tak může přejít a když něco jo tak má počkat a pak to opakovat.
7821	Dívka	12	6. třída	3	4	nějaké programy	Nikdy jsem to nezkoušel/a a ani nevím, co to je	od sourozence	Nevím, co to programování je, takže nevím	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky	vpřed,vpřed,vpřed,do leva,vpřed,vpřed,do prava,vpřed,vpře	rozhlédni se jestli nejede auto a jdi přez přechod
1924	Dívka	12	6. třída	3	3	Představím si že je to nějaké programování na počítači. .	Nikdy jsem to nezkoušel/a a ani nevím, co to je	Ne, neselkala jsem se.	Nevím, co to programování je, takže nevím	Počítačové hry, Nějaké konkrétní internetové stránky, Multimédia (muzika a filmy), Prezentace (PowerPoint)	Jdi 3krat vpřed 1krát vlevo, 3krát vpřed, 1krat vlevo a 3krát vpřed.	Nejdřive běž k přechodu potom se rozhlédni vlevo vpravo a zase vlevo, když nic nejede tak přejdi.
7225	Dívka	12	6. třída	4	4	vubec nevím o co tam jde	už jsem to slyšela ale nic o tom nevím	Od brachy	Ano	Facebook	vpřed,vpřed,vpřed,vlevo,vpřed,před,vlevo,před,před	koukni doleva a doprava jestli nic nejede jdi auprosřed silnice se ještě trochu rozhlédni
8742	Dívka	12	6. třída	3	2	nic, nevím co to je	Nikdy jsem to nezkoušel/a a ani nevím, co to je		Ano	Počítačové hry, Facebook, Multimédia (muzika a filmy)	3x vpřed, vlevo, 3x vpřed, vlevo, 3x vpřed	Podívat se 2x doleva, 2x doprava a když nic nepojede tak přejít. čyž něco pojede takpočkat a potom se znovu podivat (2x doleva, 2x doprava)
9351	Dívka	14	9. třída	4	3	Práce s určitým programem.	Zkoušel/a jsem si jednoduché grafické programování a bavilo mě to		Ano	Facebook, Nějaké konkrétní internetové stránky, 2D Grafika (fotky, a další plošné obrázky), Multimédia (muzika a filmy), Prezentace (PowerPoint), Android	Půjde o 3 políčka vpřed, pak se otočí do leva a půjde o další 3 políčka dál. Zhova se otočí do leva a popojde zase o 3 políčka a je u sýru.	
2737	Dívka	15	9. třída	3	3	využívání programů, k vytvoření prezentací, ..	Zkoušel/a jsem klasické programování v nějakém programovacím jazyce	ve škole na hodině informatiky	Ano	Facebook, Blogy, Nějaké konkrétní internetové stránky, Multimédia (muzika a filmy), Prezentace (PowerPoint), Android	vpřed, vlevo, vpřed, vlevo	zastavit se rozhlídnout se vlevo, vpravo, vlevo pak jít a koukat jestli nejede auto
1949	Chlapec	15	9. třída	4	4	nic	Nikdy jsem to nezkoušel/a a ani nevím, co to je	ve škole	Ano	Počítačové hry, Facebook, 3D Grafika, Multimédia (muzika a filmy), Prezentace (PowerPoint)	3krát vpřed pak vlevo,3krát vpřed, vlevo, a 3krát vpřed	na semafor
8684	Chlapec	15	9. třída	3	3	nevím	Nikdy jsem to nezkoušel/a a ani nevím, co to je	ne nesetkal	Nevím, co to programování je, takže nevím	Počítačové hry, Facebook	3xVPŘED,otočit vlevo,3xvpřed,otočit vlevo,3xvpřed.	První se musí podívat do prava pak do leva pak zase do prava a když nic nejede může přejít
5554	Chlapec	15	9. třída	5	5	pracování v různých programeh počítačové jazyky	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	na you tube	Ano	Počítačové hry, Facebook, Blogy, 2D Grafika (fotky, a další plošné obrázky), Tabulkový procesor (MS Excel), Textový editor (MS Word), Prezentace (PowerPoint)	3x vpřed doleva 3x vpřed doleva 3x vpřed	dávat pozor ze všech stran koukinout se doleva do prava koukinu znova do prava pomalu se rozhlížím a koukám do prava a do	
3453	Dívka	15	9. třída	3	3	Nevím	Nikdy jsem to nezkoušel/a a ani nevím, co to je	Vážně nemám potětí co to je!	Ano	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, Počítačové sítě, Prezentace (PowerPoint)	Vpřed 3x, vlevo, vpřed 3x, vlevo, vpřed 3x	Nejdřív se musíme rozhlednoud do leva do prava a zase do leva. Když nic nejede, můžeme silnici přejít. Když jezdí auta, tak čekame až nic nepojede a můžeme přejít.
2671	Dívka	15	9. třída	4	3	Otevřu nějaký program a v tom něco dělam. Jí to něco s grafy.	Zkoušel/a jsem si jednoduché grafické programování, ale nebavilo mě to		Ne	Počítačové hry, Facebook, Tvorba webových stránek, Multimédia (muzika a filmy)	Jde k sýru dopředu, když už tam skoro je tak se otočí o 90 stupnů doleva a jde zase vpřed k sýru stále jde dal až nakonec dojde k sýru	rozhlédnout se doprava, doleva a přejít silnici nejlépe po přechodu.
9200	Chlapec	15	9. třída	4	5	pracování v různých programech	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	na you tubu	Ne	Počítačové hry, Facebook	3 políčka vpřed otočí se na horu o tři políčka vpřed v levo 3 políčka v před	koukáš do leva do prava
1507	Chlapec	14	9. třída	2	4	něco instalovat	Zkoušel/a jsem si jednoduché grafické programování, ale nebavilo mě to		Nevím, co to programování je, takže nevím	Facebook	vpřed, vpřed, vpřed, vlevo, vpřed, vpřed, vlevo, vpřed, vpřed, vpřed	
1975	Dívka	15	9. třída	3	5	Nevím.	Nikdy jsem to nezkoušel/a a ani nevím, co to je	Neselkala	Nevím, co to programování je, takže nevím	Počítačové hry, Facebook, Multimédia (muzika a filmy)	Vpřed, vlevo, vlevo	Pořádně se rozhlédnout jestli nejede auto a přejít.
2454	Chlapec	15	9. třída	3	5	dělání s grafikou	Zkoušel/a jsem si jednoduché grafické programování a bavilo mě to	Hra	Je mi to jedno	Počítačové hry, Facebook, Tvorba webových stránek, 2D Grafika (fotky, a další plošné obrázky)	3x vpřed .. vlevo .. 3x vpřed .. vlevo .. 3x vpřed	Jdi na okraj silnice a několikrát se rozhlídni na obě strany, jestli něco nejede.
6415	Dívka	15	9. třída	3	3	nevím	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a	na internetu	Ano	Facebook, 2D Grafika (fotky, a další plošné obrázky), Multimédia (muzika a filmy), Textový editor (MS Word), Prezentace (PowerPoint), Android	vpřed k políčku č. 4 na políčku č 4 otočit doleva,jít v před na políčko 4 otočit doleva a vpřed	stát na kraji cilnice a podívat se do prava a doleva když nic nepojede ani z jedné strany můžeš přejít
4441	Chlapec	15	9. třída	4	3	Programuje se systém.	Něco málo jsem si o tom zjišťoval/a, ale nikdy jsem programovat nezkoušel/a		Ne	Počítačové hry, Facebook, Nějaké konkrétní internetové stránky, Multimédia (muzika a filmy), Prezentace (PowerPoint), Android	Vpřed,Vpřed,Vpřed,Vlevo,Vpřed,Vpřed,Vlevo,Vpřed,Vlevo,Vpřed	Nejdřív aby se rozhlédnu do leva a pak do prava když nic nepojede může přejít.
4497	Dívka	15	9. třída	3	4	Program v kterém něco připravjú.	Zkoušel/a jsem si jednoduché grafické programování, ale nebavilo mě to		Ne	Facebook, Počítačové sítě, Android	3x vpřed, otočí se 90 stupňů, 3x vpřed, otočí se 90 stupňů, 3x vpřed.	Rozhlédnout se doprava, doleva. Počkat až nepojede auto a nejlépe přejít po přechodě.
7480	Dívka	15	9. třída	3	3	Něco se programuje....	Nikdy jsem to nezkoušel/a a ani nevím, co to je	nikdy	Nevím, co to programování je, takže nevím	Počítačové hry, Facebook, 2D Grafika (fotky, a další plošné obrázky), Multimédia (muzika a filmy), Počítačové sítě, Textový editor (MS Word), Prezentace (PowerPoint), Android	vpřed...vlevo...vlevo	Než pujdes pres silnici tak se rozhlídni.A když nejede zadne auto tak přejdi na druhou stranu.
3290	Chlapec	15	9. třída	3	4	Programovat CNC stroje atd..	Zkoušel/a jsem si jednoduché grafické programování a bavilo mě to	Dělat svůj server	Ano	Počítačové hry, Nějaké konkrétní internetové stránky, Tvorba webových stránek, Hardware (počítače po technické stránce), Tabulkový procesor (MS Excel), Linux	Vpřed, vpřed, vpřed, vlevo, vpřed, vpřed, vlevo	

J. Výsledky výstupního dotazníku kontrolních skupin

Číselný kód	Pohlaví	Věk	Třída	Nově vysvětlení co to je "programování"	Zájem o kroužek programování	Zájem o kroužek LEGO Mindstorms	Připomínky	Popiš myšce cestu k sýru	Jak přejít silnici a nenechat se zajet?
8000	Dívka	11	6. třída	já nevím	Ne	Ne	dnes to bylo super..ale mohli byste nám to lépe vysvětlit	vpřed vpravo vlevo vpřed vpřed vlevo vpravo vlevo vpřed vpřed vlevo vlevo vpřed vpřed vpřed vpravo vpřed	rozhlednout se jestli nejede auto...nejlepe na přechodu
3714	Dívka	12	6. třída	Že se to dělá na prezentaci se tam nastavuje písmo a vkládají obrázky...	Ano	Ne		Vpřed,vpravo, vpravo, vpřed, vpřed, vlevo, vlevo, vpřed, vpřed, vlevo, vpravo, vpřed, vpřed, vlevo, vpřed.	Jako předtím, přijde k silnici podívá se 2 krát doleva a 1 do prava když nic nejede může přejít a když jo tak to opakuje.
8742	Dívka	12	6. třída	nevím, protoženěvím co to programování je.	Ne	Ne		vpřed, doprava, vpřed, doprava, 2x vpřed, doleva, vpřed, doprava, vpřed, 2x vpřed, doleva, vpřed, doprava, vpřed, doleva, 2x vpřed	rozhlédnout se 2x doprava,m 2x doleva a když nejede, tak počkat a znovu se rozhlédnout (2x doleva, 2x doprava)
7821	Dívka	12	6. třída	pořád moc nevím co to je	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ne		vpřed.do prava,vpřed,vpřed,do leva,vpřed,do leva,vpřed.vpřed,do leva,vpřed.vpřed,vpřed,do prava,vpřed	rozhlédni se na obě strany a jdi přez přechod
9795	Dívka	12	6. třída	nevím	Ne	Ne	dnes to bylo dobre, bavilo me to	vpřed,vpravo, vpřed, vlevo, vpřed, vpřed, vpravo, vpřed, vlevo,vpřed.	rozhlídni se, jestli neco nejede, jestli nic nejede tak muzes jít
7225	Dívka	12	6. třída	furt mi to není jasný	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	možná	klidná a fajn hodina	vpřed,doprava,vpřed,vpřed,doleva,vpřed,doleva,vpřed,vpřed,doleva,vpřed,doleva,doprava,doleva,před,před,doleva,vpřed	rozhlédni se,jdi,jdi
1924	Dívka	12	6. třída	Nevím co to je, takže bych to nevysvětlila.	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ne	Bylo to v pohodě.	1 krát vpřed, 1 krát vpravo, 2 krát vpřed, 1 krát vlevo, 1 krát vpravo, 1 krát v před, 1 krát vlevo, 2 krát vpřed, 1 krát vpřed, 1 krát vlevo, 1 krát vpřed, 1 krát vpřed, 1 krát vlevo, 2 krát vpřed, 1 krát vpravo, 1 krát vpřed.	Nejdřív běž k přechodu, rozhlédni se vlevo vpravo a zase vlevo, (podle toho na jaké jsi straně) a přejdi když nejede žádné auto.
5554	Chlapec	15	9. třída	nic	Ano	Ano	nic	doprava rovne doleva 2x rovne doleva nahoru doprava 3x rovne	
4441	Chlapec	15	9. třída	Nevím	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu		Vpřed,Vpravo,Vpřed,Vlevo,Vpřed,Vpřed,Vlevo,Vpřed,Vpravo,Vpřed,Vlevo,Vpřed,Vlevo,Vpřed,,Vpřed,Vpřed,Vlevo,Vpřed	Nejdřív se rozhlednot do leva pak do prava a když nic nejde tak přejít.
6415	Dívka	15	9. třída	nevím	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ano		vpřed, doprava, vpřed, doleva, vpřed, doleva,vpřed, doprava, doleva,doprředu, doprředu, doleva, dopředu, vpřed, vpřed	
9200	Chlapec	15	9. třída	nevím	Ne	Ne	v pohodě	1 políčko rovně dolu rovně 3 políčka rovně nahoru 1 políčko rovně do prava 1 políčko rovně nahoru rovně 3 políčka rovně dolu 1 políčko dolů do leva	podívat se do leva a do prava
2737	Dívka	15	9. třída	Využití programů	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ne		vpřed, vpravo, vpřed, vlevo, vpřed, vlevo, vpřed, vpravo, vpřed, vlevo, vpřed, vlevo, vpřed, vlevo, vpřed	
8684	Chlapec	15	9. třída	nevím co to je takže nevisvětlil	Ne	Ne	nemám žádné	1x vpřed,otočit v pravo,1x vpřed,otočit v levo,2x vpřed,1x vlevo,1x vpravo,2x vlevo,1x vpravo,2x vlevo,1x vlevo,1x vpravo	kouknout se do pvava pak do leva pak zase do prava a když nic nejede tak přejít
3453	Dívka	15	9. třída	Nevím	Ne	Ne	Vážně nevím co to je!!!!!!!!!!- -	uiuuz	Vpřed. v pravo. vpřed. v pravo. vpřed
9351	Dívka	14	9. třída	Nevím	Ano	Ne		O políčko do předu, pak do prava a o dvě políčka do předu. Do leva pak do prava, najoru o dvě políčka a je to	
7480	Dívka	15	9. třída	neco se programuje	Ne	Ne	nebaví me to	egghghgh	vpred...vpravo...vlevo.....vlevo ..vpravo..vlevo...vpred ..vlevo ...vpravo
2671	Dívka	15	9. třída	nějaká práce s programy	Ne	Ne	Je to nudné a zbytečné	trzrtzr	rtzrtzr
1949	Chlapec	15	9. třída	nijak	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu	nic	vpřed.vpravo,vpřed,vlevo,vpřed,vpřed,vlevo,vpřed ,vpravo,vpřed,vlevo,2krát vpřed,vlevo,vpřed,vpravo,2krát vpřed,vlevo,vpřed,vpravo,vpřed.	cekat na semafor.
3290	Chlapec	15	9. třída	nevím	Ne	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu		vpřed, vpravo, vpřed, vlevo, vpřed, vpřed, vpřed, vpravo, vpřed, vlevo, vpřed, vpřed, vpřed	
4497	Dívka	15	9. třída	Práce s programem.	Ne	Ne	Vůbec bych tyto dotazníky nedávala.	vpřed, dolů, doprava až k sýru	
1975	Dívka	15	9. třída	Nevím co to je takže bych to nikomu nevysvětlila.	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ne	Je to úplně zbytečné. Zabralo to čas.	vlevo, vlevo, vpřed, vlevo, vpřed vlevo vpřed vlevo vlevo	Rozhlídnout se
1507	Chlapec	14	9. třída	nevím	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Ne		vpřed, vpravo, vpřed, vpřed, vlevo, vpřed, vlevo, vpřed, vlevo, vpřed, vpřed, vlevo, vpřed	
2454	Chlapec	15	9. třída	dělání s grafikou	Před rozhodnutím bych si chtěl/a vyzkoušet ještě alespoň jednu úvodní ukázkovou hodinu	Před rozhodnutím bych si chtěl/a vyzkoušet alespoň jednu úvodní ukázkovou hodinu	nevím proč to děláme přijde mi to nedůležité	vpřed ..vpravo .. vpřed ..vlevo .. 2x vpřed vlevo vpřed vpravo vpřed vlevo 2x vpřed vlevo vpřed vpravo vpřed vlevo 2x vpřed vlevo vpřed vpřed	rozhlídni se a jdi ..

K. Soubor „Scratch – Instrukce pro žáky“

ODKAZ: SCRATCH.MIT.EDU
ÚKOL: Vytvoř jednoduchou hru, ve které budeš ovládat kocoura, který bude honit myš. Myš bude sama pořád dokolečka běhat po scéně a vždycky, když ji kocour chytí, získá bod. Kocoura bude pronásledovat pes, který bude také ovládaný počítačem, a když kocoura chytí, bod ubere. Kdykoliv, když nebudeš vědět, tak se přihlaš a poradím, případně se potichu zeptej spolužáka!

01.) Nastav kocourovi, aby se po stisknutí šipky nahoru na klávesnici natočil směrem nahoru a posunul se o deset kroků (po stisknutí a podržení to půjde plynule samo)

02.) Nastav kocourovi i všechny ostatní směry (tj, ještě šipku dolů, vpravo a doleva)

03.) Takhle ti ale kocour utíká ze scény, najdi a přidej nějaký příkaz, který by mu v tom zabránil
RADA: takový příkaz je na modré záložce Pohyb a potřebuješ ho přidat více než jednou

04.) Vytvoř příkaz pro restart, který po zmáčknutí mezerníku vrátí kocoura na nějakou pozici, kterou si sám/sama vybereš

05.) Přidej nový objekt, tentokrát to bude myš z knihovny objektů (hledání si můžeš zjednodušit výběrem Kategorie Zvířata, zde je to malá šedivá myš z pohledu svrchu)

06.) Stejně jako u kocoura nastav myši výchozí pozici po restartu a zmenši ji na 60%
RADA: zmenšování je na fialové záložce Vzhled

07.) Po zmáčknutí zelené vlaječky nech myš, aby se rozeběhla a po nárazu do zdi se vždy odrazila a běžela dál (takže bude běhat pořád dokolečka bez zastavení)

08.) Vytvoř příkaz, který vždy, když se myš dotkne kocoura, myš přesune na novou náhodnou pozici
RADA: Skok na pozici znáte, ale můžete do něj vložit náhodně generované číslo. Taková věc se nachází někde na zelené záložce Operátory a vybranou bublinku můžeš přetáhnout přes číslo v modrém příkazu. Náhodně generovaná čísla musíš nastavit pro X od -240 do 240 a pro Y od -180 do 180) KDYŽ NEBUDEŠ VĚDĚT, PŘIHLAŠ SE A PORADÍM

09.) Přidej ještě jeden příkaz, který myš po skočení na náhodnou pozici ještě otočí náhodným směrem
RADA: kolem dokola je to celkem 360°

10.) Na oranžové záložce DATA vytvoř novou proměnnou, u které vyber možnost pro všechny postavy a pojmenuj ji SKÓRE. Po jejím vytvoření se ti objeví několik úplně nových příkazů. Ty využij k tomu, aby se po chycení myši těsně předtím, než skočí na novou náhodnou pozici, tvoje skóre zvýšilo o jedna.

11.) Přidej nový objekt, tentokrát to bude modrý pes z knihovny objektů (hledání si můžeš opět zjednodušit výběrem Kategorie Zvířata, je to takový celý modrý pejsek)

12.) Stejně jako u kocoura a u myši, nastav i pejskovi nějakou výchozí pozici po restartu (například levý horní roh, aby byl na začátku dále od kocoura).

13.) Nastav pejskovi, aby po zapnutí hry zelenou vlaječkou začal kocoura pronásledovat.
RADA: vždy se musí natočit směrem ke kocourovi a posunout se k němu

14.) Dále nastav pejskovi, aby když kocoura chytí, ubral ze skóre jeden bod a tři vteřiny počkal, aby měl kocour šanci mu zase utéct.
RADA: tu instrukci „čekej“ najdeš na žluté záložce Ovládání

15.) Nastav, aby se po restartu skore anulovalo (tj. po restartu se skóre změní na nulu)
RADA: tento příkaz k anulování můžeš přiřadit k restartu jakékoliv postavy a stačí to udělat pouze u jedné z nich

16.) Nastav hře pozadí. Tentokrát to ale nebude pozadí z knihovny objektů, ale ze souboru, který najdeš ve složce na Rku (tedy žák) a zde ve složce Výzkum, která je úplně nahoře, je soubor Pozadí.jpg. Ten si zkopíruj na plochu a nastav ho jako pozadí u této hry. TADY KDYŽ NEBUDEŠ VĚDĚT, TAK SE PŘIHLAŠ A PORADÍM

17.) V tuto chvíli si tebou vytvořenou hru stáhni do počítače pod názvem „**Scratch – Tvé příjmení – Hra.sb2**“ a tento save si přesuň k sobě do své složky!!!

EXTRA 01.) Zkuste kocoura a psa rozhýbat (iluze pohybu se dělá střídáním dvou různých obrázků. Těmto obrázkům se říká kostýmy a tuto možnost najdete na fialové záložce Vzhled)

EXTRA 02.) Můžete si tuto hru zkusit upravit třeba pro dva hráče. Jeden bude ovládat kocoura a druhý psa nebo myš.

EXTRA 03.) Můžete si zkusit vytvořit jakoukoliv vlastní hru. S jakýmkoliv případným nápadem rád poradím.

L. Seznam odučených hodin a další detaily

PRVNÍ TÝDEN
PILOTNÍ SKUPINY
úterý 2.6. - 8.A, 3. a 4. vyučovací hodina, primární skupina dívky, ve třídě 5 žaček
úterý 2.6. - 9.C, 7. a 8. vyučovací hodina, primární skupina dívky, ve třídě 12 žaček a 3 žáci
- Nejrychlejší vyplnění dotazníku 7 min, průměr kolem 10ti minut a nejpomalejší 19 minut.
- Nejrychlejší průchod Hour of Code 32 minut, průměr 45-50 min., nejpomalejší nestihli (celkem tři žáci).
- Nejrychlejší se dostali s Lightbotem do úrovně 2-5 (celkem čtyři žáci), ale nebyl závěrečný dotazník (v ostré části výzkumu do výstupního dotazníku přidat otázku jak daleko se žáci dostali).
- Přípomínky od žáků stran srozumitelnosti nebyly, ale je potřeba pozměnit původní pořadí průběhu hodiny (jmenovitě dát ukázková videa LEGO Mindstorms až za Hour of Code před druhý dotazník, logičtěji to navazuje a žáci lépe pochopí jakým způsobem se například krokodýl ovládá).
- Nutné přidat průběžné vysvětlení jak fungují bloky opakování a podmínky!!! (videa zásadně nečtou).
- Nová updatovaná verze Hour of Code NENÍ v češtině (posledních pět nových úrovní s motivy Ice Age není přeloženo), nalezena starší originální celá česká verze kde jsou Angry Birds a Plants vs Zombies.

VLASTNÍ VÝZKUM
středa 3.6. - 6.C, 1. a 2. vyučovací hodina, sloučená skupina, 21 žáků na 16ti PC – PŘÍŠTĚ SCRATCH!
středa 3.6. - 7.A, 5. a 6. vyučovací hodina, primární skupina chlapci, ve třídě 15 žáků
čtvrtek 4.6. - 9.A, 3. a 4. vyučovací hodina, sloučená skupina, ve třídě 20 žáků – PŘÍŠTĚ SCRATCH!
čtvrtek 4.6. - 7.C, 7. a 8. vyučovací hodina, primární skupina dívky, ve třídě 14 žaček a 3 žáci
pátek 5.6. - 8.B, 5. a 6. vyučovací hodina, sloučená skupina, ve třídě 17 žáků - PŘÍŠTĚ SCRATCH!
- Zatím celkem jeden žák znal jak Hour of Code tak Lightbota a dalších pět žáků tyto projekty sice neznalo, ale bavilo je to natolik, že v řešení pokračovali i po vyplnění druhého dotazníku, kdy si mohli dělat co chtějí (jeden z nich se také ještě před zadáním výstupního dotazníku dostal ze všech žáků vůbec nejdále a to až do úrovně 3-2).
- V jedné sedmičce, kde byly jen žačky, proběhla tato dvouhodinovka jako odpoledka a už je to nebavilo, takže se sice pořád celkem slušně snažily, ale protestovaly. Proto Lightbota, který je podstatně těžší, mohly dělat ve dvojicích. Toto byla jediná výjimka, která se ale výborně osvědčila.
- Pozn. ke všem skupinám, které zkoušeli Hour of Code v obou týdnech: Žáci všech ročníků neradi čtou a vyklikávají dokonce i kratičká videa s textem. Potom velice často nechápou a zasekávají se.

DRUHÝ TÝDEN
KONTROLNÍ SKUPINY
pondělí 8.6. – 6.B, 1. a 2. vyučovací hodina, KONTROLNÍ skupina (dělali prezentace)
pondělí 8.6. – 9.B, 5. a 6. vyučovací hodina, KONTROLNÍ skupina (fotili se na tablo)

POKRAČOVÁNÍ VLASTNÍHO VÝZKUMU
středa 10.6. - 6.C, 1. a 2. vyučovací hodina, sloučená skupina, ve třídě 22 žáků, **SCRATCH**
- stihli jen úvodní ukázkový úkol, šestka je extrémně pomalá a nespolupracující, možná to je ale výjimka
středa 10.6. - 7.A, 5. a 6. vyučovací hodina, primární skupina dívky
čtvrtek 11.6. - 9.A, 3. a 4. vyučovací hodina, sloučená skupina, ve třídě 19 žáků, **SCRATCH**
- Někteří žáci stihli vše, jiní zdržovali. Zhruba třetina byla naprosto vynikající a ještě sami pokračovali a zkoušeli si různé možnosti a varianty. Třetina se Scratchem propracovala celkem bez problémů, ale také bez nějakého většího zájmu a poslední třetina byla příšerná a extrémně problematická.
- Pomalí žáci zdržují, nedá se pracovat se všemi stejně rychle. Tři skupiny (dvojice až trojice), které to extrémně bavilo sice všechno stihli a ještě se sami snažili si něco vytvořit a nebo upravit, ale stejně i jim pomalé skupiny hodinu vysloveně kazily. Řešení do příště: připravit instrukce podle kterých mohou všichni opět pracovat vlastním tempem a pomáhat jen v nejproblematičtějších pasážích.
čtvrtek 11.6. - 7.B, 5. a 6. vyčovací hodina, primární skupina dívky
pátek 12.6. - 6.A, 1. a 2. vyučovací hodina, primární skupina dívky, ve třídě 11 žaček
pátek 12.6. - 8.B, 5. a 6. vyučovací hodina, sloučená skupina, ve třídě 15 žáků, **SCRATCH**
- Příprava instrukcí se vyplatila, hodina byla naprosto vynikající, všichni pracovali, všechny to bavilo, nikdo se nezdržoval a nikdo se nenudil, podstatně lepší ohlas než v devítce.