

PROGRAMOVÁNÍ NA ZÁKLADCE



METODICKÁ PŘÍRUČKA

Tato metodická příručka slouží jako návod, jak s žáky postupovat při průchodu navrženým modelem (online výukovými materiály). Podrobněji popisuje navrhované postupy pro učitele, požadavky na nutné vybavení, způsoby ověření výstupů jednotlivých částí, možná řešení i návrh přidělování odměn.

Návrh modelu včetně pořadí aktivit svou strukturou reflektuje kurikula organizace Code.org (CS Principles Curriculum Guide) a publikace Computing At School (Computing in the national curriculum: A guide for primary teachers).

PŘÍSTUPY K MATERIÁLŮM PRO ŽÁKY A UČITELE

Přístup k online materiálu pro žáky a učitele na: <<https://www.3zskadan.cz/prgm/>>

Všechny hotové Scratch projekty pro učitele jsou dostupné ve studiu „PRGM – U“ na <<https://scratch.mit.edu/studios/4953534/>>.

Všechny Scratch projekty pro žáky jsou dostupné ve studiu „PRGM – pro žáky“ na <<https://scratch.mit.edu/studios/4927549/>>.

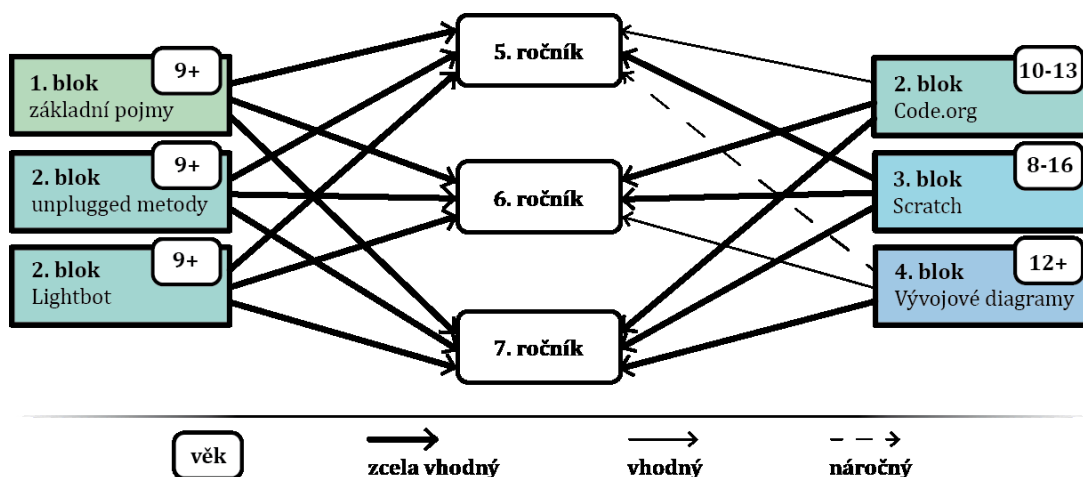
OBECNÝ CÍL

Žák si osvojí elementární znalosti a dovednosti z oblasti algoritmizace a programování s využitím vizuálních programovacích jazyků Scratch a Blockly, herních aplikací a nástroje pro vytváření vývojových diagramů PS Diagram.

PLÁN NASAZENÍ

Návrh je vytvořen v obecné rovině, jako úvod do algoritmizace a programování pro začátečníky, bez vymezení konkrétního ročníku. Jeho obsah je upraven pro věkovou skupinu 11 až 13 let.

Možnosti nasazení jednotlivých výukových bloků do ročníku jsou zobrazeny ve schématu níže (Obrázek 1).



Obrázek 1: Navrhované možnosti nasazení bloků do ročníků

OČEKÁVANÉ VÝSTUPY

Očekávané výstupy, učivo a hodinová dotace modelu (kurzu) jsou k dispozici v podobě časově-tematického plánu na <<https://www.3zskadan.cz/prgm/gamifikace/vystupy.pdf>>.

BLOKY A AKTIVITY

V úvodu každého bloku je na webu pro žáky popsán cíl, kterého by měli dosáhnout. V úvodu každé aktivity je na webu rovněž popsán:

- konkrétní cíl,
- potřebné nástroje,
- náročnost aktivity v podobě hvězdiček,
- vymezený čas,
- informace o maximálním počtu získaných puzzle bloků (viz Gamifikace),
- startovač, který by měl žáky aktivovat,
- postup – konkrétní popis činnosti.

Všichni žáci musí mít jasnou představu, co je cílem aktivit, co přesně se od nich v průběhu aktivit očekává, co konkrétně mají udělat.

Žákům dejte dostatečný prostor k testování svých programů, a to především ve Scratchi. Pečlivým testováním mohou žáci přijít na chyby, které nejsou na první pohled patrné.

Do úvodu každé aktivity jsou zařazeny vstupní požadavky na žáka. Obtížnost aktivit postupně roste, proto by bylo vhodné se vstupních požadavků držet. Žáci, kteří nezvládli předchozí aktivitu, budou mít zřejmě problémy zvládnout také aktivitu následující. Je však jen na učiteli, zda bude vstupní požadavky na žáka brát v úvahu (nebezpečí demotivace žáků).

V úvodu každé aktivity je připraven tzv. „startovač“, který by měl žáky aktivovat a přiblížit jim problém, na nějž je celá část (nebo projekt) zaměřena. V úvodu žáky seznámte s podmínkami pro získání puzzle dílků – návrhy jsou předloženy v závěrečné tabulce každé části.

V závěru každé aktivity shrňte nově získané poznatky, porovnejte jednotlivé projekty (pokud byly tvořeny), udělte puzzle dílky, odznaky či karty.

Většina aktivit umožňuje žákům pracovat ve dvojicích. Pokud to bude možné, nechte žáky řešit problémy právě ve dvojicích, a to především z důvodu osvojování komunikativních a sociálních kompetencí.

Většina aktivit se váže na tvorbu kódu, kterým je řešen určitý problém. Jeden z dvojice kód tvoří, druhý kód nahlas čte. Oba musí rozumět tomu, proč se program chová tak, jak se chová, oba z dvojice musí být aktivní.

Žáci však v případě vzájemných neshod mohou pracovat samostatně.

Žáci v rámci tvorby algoritmů a programů (především blok zaměřený na Scratch a Code.org) využívají pouze známého matematického aparátu – porovnávání celých čísel, sčítání, odčítání, násobení a dělení (zbytek po dělení) celých čísel. Například aplikace goniometrických funkcí není přípustná.

GAMIFIKACE

Žáci jsou odměňováni za úspěšné splnění jednotlivých aktivit virtuálními puzzle dílky. V závěru každé části jednoho ze čtyř bloků pak mají možnost obdržet odznak zvaný „*stickerka*“. Ten je však podmíněn počtem nasbíraných puzzle dílků. V závěru druhého, třetího a čtvrtého bloku mají možnost získat sběratelskou kartu, jejíž získání je podmíněno určitým počtem puzzle dílků. Pokud je počet nasbíraných puzzle dílků menší, žáci získávají odznak nazvaný „*Code Fighter*“. Také v průběhu jednotlivých aktivit sbírají celkem tři odznaky nazvané „*Debugger*“, a to nezávisle na počtu puzzle. K jejich zisku je zapotřebí pouze projevit aktivitu a snahu tvořit. Součástí jednotlivých částí jsou těžší aktivity nazvané „*Hard Core mód*“, které cílí především na nadané žáky. Za splnění těchto aktivit žáci získají jednu ze tří „*top programátorských*“ karet. Na závěr kurzu všichni účastníci získají finální kartu „*PRGM*“, a to bezpodmínečně.

Vše je patrné při průchodu webovou stránkou, ale kompletní gamifikační plán v podobě jednoduché infografiky je navíc k dispozici na:

<<https://www.3zskadan.cz/prgm/gamifikace/gamifikace.pdf>>.

Všechny potřebné odznaky a karty jsou v tisknutelné podobě k dispozici na:

<<https://www.3zskadan.cz/prgm/gamifikace/odmeny.pdf>>.

ZPŮSOBY ŘEŠENÍ ÚLOH

Navrhovaná řešení jednotlivých úkolů (především ve 3. bloku) nejsou jedinou variantou, jak problém vyřešit. Nebazírujte na přesně stejných postupech, na totožných programech. Naopak, nechte děti tvořit a hledat vlastní způsoby, kterými docílí správného výsledku. Případně je upozorněte na redundantní části vytvořeného programu.

OBSAH

1. blok	6
Důležité pojmy ze světa algoritmů a programování	6
2. blok	8
#1 – Origami	8
2. blok	10
#2 – Bitmapový stroj	10
2. blok	12
#3 – LEGO algoritmus	12
2. blok	14
#4 – Příšerky	14
2. blok	16
Seznámení	16
2. blok	17
úkol 1	17
2. blok	19
úkol 2	19
2. blok	21
úkol 3	21
2. blok	23
Code.org kurz	23
3. blok	27
#0 – úvod	27
3. blok	31
#1 – sedm výzev	31
3. blok	33
#2 – hra	33
3. blok	35
#3 – soutěže	35
3. blok	37
#4 – kalkulačka	37
3. blok	39
#5 – Math Masters	39
3. blok	41
#6 – podprogram s parametry	41
3. blok	43
#7 – herní prvky	43

3. blok	46
#8 – vlastní projekt.....	46
4. blok	48
Pravidla a symboly.....	48
4. blok	50
Čteme a tvoříme diagramy	50
4. blok	52
PS Diagram #1 – úprava	52
4. blok	55
PS Diagram #2 – tvorba.....	55
4. blok.....	57
PS Diagram #3 – želvička a hvězda	57

1. BLOK

Důležité pojmy ze světa algoritmů a programování

čas: 60 minut

CÍL

Žák rozumí pojmům **algoritmus**, **program**, **programování**, **programovací jazyk** a zná **vzájemné souvislosti** mezi nimi.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Nejsou vyžadovány.

OČEKÁVANÉ VÝSTUPY

Žák pochopí, co jsou to algoritmy; zná souvislost mezi algoritmem a programem.

PŘEHLED

Žákům jsou vysvětleny pojmy:

- algoritmus
 - základní vlastnostmi algoritmu, které je třeba dodržovat
 - způsoby zápisu
 - chování stroje (Sheldona) řídicího se „*algoritmem přátelství*“
- program
- programování
 - cyklus tvorby programu
- programovací jazyk
- vývojový diagram

POŽADAVKY NA HARDWAROVÉ A SOFTWARE VYBAVENÍ

Počítač s připojením k Internetu, mobilní telefon s nainstalovanou aplikací Plickers.

POSTUP

Žákům v úvodu položte otázku, zda dokáží vysvětlit některý z pojmů. Nechte žáky nad vybranými pojmy a souvislostmi mezi pojmy diskutovat. Učitel má pozici moderátora. S žáky následně pojmy shrňte a případné nejasnosti vysvětlete.

Je důležité, aby žáci pochopili význam pojmů a vzájemné vztahy mezi nimi, především pak vztah: **algoritmus – programovací jazyk – program**.

OVĚŘENÍ

Probíhá formou tvorby pojmové mapy (ve dvojicích, na papír nebo některé z dostupných online aplikací), případně formou otevřených nebo uzavřených otázek prostřednictvím mobilní aplikace **Plickers** <<https://get.plickers.com/>>.

Návrh možných otázek a správných odpovědí:

Algoritmus je...	A) postup, který řeší jen jednu konkrétní situaci. B) postup, který řeší určitý druh problému. C) programovací jazyk. D) Nevím.
Které z následujících tvrzení není pravdivé?	A) program je algoritmus zapsaný v prog. jazyce B) program vytváří programátor. C) počítačová hra není program. D) program je sada příkazů, kterou se počítač řídí, a tyto příkazy vykonává.
Mezi hlavní vlastnosti algoritmu nepatří ...	A) po každém kroku musí být jasné, co bude následovat B) lze použít na celou skupinu problémů C) nemusí nikdy skončit D) musí vyjít výsledek v souladu se zadáním
Programovací jazyk je...	A) algoritmus, zapsaný ve vývojovém diagramu. B) operační systém. C) prostředek, kterým programátor vytváří program. D) Nevím.

JINÉ UČEBNÍ TEXTY

PDF dokument shrnující pojmy 1. bloku je k dispozici na <<https://www.3zskadan.cz/prgm/doc/Pojmy.pdf>>.

Návrh odměny

za 70% úspěšnost v testu nebo
za kvalitně vytvořené pojmové mapy je udělen **odznak „New“**

Z. BLOK

Unplugged (výuka bez počítače)

#1 – Origami

čas: 30 minut

náročnost:



CÍL

Žák formuluje instrukce přesně a jednoznačně.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Žák zná vlastnosti algoritmu, chápe souvislosti mezi algoritmem, programovacím jazykem a programem.

OČEKÁVANÉ VÝSTUPY

Žák pochopí důležitost přesných a jednoznačných pokynů v programu.

Žák využívá inženýrského myšlení v aktivitách bez počítače.

PŘEHLED

Žáci se v této části zabývají popisem, který vede k vytvoření jednoduché skládačky.

Žáci by měli vyřešit problém rozkladem na nejjednodušší kroky a soustředit se na důležité části problému.

POŽADAVKY NA VYBAVENÍ

Papír (origami papír), psací potřeby.

POSTUP

Žáci nejdříve odpovídají na otázku: „*Bude náročnější origami složit nebo popsat, jak se skládá? Proč?*“ V souvislosti s odpovědí je vhodné připomenout vlastnosti algoritmů, především jednoduchost a determinovanost. Součástí startovače je úloha vybraná ze soutěže Bobřík informatiky, která je s aktivitou totožná.

Dále žáci skládají jednoduché origami z papíru a zapisují slovně pracovní postup. **Postup nesmí kreslit!** Proto je důležité, aby zvolili opravdu jednoduchou skládačku, jejíž stavbu budou schopni popsat (vlaštovka, čepice atp.). K popisu mohou využít příkladu ze startovače, vymyslet vlastní funkční způsob nebo stavbu popisovat větami. Následně si své postupy vymění se spolužáky. Ti podle obdržených postupů zkusí origami složit.

Origami představuje aktivitu, ve které se žák soustředí na to, co je opravdu důležité (abstrakce).

Instrukce pro žáky

- Vyberte si velmi jednoduchou skládačku z papíru. Na fotce je již náročný příklad.
- Zapisujte v krocích, jak se má při stavbě postupovat, aby ostatní byli schopní skládačku správně složit. Nekreslete! Zkuste vymyslet vlastní zápis.
- Během popisu je důležité, abyste postup skládání rozfázovali na velmi jednoduché kroky. V popisu kroků buďte opravdu přesní.

OVĚŘENÍ

Ověření probíhá v rámci skládání podle cizího postupu. V případě nepřesného či nejednoznačného zápisu je vhodné, aby dvojice mezi sebou prodiskutovaly, v čem je zápis problematický, případně navrhy lepší řešení – přesnější formulaci.

V případě, že žáci nemohou nalézt správné řešení, je nutný zásah pedagoga.

JINÉ UČEBNÍ TEXTY

Nejsou k dispozici.

Návrh odměny

Za splnění obou částí úkolu žák získá 3 puzzle dílky.

Za splnění jedné z částí úkolu žák získá 2 puzzle dílky.

Za snahu (nesplní ani jednu část úkolu) žák získá 1 puzzle dílek.

max. 3
puzzle dílky

ZDROJE A INSPIRACE

Ve stratovači je žákům předložena konkrétní úloha „*Folding paper*“ z mezinárodní soutěže The Beaver (v Česku Bobřík informatiky) dostupná na webu Beaver Informatics Malaysia [cit. 2018-11-08]. První aktivita je inspirována touto úlohou.

Z. BLOK

Unplugged (výuka bez počítače)

#2 – Bitmapový stroj

čas: 30 minut

náročnost:



CÍL

Žák formuluje instrukce přesně a jednoznačně.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Žák chápe nutnost přesných a jednoznačných pokynů v programu.

OČEKÁVANÉ VÝSTUPY

Žák pochopí důležitost přesných a jednoznačných pokynů v programu.

Žák využívá infromatického myšlení v aktivitách bez počítače.

PŘEHLED

Žáci se v této části zabývají tvorbou programu pomocí zadaných grafických instrukcí, jež vedou k vytvoření obrázku.

POŽADAVKY NA VYBAVENÍ

Čtverečkované papíry nebo https://www.3zskadan.cz/prgm/doc/bitmapa_bianko.pdf, psací potřeby.

POSTUP

Žáci nejdříve odpovídají na otázky: „*Jak se v počítačové grafice říká jednotlivým čtverečkům? Jaké mají vlastnosti?*“ V případě neznalosti by bylo vhodné alespoň zběžně žákům představit princip tvorby rastrového obrázku – pojmy pixel a rastrová mřížka.

Součástí startovače jsou dva obrázky k prostudování. Dejte žákům čas (1–2 minuty) na pochopení smyslu zápisu.

V obrázcích se objevuje cyklus „*opakuj to, co je v závorkách*“. Pokud žáci zcela nepochopí princip na obrázku ve startovači, je vhodné jej nahradit jednotlivými opakujícími se příkazy nebo jiným zápisem.

S žáky projděte připravený příklad s možným zápisem. Žáci pracují podle postupu zadaného na webu.

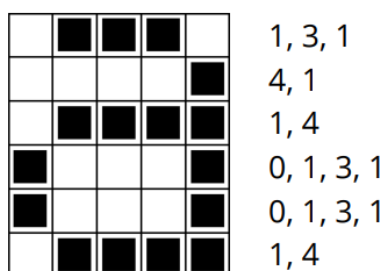
Nejprve vytvářejí vlastní obrázek, který vzniká vybarvováním jednotlivých pixelů v rastrové mřížce 12×12 px. Poté žák sestaví instrukce, jež vedou k vytvoření obrázku, a to s pomocí předem určených příkazů. Sepsané instrukce si žáci mezi sebou vymění a podle nových

instrukcí zpětně tvoří obrázky svých spolužáků. Na závěr je důležité porovnat původní obrázek s nově vzniklým. V případě neshod žáci zkontrolují instrukce a najdou chybu.

Žáci se nemusí držet jen dvoubarevného obrázku. Mohou vyzkoušet více barev, ale budou muset doplnit seznam instrukcí o změnu barvy. Také navrhovaný zápis směru postupu nemusí být jedinou ani ideální volbou. Je vhodné s žáky debatovat nad dalšími možnostmi tvorby zápisu, například svisle, po řádcích zleva, ze středu, atp.

Alternativní postup

Pokud výše zmíněný postup není pro žáky výzvou, zaměřte aktivitu na obtížnější alternativu, kterou přináší Bell, Witten a Fellow (2015, s. 16–25) v představení druhé unplugged aktivity *Color by numbers – Image representation*. Způsob zápisu je patrný na obrázku níže (Obrázek 2).



Obrázek 2: Repräsentace bitmapového obrázku s použitím číselného kódování (Bell, Witten a Fellow, 2015)

Navrhovaná alternativa může být žákům představena jako reakce na otázku: „Navržený postup vpravo není jedinou možností, která by se dala využít. Napadne vás jiný způsob?“

Tato alternativa je vhodná pro nadané děti.

OVĚŘENÍ

Ověření probíhá v rámci tvorby obrázku podle cizích instrukcí. V případě chyby je vhodné, aby dvojice mezi sebou probraly, v čem chyba tkví a společně ji opravily. V případě, že se žáci nemohou shodnout, je nutný zásah pedagoga.

JINÉ UČEBNÍ TEXTY

Nejsou k dispozici.

Návrh odměny	
Za splnění obou částí úkolu žák získá 2 puzzle dílky.	max. 2 puzzle dílky
Za splnění jedné z částí úkolu žák získá 1 puzzle dílek.	

ZDROJE A INSPIRACE

Aktivita je inspirována první lekcí kurzu D dostupnou na webu organizace Code.org [cit. 2018-11-08].

Z. BLOK

Unplugged (výuka bez počítače)

#3 – LEGO algoritmus

čas: 30 minut

náročnost:



CÍL

Žák formuluje instrukce přesně a jednoznačně.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Žák chápe nutnost přesných a jednoznačných pokynů v programu.

OČEKÁVANÉ VÝSTUPY

Žák pochopí důležitost přesných a jednoznačných pokynů v programu.

Žák využívá infromatického myšlení v aktivitách bez počítače.

PŘEHLED

Žáci se v této části zabývají vytvářením vhodného způsobu popisu, který vede k sestavení konstrukce z LEGO kostek.

Žáci by měli vyřešit problém rozkladem na nejjednodušší kroky a soustředit se na důležité části problému.

POŽADAVKY NA VYBAVENÍ

LEGO, psací potřeby, papír.

POSTUP

V rámci startovače jsou žáci pobídnuti k zápisu postupu konkrétní konstrukce. Diskutujte s žáky o vhodném a nevhodném principu zápisu, co všechno by mělo být jeho součástí... barva, popis tvaru, posun pozice aj.

POPIS BLOKU (BARVA, POSUN PO OSÁCH, OTOČENÍ)

Začátek

První dva-čtyřka (červená, pozice volná)

Na první druhá dva-čtyřka (červená, 2 vlevo, 0 vpřed, 0°)

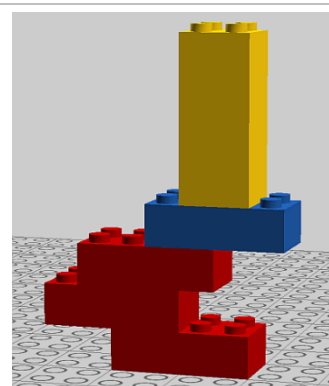
Na druhou třetí dva-čtyřka (červená, 1 vpravo, 0 vpřed, 0°)

Na třetí čtvrtá dva-čtyřka (modrá, 2 vpravo, 0 vpřed, 0°)

Na modrou dva-dvojka (žlutá, na střed (1 vpravo-1 vlevo volná), 0 vpřed, 0°)

Opakuj 3× Na žlutou dva-dvojka (žlutá, 0 vpravo, 0 vpřed, 0°)

Konec



Tabulka 1: Možný zápis

Takový zápis (Tabulka 1) by měl zaručit správné sestavení. Žáci by však měli vymyslet vlastní.

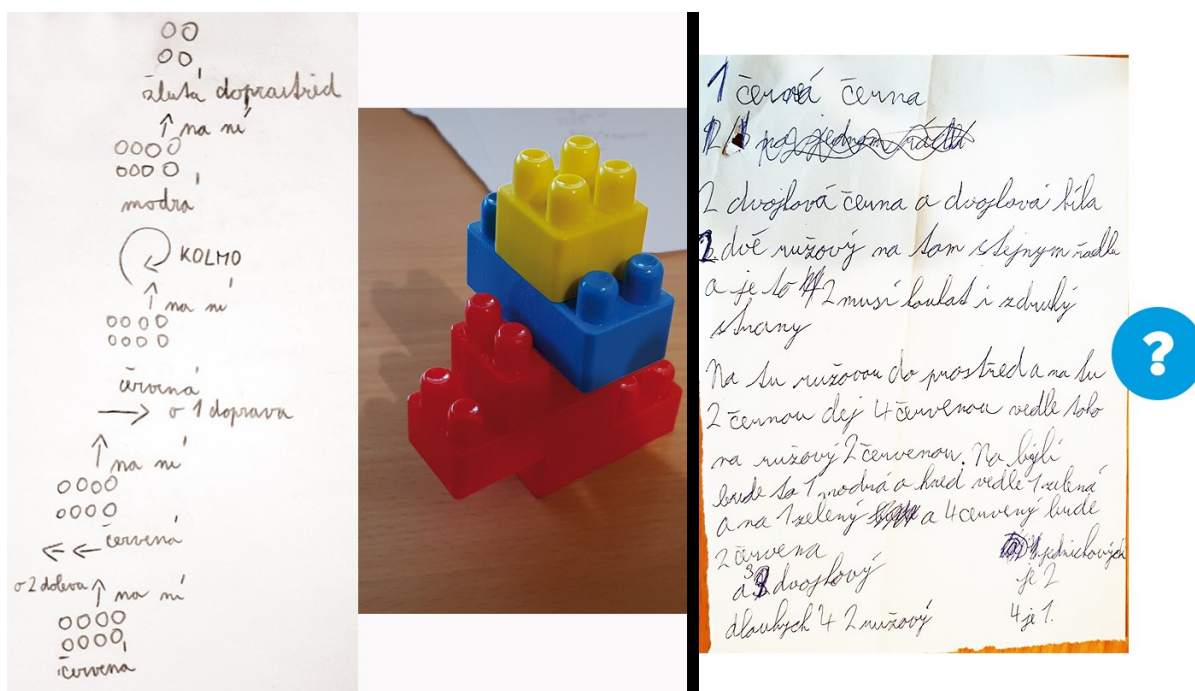
Instrukce pro žáky

Žáci pracují podle zadaného postupu. Na webu je podmínka použití maximálně 10 kostek, ale i tento počet je vysoký. Je vhodnější konstruovat s menším počtem kostek (asi 5 kostek).

Zdůrazněte žákům, že jednotlivé dílky konstrukce musí být propojeny... **bloky nesmí být zcela volně položeny**. Plán konstrukce by neměl obsahovat obrázky postavené konstrukce.

OVĚŘENÍ

Ověření probíhá v rámci stavby konstrukce z LEGO kostek podle cizího zápisu postupu. V případě nepřesného či nejednoznačného zápisu je vhodné, aby dvojice mezi sebou probraly, v čem je zápis problematický a navrhy lepší řešení – přesnější formulaci. V případě, že žáci nemohou nalézt správné řešení, je nutná podpora pedagoga.



Obrázek 3: Ukázka dobrého zápisu (vlevo) a špatného zápisu (vpravo)

Návrh odměny

Za splnění obou částí úkolu žák získá 3 puzzle dílky.

Za splnění jedné z částí úkolu žák získá 2 puzzle dílky.

Za snahu (nesplní ani jednu část úkolu) žák získá 1 puzzle dílek.

max. 3
puzzle dílky

ZDROJE A INSPIRACE

Aktivita je inspirována první kapitolou ze třetí části „CS Principles 2018: Intro to Programming“ kurikulárního dokumentu Code.org [cit. 2018-11-08].

Z. BLOK

Unplugged (výuka bez počítače)

#4 – Příšerky

čas: 15 minut

náročnost: ☆☆☆

CÍL

Žák najde pravidlo, podle kterého jsou objekty roztržiděny.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Nejsou vyžadovány.

OČEKÁVANÉ VÝSTUPY

Žák využívá inforinatického myšlení v aktivitách bez počítače.

PŘEHLED

Žáci v této části hledají pravidla, podle kterých je možné třidit jednotlivé objekty. Hledají vlastnosti, podle kterých jsou objekty popsateľné. Žáci se soustředí na to, co je důležité (abstrakce).

POŽADAVKY NA VYBAVENÍ

Skupina obrázků jednotlivých postavicek je dostupná na:

<https://www.3zskadan.cz/prgm/img/HLAVNI/priserky_nuzky.pdf>.

POSTUP

Žáci nejdřívě odpovídají na otázku: „V čem se na první pohled mohou lišit dva lidé?“ Výškou, hmotností, věkem, pohlavím aj.

Žáci odpovídají na otázku: „Podle jakého pravidla jsou roztržiděny příšerky na obrázku?“ Dále hledají jiná pravidla, podle kterých je možné příšerky třidit. Vybrané pravidlo použijí na roztržidění nastřihaných příšerek. Následně zjišťují pravidla vybraná spolužáky.

OVĚŘENÍ

Ověření probíhá v rámci celé třídy. Žák rozhodne, podle jakého pravidla roztržidil příšerky jeho spolužák.

JINÉ UČEBNÍ TEXTY

Nejsou k dispozici.

Návrh odměny

Za vytvoření vlastního pravidla třídění a správné aplikace žák získá 1 puzzle dílek.

Za nalezení pravidla použitého v příkladu spolužáka žák získá 1 puzzle dílek.

max. 2
puzzle dílky

Za 7 a více získaných puzzle dílků je udělen **odznak „Unplugged“**

Celkem
max. 10
puzzle dílků

ZDROJE A INSPIRACE

Aktivita vychází z úlohy Tomcsányiové, která byla zařazena do Bobříka informatiky 2012 a publikovaná v BOBŘÍK INFORMATIKY: Výběr z úloh národních kol soutěže 2010–2014.

Z. BLOK

Lightbot

Seznámení

celkový čas: 180 minut

CÍL

Žák vytvoří a odladí program podle konkrétních požadavků.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Nejsou vyžadovány.

OČEKÁVANÉ VÝSTUPY

Žák rozpozná opakující se vzory, používá opakování a připravené podprogramy.

Žák navrhne, napíše a odladí programy splňující konkrétní požadavky.

PŘEHLED

Aplikace Lightbot (edukační hra) je herní bludiště, kterým je třeba provést robota. Robot je řízen sekvencemi příkazů zadanými žákem. Náročnost na vytvoření funkčního kódu postupem mezi úrovněmi zdatně roste, a žák je nucen využívat například volání procedur nebo rekurze. Hra navíc nutí žáky k efektivní tvorbě kódu, a to redukcí míst pro vkládání příkazů.

Alternativa

Alternativou pro starší nebo nadané žáky může být aplikace CargoBot, dostupná na <<https://www.microsoft.com/en-us/p/cargobot/9nblggh4r05c?activetab=pivot:overviewtab>>.

POŽADAVKY NA HARDWAROVÉ VYBAVENÍ

Žák: Osobní počítač nebo tablet s nainstalovanou aplikací Lightbot (CargoBot).

POSTUP

Žáky je vhodné v první řadě seznámit s pojmem procedura, respektive podprogram, neboť volání procedur budou v rámci hry ve velké míře využívat. K tomu může posloužit na webu připravený úkol o pohybu robot.

OVĚŘENÍ

Žáci správně odpoví na otázku: *Kolik metrů robot celkem urazí?* Správná odpověď je 60 metrů.

ODKAZY

<<http://lightbot.com/hour-of-code.html>>, <<http://lightbot.com/hoclearn.html>>

Z. BLOK

Lightbot

úkol 1

čas: 60 minut

náročnost: ★★

CÍL

Žák vytvoří a odladí program podle konkrétních požadavků.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Byl seznámen s pojmem procedura.

OČEKÁVANÉ VÝSTUPY

Žák navrhne, napíše a odladí programy splňující konkrétní požadavky.

Žák rozpozná opakující se vzory, používá opakování a připravené podprogramy.

PŘEHLED

Žáci hledají sekvence příkazů, které provedou robota bludištěm.

POŽADAVKY NA HARDWAROVÉ VYBAVENÍ

Osobní počítač nebo tablet.

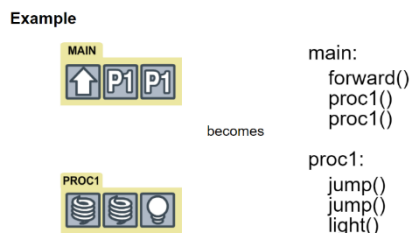
POSTUP

Každý žák spustí edukační hru Lightbot na svém počítači nebo tabletu.

K průchodu všemi 20 úrovněmi má k dispozici 1 hodinu.

Je důležité, aby pro každou ze tří finálních úrovní zhotovil PrintScreen obrazovky skýtající náhled na správně sestavené sekvence kódu.

Tyto funkční sekvence přepíše do slovní formy. Příklad je představen na obrázku níže (Obrázek 4).



Obrázek 4: Příklad přepisu do textové formy (PrintScreen z <http://lightbot.com/hoclearn.html>)

Alternativy

- Průchod hrou ve spolupracující dvojici,
- hra CargoBot jako náhrada za hru Lightbot.

OVĚŘENÍ

Žák zvládne projít všechny úrovně bez pomoci nebo s minimální pomocí učitele.

ODKAZY

Video průvodce všemi úrovněmi: <https://www.youtube.com/watch?v=R_iEX4845c>.

JINÉ UČEBNÍ TEXTY

<<https://www.3zskadan.cz/prgm/doc/lightbot.pdf>>,

<<http://lightbot.com/hoclearn.html>>

Návrh odměny

Za průchod všemi úrovněmi bez nápovědy žák získá 3 puzzle dílky.

Za průchod všemi úrovněmi s drobnou nápovědou žák získá 2 puzzle dílky.

Za průchod všemi úrovněmi s častou nápovědou žák získá 1 puzzle dílek.

Za splnění úkolu přepis příkazů ze závěrečných úrovní do textové podoby získá žák další 1 puzzle dílek.

Za testování a hledání chyb ve svých programech je všem žákům udělen **odznak „Debugger I. úrovně“**

max. 4
puzzle dílky

Z. BLOK

Lightbot

úkol 2

čas: 60 minut

náročnost: ★★☆☆

CÍL

Žák vytvoří a odladí program podle konkrétních požadavků.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Žák prošel všemi úrovněmi edukační hry Lightbot.

OČEKÁVANÉ VÝSTUPY

Žák navrhne, napíše a odladí programy splňující konkrétní požadavky.

Žák rozpozná opakující se vzory, používá opakování a připravené podprogramy.

PŘEHLED

Žáci hledají sekvence příkazů, které provedou robota bludištěm. Kromě volání procedur a využívání rekurze se ve hře objevují také podmínky, které se vykonají v případě shodnosti barev dlaždice a robota.

POŽADAVKY NA VYBAVENÍ

Osobní počítač s povoleným Flash player.

POSTUP

Žáci spustí edukační hru Lightbot 2.0. Hra je k dispozici například na <https://armorgames.com/play/6061/light-bot-20> nebo <http://www.superhry.cz/games/1734/>.

K průchodu nejméně čtyřmi úrovněmi Conditionals a čtyřmi úrovněmi Expert mají žáci vyhrazený čas (1 h).

Alternativy

- Průchod hrou ve spolupracující dvojici,
- hra CargoBot jako náhrada za hru Lightbot 2.0.

OVĚŘENÍ

Žák zvládne projít všemi šesti úrovněmi v sekci „Conditionals“ bez pomoci nebo s minimální pomocí učitele či spolužáka.

Žák zvládne projít nejméně čtyřmi úrovněmi v sekci „Expert“ bez pomoci nebo s minimální pomocí učitele či spolužáka.

ODKAZY

Video průvodce úrovněmi: <<http://armorgames.com/guide/lightbot-2-video-walkthrough>>.

Překlad návod do češtiny: <https://www.3zskadan.cz/prgm/doc/Ligtbot2-0_navod.pdf>.

JINÉ UČEBNÍ TEXTY



Obrázek 5: Návod ze hry

Návrh odměny

Za průchod všemi úrovněmi bez nápovědy nebo s drobnou nápovědou žák získá 3 puzzle dílky.

Za průchod všemi úrovněmi s častou nápovědou žák získá 2 puzzle dílky.

Za průchod několika úrovněmi žák získá 1 puzzle dílek.

max. 3
puzzle dílky

Z. BLOK

Lightbot

úkol 3

čas: 60 minut

náročnost: ★★☆☆

CÍL

Žák vytvoří a odladí program.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Žák prošel požadované úrovně edukační hry Lightbot 2.0 nebo se o to alespoň pokusil a v některých úrovních uspěl.

OČEKÁVANÉ VÝSTUPY

Žák navrhne, napíše a odladí programy splňující konkrétní požadavky.

Žák rozpozná opakující se vzory, používá opakování a připravené podprogramy.

PŘEHLED

Žák navrhne vlastní úroveň bludiště. Při tvorbě využívá a zdokonaluje tvořivé, ale také algoritmické myšlení. Žáci mohou pracovat samostatně, ale je efektivnější, aby žáci pracovali ve dvojicích.

POŽADAVKY NA HARDWAROVÉ VYBAVENÍ

Osobní počítač.

POSTUP

Každý žák spustí edukační hru Lightbot 2.0 na svém počítači nebo tabletu. Hra je k dispozici například na <<https://armorgames.com/play/6061/light-bot-20>> nebo <<http://www.superhry.cz/games/1734/>>.

K návrhu vlastní úrovně a jejímu ověření v podobě průchodu má žák vymezený čas (1 h).

Alternativa

- Průchod hrou ve spolupracující dvojici.

OVĚŘENÍ

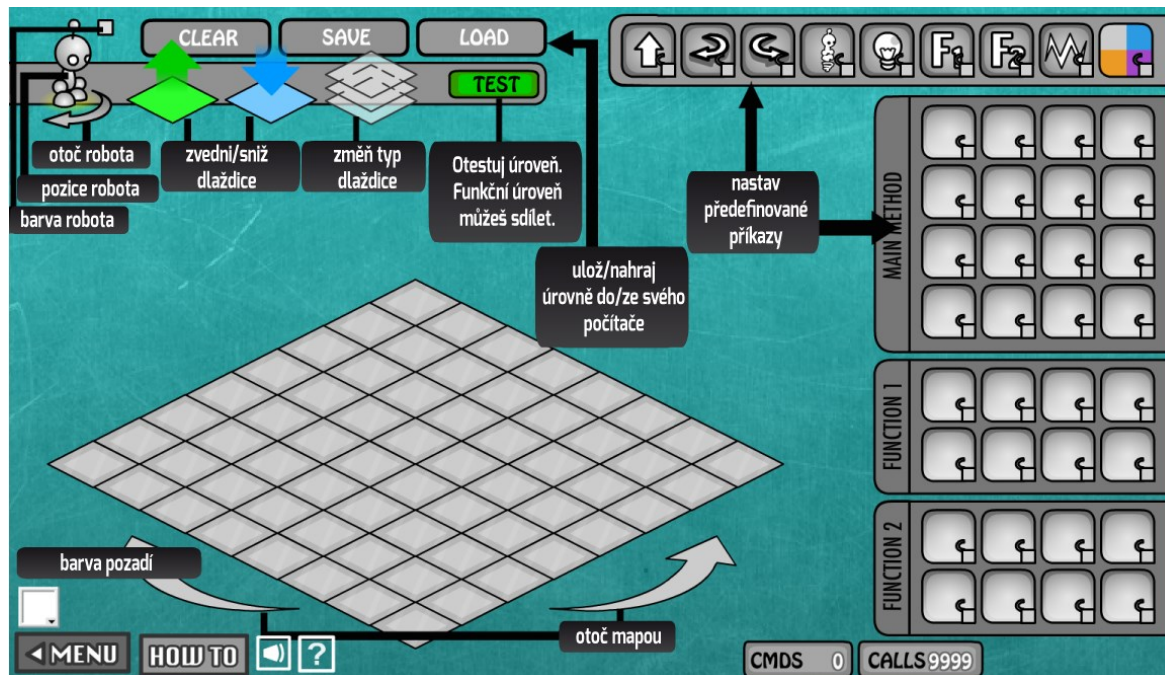
Žák dokáže vytvořit zajímavé bludiště. Navržené bludiště musí otestovat (sám bude schopen bludištěm projít, a tím ověří funkčnost úrovně).

Každý návrh musí obsahovat min. 1 podmínku (libovolné barvy), min. 1 volání procedury, přesný počet míst na kód potřebných k průchodu bludiště.

ODKAZY

Odkaz „User levels“ v hlavním menu hry Lightbot 2.0 otevře nové okno, ve kterém jsou k dispozici úrovně vytvořené dalšími hráči.

JINÉ UČEBNÍ TEXTY



Obrázek 6: Nástroje

Návrh odměny

Za vytvoření, otestování funkčnosti svého návrhu a návrhu spolužáka žák získá 3 puzzle dílky.

Za vytvoření a otestování funkčnosti svého návrhu žák získá 2 puzzle dílky.

Za vytvoření svého nebo za otestování funkčnosti spolužákovy úrovně žák získá 1 puzzle dílek.

max. 3
puzzle dílky

Za získání 7 a více puzzle dílků je udělen **odznak „Lightbot“**

celkově
max. 10
puzzle dílků

Z. BLOK

Code.org kurz

čas: 360 minut (45–60 minut na lekci)

náročnost: ★★☆☆

CÍL

Žák vytvoří a odladí program podle konkrétních požadavků.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Žák chápe nutnost přesných a jednoznačných pokynů v programu.

Žák zná pojem podprogram a chápe, k čemu se v programování využívá.

OČEKÁVANÉ VÝSTUPY

Žák navrhne, napíše a odladí programy splňující konkrétní požadavky.

Žák rozpozná opakující se vzory, používá opakování a připravené podprogramy.

Žák v blokově orientovaném jazyce sestaví program, ve kterém využívá sekvence, funkce (procedury) a vlastní funkce včetně parametrů funkcí a cyklů.

Žák upraví připravený postup pro obdobný problém; ověří správnost jím navrženého postupu, najde a opraví v něm případnou chybu.

PŘEHLED

Žák se v prostředí studia Code.org pokusí o průchod vybranými lekci kurzu. Program vytváří v block-based jazyce. Vybrané lekce jsou orientovány na želví grafiku.

POŽADAVKY NA HARDWAROVÉ VYBAVENÍ

Osobní počítač nebo tablet.

POSTUP

Teoretická část

V rámci startovače si s žáky připomeňte, k čemu se v programování využívají podprogramy (funkce/procedury).

Představte žákům pojmy proměnná a parametry funkce.

Žákům by poté mělo být jasné využití proměnné a parametrů funkce v programování, neboť je budou využívat v rámci lekcí kurzu.

Při představení proměnné by bylo vhodné, aby byli žáci hned v úvodu seznámeni s některými restrikcemi spojenými se jménem proměnné v rámci deklarace:

- název nesmí mít v prvním znaku nic jiného než písmena anglické abecedy (3prom),

- název nemá obsahovat mezery (~~treti-prom~~),
- název nesmí být stejný jako slova vyhrazená samotnému jazyku (~~main~~).

V rámci stratovače jsou představeny tři možnosti jednoduchého využití parametrů.

Praktická část

Na webu <https://studio.code.org/courses> je umožněn přístup do jednotlivých kurzů. Vyberte kurz číslo 4 (věk: 10–13 let). Existují dvě možnosti, jak s žáky kurzem projít.

První možností je výběr kurzu bez učitelského účtu, kdy žáci prochází kurzem podle pokynů učitele, případně absolvují celý kurz. Výhodou je, že se nebudete zdržovat s administrací kurzu. Přijdete ale o všechny výhody popsané níže.

Druhou možností je vytvoření učitelského účtu na Code.org a administrace kurzu. Administrací je myšlena možnost vytvoření vlastních sekcí (personalizace kurzu). Ta učiteli umožňuje vytváření vlastních tříd, přiřazení žáků do tříd, povolení programování ve dvojicích. Dále umožňuje zneviditelnit lekce, které nechcete do sekce zařadit. Velkou výhodou je možnost sledovat progres žáků v průchodu kurzem z jednoho místa.

Před zahájením kurzu mohou žáci využít krátké cvičení, které je do startovače zařazeno z důvodu snazšího přechodu z prostředí Lightbot do prostředí využívající block-based programovacího jazyku využívajícího Blockly.

Plán kurzu

Každá lekce obsahuje video, které by mělo žáky seznámit s novým pojmem, jako prostředkem využívaným ve skutečném programování. Bohužel, videa nejsou přeložena do češtiny.

V rámci několika jednotlivých cvičení ve všech lekcích jsou žákům předloženy již hotové, ale špatně fungující programy. Úkolem žáků je tyto programy opravit.

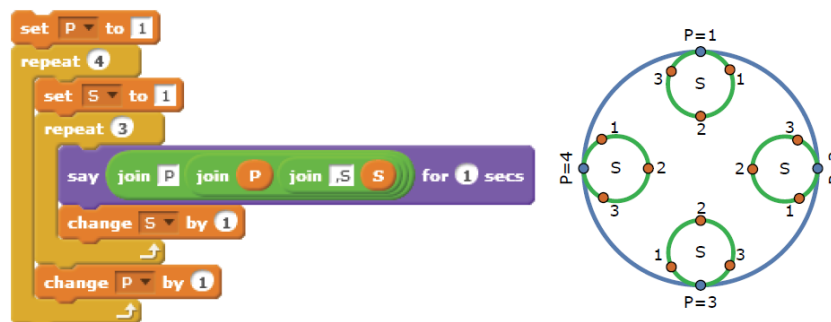
Závěrečná cvičení (tzv. „volné hry“) jsou vždy věnována volné tvorbě a zkoušení vlastních kódů.

Představte žákům možnost změny rychlosti probíhajícího programu. Díky nižší rychlosti mohou lépe identifikovat případnou chybu ve svém kódu.

Lekce 3: Umělec se skládá z 14 cvičení, k jejichž dokončení žáci musí v kódech využívat především sekvencí a cyklů.

Lekce 6: Umělec: proměnné se skládá z 16 cvičení, v nichž jsou žáci od 2. cvičení vedeni k využívání proměnných a binárních operátorů (pro výpočet velikosti úhlu nebo výpočet délky strany). V 6. lekci začínají žáci využívat vnořené cykly.

Jsou očekávané problémy s vnořenými cykly. Žákům je možné představit princip vnořených cyklů pomocí animace vytvořené ve Scratchi, kterou předkládá Marji (2014, s. 167).



Obrázek 7: Vnořený cyklus ve Scratchi – PrintScreen (Marji, 2014, s. 167)

Zajímavý způsob seznámit žáky s vnořenými cykly přináší také Bagge: <<http://code-it.co.uk/wp-content/uploads/2018/03/Nested-loops-school-year.pdf>>.

Lekce 10: Umělec: for cykly se skládá z 12 cvičení, ve kterých žáci začínají využívat for cykly. Od 4. cvičení mají žáci k dispozici všechny bloky (již nejsou k dispozici vybrané bloky, které by měly být využity k sestavení programu). Je vhodné žáky na tuto změnu upozornit.

Lekce 12: Malíř: funkce se skládá z 13 cvičení. Připomeňte žákům tvorbu programů v Lightbotovi (volání procedur). Od 3. cvičení žáci využívají zelený blok funkce. Pokud žáci chtějí funkci definovat nebo upravit, je nutné kliknout na tlačítko úpravy, které jim umožní změnu jména funkce, komentář a dále zpřístupní tělo funkce. Od 11. cvičení mají žáci přístup ke všem blokům, což může být pro některé z nich matoucí.

Lekce 14: Umělec: funkce s parametry se skládá ze 17 cvičení, ve kterých žáci opět využívají a definují funkce, a to včetně parametrů funkcí. Parametry se týkají především vlastností stran.

OVĚŘENÍ

Pochopení a vhodné využívání programovacích konstrukcí lze v reálném čase vysledovat z progresu žáků při průchodu kurzem.

Stav lekce		Stav dokončení	
Nezahájeno	Rozpracováno	Dokončeno (perfektní)	Dokončeno (too many blocks)
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Obrázek 8: Vizualizace progresu žáků v jednotlivých cvičeních (PrintScreen z webu Code.org)

MOŽNÁ ŘEŠENÍ – PRŮCHOD KURZEM

Možná řešení všech cvičení, které jsou obsahem kurzu, jsou k dispozici ve formě videí na kanálu uživatele iLearn Code [cit. 2018-11-08] na <[youtube.com](https://www.youtube.com)>.

Lekce 3: Umělec: <https://www.youtube.com/watch?v=j2PC-xtmRk>

Lekce 6: Umělec: proměnné: <https://www.youtube.com/watch?v=r0doePu5fXs>

Lekce 10: Umělec: for cykly: <https://www.youtube.com/watch?v=XG935eS99UQ>

Lekce 12: Malíř: funkce: https://www.youtube.com/watch?v=so0_CHttKRQ

Lekce 14: Umělec: funkce s parametry: <https://www.youtube.com/watch?v=gbMw8x07ORU>

JINÉ UČEBNÍ TEXTY

PDF dokument shrnující pojmy proměnná, funkce a parametry funkce je k dispozici na adrese <https://www.3zskadan.cz/prgm/doc/Code_org.pdf>.

Návrh odměny	
Za (perfektní) dokončení alespoň poloviny cvičení každé lekce žák získá 2 puzzle dílky.	max. 2 puzzle dílky
Za dokončení alespoň čtvrtiny cvičení každé lekce žák získá 1 puzzle dílek.	za průchod lekcí
Za získání 7 a více puzzle dílků je udělen odznak „Code.org“	celkově max. 10 puzzle dílků

SHRnutí

Žáci by po absolvování druhého bloku měli chápat důležitost přesných a jednoznačných pokynů v programech a také se tímto poznatkem měli řídit. Jak moc je to důležité, poznali při aktivitách bez počítače. V herním prostředí Lightbot a Lightbot 2.0 (CargoBot) se naučili rozpoznávat opakující se vzory a s využitím opakování a procedur se naučili řešit předložený problém. V průběhu tvorby správných sekvencí museli neustále program testovat a opravovat chyby. Při průchodu částí kurzu na Code.org se naučili tvořit a odladit program tak, aby splnil požadovaný účel. Programy byly tvořené již v blokově orientovaném jazyce a k jejich konstrukci žáci využívali také proměnné, vnořené cykly či funkce s parametry, tedy reálně využívané programové konstrukty a elementy.

Návrh odměny	
Za nasbíraných 26 a více puzzle dílků je udělena oboustranná sběratelská karta „Ash“	celkově max. 30 puzzle dílků
Za méně než 26 puzzle dílků je udělen odznak „Code Fighter 1“	

3. BLOK

Scratch

#0 – úvod

čas: 100 minut

náročnost: ★★☆☆

CÍL

Žák se orientuje v prostředí Scratch.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Nejsou vyžadovány.

OČEKÁVANÉ VÝSTUPY

Žák se orientuje v prostředí Scratch.

Žák v blokově orientovaném jazyce sestaví program.

Žák chápe základy Booleovy logiky – chápe význam AND, OR, NOT a jejich využití v programování.

PŘEHLED

Žáci mají za sebou již nejméně šest hodin tvorby kódu v blokově orientovaném jazyce, což by jim mělo do značné míry usnadnit přechod do edukačního prostředí Scratch.

Žáci se v úvodní lekci blíže seznámí s prostředím Scratche a vytvoří si účet. Tento krok je důležitý, neboť v prostředí budou žáci tvořit nejméně 20 hodin.

POŽADAVKY NA HARDWAROVÉ VYBAVENÍ

Osobní počítač, sluchátka.

POSTUP

V rámci startovače nechte žáky odpovědět na otázku: *Co se skrývá pod zkratkou UI?* Případně žáky nechte význam zkratky najít na Internetu. Můžete také s žáky debatovat nad otázkou: *Jaké jsou podle vás nejdůležitější vlastnosti UI?*

Žáci si vytvoří účet, při kterém se seznámí s pravidly Scratch společenství – dostupná také na <https://scratch.mit.edu/community_guidelines>.

Následně ve dvojicích prostudují prostředí Scratch online na adrese <<https://scratch.mit.edu>> s podporou staženého nebo vytištěného dvojlistu *UIScratche.pdf* dostupného na <<https://www.3zskadan.cz/prgm/doc/UIScratche.pdf>>.

Představte žákům zajímavé projekty, aby získali představu o tom, co je možné v prostředí Scratch dokázat. Děti asi nejvíce ocení hry, například:

<https://scratch.mit.edu/projects/178966496/>,

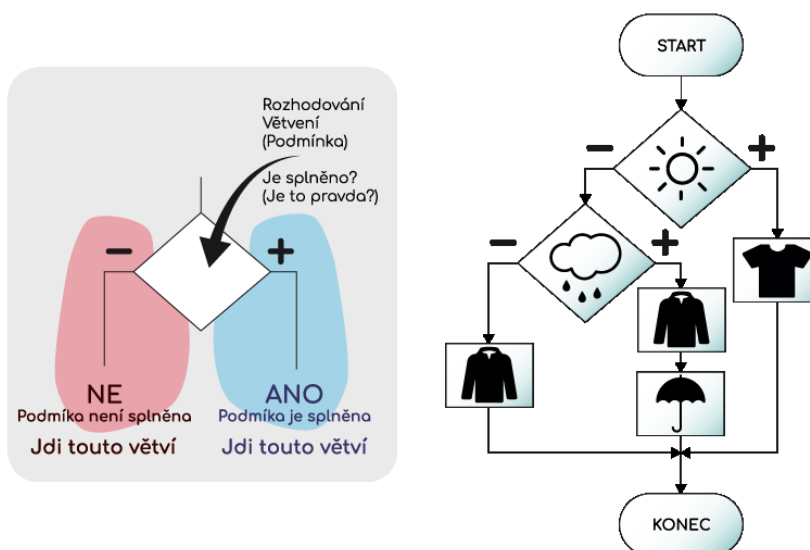
<https://scratch.mit.edu/projects/155128646/>,

<https://scratch.mit.edu/projects/74181438/>.

Společně nahlédněte, z jakých objektů jsou hry poskládány, jak vypadají jednotlivé kódy, kterými jsou objekty řízeny.

Připomeňte si s žáky pojmy proměnná a cyklus.

Seznamte žáky s novým pojmem rozhodování (podmínka). K vysvětlení pojmu je možné využít následující ilustrace (Obrázek 9).



Obrázek 9: Znárodnění rozhodování.

K bližšímu seznámení a využití proměnných, cyklů a podmínek ve Scratchi nechte žáky prostudovat připravené aktivity:

- Proměnná: <<https://scratch.mit.edu/projects/213704498/>>

Žáci by se měli ještě před stisknutím bloku pokusit určit, jak se změní hodnoty proměnných. Poté sledují, jak reagují proměnné „promenna“, „a“ a „b“ na stisknutí jednotlivých bloků. Do aktivity jsou vybrány všechny příkazy, které jsou ve Scratchi pro práci s proměnnou k dispozici.

- Cyklus: <<https://scratch.mit.edu/projects/213716175/>>

Žáci sledují, jak budou přehrány jednotlivé skupiny bloků představující hudební stupnici. Na první pohled by mělo být zřejmé, že první řešení je až nesmyslně dlouhé a mohlo by jít vyřešit jiným způsobem. Nechte žáky najít mezi čtyřmi skupinami bloků tu, která povede ke stejnému výsledku, aniž by je kliknutím spouštěli. Neměl by to již pro žáky (absolvovali kurz na Code.org) být problém.

Stejný výsledek jako první sekvence má cyklus „opakuj 4 krát“.

- Rozhodování: <<https://scratch.mit.edu/projects/213671545/>>

Opět nechte žáky nejprve přečíst si první čtyři skupiny bloků a rozhodnout, zda bude zvuk přehrán, aniž by na bloky kliknuli. Až poté zkusí, zda měli pravdu nebo ne.

Nechte žáky upravit operandy prvních čtyř bloků (pokud je třeba) tak, aby byl zvuk přehrán.

Následně nechte žáky projít stejným způsobem zbylé tři bloky s logickými operátory NOT (negace), AND (logický součin) a OR (logický součet). Nechte žáky zkoušet měnit hodnoty operandů do té doby, než pochopí princip. Zkušení žáci mohou problematiku vysvětlovat svým spolužákům.

- Spojování textu – tvoření řetězců: <<https://scratch.mit.edu/projects/213718066/>>

Vsazování bloků do sebe je jedním ze základních způsobů tvorby programu, respektive příkazu. Na tento způsob vytvoření bloků se aktivita zaměřuje. Žáci se pokusí vsadit bloky tak, aby byl výsledek totožný s předlohou. Součástí aktivity je využití proměnné.

Další aktivita je nazvaná „Najděte bloky“. Cílí na práci s jedním velmi užitečným prvkem, ve Scratchi nazývaným „blok pomoci“. Aktivita by měla také podpořit lepší orientaci mezi bloky a multibloky. Ved'te žáky k samostatnosti.

Poslední aktivita nazvaná „Výzva šesti bloků“ je zaměřena spíše na experimentování s bloky. Cílem není ani tak vytvoření funkčního programu, ale spíše na hru s bloky. Funkční program je však vítaný.

Možné řešení:



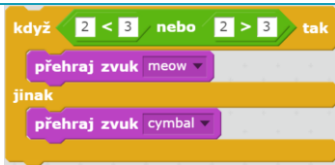
Obrázek 10: Program sestavený žákem (PrintScreen scratch.mit.edu)

OVĚŘENÍ

Ověření získaných poznatků probíhá testováním pomocí aplikace Plickers.

Návrh možných otázek a odpovědí:

Co bude řečeno?	<p>A) promenna_1 promenna_1 B) promenna_1 B promenna_1 A C) ABAA D) BA</p>
Kolikrát se zamňouká?	<p>A) ani jednou B) jednou C) třikrát D) neustále</p>
Přehraje se zvuk mňoukání?	<p>A) ANO B) NE</p>
Co se přehraje?	A) mňoukání (meow)



- B) cimbál (cymbal)
- C) mňoukání i cimbál
- D) nic

promenna_2 bude na konci rovna...



- A) 11
- B) 12
- C) 22
- D) 33

JINÉ UČEBNÍ TEXTY

Podrobný popis UI Scratche dostupné

na <https://www.3zskadan.cz/prgm/doc/UIScratche.pdf>.

Návrh odměny (za aktivitu Najděte bloky)

Za nalezení všech 5 bloků žák získá 2 puzzle dílky.

Za nalezení 3 bloků žák získá 1 puzzle dílek.

Za nejméně 80% úspěšnost v testu žáci obdrží **odznak „Hack“**

max. 2
puzzle dílky

3. BLOK

Scratch

#1 – sedm výzev

čas: 40 minut

náročnost: ★★☆☆

CÍL

Žák tvoří jednoduché programy pomocí předem připravených bloků.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Žák se orientuje v prostředí Scratch. Využívá nástroje „blok pomoci“.

OČEKÁVANÉ VÝSTUPY

Žák v blokově orientovaném jazyce sestaví program.

Žák navrhne, napíše a odladí programy splňující konkrétní požadavky.

PŘEHLED

Žáci se v první lekci blíže seznámí s vybranými bloky, které využijí při tvorbě. V každé výzvě mají předpřipravené bloky, pomocí kterých sestaví funkční program podle požadavků.

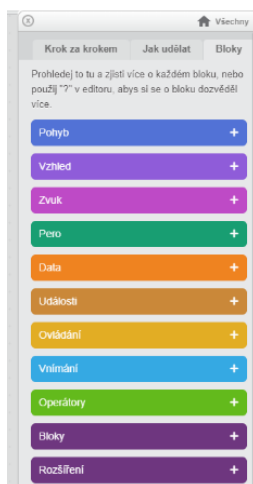
Výzvy se orientují na souřadnicový systém, směr, pohyb, random, vnímání (kolizi) a události.

POŽADAVKY NA HARDWAROVÉ VYBAVENÍ

Osobní počítač.

POSTUP

V rámci startovače nechte žáky prozkoumat multibloky dostupné pod záložkou Bloky v nápovědě, viz obrázek níže (Obrázek 11).



Obrázek 11: PrintScreen scratch.mit.edu

U všech sedmi výzev nechte žáky zkoušet tvořit správný kód ve dvojicích (samostatně). Opět je důležité, aby žáci experimentovali, četli vytvořené kódy krok po kroku a opravovali je tak, aby docílili požadovaného chování programu, měnili hodnoty a pozorovali, jak změna souvisí s chováním programu.

Alternativa pro slabší žáky

Pokud máte pocit, že některým žákům dělají tato cvičení velký problém, je vhodné zahrát roli dospělého žáka, který se pokouší o totéž, co děti. Sestavte nefunkční program, přečtěte ho, opravte, ale opět špatně. Nechte tentokrát přečíst program žáky. Společně dosáhněte postupnými opravami chyb správného výsledku. Správně sestavené programy jsou k dispozici na následujících URL:

- výzva: <<https://scratch.mit.edu/projects/261511300/>>
- výzva: <<https://scratch.mit.edu/projects/261512793/>>
- výzva: <<https://scratch.mit.edu/projects/261513597/>>

Před čtvrtou výzvou předved'te žákům, jak se bloky kopírují (pravým tlačítkem myši na bloku). Ve čtvrté výzvě budou potřebovat třikrát zkopírovat blok náhodné číslo.

- 4. výzva: <<https://scratch.mit.edu/projects/211606856/>>
- 5. výzva: <<https://scratch.mit.edu/projects/261518912/>>

Žáci v 6. výzvě budou programovat 2 objekty. Před začátkem šesté výzvy je důležité ukázat, jakým způsobem se vybírá určitý objekt. **Poprvé se téměř vždy žáci po výběru druhého objektu polekají, že se jim program smazal.** Každý objekt má vlastní prostor pro tvorbu kódu (scénáře). Vyzvěte žáky, aby vybírali objekty na přeskáčku několikrát za sebou a pozorovali při tom kód.

- 6. výzva: <<https://scratch.mit.edu/projects/261519688/>>

V 7. výzvě je vhodné, aby si žáci nakopírovali blok „čekej _“, pomocí kterého celý dialog lépe načasují. Žáci již mají úvodní část připravenou. Všechny bloky jsou v prostoru pro první objekt. Je tedy nutné některé bloky přetáhnout do prostoru scénáře druhého objektu nebo využít možnosti **batohu**.

- 7. výzva: <<https://scratch.mit.edu/projects/261530120/>>

OVĚŘENÍ

Správně sestavené funkční programy, které se chovají podle zadání.

Návrh odměny

Za nejméně 5 správně sestavených programů žák získá 2 puzzle dílky.

Za nejméně 3 správně sestavené programy žák získá 1 puzzle dílek.

max. 2
puzzle dílky

3. BLOK

Scratch

#2 – hra

čas: 80 minut

náročnost: ★★☆☆

CÍL

Žák tvoří program podle zadání.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Žák dokázal vytvořit jednoduché programy obsažené v předchozí lekci.

OČEKÁVANÉ VÝSTUPY

Žák v blokově orientovaném jazyce sestaví program.

Žák navrhne, napíše a odladí programy splňující konkrétní požadavky.

PŘEHLED

Žáci v druhé lekci absolvují postupně 4 části, které na sebe navazují. Na konci čtvrté části by měli mít vytvořenou jednoduchou hru.

POŽADAVKY NA HARDWAROVÉ VYBAVENÍ

Osobní počítač.

POSTUP

V rámci startovače si žáci mohou připomenout, co všechno již vytvořili a za pomoci jakých bloků? Kam zmizel kód, když přepnuli mezi objekty? K čemu slouží batoh nebo jak se bloky kopírují? Pokud si žáci nevzpomenou na bloky událost a náhodné číslo, nevadí. Bude jim připomenuto v nových dovednostech.

Součástí startovače jsou již hotové projekty, které se zaměřují na možnosti tvorby a využití nového bloku, využití náhody v několika konkrétních situacích. Události jsou zde využity k možnosti pohybu pomocí kláves.

- Nový blok (podprogram) na: <<https://scratch.mit.edu/projects/213784038/>>,
- události (zaměřené na pohyb pomocí stisknutí klávesy) na: <<https://scratch.mit.edu/projects/213786508/>>,
- náhoda (náhodné číslo) a některá nasazení náhody v programech na: <<https://scratch.mit.edu/projects/213789097/>>.

Opět nejprve nechte žáky určit, jak se bude kocour chovat po stisknutí bloku. Žáci by se následně měli střídát, a to tak, že 1. žák změní hodnoty a 2. žák se pokusí uhodnout, jak se změna projeví v chování programu. Poté si role vymění. Tato aktivita se týká především projektu „nový blok“ a „náhoda“.

Poté, co se žáci důkladně seznámí a pochopí stavbu nových bloků a možnosti jejich využití v programu (zkušeností z předchozích bloků Lightbot, Code.org) přejděte na tvorbu jednoduché hry. Hra se staví postupně ve čtyřech fázích. Postup tvorby hry je navržen tak, aby se v rámci jednotlivých přesunů z jedné fáze do druhé měnil částečně také princip hry, například změnou způsobu ovládání postavy. Zatímco v druhé fázi následuje kocour kurzor myši, v další fázi je již kocour ovládán klávesami. Tyto změny nutí žáky neustále upravovat kód podle nového zadání.

Je vhodné, aby žáci pracovali ve dvojicích a vzájemně mezi sebou komunikovali a spolupracovali. Pokuste se vytvořit homogenní dvojice, ve kterých budou žáci s přibližně stejnými zkušenostmi a schopnostmi.

1. část – základ: jsou předpřipravené všechny potřebné objekty i bloky. Je potřeba je jen vhodně sestavit.

Možné řešení 1. části je dostupné na: <<https://scratch.mit.edu/projects/212134436/>>.

2. část – game over: bloky a kostýmy objektů jsou předpřipravené, ale některé bloky budou muset žáci rozkopírovat.

Možné řešení 2. části je dostupné na: <<https://scratch.mit.edu/projects/212129740/>>.

3. část – odpočet: není povinná (vhodná pro dvojici nadaných žáků, kteří již mají vše ostatní hotové. Žáci zde využívají podprogram nazvaný „mizení“, změn kostýmu, efektu průhlednosti a změn velikosti.

4. část – hudba na pozadí a pohyb kurzorovými šipkami: žáci by měli využít poznatky ze startovače, případně mohou využít navržených bloků (obrázek na webu).

Možné řešení 3. a 4. části je dostupné na: <<https://scratch.mit.edu/projects/212139317/>>.

OVĚŘENÍ

Kvalitně sestavená funkční hra, která se chová podle zadání.

JINÉ UČEBNÍ TEXTY

Kód v projektech jiných uživatelů Scratche.

Návrh odměny

Za funkční hru žák získá 2 puzzle dílky.

Za částečně funkční hru žák získá 1 puzzle dílek.

max. 2
puzzle dílky

3. BLOK

Scratch

#3 – soutěže

čas: 90 minut

náročnost: ★★☆☆

CÍL

Žák upraví připravený postup pro obdobný problém.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Žák dokázal vytvořit jednoduché programy obsažené v 1. lekci.

Žák dokázal vytvořit zcela nebo částečně funkční hru v 2. lekci.

OČEKÁVANÉ VÝSTUPY

Žák řeší problémy rozkladem do menších částí.

Žák vybere vhodný algoritmus z více alternativ, které řeší stejný problém.

Žák navrhne, napíše a odladí program splňující konkrétní požadavky.

Žák upraví připravený postup pro obdobný problém; ověří správnost jím navrženého postupu, najde a opraví v něm případnou chybu.

PŘEHLED

Žáci ve třetí lekci vytváří funkční soutěže, založené na náhodě.

POŽADAVKY NA HARDWAROVÉ VYBAVENÍ

Osobní počítač.

POSTUP

Do lekce byly vybrány dvě soutěže/hry, které závisí na náhodě, a to hra „Hod mincí“ a „Na jaké číslo myslím?“

V rámci startovačů nechte žáky odpovědět na otázky: „Kolik variant může nastat při hodu mincí? Jak by mohl probíhat scénář hry? Jak naplánovat strategii k uhodnutí myšleného čísla? Existuje nějaký algoritmus usnadňující uhodnutí čísla?“

1. Hod mincí

Žáci mají v projektu „Hod mincí“ již předpřipravené části bloků, které mají uspořádat do funkčního programu a následně přidat tlačítko pro novou hru.

Šikovné děti nechte ošetřit možnost jiného vstupu než je „panna nebo orel.“

Možné řešení: <<https://scratch.mit.edu/projects/212358187/>>.

2. Na jaké číslo myslím

Žáci mají v projektu „Na jaké číslo myslím“ předpřipravenou část kódu. Kód ale musí upravit a dodělat podle zadání.

Ve vhodné, aby si již hotový kód nejprve přečetli a rozuměli, co konkrétně dělá.

Šikovné děti vypracují bonusovou část: „Přidej kód, který bude v závěrečném výpisu informovat o tom, na kolikátý pokus bylo číslo uhodnuto.“

Možné řešení: <<https://scratch.mit.edu/projects/212336558/>>.

OVĚŘENÍ

Obě soutěže/hry jsou funkční. Programy se chovají podle požadavků.

JINÉ UČEBNÍ TEXTY

Kód v projektech jiných uživatelů Scratche.

Návrh odměny

Za oba funkční programy žák získá 2 puzzle dílky.

Za jeden funkční program nebo oba částečně funkční programy žák získá 1 puzzle dílek.

max. 2
puzzle dílky

3. BLOK

Scratch

#4 – kalkulačka

čas: 60 minut

náročnost: ★★☆☆

CÍL

Žák navrhne, napíše a odladí programy splňující konkrétní požadavky.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Žák dokázal vytvořit zcela nebo částečně funkční programy soutěží z předchozí lekce, ve kterých využíval proměnné, rozhodování a operátory.

OČEKÁVANÉ VÝSTUPY

Žák řeší problémy rozkladem do menších částí.

Žák rozpozná opakující se vzory, používá opakování a připravené podprogramy; používá události ke spouštění podprogramů.

Žák vybere vhodný algoritmus z více alternativ, které řeší stejný problém.

PŘEHLED

Žáci ve čtvrté lekci navrhnu funkční prvky kalkulačky, vytvoří a odladí funkční program pro kalkulačku. V základu dokáže kalkulačka sčítat, odčítat, násobit a dělit dvě čísla, která zadá uživatel kalkulačky.

POŽADAVKY NA HARDWAROVÉ VYBAVENÍ

Osobní počítač.

POSTUP

V rámci startovače debatujte s žáky nad otázkou „*Jak funguje kalkulačka?*“, a to co možná nejdetailněji. Nechte žáky ve skupinách (nejméně dvojicích) navrhnout na papír jednoduchý princip, který odpovídá zadání. Můžete žáky nechat prostudovat kódy ostatních uživatelů.

Žáci ve dvojicích vytvoří funkční kalkulačku. Žáci mají nacházet opakující se vzory a v programu využívat blok „*Nový blok*“, který budou v hlavním programu volat.

Poznámka: Scratch neobsahuje „switch“. Ved'te žáky již při návrhu funkčnosti kalkulačky spíše k podmínce „if-else“ nebo k využití „událostí“.

Šikovní žáci by měli ošetřit možnosti nesprávných vstupů (včetně nuly v děliteli).

Možné řešení: <<https://scratch.mit.edu/projects/261816085/>>.

OVĚŘENÍ

Navržený program je funkční – dokáže sečíst, odečíst, vynásobit a vydělit 2 čísla.

JINÉ UČEBNÍ TEXTY

Kódy v projektech jiných uživatelů Scratche.

Návrh odměny

Za plně funkční kalkulačku žák získá 2 puzzle dílky.

Za částečně funkční kalkulačku žák získá 1 puzzle dílek.

Za testování a hledání chyb v programu je všem žákům udělen
odznak „Debugger II. úrovně“

max. 2
puzzle dílky

3. BLOK

Scratch

#5 – Math Masters

čas: 120 minut

náročnost: ★★ ★

CÍL

Žák navrhne, napíše a odladí programy splňující konkrétní požadavky.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Žák dokázal sám nebo s pomocí učitele vytvořit funkční kalkulačku.

OČEKÁVANÉ VÝSTUPY

Žák řeší problémy rozkladem do menších částí.

Žák navrhne, napíše a odladí programy splňující konkrétní požadavky.

Žák v blokově orientovaném jazyce sestaví program, ve kterém využívá sekvence, funkce (procedury) a vlastní funkce včetně parametrů funkcí, cyklů a náhodných čísel.

PŘEHLED

Žáci čekají v páté lekci dva povinné a jeden dobrovolný projekt. Žáci již nemají předpřipravené žádné části programu ani vybrané bloky. Veškerý kód tvoří od počátku sami. V projektech jsou k dispozici jen objekty využívané v ukázkovém příkladu.

POŽADAVKY NA HARDWAROVÉ VYBAVENÍ

Osobní počítač.

POSTUP

Odpočet

V aktivitě „**Odpočet**“ se žáci zabývají tvorbou časového limitu. V rámci startovače diskutujte způsoby, jak může fungovat časový limit, k čemu se využívá, jak fungují kuchyňské minutky. Jaký je rozdíl mezi minutkami (čas je odpočítáván k nule) a běžícím časem.

Úkolem žáků je vytvořit program, který bude fungovat oběma způsoby naráz, a to od/k zadanému číslu.

Možné řešení: <<https://scratch.mit.edu/projects/213073338/>>

V tomto řešení není ošetřen neplatný vstup, například písmena. Ošetření by mohlo být možnou alternativou pro žáky, kteří mají zadanou aktivitu hotovou.

Varianta pro slabší žáky:

Žáci vytvoří program, který bude pouze odpočítávat čas, například od 20 k 0.

Sudá – lichá

Aktivita „**Sudá – lichá**“ do značné míry zasahuje do učiva matematiky. Úkolem žáků je vytvořit program, který by našel uplatnění právě v hodinách matematiky v nižších ročnících při procvičování sudých a lichých čísel.

Varianty:

Slabší žáci pracují podle zadání 1. programu.

Možné řešení 1. programu: <<https://scratch.mit.edu/projects/213654023/>>.

Program však určí také záporná čísla, není odladěn. Dalším počinem žáků by mohlo být nastavení podmínky, která by eliminovala vstup pro jiná než přirozená čísla.

Ostatní pracují na 2. programu podle zadání. Důležitým elementem 2. programu je aplikace znalostí vlastností sudých a lichých čísel. Začněte tedy otázkou: „*Jakou společnou vlastnost mají všechna sudá čísla?*“ Žáci by měli přijít na to, že všechna sudá čísla jsou beze zbytku dělitelná číslem 2.

Možné řešení 2. programu: <<https://scratch.mit.edu/projects/213090208/>>.

Většina kódu 2. programu je ve scénáři pro pozadí.

Poslední a dobrovolnou aktivitou v 5. lekci je „*Math Master*“. Jedná se o aktivitu „*HardCore mód*“ určenou především nadaným žákům. Náročnost takových aktivit je nastavena na 4 hvězdy. Všichni úspěšní tvůrci funkčního programu jsou odměněni tzv. top kartou „*HardCore*“.

Pokud žáci bez větších problémů vytvořili v předchozí lekci funkční kalkulačku, neměla by pro ně tvorba tohoto kódu neřešitelná. Naopak, stačí jen postupovat opačným směrem.

Možné řešení této aktivity je dostupné na: <<https://scratch.mit.edu/projects/213584605/>>.

OVĚŘENÍ

Navržené programy jsou funkční a podle požadavků.

JINÉ UČEBNÍ TEXTY

Kód v projektech jiných uživatelů Scratche. Je možné, že žáci narazí na řešení rekurzí, která bude probírána v následující lekci.

Návrh odměny

Za oba plně funkční programy (1. i 2.) žák získá 3 puzzle dílky.

Za plně funkční program (1. nebo 2.) žák získá 2 puzzle dílky.

Za částečně funkční program žák získá 1 puzzle dílek.

Za plně funkční *HardCore mód – Math Master* žák obdrží **top kartu** „*HardCore*“ podle vlastního výběru.

max. 3
puzzle dílky

3. BLOK

Scratch

#6 – podprogram s parametry

čas: 180 minut

náročnost: ★★ ★

CÍL

Žák navrhne, napíše a odladí programy splňující konkrétní požadavky.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Žák dokázal sám nebo s pomocí vytvořit funkční program odpočet.

Žák absolvoval lekci č. 14 v kurzu Code.org.

OČEKÁVANÉ VÝSTUPY

Žák v blokově orientovaném jazyce sestaví program, ve kterém využívá sekvence, funkce (procedury) a vlastní funkce včetně parametrů funkcí a cyklů.

PŘEHLED

Žáci v této lekci aplikují zkušenosti z oblasti podprogramů a podprogramů s parametry v tvorbě vlastního programu.

POŽADAVKY NA HARDWAROVÉ VYBAVENÍ

Osobní počítač.

POSTUP

V úvodu si s žáky připomeňte, co jsou a k čemu již využívali podprogramy/procedury/funkce.

1. úloha – n-úhelníky je určena všem žákům. Cílem je vytvořit vlastní program, který bude kreslit pravidelný n-úhelník. Program bude umožňovat nastavení počtu stran, délku stran a tloušťku pera. Žáci využijí „Nový blok“ s parametry.

Nápověda pro žáky (pokud bude potřeba):

1. Úhel roviny je 360° .
2. Jeden úhel rovnostranného trojúhelníku má 60° , resp. vedlejší úhel má velikost 120° (3 strany).
3. Jeden vnitřní úhel čtverce má 90° (4 strany).
4. Jeden vnitřní úhel pravidelného šestiúhelníku má 120° , resp. 60° (6 stran).

Možné řešení: <<https://scratch.mit.edu/projects/194698897/>>.

2. úloha – rekurze je určena spíše pro zdatné žáky. Společně se zaměřte na rekurzi – volání funkce v těle samotné funkce. Žáci by si mohli vzpomenout na Lightbota. Mohli by již zaznamenat, že opakování v Lightbotovi vlastně nebyly cykly (takové, jak je nyní znají), ale

nekonečné rekurze v rámci jedné procedury, které ukončil buď program při splnění úrovně, nebo samotný hráč.

Úkolem žáků je detailněji popsat již hotový program.

Nadaní žáci by mohli využít nové znalosti a program odpočet (z 5. lekce) pomocí rekurze sestavit, ne jen okomentovat.

Možné řešení: <<https://scratch.mit.edu/projects/213175381/>>.

Poslední a dobrovolnou aktivitou v 6. lekci je „*vykreslení n-úhelníků*“ s využitím rekurze. Jedná se o aktivitu „*HardCore mód*“ určenou především nadaným žákům. Náročnost aktivity je nastavena na 4 hvězdy. Všichni úspěšní tvůrci funkčního programu jsou odměněni top kartou „*HardCore*“ podle vlastního výběru.

V úvodu s žáky proberte, kdy je vhodné rekurzi ukončit, např. otázkou: „*Kolik nejméně stran může mít n-úhelník, aby to byl n-úhelník?*“

Možné řešení: <<https://scratch.mit.edu/projects/213466801/>>.

OVĚŘENÍ

První program je funkční. Správně reaguje na hodnoty zadávané uživatelem.

JINÉ UČEBNÍ TEXTY

Kód v projektech jiných uživatelů Scratche.

Návrh odměny

Za plně funkční 1. program a správně komentovaný 2. program žák získá 3 puzzle dílky.

Za plně funkční 1. program nebo správně komentovaný 2. program žák získá 2 puzzle dílky.

Za částečně funkční program žák získá 1 puzzle dílek.

Za plně funkční *HardCore mód* – vykreslení *n-úhelníků* žák obdrží **top kartu „HardCore“** podle vlastního výběru.

max. 3
puzzle dílky

3. BLOK

Scratch

#7 – herní prvky

čas: 180 minut

náročnost: ★★★★★

CÍL

Žák navrhne, napíše a odladí programy.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Žák dokáže spustit scénář, využívá vhodné události, umí pracovat s proměnnou a cykly, v řídicích strukturách využívají logické operátory

OČEKÁVANÉ VÝSTUPY

Žák v blokově orientovaném jazyce sestaví program, ve kterém využívá sekvence, funkce (procedury) a vlastní funkce včetně parametrů funkcí, cyklů a náhodných čísel.

PŘEHLED

Žáci v této lekci aplikují různé zkušenosti z předchozích lekcí. V závěru lekce by žáci měli dokázat napsat funkční hru využívající pohybu pozadí (paralax efekt), ovládání pohybu postavy, kolizí s objekty, a dalších elementů her.

POŽADAVKY NA HARDWAROVÉ VYBAVENÍ

Osobní počítač.

POSTUP

V úvodu si s žáky diskutujte nad tím, jaké herní elementy znají.

V rámci modelu jsou některé z elementů uvedeny, ale nemusí to být právě ty, o které by Vaši žáci stáli.

Do modelu byly zařazeny tyto konkrétní elementy: změna hráče, paralax, hudba v pozadí, zvuky, skok, úhybný manévr, nepřátelé a překážky, kolize s objekty, „life bar“, počítadlo času, nová úroveň, konec hry.

Jednotlivé projekty mohou žáci řešit postupně, nadaní žáci podle vlastního výběru. Nechte nadaným žákům volnost ve výběru, ale upozorněte je na postupnou návaznost a zvyšující se obtížnost jednotlivých projektů.

1. prvek – změna postavy: změna kostýmu po stisknutí klávesy.

Velmi snadná aktivita, vhodná pro slabší žáky.

Možné řešení: <<https://scratch.mit.edu/projects/213501959/>>.

2. prvek – parallax: pozadí je složeno ze stromů, mraků, kamenů a kopců, z nichž je každý v určité úrovni (hloubce). Objevují se v náhodném časovém intervalu, je využíváno **klonování objektů**. Samotná postava pouze mění kostým, což vyvolává dojem chůze. Žáci mají k dispozici seznam bloků pro 1 objekt pozadí.

Snadná aktivita, vhodná také pro slabší žáky.

Možné řešení: <<https://scratch.mit.edu/projects/213493301/>>.

3. prvek – skok a hudba: hudba je přidána ke scéně (pozadí). Skok je vytvořen jako opakování s řídicí proměnnou. Nechte žáky prozkoumat projekty obsahující skoky, a které ve svých programech skoky vhodně řeší (stisknutím klávesy, interaktivního objektu, tlačítka myši). Slabší žáci mohou kód přepsat.

Snadná aktivita, vhodná také pro slabší žáky.

Možné řešení: <<https://scratch.mit.edu/projects/213507741/>>.

4. prvek – ninja pohyb a zvuk: aplikujte stejný postup.

Snadná aktivita, vhodná také pro slabší žáky.

Možné řešení: <<https://scratch.mit.edu/projects/213530389/>>.

5. prvek – nepřátelé, kolize a konec hry: aby hra měla smysl, je třeba přidat překážky. Jednou z variant je přidat do scény další objekt/objekty a jejich klony, kterým se musí postava vyhýbat.

Kolize je řešená podmínkou „dotýkáš se objektu1 || objektu2“ v podprogramu „hit“. Je testována při každé změně kostýmu hlavní postavy.

Ubývající life bar je řešen změnou kostýmu. Poté, co dojde k poslední změně (prázdný life bar), je vyslaná zpráva, následovaná změnou pozadí a ukončením hry.

Obtížná aktivita (velmi obtížná při zařazení více kolizních objektů), vhodná pro nadané.

Možné řešení: <<https://scratch.mit.edu/projects/215134345/>>.

6. prvek – další level: do hry je přidán časovač/krokoměr. Po 20 sekundách nebo metrech vyšle zprávu.

Snadná aktivita, vhodná také pro slabší žáky.

Možné řešení: <<https://scratch.mit.edu/projects/213558416/>>.

OVĚŘENÍ

1. Navržený program je funkční. Správně reaguje na stisk kláves.
2. Žák vytvoří animaci představující pohyb postavy s ubíhajícími objekty v popředí a pozadí.
3. a 4. Postava vhodně reaguje.
5. Do animace přidán alespoň jeden další kolizní objekt jako překážka. Postava na kolizi vhodně reaguje.
6. Po 20 sekundách vysílá zprávu, která vyvolává změny.

JINÉ UČEBNÍ TEXTY

Kódy v projektech jiných uživatelů Scratche.

Návrh odměny

Za 5 (navrhovaných/vlastních) funkčních herních prvků žák získá 4 puzzle dílky.

Za 4 (navrhované/vlastní) funkční herní prvky žák získá 3 puzzle dílky.

Za 3 a 2 (navrhované/vlastní) funkční herní prvky žák získá 2 puzzle dílky.

Za 1 (navrhovaný/vlastní) funkční herní prvek žák získá 1 puzzle dílek.

max. 4
puzzle dílky

3. BLOK

Scratch

#8 – vlastní projekt

čas: 120 minut

náročnost: ★★★★★

CÍL

Žák tvoří projekt podle vlastního uvážení.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Žák je schopný pracovat samostatně/ve skupině v prostředí Scratch. Při tvorbě využívá sekvence, řídicí struktury, proměnné, procedury (s parametry), spojování řetězců, logických spojek, zastavení programu, řízení programu (paralelizace), náhodné číslo.

OČEKÁVANÉ VÝSTUPY

Žák v blokově orientovaném jazyce sestaví program, ve kterém využívá sekvence, funkce (procedury) a vlastní funkce včetně parametrů funkcí, cyklů a náhodných čísel.

PŘEHLED

Žáci v této lekci aplikují všechny své zkušenosti z předchozích lekcí. Pracují ve dvojicích nebo samostatně. Alternativou může být programování ve více lidech, kdy každý tvoří jen určitou část programu.

POŽADAVKY NA HARDWAROVÉ VYBAVENÍ

Osobní počítač.

POSTUP

V rámci startovače si s dětmi připomeňte, co všechno již ve Scratchi udělali a co se naučili – shrňte.

Žáci si vyberou vlastní projekt: interaktivní příběh, simulaci jevu (např. fyzikální pokus), vlastní jednoduchou hru aj.

Napíší nebo nakreslí na papír scénář, a to včetně funkčních prvků. Podle scénáře vytvoří program. Hotové programy si vzájemně otestují (beta testing). Případné chyby odladí. Program následně zanalyzují online softwarem Dr. Scratch <<http://www.drscratch.org/>>. V případě nespokojenosti s výsledkem analýzy znovu upraví a zanalyzují.

Alternativa pro slabší žáky

Žáci si vyberou projekt jiného Scratchera, který remixují podle vlastního scénáře.

OVĚŘENÍ

V závěru vytvořené práce představí ostatním spolužáků, a to včetně využitých struktury programu.

JINÉ UČEBNÍ TEXTY

Kód v projektech jiných uživatelů Scratche.

Návrh odměny	
Za vlastní/remixovaný plně funkční program s komentáři žák získá 4 puzzle dílky.	max. 4 puzzle dílky
Za vlastní/remixovaný plně funkční program bez komentářů žák získá 3 puzzle dílky.	
Za částečně funkční vlastní/remixovaný program s/bez komentáře žák získá 2 puzzle dílek.	
Pokud se žák snažil, ale program není funkční, žák získá 1 puzzle dílek.	
Za plně funkční program, který byl v Dr. Scratch ohodnocen jako „Master!“ žák obdrží top kartu „HardCore“ podle vlastního výběru.	

SHRNUTÍ

Žáci by po absolvování třetího bloku měli být schopni tvořit programy v blokově orientovaném jazyce. Měli by chápat, k čemu se využívají základní programové konstrukty a elementy – proměnné, cykly, příkazy složené z více bloků, podprogramy s parametry, podmínky využívající operátory aj. Žáci by měli chápat důležitost testování i ladění programu a využívat jich k úspěšnému vytvoření vlastních funkčních projektů.

Žáci by měli pochopit, že Scratch mohou využívat také v rámci jiných předmětů, například v rámci matematiky, fyziky, ale i hudební či výtvarné výchovy. Zde však již záleží na učitelích jednotlivých předmětů, zda žáky v práci budou podporovat a umožní jim v prostředí Scratche řešit problémy spojené s konkrétním předmětem.

Návrh odměny	
Za nasbíraných 46 a více puzzle dílků je udělena oboustranná sběratelská karta „Cat“	celkově max. 53 puzzle dílků
Za méně než 46 puzzle dílků je udělen odznak „Code Fighter 2“	

4. BLOK

Vývojové diagramy

Pravidla a symboly

čas: 30 minut

CÍL

Žák se seznámí s pravidly a symbolikou používanou při tvorbě vývojových diagramů.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Žák zná vlastnosti algoritmu, chápe souvislosti mezi algoritmem, programovacím jazykem a programem.

PŘEHLED

Žáci se v této části blíže seznamují s pravidly a symbolikou využívanou při tvorbě vývojového diagramu.

POŽADAVKY NA VYBAVENÍ

Žádné (případně papíry, psací potřeby).

POSTUP

Žáci jsou blíže seznámeni s pojmem vývojový diagram, pravidly jeho tvorby.

Žáky nejdříve nechte odpovědět na otázky: „*Jaké znáš základní vlastnosti algoritmů? K čemu se využívají vývojové diagramy?*“

Žáci by si měli určitě vzpomenout na jednoduchost, determinovanost a konečnost. Připomeňte si ještě zbývající dvě (výslednost, univerzálnost). Případně přehrajte znovu video „*Algoritmus přátelství.*“

Nechte žáky prohlédnout obrázek diagramu přípravy míchaných vajíček od Lessnera a porovnat, zda jsou splněny všechny vlastnosti algoritmu. Žáci mohou vývojový diagram porovnat s animací „*Představa algoritmu – práce s daty*“ a nalézt podobnosti, respektive určit, co by bylo konkrétně schováno ve stroji „*ALOGO R-T-M-S 01.*“

OVĚŘENÍ

Žáci naleznou shody v obrázcích, dokáží určit směr toku dat ve vývojovém diagramu.

JINÉ UČEBNÍ TEXTY

Každý žák má k dispozici (vytištěný/v elektronické podobě) dokument

„*Diagramy_symboly.pdf*“ dostupný

na <https://www.3zskadan.cz/prgm/doc/diagramy_symboly.pdf>.

Bez odměn

4. BLOK

Vývojové diagramy

Čteme a tvoříme diagramy

čas: 20 minut

náročnost:



CÍL

Žák využívá vývojový diagram.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Žák byl seznámen s pravidly a symbolikou využívanou při tvorbě vývojových diagramů.

OČEKÁVANÉ VÝSTUPY

Žák vybere vhodný algoritmus z více alternativ, které řeší stejný problém.

Žák řeší problémy rozkladem do menších částí.

PŘEHLED

Žáci budou číst a vytvářet vývojové diagramy.

POŽADAVKY NA VYBAVENÍ

Papíry, psací potřeby, dokument „*Diagramy_symboly.pdf*“

POSTUP

Čtení

Žáci pracují ve dvojicích (samostatně). Pokusí se slovně vyjádřit, co představují vývojové diagramy – „*První vývojový diagram*“ a „*Druhý vývojový diagram*“ na webu a jaké jsou jednotlivé kroky.

Možné odpovědi: První může představovat algoritmus pro výpočet obsahu pravoúhlého trojúhelníku, druhý pak přechod silnice přes přechod pro chodce se semaforem.

Tvorba

Před začátkem znovu upozorněte na důležitost rozkladu postupu na elementární kroky.

Žáci pracují ve dvojicích. Využívají dokument „*Diagramy_symboly.pdf*“.

1. vývojový diagram – čištění zubů: žáci vytvoří vývojový diagram, který povede k vyčištění zubů.

Jedná se o úkol nižší náročnosti, který je vhodný pro slabší žáky.

2. vývojový diagram – pokoj: žáci vytvoří diagram, který povede k přechodu z jednoho pokoje do druhého. Mezi oběma pokoji jsou (zamčené) dveře. Přecházející osoba má u sebe svazek klíčů.

Vytvořené diagramy si žáci ve skupinách vymění a zkontrolují správnost diagramů.

V závěru společně se všemi žáky najdete (nebo navrhněte) nejlepší vývojový diagram pro každý z obou problémů.

OVĚŘENÍ

Žák je schopen přečíst jednotlivé kroky popsané vývojovým diagramem. Dokázal sledovat směr toku dat v algoritmu. Žák dokázal správně (s drobnými nedostatky) navrhnout vývojový diagram řešící konkrétní problém ze života.

JINÉ UČEBNÍ TEXTY

Každý žák má k dispozici vytištěný (v elektronické podobě) dokument Diagramy_symboly.pdf, který je dostupný na <https://www.3zskadan.cz/prgm/doc/diagramy_symboly.pdf>.

Návrh odměny

Za splnění obou částí úkolu žák získá 1 puzzle dílek.

max. 1
puzzle dílek

4. BLOK

Vývojové diagramy

PS Diagram #1 – úprava

čas: 45 minut

náročnost:



CÍL

Žák dokáže upravit algoritmus prezentovaný vývojovým diagramem.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Žák je schopen přečíst jednotlivé kroky popsané vývojovým diagramem. Žák dokáže najít chybu a upravit již hotový vývojový diagram.

OČEKÁVANÉ VÝSTUPY

Žák upraví připravený postup pro obdobný problém; ověří správnost jím navrženého postupu, najde a opraví v něm případnou chybu.

PŘEHLED

Žáci budou upravovat chybné vývojové diagramy.

POŽADAVKY NA HARDWARE, SOFTWARE A VYBAVENÍ

Osobní počítač, nainstalovaná aplikace PS Diagram, dokument „*Diagramy_symboly.pdf*“, „*UI_PS_Diagram.pdf*“ dostupný na <https://www.3zskadan.cz/prgm/doc/UI_PS_Diagram.pdf>.

Aplikace PS Diagram je dostupná na webu autora <<http://www.psdiagram.cz/>> nebo je k dispozici na <https://www.3zskadan.cz/prgm/doc/PS_Diagram_1.4.2.1.zip>.

Stažené diagramy s chybami: „*1PS.zip*“, „*2PS.zip*“, „*3PS.zip*“.

POSTUP

Nejprve s žáky prozkoumejte prostředí aplikace PS Diagramu. Je intuitivní, user-friendly. Ze začátku by žákům mohlo dělat problém rozlišovat mezi dvěma režimy – editační a animační režim (přechod algoritmu). Přepínačem mezi režimy je kouzelná hůlka. Pro bližší seznámení a orientaci v prostředí využijte dokument „*UI_PS_Diagram.pdf*“ nebo video autora, v němž práci s PS Diagramem sám popisuje <<https://www.youtube.com/watch?v=moGdMVwN9-Y>>.

Žáci hledají chyby či nedostatky ve vývojových diagramech.

1. příklad: úkolem žáků v prvním příkladu „*1PS.zip*“ je opravit diagram tak, aby vypisoval obsah obdélníku podle zadaných délek stran.

Dále by měli žáci diagram upravit tak, aby kromě obsahu spočítal a vypsal také obvod.

Poznámka: Desetinná čárka se v PS Diagramu zapisuje jako desetinná tečka: 1.2 (1,2).

Úkol je velmi snadný, vhodný pro slabší žáky.

Navrhované řešení: <https://www.3zskadan.cz/prgm/doc/1PS_r.zip>.

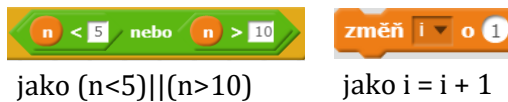
2. příklad: úkolem žáků ve druhém příkladu „2PS.zip“ je opravit diagram tak, aby seřadil dvě čísla sestupně a přidat řešení při rovnosti čísel.

Úkol je opět velmi snadný.

Navrhované řešení: <https://www.3zskadan.cz/prgm/doc/2PS_r.zip>.

3. příklad: je složen ze dvou částí zaměřených především na nadané žáky.

Pokud se rozhodnete s žáky příklad absolvovat, je nutné jim předem představit odlišný zápis reprezentovaný v textově orientovaných programovacích jazycích rodiny C, například:



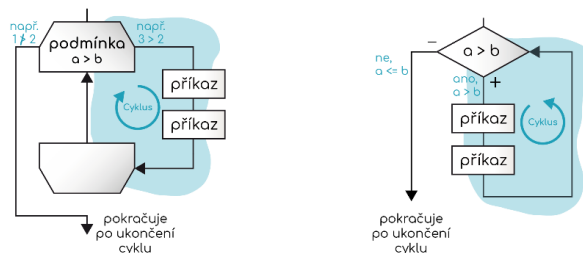
Navrhovaná řešení: <https://www.3zskadan.cz/prgm/doc/3PS_r.zip>,

<https://www.3zskadan.cz/prgm/doc/4PS_r.zip>.

Dále žáci by měli být seznámeni se způsobem tvorby cyklu, přinejmenším prostudováním obrázku představujícího cykly s podmínkou na začátku a na konci:

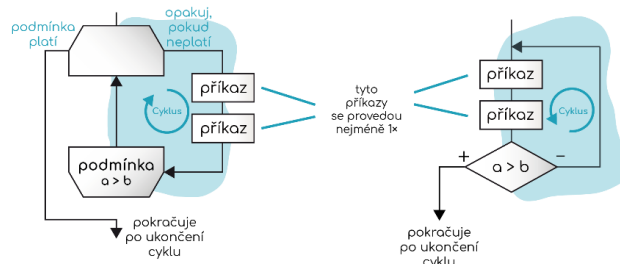
Cyklus s podmínkou na začátku

1. nejprve se vyhodnocuje podmínka
2. pokud je podmínka splněna, provádí se tělo cyklu
 - nemusí se vykonat nikdy (nesplnění podmínky při vstupu do cyklu)
 - může se opakovat do nekonečna (podmínka je stále splněna) = PROBLÉM!



Cyklus s podmínkou na konci

1. nejprve se provede první tělo cyklu
2. potom se vyhodnocuje podmínka
 - vykoná se nejméně 1 (i když není splněna podmínka)
 - může se opakovat do nekonečna (podmínka není stále splněna) = PROBLÉM!



Obrázek 12: Cyklus s podmínkou na začátku a na konci

OVĚŘENÍ

Žák dokáže ve vývojových diagramech najít chyby a opravit je.

JINÉ UČEBNÍ TEXTY

Každý žák má k dispozici vytištěný (v elektronické podobě) dokument

„*Diagramy_symboly.pdf*“ dostupný

na <https://www.3zskadan.cz/prgm/doc/diagramy_symboly.pdf> a dokument

„*UI_PS_Diagram.pdf*“ dostupný

na <https://www.3zskadan.cz/prgm/doc/UI_PS_Diagram.pdf>.

Návrh odměny

Za splnění všech tří úkolů žák získá 3 puzzle dílky.

Za splnění dvou úkolů žák získá 2 puzzle dílky.

Za splnění jednoho úkolu žák získá 1 puzzle dílek.

max. 3
puzzle dílky

4. BLOK

Vývojové diagramy

PS Diagram #2 – tvorba

čas: 60 minut

náročnost: ★★

CÍL

Žák navrhuje vývojový diagram.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Žák dokázal upravit vývojový diagram.

OČEKÁVANÉ VÝSTUPY

Žák vybere vhodný algoritmus z více alternativ, které řeší stejný problém.

Žák řeší problémy rozkladem do menších částí.

Žák popíše jednoduchý problém, navrhne a popíše jednotlivé kroky jeho řešení.

PŘEHLED

Žáci budou vytvářet vývojové diagramy.

POŽADAVKY NA VYBAVENÍ

Papíry, psací potřeby, osobní počítač s instalací PS Diagram, dokument „*Diagramy_symbols.pdf*“ a dokument „*UI_PS_Diagram.pdf*“.

POSTUP

V úvodu nechte žáky přemýšlet nad možnostmi, které mohou nastat při porovnávání dvou čísel.

Poté společně diskutujte nad počtem možností, které mohou nastat při porovnávání tří čísel.

1. příklad: žáci vytvoří vývojový diagram, který porovná dvě čísla zadaná uživatelem a vypíše výsledek.

Jedná se o úkol nižší náročnosti, který je vhodný pro slabší žáky.

Navrhované řešení: <https://www.3zskadan.cz/prgm/doc/5PS_r.zip>.

2. příklad: žáci vytvoří vývojový diagram, který porovná tři čísla zadaná uživatelem a vypíše výsledek.

Tento úkol je již náročnější, vhodný pro šikovné děti.

Navrhované řešení: <https://www.3zskadan.cz/prgm/doc/6PS_r.zip>

Správnost svých diagramů mohou žáci ověřit ve Scratchi.

V závěru společně se všemi žáky diskutujte nad algoritmem řešícím porovnávání většího počtu čísel, například 10 čísel. Cílem diskuze jsou pole a třídící algoritmy, například bubble sort, merge sort aj.

Vizualizace bubble sort je dostupná například na:

<<https://www.algoritmy.net/article/3/Bubble-sort>>.

Merge sort v podání lidového tance na:

<<https://www.youtube.com/watch?v=XaqR3G NVoo>>.

OVĚŘENÍ

Žák je schopen dokázat správně (s drobnými nedostatky) navrhnout vývojový diagram řešící třídění/porovnávání.

JINÉ UČEBNÍ TEXTY

Dokumenty „*Diagramy_symboly.pdf*“ a „*UI_PS_Diagram.pdf*“.

Návrh odměny

Za splnění obou úkolů žák získá 3 puzzle dílky.

Za splnění jednoho z úkolů žák získá 2 puzzle dílky.

Za částečné splnění jednoho z úkolů žák získá 1 puzzle dílek.

max. 3
puzzle dílky

Za hledání chyb a oprav ve vývojových diagramech je všem žákům udělen **odznak „Debugger III. úrovně“**

4. BLOK

Vývojové diagramy

PS Diagram #3 – želvička a hvězda

čas: 60 minut

náročnost:



CÍL

Žák navrhuje vývojový diagram.

VSTUPNÍ POŽADAVKY NA ŽÁKA

Žák je schopen navrhovat vývojové diagramy v aplikaci PS Diagram nebo na papíru.

OČEKÁVANÉ VÝSTUPY

Žák řeší problémy rozkladem do menších částí.

Žák popíše jednoduchý problém, navrhne a popíše jednotlivé kroky jeho řešení.

PŘEHLED

Žáci budou vytvářet vývojový diagram. Jeho funkčnost ověří v prostředí Scratch nebo Code.org. Aktivita je zaměřena na možnost využití vývojového diagramu při návrhu programu.

POŽADAVKY NA HARDWARE, SOFTWARE A VYBAVENÍ

Psací potřeby, papír, osobní počítač, nainstalovaná aplikace PS Diagram, dokument „Diagramy_symbols.pdf“, „UI_PS_Diagram.pdf“.

POSTUP

Poslední úkol opět využívá želví grafiky. Pokud uznáte za vhodné, připomeňte si s žáky na úvod princip výpočtu úhlů, podle kterých se má želvička řídit, aby vytvořila obrazec hvězdy.

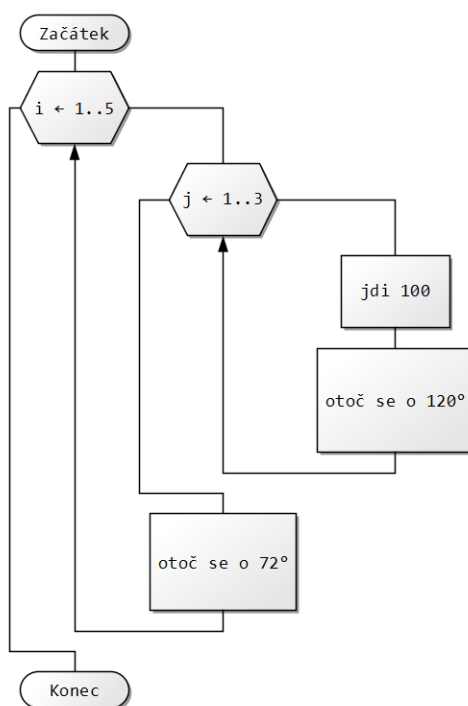
Poznámka: Je možné využít zcela jiných obrázců.

Následně navrhnu v PS Diagramu nebo na papíru vývojový diagram, který bude řešit problém kresby hvězdy.

Správnost svých diagramů žáci ověří ve Scratchi nebo na Code.org.

Převod: 10 cm = 1 krok = 10 px.

Možné řešení:



Obrázek 13: PrintScreen diagramu navrženého v aplikaci PS Diagram

Šikovní žáci mohou funkčnost dále modifikovat – uživatelem zadávaná délka strany, každý trojúhelník jinou barvou, využití celého programu jako funkce pro kreslení hvězdiček (s využitím náhodného čísla pro velikost, barvu i počet cípů) atd.

OVĚŘENÍ

Žák dokázal navrhnout vývojový diagram, prostřednictvím kterého je možné nakreslit zadaný obrazec a svůj návrh ověřil v block-based programovacím jazyku.

JINÉ UČEBNÍ TEXTY

Každý žák má k dispozici dokumenty „Diagramy_symboly.pdf“ a „UI_PS_Diagram.pdf“.

Návrh odměny	
Za splnění obou částí úkolu žák získá 3 puzzle dílky.	max. 3 puzzle dílky
Za správný návrh vývojového diagramu žák získá 2 puzzle dílky.	
Za správný návrh vývojového diagramu (s pomocí učitele) žák získá 1 puzzle dílek.	
Za získání 7 a více puzzle dílků je udělen odznak „Flowchart“	celkově max. 10 puzzle dílků

SHRNUTÍ

Žáci by po absolvování čtvrtého bloku měli být schopni vytvářet vývojové diagramy podle pravidel. Měli by chápat a při tvorbě užívat správnou symboliku. Žákům by mělo být jasné,

k čemu se vývojové diagramy využívají a proč. Vývojové diagramy by měli umět vytvořit jak na papíru, tak v prostředí vybraného softwaru.

Návrh odměny

Za nasbíraných 53 a více puzzle dílků je udělena oboustranná sběratelská **karta „Dew“**

Za méně než 53 puzzle dílků je udělen **odznak „Code Fighter 3“**

**celkově
max. 63
puzzle dílků**

FINÁLE

V rámci posledních hodin by žáci mohli být seznámeni s alternativními zdroji, které mohou využívat při sebevzdělávání doma. Za zmínku rozhodně stojí Kodu a robotické stavebnice LEGO.

Je velmi nepravděpodobné, že každý žák dosáhl počtem puzzle dílků na odměnu v podobě karty, proto na závěr udělte všem žákům finální kartu „PRGM“ bez závislosti na nasbíraných puzzle dílcích, jako odměnu za snahu a práci v rámci svých možností.

ZDROJE

Beaver Informatics Malaysia: Malaysian computational thinking and informatics contest [online]. [cit. 2018-11-08]. Dostupné z: <http://beaver.my>

Code.org - Course D (2018): Programování Na Milimetrovém Papíru #1 [online]. [cit. 2018-11-08]. Dostupné z: <https://studio.code.org/s/coursed-2018/stage/1/puzzle/1>

BELL, Tim, Ian H. WITTEN a Mike FELLOWS. *CS Unplugged: An enrichment and extension programme for primary-aged students* [online]. 2015 [cit. 2018-10-20]. Dostupné z: https://classic.csunplugged.org/wp-content/uploads/2015/03/CSUnplugged_OS_2015_v3.1.pdf

CS Principles 2018: Intro to Programming [online]. 2018 [cit. 2018-11-08]. Dostupné z: <https://curriculum.code.org/csp-18/unit3.pdf>

TOMCSÁNYIOVÁ, Monika. Potvůrky. In: *BOBŘÍK INFORMATIKY: Výběr z úloh národních kol soutěže 2010–2014* [online]. Praha: NÚV, 2015 [cit. 2018-11-11]. ISBN 978-80-7481-142-5. Dostupné z: <http://www.nuv.cz/uploads/Publikace/bobrik2017.pdf>

Code.org- Course 4. In: Youtube [online]. [vid. 2018-11-08]. Playlist uživatele iLearn Code. Dostupné z: <https://www.youtube.com/channel/UCMdrRUTHi8mpPC3ia1JTQbIA/playlists>

MARJI, Majed. *Learn to Program with Scratch: A Visual Introduction to Programming with Games, Art, Science, and Math*. San Francisco: No Starch Press, 2014. ISBN 978-1-59327-543-3.