



Microsoft

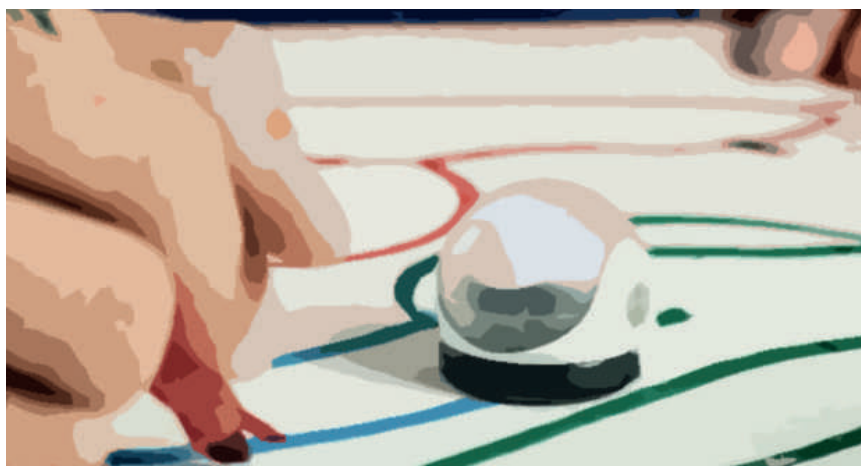
a



czechitas

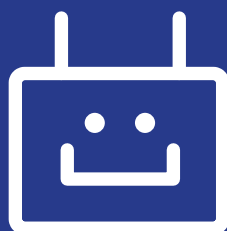
#NewGeneration

uvádí



PROGRAMUJU OZOBOTY

Tutoriál pro první kroky s Ozoboty



Programování Ozobota

Co jsou Ozoboti, jaké existují druhy i jaké jsou mezi nimi rozdíly, víš již z předchozího materiálu *Začínáme s Ozboty*. Určitě tě ale také zajímá, jak lze Ozoboty programovat. V minulém dílu jsme se společně učili, jak takového Ozobota naprogramovat pomocí papíru a fixek, v tomto materiálu se zaměříme na programování pomocí počítače a tabletu.

Přestože jsme v dílu *Začínáme s Ozboty* mnohé naznačili, měli bychom si před začátkem vysvětlit ještě jednu věc. A sice samotný pojem programování.

Co je programování?

Slovo programování se vedle slova *smart* neboli *chytrý* dnes skloňuje úplně všude. Mluví se o chytrých telefonech, chytrých městech, programování domácností, aut i počítačových her. Určitě slovo *programovat* neuteklo ani tvým uším. Pojďme se společně na tento termín podívat a také si rovnou vyzkoušet napsat první kód. Hezky pomalu a postupně.

Jak na definici?

Hravě! Pojďme se seznámit s programováním za pomoci něčeho, co důvěrně znáš. Třeba čištění zubů. Vezmi si papír a tužku a sepiš, jak takové čištění zubů probíhá. Představ si, že tuto činnost představuješ někomu, kdo si zuby nikdy nečistil, takže nesmíš vynechat žádný detail.

Až postup sepíšeš, zkus jej zkontrolovat ještě s někým jiným, jestli ti nechybí nějaký krok. Se sourozencem, rodiči, kamarády nebo vyučujícími ve škole. Společně se ujistěte, že postup obsahuje následující aktivity (ne nutně v tomto pořadí):

uchopení kartáčku namočení kartáčku odplivnutí
vykloktání pohyb rukou shora dolů přiložení kartáčku na zuby
nanesení pasty na kartáček pohyb rukou zdola nahoru

To není všechno. Než si totiž může někdo vyčistit zuby, je potřeba se ujistit, že si má čím zuby vyčistit. V celém postupu proto nesmí chybět aktivity jako

ujištění se, že mám (svůj) kartáček
kontrola, že mám k dispozici pastu (a tuba není prázdná)
instrukce, jak vysoko zvednout ruku s kartáčkem

Ještě nekončíme. Je potřeba také vymezit dobu trvání čištění zubů. Kdy je potřeba činnost ukončit? Jak budeš čas měřit?

Rozrůstá se ti seznam aktivit?

Na konci tohoto cvičení budeš mít k dispozici celý návod na to, jak si lidé čistí zuby. Takže pokud přijdeš například za babičkou, budeš jí číst tento návod a ona bude dodržovat pokyny, měla by si umět správně vyčistit zuby. A to tak, aniž by se pobryndala nebo aniž by se trefila kartáčkem do oka. Vyzkoušej, jestli to funguje!

Pokud v seznamu najdeš chyбку, hned ji naprav a znovu otestuj, jestli si tentokrát babička zuby dočistí přesně podle pokynů. Opravování a testování opakuj, dokud nebude čištění zoubků perfektní a dokud by tě za výkon podaný na základě instrukcí nepochválila paní zubařka nebo pan zubař.

Kam jsme se to dostali?

Máš za sebou tvorbu prvního kódu/programu, který můžeš nazvat *Návod na čištění zubů*. A takhle vlastně funguje programování.

Programování samotné totiž znamená **zadávání úkolů počítači, který zadané úkoly plní**. Stejně jako babička dodržuje tvoje instrukce při čištění zubů. Instrukce musí popisovat krok po kroku, co je potřeba udělat.

Pokud není kód zcela v pořádku, budou si počítače (i babička) stěžovat, že úkolu nerozumí. Je třeba proto vyhledat chybu a opravit ji. Odstraňování chyb se v angličtině jmenuje **debugging**. Neoddělitelnou součástí programování je také **testování**, jestli vše funguje, jak má. Nezapomínej na něj, ušetří ti spoustu starostí.

Možná tě napadá zásadní otázka: *K čemu je to všechno dobré? Nebo: Musím zrovna já umět programovat?*

V dnešním světě jsme obklopeni počítačovými programy. Ty ovládají techniku kolem nás – od mobilního telefonu přes ledničku až po auto. Malé počítače jsou součástí hraček, dronů i chytrých hodinek. Všechny tyto věci následují instrukce, co mají (krok po kroku!) dělat (a kdy). Instrukcemi není nic jiného, než počítačový kód, program, vytvořený programátory a programátorkami, kterým se také říká vývojáři či developři (od anglického slovesa develop).

Příklady

Všechno, co děláš na počítači, ať již jde o vyhledávání na internetu, psaní ve Wordu, kreslení v Malování nebo pouštění hudby, funguje kódu, který někdo napsal. Někdo, kdo ti ohromně ušetřil práci – třeba jen s výběrem obrázků a pozadí do Powerpointové prezentace.

Stejně fungují také počítačové hry nebo herní konzole. Pokud máš doma Xbox, můžeš říci, že máš jen jiný druh počítače. Sice na něm asi nenapíšeš slohovou práci, protože má jiné vlastnosti (a je jinak naprogramovaný, než notebook nebo stolní počítač), ale pořád funguje na stejném principu. Programy a hry jsou opět kódy, stejně tak grafika, ovládání hry a použité zvuky fungují jen na základě programu (kódu).

Programování je také součástí praček. Pomoz rodičům vyprat prádlo a prozkoumej, kolik programů má vaše pračka. Programy nastavují teplotu vody a rychlost otáčení bubny. Často také upravují také množství použité vody a délku praní podle toho, kolik prádla se v pračce nachází.

Doplň další příklady programovatelných věcí:

_____	_____
_____	_____

A jelikož programování je všude kolem nás, je docela šikovné mu nejen rozumět, jak funguje, ale také ovládat alespoň jeden programovací jazyk.

Alespoň jeden?

Přesně tak! Programovacích jazyků je celá spousta. Liší se svými vlastnostmi i použitím. Než se ale k rozdělování programovacích jazyků dostaneme, je dobré si vyzkoušet samotné programování na něčem podobně jednoduchém, jako je čištění zubů. Vyzkoušej si proto s tímto tutoriálem, jestli tě programování bude bavit. Poté teprve přijde na řadu volba programovacího jazyka a jeho studium.

Tip: ať již budeš programovat Ozoboty, hry ve Scratchi, v KODU nebo v rámci Hodiny Kódu na stránkách code.org, nezapomínej se vedleprogramování učit ještě jednu důležitou věc. Angličtinu.

Kromě toho, že spousta programovacích jazyků přebírá slovní zásobu z angličtiny, takže pokud budeš chtít příkazům dobře rozumět, tomuto jazyku se nevyhneš, tak návody na internetu jsou ve větším množství anglicky než česky. Zkus si napsat do vyhledávače na YouTube *How to program in...* (název jazyka, například *Java*) a uvidíš, kolik tutoriálů na tebe vyskočí. Kdykoli si tedy nebudeš ve své programátorské kariéře vědět rady, zkus si na internetu vyhledat pomoc.

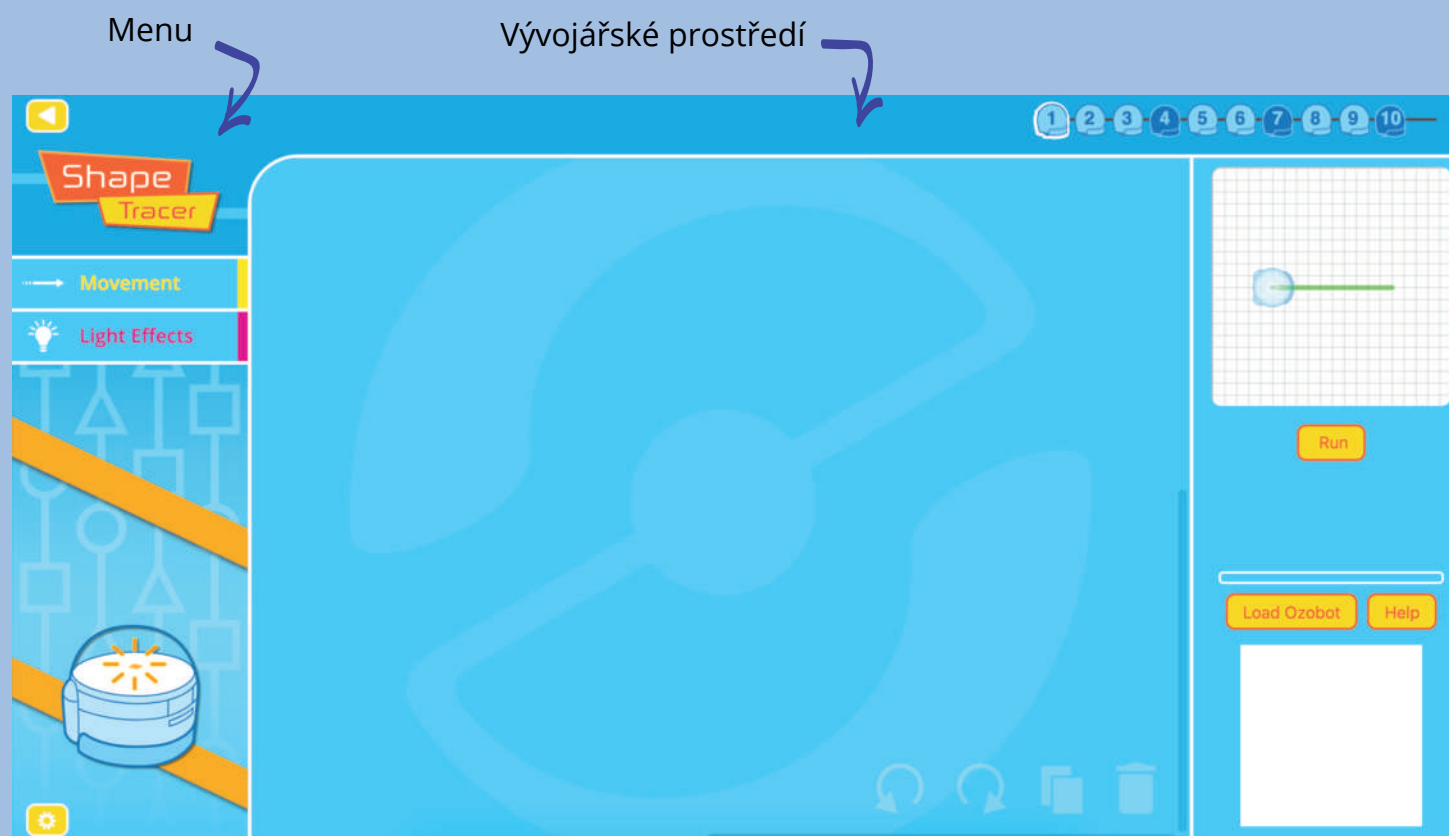
Možná již podobné vyhledávání znáš, jestli hraješ počítačové hry. Hráči PC her si také často hledají návody, jak překonat kterou překážku.

Ozo blockly tutoriály

Nejen vyhledávání, ale i některé tréninky jsou dostupné především v angličtině. Třeba právě programování s Ozoboty v Ozo blockly. Vůbec se jej neboj, angličtina je v něm jednoduchá a my ti s ní hlavně pomůžeme. Jakmile se jednou zorientuješ, úplně zapomeneš, že program není česky. Podívej se na stránky her [OzoBlockly](https://ozoblockly.com) a začni postupně odemykat nové úrovně. Začni s Shape Tracer (1) (klikni na *Play now!*).

Dostaneš se na stránku, kde se uprostřed nachází tvé vývojářské prostředí, tedy plocha, kam patří tvůj kód. Po jeho levé straně máš malá menu, v nichž najdeš kódy. Výhodou je, že kódy jsou již předepsané, takže se ti nemůže stát, že uděláš chybu. Zpočátku máš k dispozici dva typy příkazů, jedny určují, jak se bude Ozobot pohybovat (*Movement* označený šipkou), druhý Ozobotovi říká, jak má u pohybu svítit (*Light Effects* s ikonou žárovky).

Vpravo nahoře najdeš označení kol, kde se nacházíš a pod ním zadání úkolu. Zadání se úkolu také mění na pole ukazující tvůj kód v praxi, stačí stisknout tlačítko *Run*.



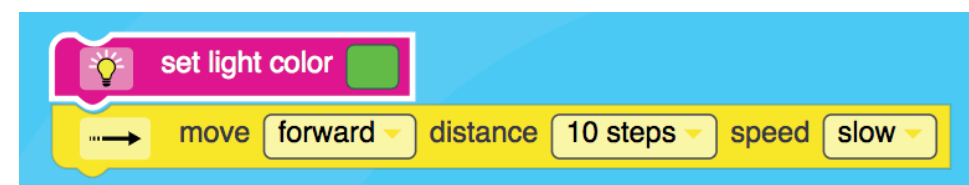
Kód, který vytvoříš, můžeš samozřejmě nahrát do svého Ozobota. Stačí přiložit Ozobota k spodnímu bílému čtverci a stisknout Load Ozobot. Nezapomeň, že při tom musíš mít jas obrazovky na maximum a s Ozobotem nesmíš hýbat po celou dobu nahrávání.

Jak bude vypadat kód v prvních kolech?

Prvním našim úkolem je posunout Ozobota o 10 kroků dopředu. U toho všeho musí svítit zeleně (když si spočítáš kostičky na obrázku vpravo nahoře, zjistíš, že jich je 10).

Tvůj kód bude následující:
nastav barvu světla na zelenou
posuň se dopředu o vzdálenost 10 kroků (jed' pomalu)

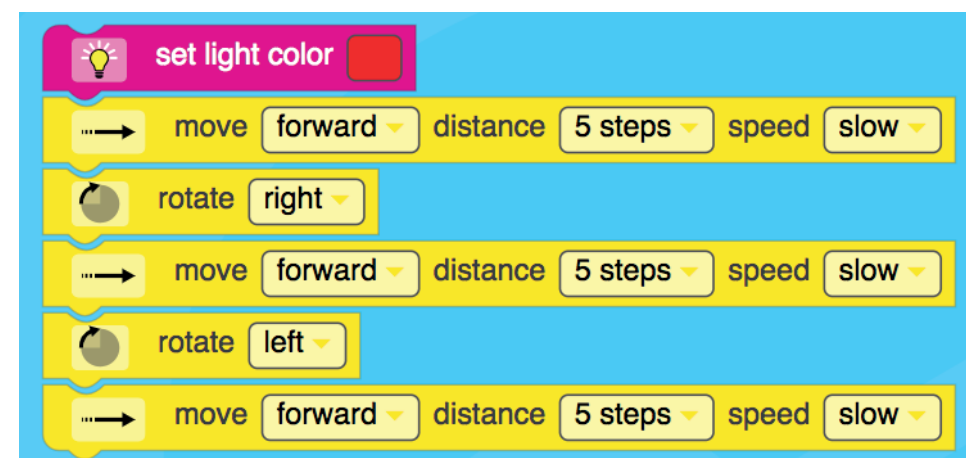
Nezapomeň zmáčknout *Run*!



Do nového kola se posuneš, pokud vyzvoříš správný kód, přehraješ jej a stiskneš tlačítko *Next Level*.

Jak si poradit s druhým kolem?

Musíš jít o pět kroků dopředu, zatočit vpravo... Vymysli si postup a pokud nevíš, jestli je správný, klidně si jej vyzkoušej na vlastní kůži. vstaň ze židle a zkus se projít tak, jak zní tvůj kód. Bude tvoje trasa shodná se zadáním pro Ozobota?



Aby se nestalo, že neporozumíš slovní zásobě u kódů, máme pro tebe pár slovíček do zásoby.

Move forward – jdi dopředu (pohni se dopředu)

Move backward – jdi dozadu

Distance 1 step – posuň se o 1 krok

Speed slow – pomalu

Speed medium – středně rychle

Speed fast – rychle

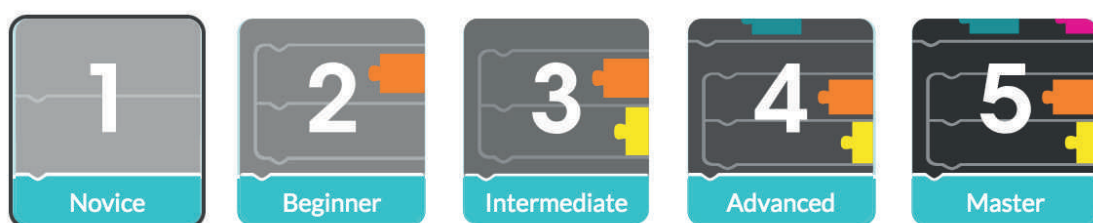
Speed very fast – velmi rychle

Rotate slight left/right – otoč se lehce doleva/doprava (o 45°)

Podařilo se ti pomocí slovníčku dokončit první online tutoriál? Gratulujeme!

Tip: Chceš během prvního tréninku angličtinu úplně vynechat? I pro tebe máme řešení!

Místo ptvního tutoriálu *Shape tracer 1* zvol [OzoBlockly editor](#) a vyber si svou úroveň. Pro první kroky (bez angličtiny) doporučujeme nováčkovský režim *NOVICE*.



Až vytvoříš svých prvních pár řádků kódu, tak se budeš moci posouvat dál přes začátečníka, středně pokročilého, pokročilého až po naprostého mistra ve skládání kódu. A to je výzva, nemyslíš?

Jak se zorientovat v OzoBlockly?

Největší část obrazovky zabírá tvá hezní plocha – místo, kde budeš vytvářet kód.

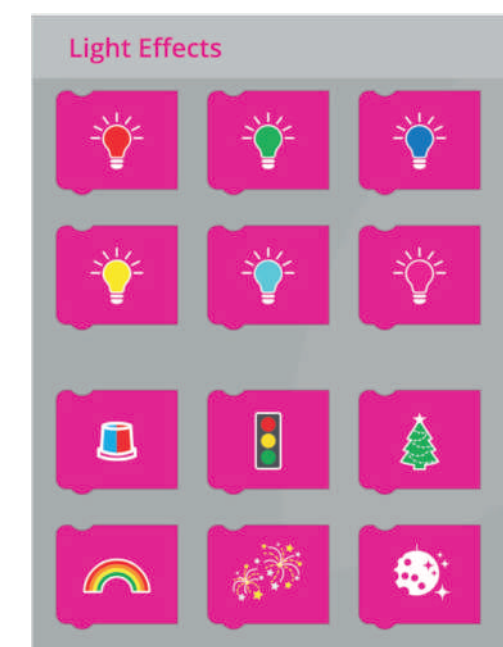
V levé části obrazovky najdeš menu – a zde si vybíráš úroveň, s níž pracuješ (čísla od 1 do 5) a také nástroje (příkazy), které máš k dispozici.



Movement – pohyb

Jednotlivé kousky kódu říkají, o kolik kroků (a případně jakým směrem) se s Ozobotem vydáš. Pojedeš tedy dopředu nebo dozadu? Chceš se s Ozobotem otočit?

Pokud chceš, aby Ozobot svítil nebo blikal, využiješ *Light Effects*, tedy světelné efekty.



Co takhle dát robotovi chvilku pauzu na oddych? Vyber *Wait* a Ozobot několik vteřin vyčká.

Určitě si také všimni ikonek v pravém dolním rohu – šipku doleva či doprava znázorňující krok zpět či krok vpřed jistě nemusíme představovat. Stejně tak koš.

Dva archy papíru slouží pro duplikaci. Pokud budeš chtít nějaký kód zkopírovat, jednoduše k tomu použij tuto ikonku.

Ikona složky s Ozobotem pak umožňuje práci se souborem – jeho uložení i sdílení.



A teď již se pusť do práce.

Úkol 1:

Nech Ozobota ujít 5 kroků rovně. Celou dobu bude robot svítit červeně.

Úkol 2:

Ozobot ujde 5 kroků rovně (a svítí červeně), poté zahne o 90°, začne svítit zeleně a ujde 10 kroků rovně.

Úkol 3:

Ozobot couvá o 4 kroky, poté se na 2 vteřiny zastaví, poté se rozbliká jako ohňostroj a začne se točit ve spirále po směru hodinových ručiček

Jak zařídit, aby tvůj Ozobot porozuměl kódu z počítače? Přesně k tomu slouží FLASHING v levém dolním rohu obrazovky.

Od druhého kola se i v tomto editoru vrátíš k angličtině. Věříme, že si však již s kódem více rozumíš a nebude ti dělat problémy. Můžeš se proto vrátit k *Shape Tracer 1* a dokončit všechny úlohy.

Shape Tracer 2 poté přináší nové výzvy a my se společně podíváme na novou látku, kterou jsou CYKLY.

Když se podíváš na zadání hned prvního úkolu, napadne tě to, co nás? Jednotlivé kroky se opakují. A to je dobře. Umožní nám společně přejít právě k cyklům, protože neustále zadávat *Jdi pět kroků rovně, zatoč doprava, jdi pět kroků rovně, zatoč doprava...* by byla nuda.

Cykly jsou skvělým nástrojem pro usnadnění práce a šetření času. My lidé jsme tak trochu líní dělat repetitivní (tedy opakující se) činnosti. Nebaví nás pořád dokola psát to samé či dělat to samé. Počítačům to však nevadí, zvládají dělat to samé prakticky donekonečna. A jelikož i roboti jsou řízeni malými počítači, jejich programování nebude o nic jinější.

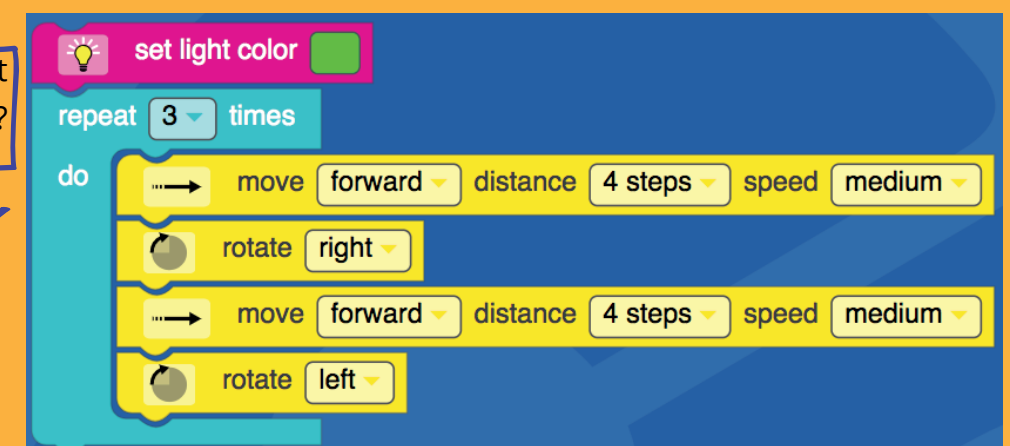
V rámci jednoho cyklu můžeš vybrat činnost nebo činnosti, které se budou opakovat. Pak už jen zvolíš, jak dlouho se mají opakovat, jestli dvakrát, osmkrát nebo třeba donekonečna.

TIP: anglická slovíčka týkající se cyklů:

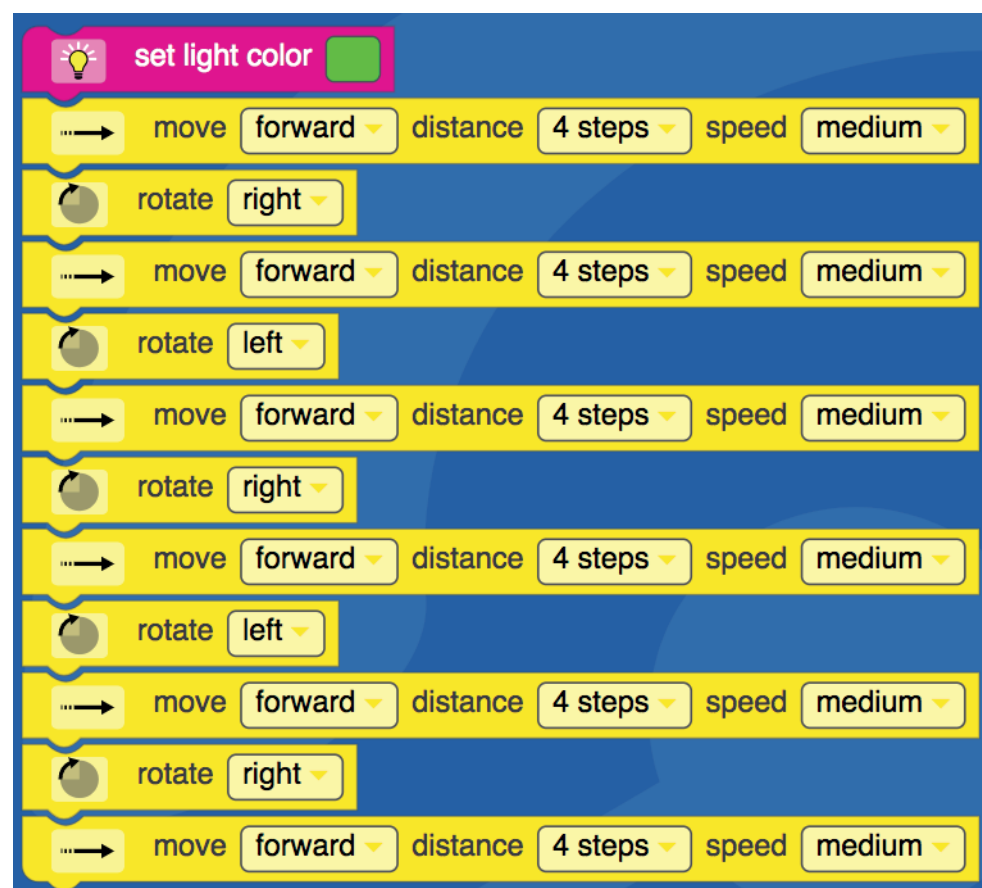
repeat 2 times – opakuj dvakrát

repeat forever – opakuj donekonečna

Jak bude vypadat kód u prvního úkolu?



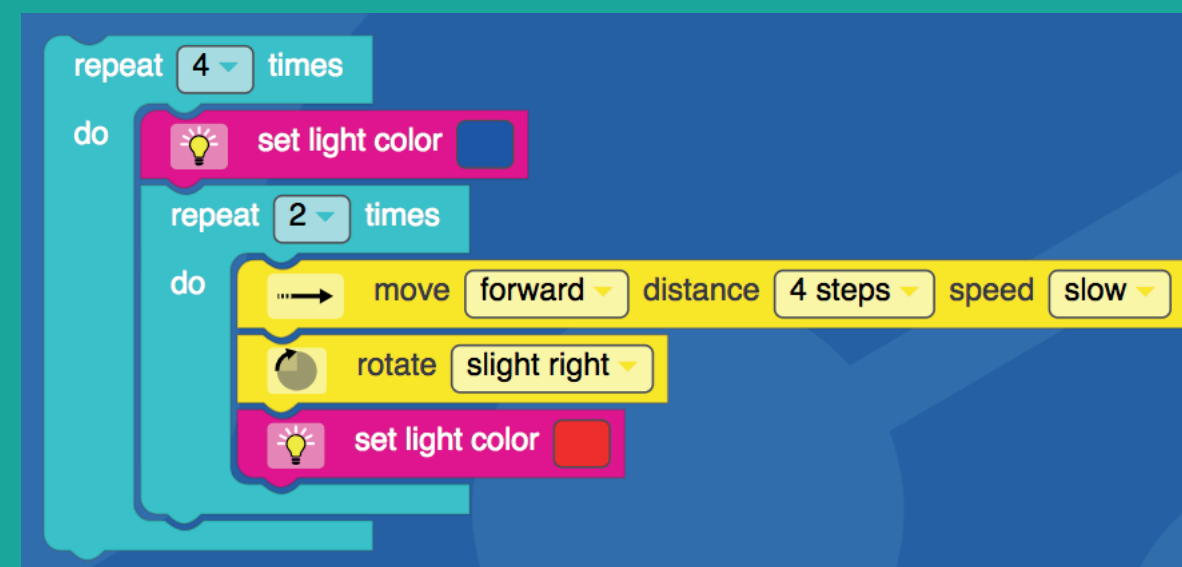
Kolik práce se ti podařilo ušetřit? Podívej se, jak by vypadal ten stejný kód bez použití cyklů.



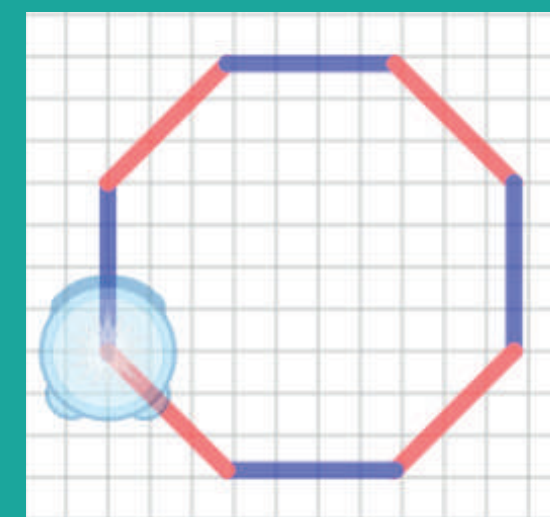
Cykly se samozřejmě nepoužívají jen kvůli lenosti lidí, kteří kód vytváří. Jejich největším přínosem je čitelnost – jak pro autory kódu, tak pro někoho, kdo by kód četl. Uznej, že je i pro tebe jednodušší si představit, jakým směrem se Ozobot vydá i jak dlouhá bude jeho trasa, použijeme-li cyklus a povíme-li ti počet opakování. Nemusíš pak trávit čas počítáním řádků kódu a hledáním rozdílů a stejných vzorců. Bude pro tebe jednodušší kontrola vlastního kódu i kódu někoho jiného.

A co když chceš opakovat stejné kroky i v rámci nějakého cyklu? Pojďme se podívat na **vnořené cykly**, tedy jednoduše cykly uvnitř jiných cyklů.

Představ si, že se ti v postupu několikrát opakuje sada stejných kroků. Například když chceš s Ozobotem obkreslit osmiúhelník, jehož stany budou střídavě modré a červené. Osmiúhelník se ti podaří vytvořit za použití 45° úhlu (*rotate slight left/right*). Ozobot bude muset svítit nějakou barvou, jít rovně, zabočit, změnit barvu, opět pokračovat rovně, zabočit... a tak dále.



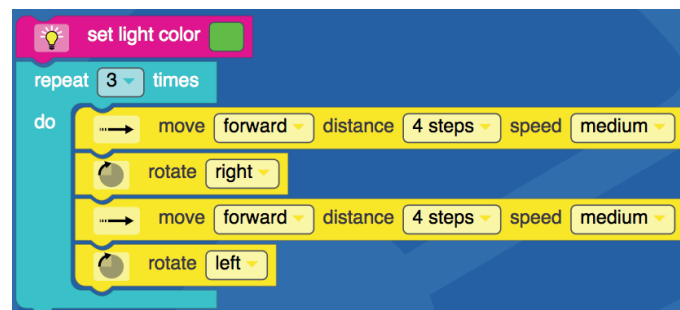
Takto bude vypadat trasa



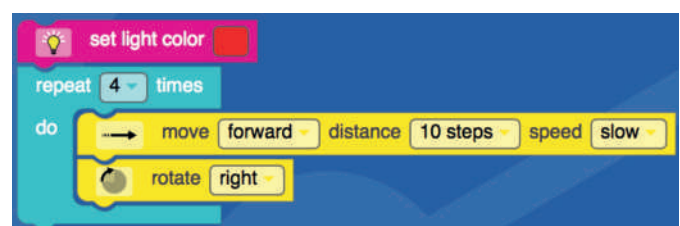
Podařilo se ti dokončit tutoriál *Shape Tracer 2*?

Pokud ano, gratulujeme! Pokud však s některým z úkolů nemůžeš hnout, mrkni na následující dvojstránku. Připravili jsme pro tebe kontrolní stranu s řešením.

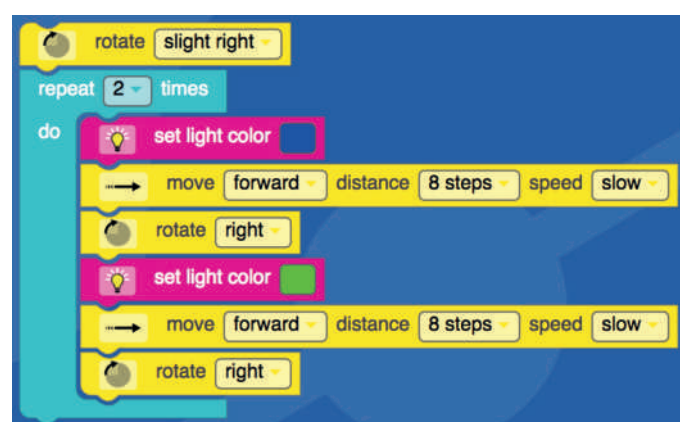
1. kolo



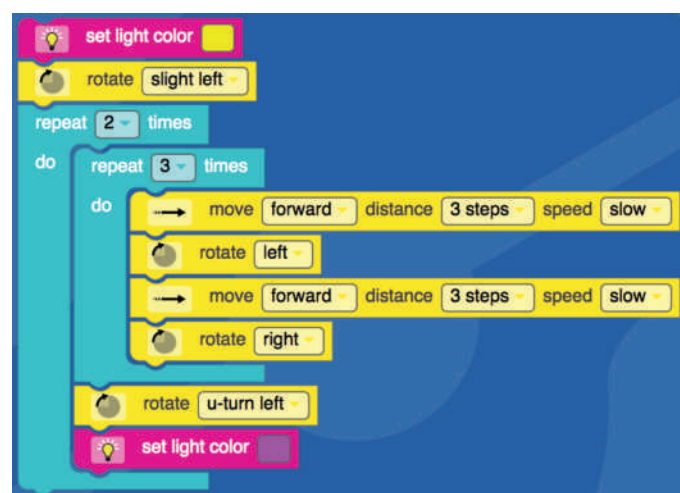
2. kolo



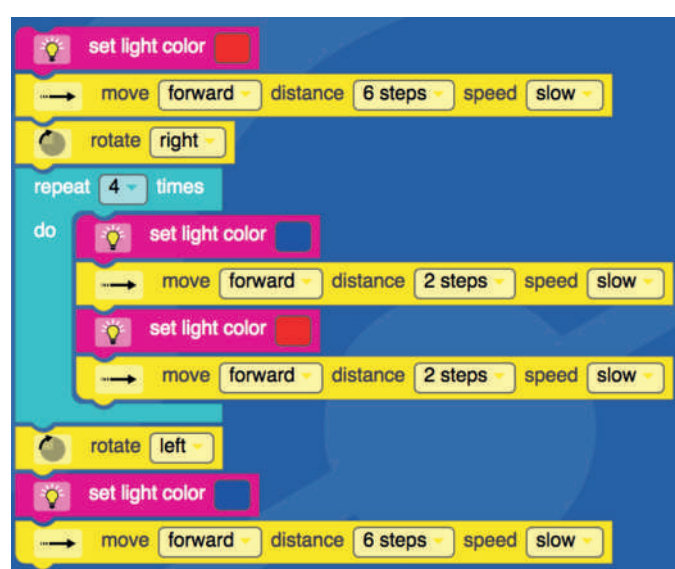
3. kolo



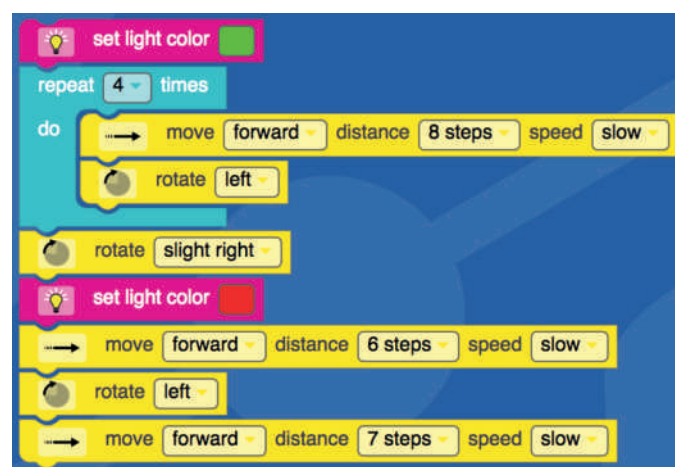
4. kolo



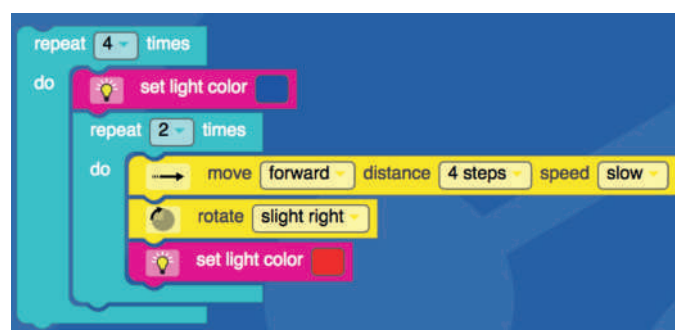
5. kolo



6. kolo



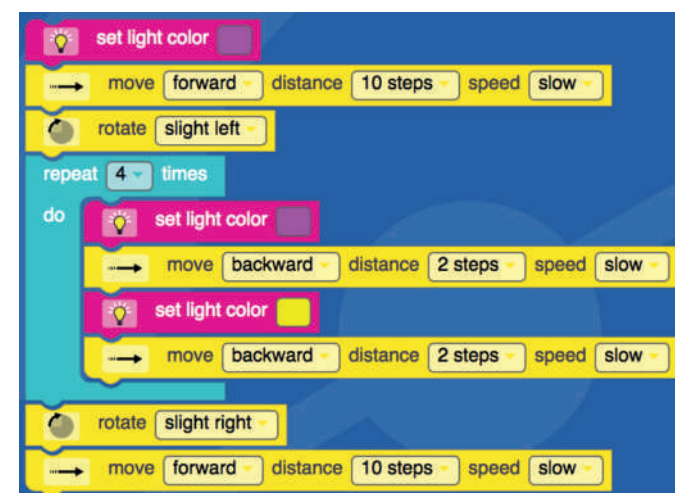
7. kolo



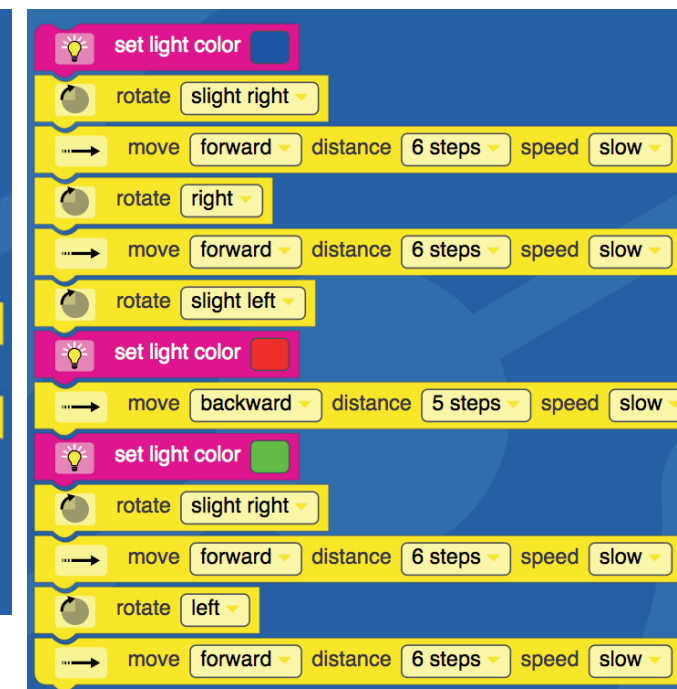
8. kolo – možné řešení



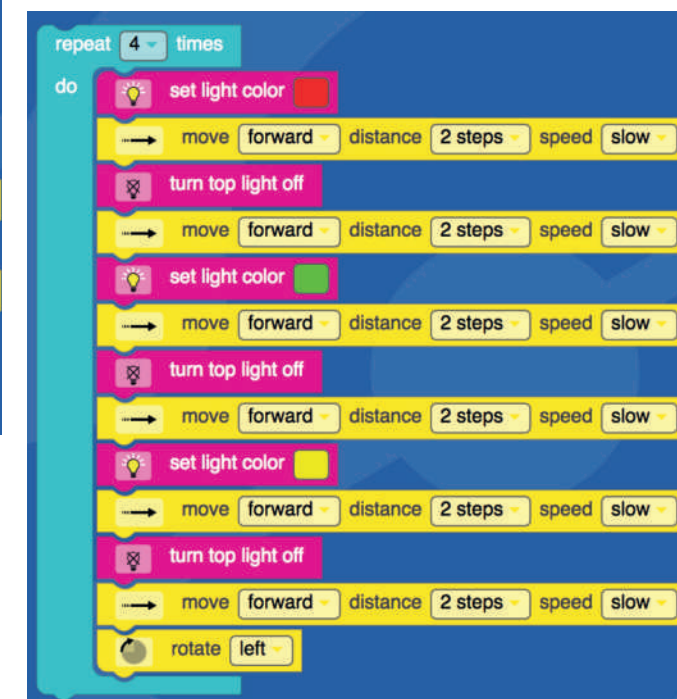
8. kolo – lepší řešení (kratší kód)



9. kolo



10. kolo



Cesta do školy.

Za 50 metrů odbočte doleva... GPS navigaci jistě nemusíme představovat. Asi všichni jsme zažili, že potřebujeme dojet autem na nové místo, tak zapneme navigaci a posloucháme její instrukce. Dnes to dělají skoro všichni rodiče i řidiči taxi. A proto si na podobnou navigaci zahrajeme.

Tento úkol se nejlépe řeší ve dvojici. Tedy ve trojici, třetím členem týmu bude samozřejmě Ozobot.

Co budeš potřebovat?

Tužku, fixy, bílé papíry, Ozobota a Ozo Blockly editor, vytištěnou mapu svého bydliště, pravítko s ryskou. Kamarádku nebo kamaráda.

Cíl hry: Naviguj svého řidiče cestou z domu do školy.

Postup: Vezmi si mapu svého bydliště a vyznač fixou cestu do školy. Viditelně označ start i cíl. Snaž se najít co nejkratší a nejrychlejší cestu. Poté si do mapy tužkou narýsuj čtverečkovitou síť. Co jeden čtvereček, to jeden krok Ozobota. Označ si také úseky, kde je potřeba dbát zvýšené opatrnosti – na křižovatkách či před školou, stejně tak si označ přechody pro chodce, kde je potřeba vyčkat pár vteřin a dát chodcům přednost, nebo železniční přejezdy.

Startujeme, nasedat! Teď již můžeš navigovat svého řidiče. Seznam jej předem s cílem své cesty a než se rozjedete, popiš, jaká cesta vás čeká. A pak již můžete vyjet. Vždy řidiči říkej, kolik kroků musí udělat kterým směrem. Nezapomeň – spolujezdec nikdy nesmí zasahovat řidiči do práce, proto jediný, kdo může ovládat editor Ozo Blockly je řidič. Tvým úkolem je naopak dbát na bezpečnou navigaci – upozorňuj předem na vyznačené úseky, kde je potřeba zvýšit opatrnost.

Jaká je povolená rychlost v obci a mimo obec?

Dojíždíš zdaleka? Opouštíš město? Auta mají mimo obec povolenou rychlost 90 kilometrů v hodině. Dovol svému řidiči jet s Ozobotem rychle. Ve městě je poté možno jezdit padesátikilometrovou rychlostí – v tomto případě nezapomeň upozornit na jízdu v obci (a zvolte střední rychlost). V některých zónách se ve městech jezdí ještě pomaleji, třeba třicetikilometrovou rychlostí. V tomto úseku jeďte s Ozobotem opravdu pomalu.

Upozornění/opakování:

Spolujezdec nikdy nesahá řidiči do řízení. Stejně tak řidič nikdy nekouká do mapy, ale vždy pozorně sleduje svou cestu.

Řidič je tvůrcem kódu. Zatímco spolujezdec říká, kterým směrem se výprava vydá (a jak dlouho ještě pojedou rovně, za jak dlouho kam zatočí).

...Dorazili jste do cíle? Nahrajte kód do Ozobota, toho pak posadte na bílý papír a sledujte jeho dráhu. Porovnejte jeho cestu s cestou vyznačenou v mapě.

Pokud nevíte, zkuste na Ozobota připevnit lehkou fixu. Ozobot tak bude během své jízdy kreslit trasu.

Po skončení jízdy si projděte kód. Opakují se během cesty některé kroky tak, že je možné použít cykly? Pokud ano, zjednodušte kód.

Nakonec ještě vytvořte kód pro cestu zpět. V čem se bude lišit?

Nezapomeňte se vystřídat. Kdo doposud navigoval teď bude řídit – třeba do kina nebo do parku. Nebo k babičce na návštěvu.

Pro rodiče/učitele:

Tato úloha rozvíjí kreativní a logické myšlení a spočívá v hledání vhodného řešení. Neexistuje jedno správné řešení.

V rámci úlohy je vhodné vysvětlit orientaci v mapách a také pravidla silničního provozu.

Příběhy číslo dva.

V předchozím tutoriálu jsme si vyprávěli pohádky a příběhy pomocí Ozobotů. Rozkreslovali jsme si scénář a určovali, co má která postava udělat a kdy. Samotný příběh jsme pak programovali pomocí Ozo kódů.

Co takhle vyprávět pohádky za pomoci programování v Ozo Blockly?

Co budeš potřebovat:

Scénář pohádky. Postup, jak na něj, se dozvíš v tutoriálu *Začínáme s Ozoboty*. Fixy, bílé papíry a samozřejmě Ozobota a editor Ozo Blockly.

Cíl: Vytvořit kód, pomocí něhož se bude Ozobot pohybovat, zatímco vyprávěč bude vyprávět pohádku.

Jak na to?

Nakresli si trasu, po níž se mají postavy pohybovat. Připrav si plánec s vyznačenými startovními pozicemi všech postav. Pozor, tentokrát bude trasa více geometrická, jelikož programování v Ozo Blockly ti umožňuje s Ozobotem jezdit rovně, zatáčet o 45° nebo zatáčet o 90°.

Až budeš mít připravenou mapu, začni s vytvářením kódů v Ozo Blockly. Zatímco minule bylo potřeba správně vychytat Ozo kódy pro úpravu rychlosti a čekání, teď je zapotřebí do kódu dopsat také veškerou navigaci. Od jízdy dopředu nebo dozadu přes všechny zatáčky a otočení o 180°.

Troufneš si na vyprávění Červené Karkulky pomocí Ozo Blockly?

Tip: Stejně jako v minulém díle, i v tomto ti doporučíme pracovat po kapitolách. Rozděl si příběh do jednotlivých kroků. Nechtěj hned vyprávět celou pohádku najednou.

Tip 2: Ušetři si práci pomocí cyklů. Jsou některé kroky, které se ve vyprávění opakují? Mysli na čitelnost kódu (a také pohodlnost a lenost tvůrců) a nahraď všechny opakující se kroky pomocí LOOPS (cyklů).

Geometrie v kapse.

Pojď si procvičit geometrii a zopakovat si úhly. U kterých obrazců se může objevit úhel 90°? A můžeš mít trojúhelník, jehož dvě strany svírají úhel 45°? Je možné tento trojúhelník nakreslit s Ozobotem?

1. Obkresli s Ozobotem čtverec.

Vytvoř kód pro čtverec s délkou strany $a = 5$ (kroků).

2. Obkresli s Ozobotem obdélník.

Vytvoř kód pro obdélník s délkou strany $a = 5$ (kroků) a s délkou strany $b = 8$ (kroků).

3. Obkresli s Ozobotem pravidelný osmiúhelník.

Délka strany $a = 4$ kroky.
Jaký úhel použiješ?

4. Obkresli s Ozobotem pravoúhlý trojúhelník.

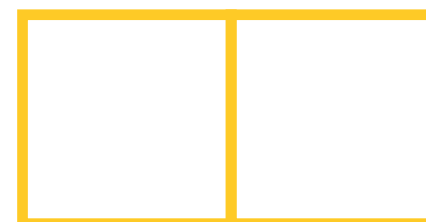
Délka strany a bude 5 kroků.
Jaká bude délka stran b , c ?
Jaké použiješ úhly?

5. Obkresli s Ozobotem přerušovanou čáru (přímku).

Rozsviť Ozobota vždy na 3 kroky, poté bude 3 kroky bez světla. Opakuj šestkrát.

6. Obkresli s Ozobotem obdélník, uvnitř něhož budou dva čtverce. Délka strany $a = 5$ kroků.

Jak dlouhá bude strana b ?



Tento materiál vznikl v rámci projektu Akademie programování, na němž spolupracují organizace Czechitas s firmou Microsoft.

Autorkou tutoriálu je Pavla Randáková působící v organizaci Czechitas na pozici Youth Education specialist. Texty a úlohy vznikly mimo jiné za pomoci komunity Ozobot.

Našli jste v textu nesrovnalosti? Veškeré dotazy, náměty a komentáře prosím směřujte na paja@czechitas.cz

