


LEKCE 7: ČÍSLA A PROMĚNNÉ

Než začnete

Už víme, že každý předmět má svoje unikátní číslo, podle kterého můžeme přesně určit o jaký předmět se jedná. Čísla už používáme na spoustu věcí, například souřadnice, počet opakování bloků příkazů a bude toho ještě víc. Jistě uznáte, že s čísly se tedy pracuje opravdu hodně. Koneckonců, kdybychom to vzali opravdu hodně do hloubky, tak počítač ani nic jiného nezná. Protože jsou čísla tak důležitá, dnešní lekce bude jenom o číslech.

Předmět podle čísla

Víme, že každý předmět má své číslo. Ale co když máme jen to číslo? Později to bude třeba nějaké číslo, které my ani neznáme. Budeme potřebovat určit předmět podle čísla. K tomu se nám bude hodit obecný **symbol předmětu**. Najdete ho skoro na konci šestého řádku nabídky příkazů. Když dáte do programu tuto ikonku a za ni číslo, program se zachová stejně, jako kdybyste tam dali rovnou předmět s příslušným číslem. Říkáte si k čemu to tedy je? Někdy třeba nebudeme znát to číslo a bude ho znát jenom program, takže předmět nemůžeme předem najít v bance. Třeba když budeme generovat náhodný předmět. Už teď můžeme tento příkaz použít, když chceme **smazat všechny předměty** na políčku před Baltíkem.

 **0** Čaruj předmět 0 = smaž všechny předměty na políčku před Baltíkem

Náhodné číslo

K vygenerování náhodného čísla nám slouží příkaz, který vypadá jako **hrací kostka**. Najdete ho ve třetím řádku nabídky příkazů. Sám o sobě příkaz náhodně generuje desetinné číslo od 0 do 1 se spoustou desetinných míst. My obvykle budeme chtít jiný rozsah možných hodnot. Pokud za kostku dáme číslo, vybírá se náhodně celé číslo **od 0 do čísla o jedno menšího**, než je to zadané číslo. Takže když zadám náhodné číslo 15, vybírá se z čísel od 0 do 14. Následující příkaz vloží zvonek (předmět č. 14) na obrazovku na náhodné souřadnice. Vpomeňte si, že souřadnice políček číslujeme od 0 do 14 (souřadnice X) a od 0 do 9 (souřadnice Y). Vyzkoušejte si to a spusťte program vícrát, ať vidíte, že se zvonek opravdu objevuje na různých náhodně vybraných místech.



Ale co, kdybychom chtěli vybírat z jiného intervalu než od nuly? I to samozřejmě jde. Za kostku dejte dvě čísla označující **minimum** a **maximum** a mezi ně dvě tečky smybolizující interval. Následující příklad dává opět předmět na náhodné souřadnice, ale tentokrát budeme dávat i **náhodný předmět**. Využijeme tady taky příkaz získání předmětu podle čísla. Takže následující příklad vloží na náhodné místo náhodný předmět s číslem od 8 do 26. Podívejte se do banky, které všechny předměty to můžou být. A rovnou vložíme třeba 20 náhodných předmětů naráz. Vyzkoušejte.



Zkuste spočítat vložené předměty. Někdy jich nebude úplně dvacet, protože se některé vloží tam, kde už je jiný předmět a tím ho zakryjí. Dá se zajistit, aby se předměty nepřekrývaly, ale k tomu se dostaneme až v příští lekci.

Úkol na procvičení

Zobrazte na náhodném místě obrazovky jeden náhodný předmět s číslem od 8 do 26. Pokaždé, když se stiskne nějaká klávesa nebo se klikne na tlačítko myši, předmět změní své místo (opět náhodně), ale je to pořád ten samý předmět.

Nápověda: Použijte nekonečný cyklus (nekonečné opakování závorky). Použijte animovaný předmět.

Úkol na procvičení – těžší varianta

Zadání je stejné jako v předchozím úkolu, ale navíc po každém pátém kliknutí se změní i daný předmět (opět bude mít číslo mezi 8 a 26).

Nápověda: Bude potřeba správně použít vnořené bloky příkazů.

Proměnné

Dostali jsme se k další velmi důležité části. Dalo by se říct, že proměnné jsou možná nejdůležitější věcí v každém programovacím jazyku. Proměnná je (odborně řečeno) označené místo paměti počítače. Jednodušeji řečeno, je to způsob, jak si program může některé věci **pamatovat**. Je to jakýsi šuplíček, kam si program ukládá všechny informace. Proto taky mají proměnné v Baltíku podobu šuplíčku.

Nabídka konstant a proměnných

Nahoře je ikonka, která nám otevře **nabídku proměnných a konstant**.



Můžeme přepínat záložky v dolní části okna. **Konstanty** (vypadají jako list pergamenu) mají předdefinovanou hodnotu, která se nedá měnit v průběhu programu, občas se hodí, ale nebudeme je používat často. Naproti tomu **proměnné** (šuplíčky a košíčky) jsou místa, kam můžeme **ukládat hodnoty** v průběhu programu, a tyto hodnoty později také **měnit** (odtud název proměnná). Na rozdíl od konstant budeme proměnné používat opravdu hodně.

Datové typy

Datový typ proměnné určuje, jakou hodnotu do ní můžeme vložit. Datový typ poznáme v Baltíku podle **barvy** proměnné. **Zelené** proměnné uchovávají reálná (tedy desetinná) čísla. **Žluté** proměnné si mohou pamatovat text. Do **modrých** proměnných ukládáme celá čísla. Vzpomeňte si na literály, ty fungují stejně. A stejné datové typy i barvy mají také konstanty.

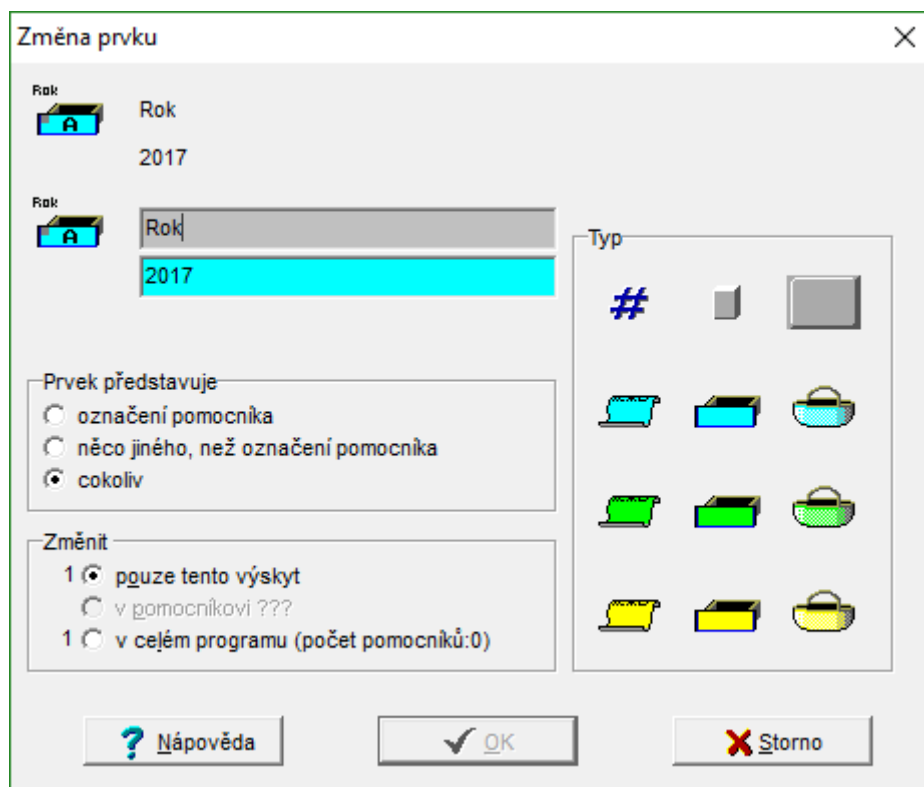
K textu se vrátíme někdy příště. Reálná a celá čísla fungují celkem podobně. Dají se také mezi sebou navzájem **převádět**. Když převádíte celé číslo na reálné, není v tom žádný problém, protože z matematického hlediska každé celé číslo je zároveň i reálné akorát s nulovou desetinnou částí (např. $1 = 1,0$). Jde převádět i reálná čísla na celá, Baltík (i jiné programovací jazyky) to dělá tak, že prostě **usekne desetinnou část**, takže když 0,1 převedeme na celé číslo, tak budeme mít 0. Pozor, **nejedná se o zaokrouhlování**. I 0,9 je po převedení na celé číslo pořád 0.

Globální a lokální proměnné

Globální proměnné jsou v našem případě šuplíčky. Stejná proměnná platí všude v celém programu. Lokální proměnné (košíčky) fungují pouze v pomocnících (neboli podprogramech), ale k těm se dostaneme až později. Zatím tedy budeme používat pouze globální proměnné.

Pojmenování proměnné

Každá proměnná musí mít **jméno**. Některé proměnné už máme pojmenované dopředu (proměnné s čísly) a některé si můžeme pojmenovat sami (proměnné s písmeny). Když na některou nepojmenovanou kliknete a vložíte ji do programu, objeví se **dialog**, který vám umožní ji pojmenovat. Každá proměnná má v celém programu jedno jméno a nemůže ho za běhu programu měnit. Pokud chcete jiné jméno, použijte jinou proměnnou. Dialog vám také umožní definovat **počáteční hodnotu** proměnné. Tato hodnota bude v proměnné od začátku programu do té doby, než ji změníte.



Vložení hodnoty do proměnné

Teoretický úvod máme za sebou. Pustíme se konečně do programování. Hodnotu do proměnné vložíme pomocí nám známého příkazu **vložení** v podobě šipky. Hodnota bude v proměnné uložena od chvíle, kdy se vykoná příkaz vložení až do skončení programu, nebo dokud ji nezměníme. Následující příklad vloží hodnotu 5 do proměnné A.



Zobrazení hodnoty proměnné

Taky by se nám hodilo při běhu programu zjistit, co v proměnné vlastně je. To můžeme udělat dvěma způsoby. Když spustíme program, hodnoty všech proměnných můžeme zobrazit tlačítkem



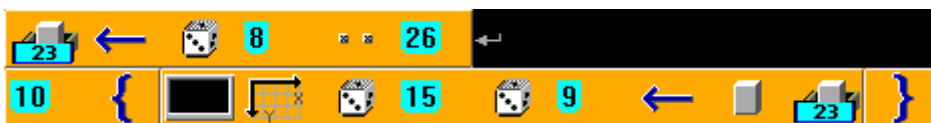
Ukázat proměnné, které se nachází v levém dolním rohu okna spuštěného programu.

Nebo můžeme také **vypsat** hodnotu proměnné na obrazovku. Hodnota se vypíše do horního levého rohu (souřadnice 0, 0). Nebo pokud chcete, můžete učít souřadnice, kam se hodnota vypíše.



Použití proměnné

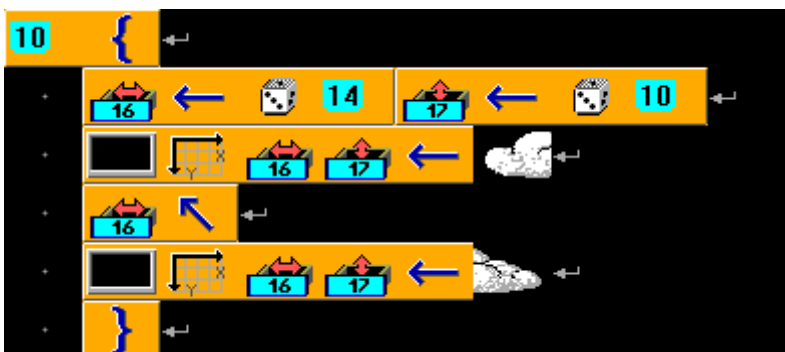
Proměnné můžete použít v programu kdekoli, kde byste použili literál. Ale k čemu si to teda vlastně komplikovat, když můžeme jednoduše použít literál? Zkrátka, nejde to úplně vždy. Vzpomeňte si, jak jsme vkládali náhodné předměty na náhodné místo, ale co kdybych tentokrát chtěl náhodně vybrat jeden předmět a ten pak desetkrát vložit na obrazovku na deset náhodných míst. Budeme muset vybrat náhodné číslo předmětu, to si zapamatovat a pak už vkládat pokaždé stejný předmět.



***Poznámka:** Použil jsem proměnnou 23, která se jmenuje předmět, a je tedy určena k ukládání čísla předmětu, ale můžete místo ní použít kteroukoli jinou celočíselnou proměnnou.*

Změna hodnoty

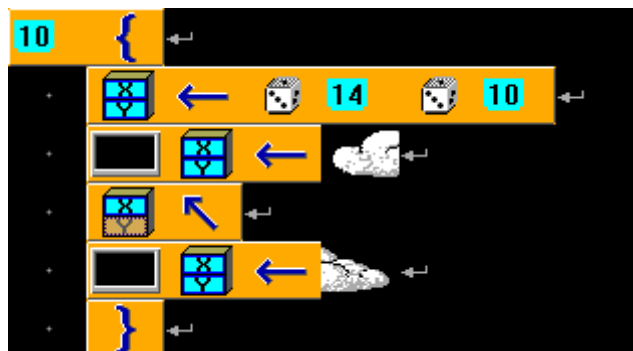
Mějme následující příklad: Na náhodná místa budeme dávat mráčky složené z předmětů 66 a 67. V takovém případě si musíme pokaždé zapamatovat souřadnice X i Y, aby byly oba předměty vedle sebe. Ještě také musíme mezi vložení jednotlivých částí upravit hodnotu X, aby se jednotlivé části nepřekrývaly. K tomu můžeme použít příkaz **zvětšení** (šikmá šipka směrem nahoru), příkaz najdete v nabídce příkazů hned vedle šipky pro vložení. Pokud dáte příkaz za proměnnou, její hodnota se **zvětší o 1**. Pokud chcete k proměnné přičíst jiné číslo než 1, můžete to číslo přidat za příkaz zvětšení. Stejně funguje i příkaz **zmenšení**. Zkuste si to.



Všimněte si, že je **menší rozsah** pro souřadnici X (pouze od 0 do 13), protože jinak by mrak mohl sahat za okraj obrazovky.

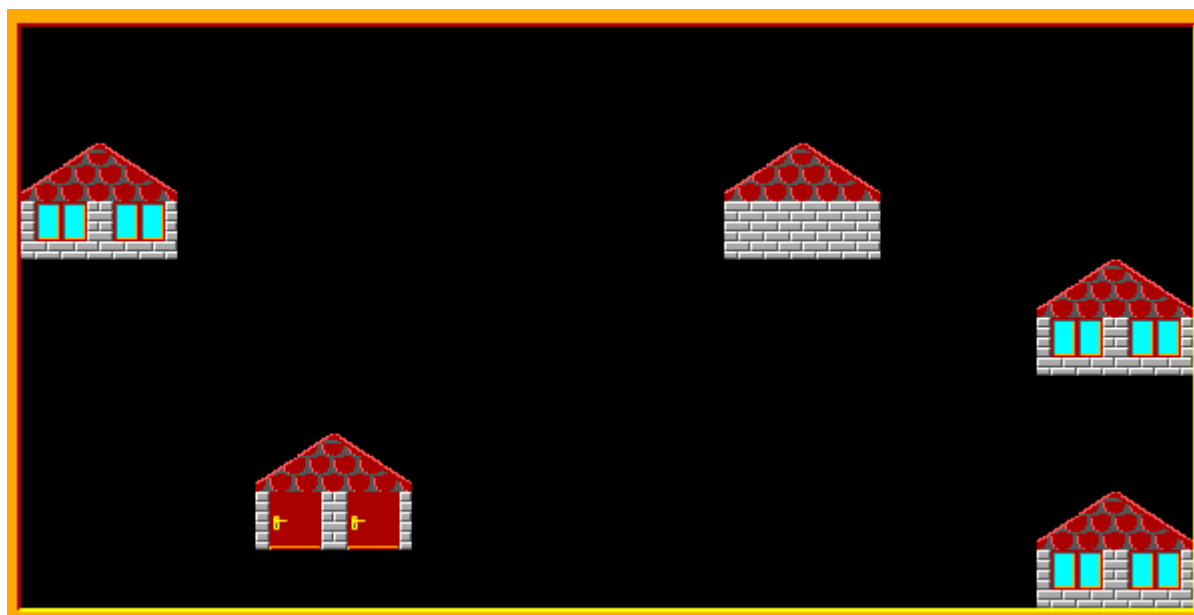
Souřadnice

Pokud chci v proměnných uchovávat **souřadnice**, můžu na to použít **speciální proměnné** pro to připravené. Najdete je ve spodním řádku nabídky proměnných. Mají tu výhodu, že k nim můžete přistupovat jak **jednotlivě**, tak i k **oběma naráz**. Navíc pokud s nimi zadáváte souřadnice, nemusíte před ně dávat značku souřadnic. Předchozí příklad by tedy s použitím těchto souřadnicových proměnných vypadal takto.



Úkol na procvičení

Udělejte program, ve kterém se objeví pět jednoduchých domků na náhodných místech obrazovky. Budou mít jednoduchou střechu (předměty č. 52 a 53) a políčka dole můžou být okno, zeď nebo dveře (předměty č. 2 až 4). Spodek domku bude taky náhodný, ale obě políčka musí být stejná.



Nápověda: Nezapomeňte správně určit rozsahy jednotlivých náhodných čísel souřadnic, aby domek nesahal za okraj obrazovky.

Zapamatujte si

Už byste měli umět použít předmět, když **znáte jen jeho číslo**, taky **generovat náhodná celá čísla z určitého rozsahu**. Dále bude i příštích lekcích hodně potřebovat práci s **proměnnými**. Měli byste zvládnout **vložit do proměnné hodnotu** nebo ji **upravit** a potom ji taky k něčemu **použít**.

Co nás čeká a nemine

Příště naučíme naše programy přemýšlet. Povíme si něco o tom, jak fungují podmínky a co všechno se dá s jejich pomocí udělat.