



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Karel, parametry procedur a příkazů

Číslo projektu	CZ.1.07/1.5.00/34.0950
Kódování materiálu	vy_32_INOVACE_inf3_prg04
Označení materiálu	prg04_karel_parametry
Název školy	Gymnázium Kladno
Autor	Mgr. Pelikánová Lucie
Anotace	Výukový materiál Algoritmizace a programování, materiál je určen pro LMS systém, lze jej využít také pro samostatnou práci žáků. Tato část kurzu seznamuje s použitím náhodných čísel a parametrů příkazů v prostředí programovacího jazyka Karel.
Předmět	Informatika a výpočetní technika
Tematická oblast	Algoritmizace a programování
Téma	Karel, parametry procedur a příkazů
Očekávané výstupy	Žák vygeneruje náhodné číslo, které použije pro řízení cyklu repeat. Seznámí se používáním parametrů. Parametry zadané uživatelem předává pomocí dialogového okna.
Klíčová slova	generátor náhodných čísel, random, parametr, skutečný parametr, formální parametr
Druh učebního materiálu	výukový kurz (lekce)
Ročník	3
Cílová skupina	vyšší stupeň osmiletého gymnázia, čtyřleté gymnázium
Ověřeno	27. 5. 2013, O7
Pokud není uvedeno jinak, uvedený materiál je z vlastních zdrojů autora.	

Metodický pokyn

Vstupní soubory a složky:

prg04_karel_parametry.pdf	výukový text
prg04_karel_parametry.krl	praktické ukázky v prostředí Karel

Výstupy (řešení):

Tato ukázková řešení nejsou určena pro žáky, ale jsou pro učitele.

prg04_karel_parametry_reseni.pdf	řešení zadaných úloh
reseni/stena.krl, plot.krl	řešené příklady

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Algoritmizace

Karel – parametry

prg04

Téma: Karel, parametry procedur a příkazů

Náhodná čísla

Nejprve si ukážeme jak v Karlovi pracovat s generátorem náhodných čísel. Ke generování náhodných čísel se používá příkaz `random(n)`, v závorce je **parametr** `n`, který určuje rozsah, ze kterého generujeme náhodná čísla.

Příklad

`random(10)` – vygeneruje náhodné číslo z intervalu $<0,10)$, pozor nejvyšší možné číslo je v tomto případě 9.

Praktické cvičení č. 1

Pomocí cyklu `repeat` a generátoru náhodných čísel nechte Karla položit náhodný počet cihliček z rozsahu 0 – 9. Pojmenujte tento příkaz `poloz_nahoda`.

Co je to parametr?

Údajům, které se mají při různých provedeních příkazu nebo procedury měnit, říkáme **parametry**. Parametry, se kterými se má příkaz nebo procedura skutečně provádět, se používají až při volání a nazývají se **skutečné parametry**.

Procedury s parametry

Používáme tehdy, když potřebujeme několikrát provádět týž algoritmus (postup), ale pokaždé s jinými hodnotami. V okamžiku navrhování procedury její skutečné parametry neznáme a tělo procedury tedy píšeme s použitím tzv. **formálních parametrů**.

Příklad

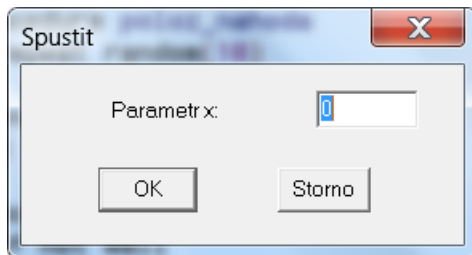
Napišeme proceduru, pomocí které necháme udělat Karla několik opatrných kroků. To kolik jich bude, zadá uživatel až při vykonávání této procedury (skutečné parametry). Při navrhování (deklaraci) procedury použijeme formální parametr `x`. Pozor místo obyčejného kroku (step), použijte tzv. opatrný krok (step?), pokud je před Karlem zeď, krok neudělá.

Proceduru nazvěte: `jdi(x)` – `x` je parametrem této procedury (formální), určuje kolik kroků se vykoná. Místo skutečného počtu kroků, uvnitř této procedury používáme `x`.

```
procedure step?  
  if not wall  
    step  
  end  
end
```

```
procedure jdi(x)  
  repeat x  
    step?  
  end  
end
```

Za x v tomto případě dosadí až uživatel při volání této procedury. Použije se speciální dialogové okno, viz obrázek.



Jakou hodnotu napíše, tolik kroků Karel ujde (pokud mu nepřekáží zeď).

Zídka z dílků

Celé vyplněné políčko budeme označovat v této lekci jako dílek. Nejprve naprogramujeme dílek.

```
procedure dilek
  repeat 9
    put
  end
end
```

Zídka z dílků bude mít proměnnou délku, kterou zadá uživatel (předpokládáme, že se zídka vejde na šířku místnosti).

```
procedure zidka(pd)
  repeat pd
    dilek
  step
  end
end
```

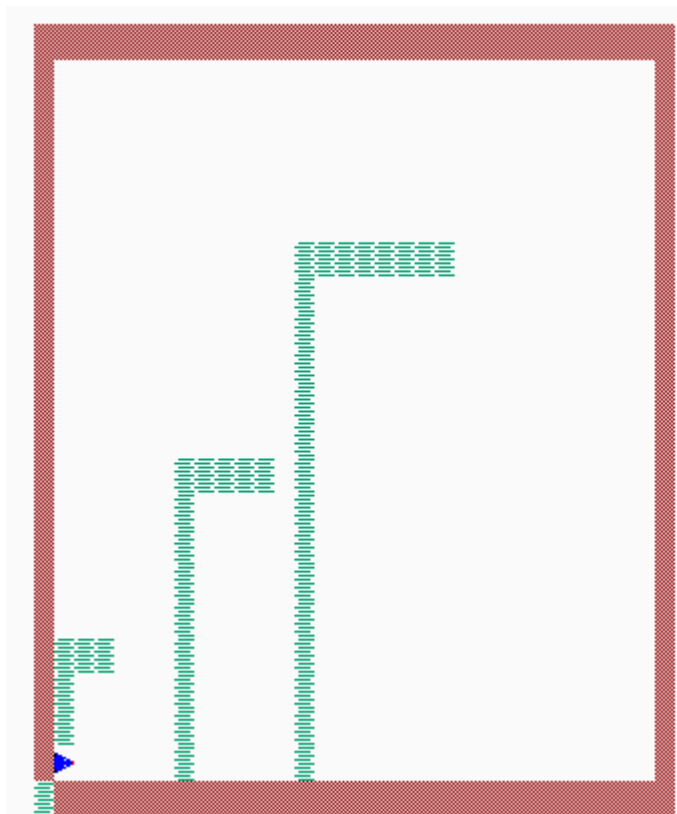
Předávání parametrů mezi procedurami

Skutečný parametr nemusí zadávat pouze uživatel, ale i jiná procedura. Jako příklad si ukážeme stavbu „Šibenice“. Šibenice se bude skládat z nohy a ramena. Nohu budou tvořit dílky postavené na sebe. Rameno je zídka. Rameno bude o 2 dílky kratší než noha. Ukážeme si, jak vytvořit dvě šibenice malou a velkou. Obě se budou stavět z místa, kde právě bude postaven Karel. Nebudeme se zabývat situací, že se šibenice nevejde.

Šibenice

```
procedure noha(xn)
  nasever
  repeat xn
    dilek
  step
  end
  na_rameno
end
```

```
procedure Sibenice(x)
  noha(x)
  rameno(x/3)
end
```



Také je možné stavět jen jednu šibenici a nechat uživatele, aby její velikost zadal. Pozor na předání parametrů, formální parametr musíme zmenšit o 2.

```
procedure Sibenice(x)
  noha(x)
  rameno(x/3)
end
```

Můžeme také speciálním příkazem postavit více šibenice za sebou.

```
procedure Stavime_sibenice
  Sibenice(4)
  home
  jdi(8)
  Sibenice(8)
  home
  jdi(15)
  Sibenice(15)
  home
end
```

Procedura nemusí mít pouze jeden parametr, předávat je možno i několik parametrů současně viz praktické cvičení

Praktické cvičení č. 2

1. Stěna

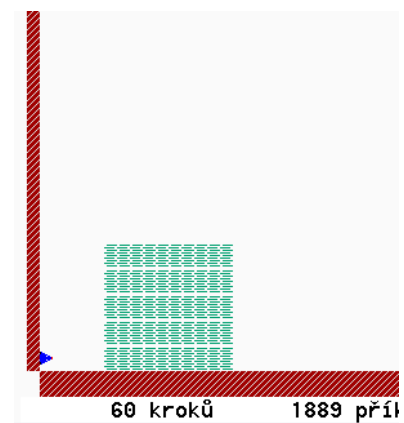
Karel pomocí cihlíček naskládaných do dílků postaví stěnu o rozměrech $m \times n$ políček.

Parametry m a n zadá uživatel.

Jednodušší varianta je, že stěna nepřiléhá přímo ke zdi ani na jedné straně místnosti. Naprogramujte tuto variantu. Předpokládejte na začátku prázdnou místnost.

```
procedure stena(m,n)
  ...
  ...
end
```

Vpravo vidíte ukázkou postavené stěny o rozměrech 10 x 5.



Při přechodu na novou řádku směrem vzhůru řešte zvlášť situaci, kdy je Karel obrácen na východ a kdy na západ (pomocí podmínky).

2. Čtvercový plot

Karel postaví čtvercový plot okolo zahrádky. Velikost jedné strany plotu zadá uživatel. Příkaz plot se naprogramuje s formálním parametrem označeným jako px . Plot bude tvořen vyplněnými políčky s cihlíčkami, které označte jako dílky plotu.

Pozor! Vzhledem k nepochopitelnému poměru mezi výškou a šířkou políček nebude zahrádka vypadat jako čtvercová, ale rozhodující je počet těchto vyplněných polí.

Nápověda

Plot postavte pomocí čtyř stran, rozmyslete, kolik políček bude strana mít. Nezapomeňte, že jeden rohový dílek patří dvěma stranám.



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Algoritmizace

Karel – parametry řešení

prg04

Karel – parametry řešení

(1)

```
procedure poloz_nahoda
  repeat random(10)
    put
  end
end
```

(2.1)

Stěna

```
procedure dilek
  repeat 9
    put
  end
end
```

```
procedure zidka(pd)
  repeat pd
    dilek
    step
  end
end
```

```
procedure prejdi
  if east
    left
    step
    left
    step
  else
    right
    step
    right
    step
  end
end
```

```
procedure stena(m,n)
  repeat n
    zidka m
    prejdi
  end
  home
end
```

(2.2)

Plot

```

procedure plot(px)
  repeat 4
    stranap(px)
  end
end

procedure dilek
  repeat 9
    put
  end
end

procedure stranap(px)
  repeat px-1
    dilek
    step
  end
  left
end

```