



evropský  
sociální  
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,  
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání  
pro konkurenceschopnost

## INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

### Karel, proměnné a cykly

Číslo projektu	CZ.1.07/1.5.00/34.0950
Kódování materiálu	vy_32_INOVACE_inf3_prg06
Označení materiálu	prg06_karel_cykly
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 používáním celočíselných proměnných a s vytvořením a použitím cyklu, který je řízen podmínkou.
Předmět	Informatika a výpočetní technika
Tematická oblast	Algoritmizace a programování
Téma	Karel, proměnné a cykly
Očekávané výstupy	Žák dokáže použít celočíselnou proměnnou včetně základních operací s touto proměnnou. Vytvoří a použije cyklus s podmínkou na začátku (while)
Klíčová slova	proměnná, integer, operace s číselnými hodnotami, cyklus s podmínkou, while,
Druh učebního materiálu	výukový kurz (lekce)
Ročník	3
Cílová skupina	vyšší stupeň osmiletého gymnázia, čtyřleté gymnázium
Ověřeno	4. 6. 2013, 07
Pokud není uvedeno jinak, uvedený materiál je z vlastních zdrojů autora.	

### Metodický pokyn

#### Vstupní soubory a složky:

prg06_karel_cykly.pdf	výukový text
-----------------------	--------------

#### Výstupy (řešení):

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

prg06_karel_cykly_reseni.pdf	řešení zadaných úloh
reseni/karel_ukol.krl,	řešený příklad
reseni/cyklus_s_podminkou.krl	řešené příklady druhá část



evropský  
sociální  
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,  
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání  
pro konkurenceschopnost

## INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Algoritmizace

Karel – proměnné, cykly

prg06

### Téma: Karel, proměnné a cykly

#### Použití číselných proměnných

Karel může pracovat s celočíselnými hodnotami. Proměnné se vytvářejí příkazem `INTEGER`, za kterým následují názvy proměnných oddělené čárkami. Když chcete uložit číslo do proměnné, pak napište název proměnné, rovnítko a číselný výraz.

##### Příklad

```
INTEGER Pocet, delka
```

```
Pocet = 5
```

```
delka = 10
```

Když chcete přečíst hodnotu proměnné, pak stačí napsat název proměnné kdekoli v číselném výrazu.

##### Příklad

```
repeat Pocet
```

```
    jdi (delka)
```

```
end
```

Příkaz `INTEGER` lze použít kdekoli uvnitř procedury. Navíc můžete hned v příkaze `INTEGER` do proměnné přiřadit hodnotu. Stačí za proměnnou napsat rovnítko a číselný výraz. Když to neuděláte, pak bude při vytvoření proměnná obsahovat nějaké nesmysly. **Při skončení procedury zanikají všechny její proměnné. Není možné vytvářet globální proměnné.** To jsou proměnné, které by měly platnost ve všech procedurách našeho programu.

##### Příklad

```
INTEGER Pocet = 10
```

```
INTEGER delka = 5
```

#### Operace s číselnými hodnotami podle priorit

`*` (násobení), `/` (celočíselné dělení), `%` (zbytek po dělení)

`+` (sčítání), `-` (odčítání)

Operace `%` počítá zbytek po dělení (někdy označováno jako `mod`)

Příklad  $7 \% 3 = 1$ ,  $20 \% 5 = 0$ ,  $12 \% 5 = 2$  atd.

### Praktické cvičení č. 1

1. Postupujte podle návodu:

- nový příkaz (proceduru) pojmenujte `ukol`
- vytvořte proměnnou `Pkr` (počet kroků)
- pomocí generátoru náhodných čísel uložte do `Pkr` číslo z intervalu  $<0;15)$
- nechte Karla odejít z domovské pozice na toto místo na spodní řádce (`jdi`)
- políčko, na které Karel došel, vyplňte celé cihličkami (`dilek`)

- Karla otočte a nechte ho přiblížit se k domovu na poloviční vzdálenost, může zůstat otočený na západ (otoc, jdi)

Nezapomeňte strukturovat, použijte pomocné příkazy (jdi, dilek, otoc), cihličky pokládejte opatrným put (put?).

## Cykly s podmínkou

### Cyklus s podmínkou na začátku:

```
While podmínka
    příkazy
end
```

Podmínky jsou stejné jako u podmíněného příkazu. Dokud platí podmínka, vykonávají se příkazy v těle cyklu. Pokud podmínka přestane platit, vykonávají se příkazy následující za cyklem.

#### Příklad 1

Dojděte s Karlem ke zdi – příkaz ke\_zdi

#### Řešení

```
procedure ke_zdi
    while not (wall)
        step
    end
end
```

Zatímco není zed', dělej krok.

#### Příklad 2

Předělejte příkaz na\_sever pomocí cyklu s podmínkou na začátku.

Dokud není Karel otočen na sever, udělá vlevo vbok.

```
procedure na_sever
    while not (north)
        left
    end
end
```

### Praktické cvičení č. 2

1. Obejděte s Karlem místnost (okolo)  
Nápověda: Opakujte příkaz ke\_zdi, u zdi odbočte
2. Pomocí cyklu s podmínkou (space) doplňte počet značek pod Karlem na maximum (doplň)
3. Vysbírejte s Karlem všechny značky na jednom místě seber\_vse
4. Náhodně generujte počet cihliček a postupně jimi vyplňujte celou dolní řadu (pokladej)  
Nápověda: pozor na začátek a konec, jednou budete muset položit značky mimo cyklus.  
V cyklu kontrolujte, zda není Karel u zdi.
5. Vysbírejte s Karlem všechny značky na dolní řádce (sbirej)
6. Zkombinujte úkol 4 a 5, Karel jedním směrem značky pokládá, druhým je sbírá (pokladej\_sbirej)



evropský  
sociální  
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,  
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání  
pro konkurenceschopnost

## INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Algoritmizace

Karel – proměnné, cykly řešení

prg06

### Karel – proměnné, cykly řešení

(1)

```
procedure Ukol
  integer Pkr
  Pkr = random(15)
  home
  jdi(Pkr)
  dilek
  otoc
  Pkr = Pkr/2
  jdi(Pkr)
end
```

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

```
procedure put?
  if space
    put
  end
end
```

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

```
procedure otoc
  left
  left
end
```

## (2.1)

### okolo

```
procedure okolo
  repeat 4
    ke_zdi
    left
  end
end
procedure ke_zdi
  while not(wall)
    step
  end
end
```

## (2.2)

### doplň

```
procedure doplň
  while space
    put
  end
end
```

## (2.3)

### seber\_vse

```
procedure seber_vse
  while mark
    pick
  end
end
```

## (2.4)

### pokladej

```
procedure dilek2(x)
  //x určuje, kolik cihliček bude položeno
  seber_vse
  repeat x
    put
  end
end

procedure pokladej
  //celou spodní
  //řádku vyplní náhodnými počty cihliček
  integer kolik = random(10)
  dilek2(kolik)
  while not(wall)
    step
    kolik = random(10)
    dilek2(kolik)
  end
end
```

## (2.5)

### **sbirej**

```

procedure sbirej
  //sebere všechny cihličky na sspodní
  //řádce
  seber_vse
  while not(wall)
    step
    seber_vse
  end
end

```

```

procedure seber_vse
  while mark
    pick
  end
end

```

## (2.6)

### **pokladej\_sbirej**

```

procedure pokladej_sbirej
  pokladej
  otoc
  sbirej
end

```

```

procedure otoc
  left
  left
end

```