



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Karel, parametry procedur pro pokročilé

Číslo projektu	CZ.1.07/1.5.00/34.0950
Kódování materiálu	vy_32_INOVACE_inf3_prg07
Označení materiálu	prg07_karel_parametry_pokrocile
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 lekce je určena pouze pro pokročilé žáky, které seznamuje s různými způsoby předávání parametrů (hodnotou a odkazem). Součástí je jeden řešený příklad pro žáka a námět na samostatnou úlohu.
Předmět	Informatika a výpočetní technika
Tematická oblast	Algoritmizace a programování
Téma	Karel, parametry procedur pro pokročilé
Očekávané výstupy	Žák se seznámí se dvěma způsoby předání parametrů procedurám – hodnotou a odkazem. Probranou látku procvičí na řešeném příkladu.
Klíčová slova	parametr předaný odkazem, parametr předaný hodnotou, parametry procedur
Druh učebního materiálu	výukový kurz (lekce)
Ročník	3
Cílová skupina	vyšší stupeň osmiletého gymnázia, čtyřleté gymnázium
Ověřeno	11. 6. 2013, O7
Pokud není uvedeno jinak, uvedený materiál je z vlastních zdrojů autora.	

Metodický pokyn

Vstupní soubory a složky:

prg07_karel_parametry_pokrocile.pdf	výukový text
karel_parametry_pokrocile.krl	ukázkový příklad

Výstupy (řešení):

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

prg07_parametry_pokrocile_reseni.pdf	řešení zadaných úloh
reseni/cihla_ridi.krl	řešený příklad

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Algoritmizace

Karel – parametry, pokročilé

prg07

Téma: Karel, parametry procedur pro pokročilé

V základní výuce o parametrech procedur jsme používali pouze jeden druh parametrů tzv. **parametry volané hodnotou**. Při realizaci programu musel v okamžiku volání procedury již být formální parametr nahrazen skutečným parametrem, tedy v případě Karla konkrétním celým číslem. V tomto případě nebylo možné, aby Karel v proceduře něco spočítal a svůj výsledek předal proceduře, která ho volala. Takové použití parametrů, při kterém parametry mohou vracet vypočtené hodnoty, se nazývá **volání parametrů odkazem**.

Parametr volaný hodnotou

Při volání parametrů hodnotou se vytvoří nové paměťové místo, do kterého se při volání uloží kopie hodnoty parametru. Pod skončení procedury, toto paměťové místo zanikne.

Parametr volaný odkazem

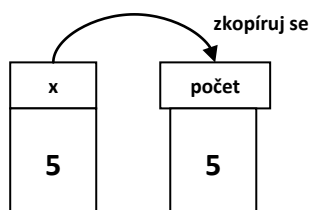
Při volání parametrů odkazem se **nevytvoří** nové paměťové místo, ale vytvoří se pouze odkaz, který nese jméno parametru, ale ukazuje na původní paměťové místo. Pod skončení procedury se v proměnné, pomocí které se procedura volala, uchová změněná (pokud se mění) hodnota proměnné. Je nutné tuto proměnnou označit speciálním znakem &, např. &pocet

Obrázek ukáže rozdíl.

Parametr volaný hodnotou

jdi_sp(x), kde x = 5

procedure jdi_sp(pocet)

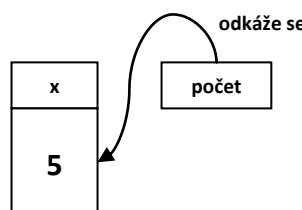


po skončení procedury se
parametr x nezměnil

Parametr volaný odkazem

jdi_sp(x), kde x = 5

procedure jdi_sp(&pocet)



po skončení procedury se
může parametr x změnit,
záleží na proceduře jdi_sp

Příklad

Procedura **jdi_a_stav** dostane parametr x, tento parametr jednou předá proceduře **jdi_sp1**, pro kterou je volán hodnotou. Procedura udělá x kroků a nastaví x na číslo 9, ale toto nastavení se neprojeví. Počet cihlíček, které procedura postaví, je pořád roven původní hodnotě x. Procedura

jdi_sp2 má parametr předávaný odkazem, po jejím skončení je původní parametr přepsán na hodnotu 9.

```
procedure jdi_a_stav(x)
//uživatel zadá x
//procedura volá jdi_sp1 a použije x
//jako parametr
//po chůzi se postaví původně zadaný počet cihel
//program může i havarovat x>9
    jdi_sp1(x)
    repeat x
        put
    end

//jdi_sp2, změni parametr na 9
//po chůzi se vždy postaví přesně 9 cihel
    jdi_sp2(x)
    repeat x
        put
    end
    step
end
```

```
procedure jdi_sp1(pocet)
//tato procedura udělá x kroků
//a pocet změni na 9, změna se nikde neprojeví
    repeat pocet
        step
    end
    pocet = 9
end
```

```
procedure jdi_sp2(&pocet)
//tato procedura udělá x kroků
//a x změni na 9
    repeat pocet
        step
    end
    pocet = 9
end
```

Porovnejte použití obou procedur.

Karel se řídí podle cihel

Využijeme znalost předávání parametrů odkazem, necháme Karla počítat cihly a podle toho, kolik cihel napočítá (sebere), tolik kroků udělá. Sbírání a chůzi rozdělte do různých příkazů (procedur). Uložte jako cihla_ridi.



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Algoritmizace

Karel – parametry, pokročilé – řešení

prg07

Karel – parametry, pokročilé – řešení

(1)

cihla_ridi

```
procedure cihla_ridi
  integer pk
  //pomocný příkaz, umístí Karla, položí cihly
  priprava
  pocitej(pk)
  repeat pk
    step?
  end
end
```

```
procedure pocitej(&pc)
  pc = 0
  while mark
    pick
    pc=pc+1
  end
end
```

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

```
procedure priprava
  //pouze pomocné, lze řešit i jinak
  home
  step
  repeat random(9)
    put
  end
  home
  step
end
```