

PODPROGRAMY – PROCEDURY A FUNKCE

Programy bez podprogramů

Příklady:

- a) Napište program, který na obrazovku nakreslí čáru složenou ze znaků pomlčka.

```
program Cara;  
var  
  i: integer;  
begin  
for i:=1 to 70 do write('-');  
writeln;  
end.
```

- b) Napište program, který na obrazovku nakreslí X čar složených ze znaků pomlčka (při praktickém použití se jednotlivé čáry mohou vyskytovat na různých místech v programu).

```
program Cary;  
var  
  i: integer;  
begin  
for i:=1 to 70 do write('-');  
writeln;  
for i:=1 to 70 do write('-');  
writeln;  
for i:=1 to 70 do write('-');  
writeln;  
for i:=1 to 70 do write('-');  
writeln;  
for i:=1 to 70 do write('-');  
writeln;  
readln;  
end.
```

- c) Napište program, který vypočítá N-tou mocninu čísla X (X^N , kde X je reálné a N je celé kladné).

```
program Mocnina;  
var  
  Zaklad, Vysledek: real;  
  Exponent, i: integer;  
  
begin  
write('Zadej cislo: ');  
readln(Zaklad);  
write('Zadej mocninu: ');  
readln(Exponent);  
Vysledek:=1;  
for i:=1 to Exponent do  
  Vysledek:=Vysledek*Zaklad;  
writeln(Exponent, '. mocnina cisla ', Zaklad:10:2, ' je ', Vysledek:10:2);  
end.
```

Nevýhoda: Opakovaný zápis celého kódu na více místech programu.

Podprogramy umožňují rozložit program na dílčí úlohy. Přispívají tak k hierarchické struktuře programu a tím lepší přehlednosti a srozumitelnosti programu. Podprogramy se dělí na procedury a funkce.

Procedura je popis algoritmu, který nějakým způsobem transformuje data. **Funkce** popisuje algoritmus výpočtu jedné hodnoty, která může být použita jako operand ve výrazu.

Procedury

a) Procedura bez parametrů

```
program Cary;  
procedure OddelCarou;  
  var  
    i: integer;  
  begin  
    for i:=1 to 70 do write('-');  
    writeln;  
  end;  
  
begin  
  OddelCarou;  
  OddelCarou;  
  OddelCarou;  
end.
```

Výhoda: pro opakované použití procedury na více místech programu pouze zavolám jméno procedury.

Procedura se skládá z hlavičky (klíčového slova **procedure**, jména procedury, případně lokální deklarace) a těla ukončeného středníkem. U procedury s parametry (viz dále) se v hlavičce ještě uvede seznam specifikací parametrů. Jednotlivé specifikace parametrů se oddělují středníkem.

```
  procedure Identifikátor (seznam specifikací parametrů);  
    lokální deklarace  
  begin  
    posloupnost příkazů  
  end;
```

Pro vyvolání procedury se použije **příkaz procedury**, který je tvořen jménem procedury (a případně seznamem skutečných parametrů u procedury s parametry).

```
program Mocninal;  
var  
  Zaklad, Vysledek: real;  
  Exponent, i: integer;  
  
procedure Mocnina;  
  begin  
    Vysledek:=1;  
    for i:=1 to Exponent do  
      Vysledek:=Vysledek*Zaklad;  
    end;  
  
begin  
  write('Zadej cislo: ');  
  readln(Zaklad);  
  write('Zadej mocninu: ');  
  readln(Exponent);  
  Mocnina;  
  writeln(Exponent, '. mocnina cisla ', Zaklad:10:2, ' je ', Vysledek:10:2);  
end.
```

Použití proměnných z hlavního programu (tzv. **globálních proměnných**) v proceduře není dobré řešení. Vnitřek procedury by měl být nezávislý na globálních proměnných (globální proměnné by neměly být procedurou ovlivňovány ani používány). Procedura by měla mít všechny proměnné, které používá, **deklarovány lokálně**. Globální proměnné existují po celou dobu provádění programu, lokální pouze po dobu provádění dané procedury.

b) Procedura s parametry

Parametry volané hodnotou se používají pro vstup, vytváří se jejich kopie, se kterou procedura pracuje. Obsah proměnných dosazených jako parametry zůstane nedotčen.

Parametry volané odkazem se obvykle používají pro výstup, procedura pracuje přímo s dosazenými (například se do nich ukládá výsledek operace).

Teoreticky bychom mohli používat pouze parametry volané odkazem, ale není dobré během provádění procedury měnit původní vstupní hodnoty, které můžeme ještě v programu potřebovat. Jako parametry volané hodnotou mohou být dosazeny i konstanty, což u parametrů volaných odkazem není možné. Např.

```
Mocnina(10, 3, Vysledek);
```

```
program Mocnina2;
var
  Zaklad, Vysledek: real;
  Exponent, i: integer;
procedure Mocnina(Z: real; E: integer; var V: real);
begin
  V:=1;
  for i:=1 to E do
    V:=V*Z;
  end;
begin
write('Zadej cislo: ');
readln(Zaklad);
write('Zadej mocninu: ');
readln(Exponent);
Mocnina(Zaklad, Exponent, Vysledek);
writeln(Exponent, '. mocnina cisla ', Zaklad:10:2, ' je ', Vysledek:10:2);
end.
```

Z, E a V je tzv. **formální parametry** (zástupné, za které se dosadí skutečné parametry v okamžiku volání procedury).

Při volání funkce Mocnina jsou Základ, Exponent a Výsledek tzv. **skutečné parametry**.

Použití proměnné **i** z hlavního programu (tzv. globálních proměnných) v proceduře opět není dobré řešení. Zejména proměnné cyklu by procedury měly mít deklarované lokálně, aby nedošlo k nechtěné změně proměnné cyklu (mohla by způsobit např. nekonečný cyklus).

Procedura s parametry + lokální deklarace

```
program Mocnina3;
var
  Zaklad, Vysledek: real;
  Exponent: integer;
procedure Mocnina(Z: real; E: integer; var V: real);
var
  i: integer;
begin
  V:=1;
  for i:=1 to E do
    V:=V*Z;
  end;
begin
write('Zadej cislo: ');
readln(Zaklad);
write('Zadej mocninu: ');
readln(Exponent);
Mocnina(Zaklad, Exponent, Vysledek);
writeln(Exponent, '. mocnina cisla ', Zaklad:10:2, ' je ', Vysledek:10:2);
end.
```

Opakované volání procedury

```
program Mocnina4;
var
  Zaklad, Vysledek: real;
  Exponent: integer;

procedure Mocnina(Z: real; E: integer; var V: real);
var
  i: integer;
begin
  V:=1;
  for i:=1 to E do
    V:=V*Z;
  end;

begin
write('Zadej cislo: ');
readln(Zaklad);
write('Zadej mocninu: ');
readln(Exponent);
Mocnina(Zaklad, Exponent, Vysledek);
writeln(Exponent, '. mocnina cisla ', Zaklad:10:2, ' je ', Vysledek:10:2);

Zaklad := 10;
Exponent := 3;
Mocnina(Zaklad, Exponent, Vysledek);
writeln(Exponent, '. mocnina cisla ', Zaklad:10:2, ' je ', Vysledek:10:2);

Mocnina(10, 3, Vysledek);
writeln('3. mocnina cisla 10 je ', Vysledek:10:2);
end.
```

Funkce

Použití funkce na pravé straně jakéhokoli výrazu, kde se očekává hodnota daného typu

```
program Mocnina5;
var
  Zaklad, Vysledek: real;
  Exponent: integer;

function Mocnina(Z: real; E: integer) : real;
var
  i: integer;
  V: real;
begin
  V:=1;
  for i:=1 to E do
    V:=V*Z;
  Mocnina := V;
end;

begin
write('Zadej cislo: ');
readln(Zaklad);
write('Zadej mocninu: ');
readln(Exponent);
Vysledek := Mocnina(Zaklad, Exponent);
writeln(Exponent, '. mocnina cisla ', Zaklad:10:2, ' je ', Vysledek:10:2);
end.
```

Funkce je algoritmus, jehož provedením se vypočte hodnota určitého typu. Funkce se skládá z hlavičky (klíčového slova **function**, jména funkce, specifikace parametrů a **typu funkce** – typ hodnoty, kterou funkce počítá – a lokální deklarace) a těla ukončeného středníkem. Typ funkce nesmí být strukturovaný typ. V těle funkce se musí vyskytovat **přiřazovací příkaz**, kterým se přiřazuje hodnota identifikátoru (jména) funkce a jehož provedením se definuje výsledná hodnota.

```
function Identifikátor (seznam specifikací parametrů):typ funkce;  
    lokální deklarace  
begin  
    posloupnost příkazů  
end;
```

Pro vyvolání funkce se použije **zápis funkce**, který je tvořen jménem funkce (a případně seznamem skutečných parametrů u procedury s parametry). Zápis funkce musí být použit vždy v nějaké konstrukci, která předepíše, co se s výslednou funkční hodnotou provede.

Výhoda: pro opakované použití funkce na více místech programu pouze zavolám jméno funkce.

Použití funkce přímo ve výstupu

```
program Mocnina6;  
var  
    Zaklad: real;  
    Exponent: integer;  
  
function Mocnina(Z: real; E: integer) : real;  
    var  
        i: integer;  
        V: real;  
    begin  
        V:=1;  
        for i:=1 to E do  
            V:=V*Z;  
        Mocnina := V;  
    end;  
  
begin  
    write('Zadej cislo: ');  
    readln(Zaklad);  
    write('Zadej mocninu: ');  
    readln(Exponent);  
    writeln(Exponent, '. mocnina cisla ', Zaklad:10:2, ' je ', Mocnina(Zaklad,  
Exponent):10:2);  
end.
```

Použití funkce a procedury

```
program Mocnina7;
var
  Zaklad: real;
  Exponent: integer;

function Mocnina(Z: real; E: integer) : real;
var
  i: integer;
  V: real;
begin
  V:=1;
  for i:=1 to E do
    V:=V*Z;
  Mocnina := V;
end;

procedure OddelCarou;
var
  i: integer;
begin
  for i:=1 to 70 do write('-');
  writeln;
end;

begin
  OddelCarou;
  write('Zadej cislo: ');
  readln(Zaklad);
  write('Zadej mocninu: ');
  readln(Exponent);
  writeln(Exponent, '. mocnina cisla ', Zaklad:10:2, ' je ', Mocnina(Zaklad,
  Exponent):10:2);
  OddelCarou;
  readln;
end.
```

Zpracovala: Ing. Simona Martínková, březen 2007