

## PODPROGRAMY (PROCEDÚRY a FUNKCIE)

Pri tvorbe väčších programov je vhodné si celú úlohu najprv rozdeliť na menšie podúlohy. Potom riešime každú takúto podúlohu zvlášť a pritom dbáme nato, aby to dokopy dávalo riešenie našej pôvodnej úlohy. Často pre každú takúto podúlohu vytvoríme samostatnú podprogram – *zadefinujeme podprogram*. Podprogram pritom dostáva meno, aby sme sa pri riešení úlohy mohli na tento podprogram odvolávať práve týmto menom – budeme tomu hovoriť *volanie podprogramu*. Znamená to, že ak na nejakom mieste programu uvedieme meno podprogramu, na tomto mieste sa použije (zavolá) príslušný podprogram.

Keď už raz zadefinujeme nejaký podprogram, tak ho potom už môžeme volať na viacerých miestach. Definovali sme ho raz ale používame ho veľa krát. Ak pri ladení programu v ňom nájdeme nejakú chybu, tak ju opravíme len raz. Programátori často, po napísaní podprogramu, ho najprv samostatne odladia a až potom ho začnú používať v svojom hlavnom programe. Rozdelenie úlohy na podprogramy má aj tú výhodu, že na týchto rozdelených podúlohách môže naraz pracovať viac programátorov.

poznáme dva typy podprogramov:

**procedúra** – postupnosť príkazov, ktoré riešia nejakú podúlohu

**funkcia** – postupnosť príkazov, ktoré majú za úlohu vypočítať výsledok

### PROCEDÚRY:

... aby sme mohli procedúru používať, musíme ju najprv deklarovať. Deklaráciu umiestňujeme v programe do časti deklarácii a definícií. Deklarácia je tvorená z hlavičky a tela. Má presne definovanú organizáciu, je možné ju deklarovať v rôznych zjednodušených tvaroch:

1. *bez parametrov a lokálnych objektov*
2. *bez parametrov s lokálnymi objektmi*
3. *s parametrami*

**lokálne premenné** – premenné definované v deklaračnej časti procedury, využívajú sa len počas výpočtu v danej procedúre

**globálne premenné** – premenné, ktoré sú definované v deklaračnej časti programu a existujú počas celého programu

1. **bez parametrov a lokálnych objektov:**

deklarácia procedúry má tvar:

```
procedure m; {hlavička = slovo procedure + meno procedúry}
begin      {telo = zložený príkaz}
prikaz1;
prikaz2;
....
end;
```

napr. Napíšte podprogram, ktorý vymení hodnoty dvoch premenných a,b.

*procedure vymena;*

*begin*

*pom:=a;*

*a:=b;*

*b:=pom;*

*end;*

volanie procedúry v tele programu pozostáva iba z jej mena

napr: *vymena;*

### **PRIKLAD:**

**Napíšte program s použitím procedúry, ktorá vymení navzájom hodnoty dvoch čísel**  
*navymenu.pas*

*program navymenu;*

*var a, b, pom: integer;*

*procedure vymena;*

*begin*

*pom:=a;*

*a:=b;*

*b:=pom;*

*end;*

*begin*

*write( 'zadaj cisla a, b');*

*readln (a,b);*

*writeln( ' hodnoty a, b pred vymenou a=', a, ', b= ',b);*

*vymena;*

*writeln( ' hodnoty a, b po vymene a=', a, ', b= ',b);*

*readln;*

*end.*

## **2. bez parametrov s lokálnymi objektmi:**

### **PRIKLAD:**

*program navymenu;*

*var a, b: integer;*

*procedure vymena; {hlavička =slovo procedure + meno procedúry}*

*var pom: integer;     {definovanie lokalnych premennych}*

*begin*

*pom:=a;*

*a:=b;*

*b:=pom;*

*end;*

*begin*

```

write( 'zadaj cisla a, b');
readln (a,b);

writeln( ' hodnoty a, b pred vymenou a=', a, ', b= ',b);
vymena;
writeln( ' hodnoty a, b po vymene a=', a, ', b= ',b);
readln;
end.

```

### 3. s parametrami

... procedúry bez parametrov „komunikovali“ s okolím iba cez globálne premenné. To obmedzuje ich použitie. Procedúry s parametrami umožňujú „priväzať“ do tela procedúry údaje, ktoré má spracovať cez parametre. Cez parametre sa aj „vyvážajú“ výsledky. Rovnako ako v procedúrach bez parametrov nemali v čase deklarácie premenné použité v ich telách priradené hodnoty, takisto nemajú v čase deklarácie priradené hodnoty ani parametre. Hovoríme preto o tzv. formálnych parametroch. V príkaze procedúry už musíme parametre „poznať“, v príkaze procedúry vystupujú tzv. skutočné parametre.

deklarácia procedúry ma tvar:

```

procedure meno(s1,s2,...sn: typ premennej); {hlavička }
begin..... {telo}
prikazy.....
End;

```

#### PRIKLAD:

Napište program s použitím procedúry, ktorý vás pozdraví

```

program pozdrav;
procedure ahoj(kto:string);
begin
  writeln('Ahooooooj ',kto);
end;

begin
  ahoj('zuz'); (* vypíše "Ahoooooooj Zuz" *)
readln;
end.

```

## FUNKCIE

...funkcia je špeciálna procedúra, ktorá dáva ako výsledok **jedinú hodnotu!!!**

*Deklarácia funkcie:*

```
function meno (s1,s2,...,sn:integer):t; {hlavička funkcie}

begin
prikazy.....          {telo funkcie}
end;
```

*meno: meno f-cie*

*s1,s2,...,sn : skutočné parametre*

*t: typ výslednej hodnoty funkcie*

### PRIKLAD:

**Napíšte program, ktorý bude počítat' obsah trojuholníka**

```
program obsah;
var b,vb:integer;
```

```
function obsahtroj(a, va:real):real;
var
begin
obsahtroj:=a*va/2;
end;
```

```
begin
Write('zadaj stranu a vysku');
read(b,vb);
write('obsah je ',obsahtroj(b,vb):4:2,' ');
readln;
readln;
end.
```