

## Podprogramy – procedury a funkce

### Deklarace a volání procedur bez parametrů

Deklarace procedury se skládá z hlavičky procedury, těla procedury a končí středníkem. Deklarace jednoduché procedury, která nemá parametry a jejíž tělo je tvořeno pouze složeným příkazem má tvar:

```
Procedure identifikátor;  
lokální deklarace;  
begin  
    pospoupanost příkazů;  
end;
```

Ve složeném příkazu, který tvoří tělo takto deklarované procedury a jejíž deklarace předchází deklaraci této procedury. Pomocí těchto objektů, které nazýváme nelokální objekty může procedura komunikovat s okolím. Složenému příkazu v těle procedury mohou předcházet lokální deklarace (např. deklarace konstant a proměnných), které platí pouze pro danou proceduru a nemají účinek vně procedury.

Příklad:

```
procedure UkazZpravu;  
begin  
    ShowMessage ('prave probiha procedura ukaz zpravu');  
    Beep;  
end;
```

### Deklarace a volání procedury s parametry volanými hodnotou

```
Procedure název procedury (seznam formálních parametrů volaných hodnotou);  
Deklarace lokálních proměnných;  
begin  
    P1;  
    P2;  
    Pn;  
end;
```

Je – li formálních parametrů více, oddělují se středníkem.

Příklad (A : Integer; B :Real);

Je – li formálních parametrů více a jsou li stejného typu, oddělují se čárkou.

Příklad : (A,B:Integer);

Formální parametry volané hodnotou představují v těle procedury odpovídající lokální proměnné, do kterých se po zavolání procedury nakopírují skutečné hodnoty. Volání procedury s parametry volanými hodnotou:

**název procedury** (seznam skutečných parametrů);

Zásady:1) Počet skutečných a formálních parametrů musí být stejný

2) Odpovídající si skutečné a formální parametry musí být stejného typu

## Deklarace a volání procedury s parametry volanými odkazem

**Procedure** název procedury (seznam formálních parametrů volaných hodnotou a odkazem );

Deklarace lokálních proměnných;

**begin**

P1;

P2;

Pn;

**end;**

Je-li formálních parametrů více, oddělují se středníkem.

Příklad (A : Integer;B : Real;var C : Real);

Proměnné A a B jsou formální parametry volané hodnotou. Proměnná C je formální parametr volaný odkazem. Je-li formálních parametrů více a jsou li stejného typu, oddělují se čárkou. Formální parametry volané hodnotou představují v těle procedury odpovídající lokální proměnné, do kterých se po zavolání procedury nakopírují skutečné hodnoty. Formální parametr volaný odkazem představuje v těle procedury vždy tu konkrétní proměnnou, která je určena skutečným parametrem a jejíž adresa umístění v paměti se vypočte na začátku provedení procedury. Z tohoto rozdílu vyplývá i nejčastější použití těchto parametrů. Pomocí parametrů volaných hodnotou reprezentujeme v proceduře její vstupní hodnoty, zatímco pomocí parametrů volaných odkazem reprezentujeme v proceduře její výstupní proměnné, jejichž hodnoty budou provedením procedury definovány. Specifikace parametrů volaných odkazem začíná klíčovým slovem var. Každý parametr volaný odkazem představuje v těle procedury proměnnou, jejíž typ je dán typem formálního parametru a která je určena skutečným parametrem. Přípustným skutečným parametrem je tedy pouze proměnná, jejíž typ je totožný s typem formálního parametru. Volání procedury s parametry volanými hodnotou i odkazem:

**název procedury** (seznam skutečných parametrů);

Zásady:1) Počet skutečných a formálních parametrů musí být stejný

2) Odpovídající si skutečné a formální parametry musí být stejného typu

3) Hodnoty skutečných parametrů volaných hodnotou se po zavolání procedury nakopírují do odpovídajících lokálních proměnných.

4) Skutečný parametr volaný odkazem (je globální proměnná Z) bude po zavolání procedury ukazovat na stejné místo v paměti jako formální parametr volaný odkazem C.

5) Pomocí formálního parametru volaného odkazem procedura předává výsledek mimo proceduru k dalšímu zpracování.

Příklad:

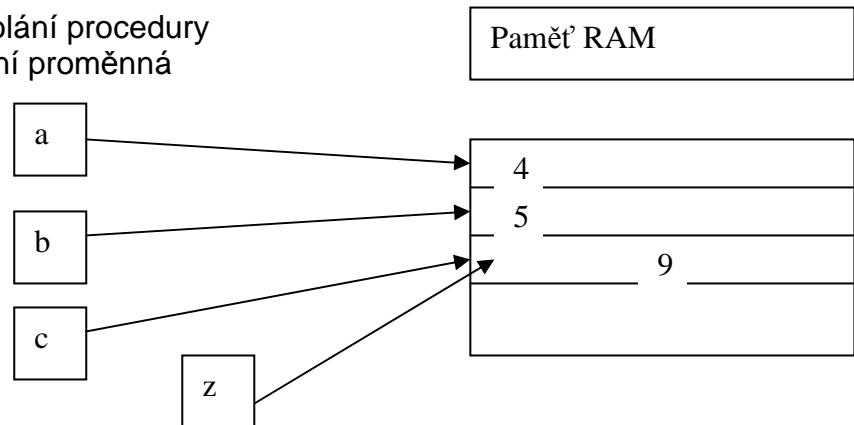
**procedure** soucet (a,b:real;var c:real);//a,b,c jsou lokální proměnné

**begin**

c:=a+b;

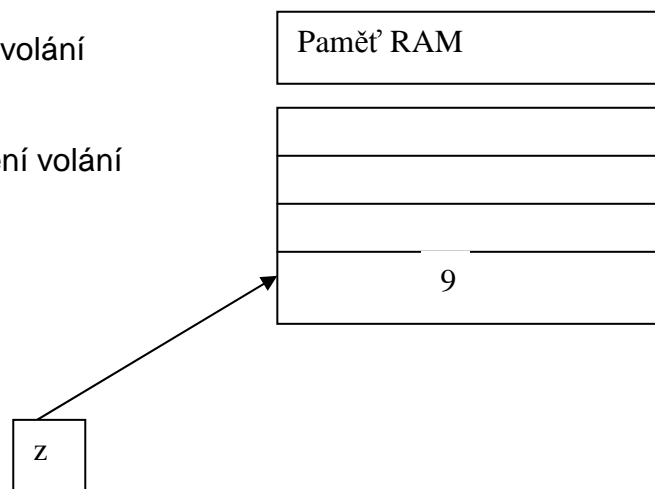
**end;**

Obsah paměti RAM po zavolání procedury  
Soucet (4,5,Z); //Z je globální proměnná



Obsah paměti RAM po ukončení volání  
procedury :

Lokální proměnné a,b,c po ukončení volání  
zanikají ,  
globální proměnná z zůstává!!!



### Deklarace a volání funkcí

Jako funkce se v programu zapisují algoritmy, jejichž provedením se vypočte hodnota určitého typu. V hlavičce funkce se definuje identifikátor funkce, do kulatých závorek se specifikují parametry funkce podobně jako u procedur. Za kulatými závorkami se píše dvojtečka a za ní se specifikuje typ hodnoty, kterou funkce vypočítá. Deklarace funkce:

**function** identifikátor (seznam specifikací parametrů) : typ funkce;

lokální deklarace;

**begin**

posloupnost příkazů;

**end;**

Typ funkce nesmí být strukturovaný typ a musí být označen identifikátorem typu. Specifikace parametrů mají stejný význam jako u procedur. V těle funkce musí být přiřazovací příkaz, který přiřadí vypočítanou funkční hodnotu identifikátoru funkce. Při volání funkce se přípustnost skutečných parametrů, jejich substituce za formální parametry a způsob provedení těla funkce při jejím vyvolání řídí stejnými pravidly

jako pro procedury. Zdůrazněme však, že zatímco vyvolání procedury je příkaz, zápis volání funkce je výraz označující hodnotu, která je výsledkem provedení těla funkce. Zápis funkce tedy musí být použit vždy v nějaké konstrukci, která předepisuje, co se s výslednou funkční hodnotou provede.

Volání funkce:

proměnná:=**název funkce** (seznam skutečných parametrů);