



MINISTERSTVO ŠKOLSTVÍ,  
MLÁDEŽE A TĚLOVÝCHOVY



**OP Vzdělávání  
pro konkurenceschopnost**

INVESTICE  
DO ROZVOJE  
VZDĚLÁVÁNÍ

**Střední průmyslová škola a Vyšší odborná škola technická Brno, Sokolská 1**

**Šablona: Inovace a zkvalitnění výuky prostřednictvím ICT**

**Název: Programování se strukturovanými údaji**

**Téma: Funkce**

**Autor: Ing. Hodál Jaroslav, Ph.D.**

**Číslo: VY\_32\_INOVACE\_26-17**

**Anotace:** *Materiál popisuje definici a použití funkcí v jazyce Pascal.  
Materiál je určen pro 3. a 4. ročník oboru strojírenství a technické lyceum.  
Vytvořeno: květen 2013.*

# 17. Funkce

- funkce je podprogram (samostatná část programu), která obsahuje blok příkazů obvykle řešící nějakou ucelenou jednodušší úlohu
- funkci použijeme tehdy, pokud chceme osamostatnit část kódu programu, která má jediný nebo jeden dominantní výsledek
- pokud nemá podprogram žádný výsledek, je vhodné použít místo funkce proceduru (viz 26-18 – Procedury)
- definováním funkce vytváříme nový jednoduchý příkaz, který bude pojmenovaný stejně jako funkce
- pomocí tohoto názvu funkci později voláme

# Použití a syntaxe procedur

- pro přenesení informací mezi programem a funkcí můžeme použít buď globální proměnné programu (méně vhodné) nebo rozhraní tvořené tzv. parametry (lepší řešení, viz 25-18), každá **funkce má** však také jeden implicitní **výsledek**
- obecný zápis funkce:

```
function název (parametry) : typ_výsledku ;
```

```
    deklarace konstant
```

```
    definice datových typů
```

```
    deklarace proměnných
```

```
    podprogramy
```

```
begin
```

```
    příkazy...
```

```
end;
```

# Příklady funkcí (1)

```
program test1;  
var x:byte;  
function soucet(a,b:byte):word;  
begin  
    soucet := a+b;  
end;  
  
begin  
    readln(x) ;  
    writeln(soucet(x,5) ) ;  
    ...  
end.
```

- funkce se dvěma vstupními daty a výsledkem

# Příklady funkcí (2)

```
procedure kvr(a,b,c:shortint;  
             var x1,x2:real):byte;  
var d: integer;  
begin  
  d := sqr(b)-4*a*c;  
  if d>0 then begin  
    kvr:=2; x1:= (-b-sqrt(d))/(2*a);  
    x2:= (-b+sqrt(d))/(2*a);  
  end else  
    if d=0 then begin  
      kvr := 1; x1 := (-b)/(2*a);  
    end else  
      kvr := 0;  
end;
```

- funkce s jedním dominantním výsledkem předávaným implicitně a dvěma dalšími předávanými přes parametry

# Příklady funkcí (3)

```
program test2;  
var v1, v2: real;  
procedure kvr ...  
    ...  
end;  
  
begin  
    case kvr(-1,2,5,v1,v2) of  
        0: writeln('Nemá řešení');  
        1: writeln('1 řešení - ', v1);  
        2: writeln('2 řešení - ', v1, ' ' v2);  
    end;  
end.
```

- použití funkce z předešlé ukázky