



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: Procedury

Autor: Ing. Hodál Jaroslav, Ph.D.

Číslo: VY_32_INOVACE_26–16

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

16. Procedurey

- procedura je podprogram (samostatná část programu), která obsahuje blok příkazů obvykle řešící nějakou ucelenou jednodušší úlohu
- proceduru použijeme tehdy, pokud chceme osamostatnit část kódu programu, která nemá žádný nebo naopak má několik (více než 1) stejně důležitých výsledků
- pokud by měl podprogram jeden výsledek, je výhodnější použít místo procedury funkci (viz 26-17 – Funkce)
- definováním procedury vytváříme nový jednoduchý příkaz, který bude pojmenovaný stejně jako procedura
- pomocí tohoto názvu proceduru později voláme

Použití a syntaxe procedur

- pro přenesení informací z hlavního programu do procedury a z procedury zpět do hlavního programu můžeme použít buď globální proměnné programu (méně vhodné) nebo rozhraní tvořené tzv. parametry (lepší řešení, viz 25-18)
- obecný zápis procedury:

procedure **název** (**parametry**) ;

deklarace konstant

definice datových typů

deklarace proměnných

podprogramy

begin

příkazy...

end;

Příklady procedur (1)

```
program test1;  
  procedure chyba;  
  begin  
    writeln('V programu nastala chyba!');  
  end;  
  
begin  
  ...  
  chyba;  
  ...  
end.
```

- procedura bez vstupních a výstupních dat

Příklady procedur (2)

```
program test2;  
var k: byte;  
procedure chyba;  
begin  
    writeln('Nastala chyba číslo ', k, '!');  
end;  
begin  
    ...  
    k := 5;  
    chyba;  
end.
```

- procedura se vstupní informací získanou prostřednictvím globální proměnné

Příklady procedur (3)

```
program test3;  
procedure chyba(kod:byte) ;  
begin  
    writeln('Nastala chyba číslo ',kod, '!') ;  
end;  
  
begin  
    ...  
    chyba (5) ;  
    ...  
end.
```

- procedura se vstupní informací získanou prostřednictvím parametru

Příklady procedur (4)

```
program test4;  
var x:byte; z:word;  
procedure soucet(a,b:byte; var c:word);  
begin  
    c := a+b;  
end;  
begin  
    readln(x); readln(y);  
    soucet(x,5,y);  
    writeln(y);  
end.
```

- procedura se dvěma vstupními informacemi a jednou výstupní předávanými prostřednictvím parametrů