



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: Podprogramy**

**Autor: Ing. Hodál Jaroslav, Ph.D.**

**Číslo: VY\_32\_INOVACE\_26–14**

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

# 14. Podprogramy

- podprogramy představují možnost, jak lze programovací jazyk doplnit o nové **vlastní jednoduché příkazy**
- všechny jednoduché příkazy (viz 25-07) jsou vlastně podprogramy, které však za nás předem nachystali autoři vývojového prostředí
- jejich název funguje jako **náhrada** (substituce) za celý blok programového kódu, který může být tvořen libovolným počtem příkazů
- pomocí podprogramů lze tedy zjednodušit takové programy, ve kterých se **opakovaně vyskytuje** identická **sekvence příkazů**

# Použití podprogramů

- kdyby se opakovaná sekvence vyskytovala několikrát po sobě **na stejném místě** programu, bylo by možné program zjednodušit použitím **cyklů** (viz 25-15, 18 a 19)
- v situaci, kdy jsou výskyty zmíněné sekvence „roztroušeny“ po celém programu už ale cykly využít nelze, a pak je jedinou cestou ke zjednodušení kódu právě použití podprogramů
- podprogramy jsou vytvářeny pomocí konstrukcí, které představují jednodušší, téměř nezávislé **programy v programu** hlavním
- programový kód umístěný v podprogramu již nemusíme mít v těle hlavního programu, jeho provedení v něm si vynutíme zapsáním **názvu** podprogramu jako příkazu

# Druhy podprogramů

- jazyk Pascal rozlišuje dva druhy podprogramů
  - **procedury**
  - **funkce**
- některé programovací jazyky používají jen jeden druh podprogramů (obvykle funkce), starší nižší jazyky podprogramy dokonce vůbec neměly
- oba druhy podprogramů mají mnoho společného, hlavní rozdíl je v tom, že **funkce má** vždy jeden **přímý výsledek** zatímco **procedura jej nemá**
- z toho vyplývá také odlišný způsob volání funkcí a procedur
- oba druhy podprogramů mohou dále poskytovat libovolný počet nepřímých výsledků

# Struktura podprogramů

- podprogram se chová i zapisuje jako program v programu
- má tedy svoji **hlavičku** a **tělo** stejně jako hlavní program
- v hlavičce jsou deklarace a definice stejných prostředků, jaké může mít běžný program (proměnné, konstanty, datové typy a dokonce i podprogramy) – tzv. **lokální**
- v těle podprogramu, které je i zde ohraničeno slůvky **begin** a **end**, jsou pak všechny příkazy podprogramu
- za **end** je v podprogramu ale středník (tečka se dává pouze na konec těla hlavního programu)
- protože je součástí kódu hlavního programu, lze v něm používat všechny prostředky, které hlavní program deklaruje pro svoji potřebu – tzv. **globální** (viz 26-15)

# Umístění podprogramů

- podprogramy se obvykle zapisují **na konci deklarační části** (hlavičky) programu (viz 25-05 - Základní kostra programu)
- podprogram může jako svůj lokální prostředek obsahovat i další **vnořený podprogram**
- tento vnořený podprogram se pak podle podobných pravidel umístí do hlavičky svého nadřazeného podprogramu (který se vůči němu chová podobně jako hlavní program ke svému podprogramu)
- počet úrovní vnoření podprogramů je omezen, ale v praxi se více než dvě úrovně vnoření vyskytují jen zřídka
- podprogram lze za určitých okolností z hlavního programu dokonce **vyjmout** a umístit **do samostatného souboru** (viz 26-19 – Programové jednotky)

# Příklad procedura vs. funkce

```
program pro;  
var a,b,c:byte;  
procedure soucet;  
begin  
    c := a+b;  
end;
```

```
begin  
    readln(a);  
    readln(b);  
    soucet;  
    writeln(c);  
end.
```

```
program fce;  
var a,b:byte;  
function soucet:word;  
begin  
    soucet := a+b;  
end;
```

```
begin  
    readln(a);  
    readln(b);  
    writeln(soucet);  
end.
```