



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: Základy programování a algoritmizace úloh**

**Téma: Vícenásobné větvení**

**Autor: Ing. Hodál Jaroslav, Ph.D.**

**Číslo: VY\_32\_INOVACE\_25-14**

**Anotace:** *Materiál popisuje použití příkazu pro vícenásobné větvení programu. Materiál je určen pro 3. a 4. ročník oboru strojírenství a technické lyceum. Vytvořeno: září 2012.*

# 14. Vícenásobné větvení

- v situaci, kdy potřebujeme program rozvětvit na větší množství větví, lze kromě struktury složené z mnoha do sebe vnořených podmíněných příkazů použít také příkaz vícenásobného větvení
- rozvětvuje program na téměř libovolný počet větví
- není založen na testování pravdivosti jednoho tvrzení jako podmíněný příkaz, ale obsahuje celý systém dotazů na rovnost dvou hodnot
- jeho syntaxe je poměrně dost volná, a proto obsahuje kvůli snazšímu vyhodnocování některé zvláštnosti
- v každé větvi lze provést vždy jen jeden příkaz, více příkazů se musí uzavřít do složeného příkazu

# Zápis příkazu vícenásobného větvení

```
case výraz of
```

```
    hodnota1: příkaz1;
```

```
    hodnota2: příkaz2;
```

```
    hodnota3a, hodnota3b: příkaz3;
```

```
    hodnota4a..hodnota4b: příkaz4;
```

```
    hodnota5a, hodnota5b..hodnota5c:  
        příkaz5;
```

```
    . . .
```

```
else
```

```
    příkaz;
```

```
end;
```

# Fungování příkazu

- **výraz**, který požadujeme za slůvkem `case`, musí mít výsledek **ordinálního typu**
- výsledek výrazu je postupně porovnáván s hodnotami, které jsme uvedli v jednotlivých větvích příkazu
- hodnoty musí být stejného typu, jako výsledek výrazu
- větví může být téměř libovolný počet
- pokud je v některé větvi nalezena hodnota rovná výsledku výrazu, provede se příkaz uvedený v příslušné větvi
- celý příkaz pak rovnou končí, nehledají se další shody
- **duplicita** hodnot v různých větvích **je nepřipustná**
- není-li nalezena shoda, provede se příkaz ve větvi `else`
- větev `else` lze vynechat

# Význam jednotlivých zápisů

- ve větvi `hodnota1: příkaz1;` se porovnává s výsledkem výrazu jen uvedená jedna hodnota
- příkaz ve větvi `hodnota3a, hodnota3b: příkaz3;` se provede tehdy, pokud výsledek výrazu bude roven jedné z uvedených dvou hodnot (hodnot může být i více; oddělují se od sebe čárkou)
- příkaz ve větvi `hodnota4a..hodnota4b: příkaz4;` se provede tehdy, pokud výsledek výrazu bude roven kterékoliv hodnotě z intervalu `<hodnota4a, hodnota4b>`
- předchozí dva zápisy lze i kombinovat `hodnota5a, hodnota5b..hodnota5c: příkaz5;`