



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: Podmíněný příkaz**

**Autor: Ing. Hodál Jaroslav, Ph.D.**

**Číslo: VY\_32\_INOVACE\_25-12**

**Anotace:** *Materiál popisuje k čemu slouží při vytváření programů podmíněný příkaz. Uvádí jeho dvě varianty a věnuje se i jeho složitějšímu použití k rozsáhlejšímu větvení programu. Materiál je určen pro 3. a 4. ročník oboru strojírenství a technické lyceum. Vytvořeno: srpen 2012.*

# 12. Podmíněný příkaz

- podmíněný příkaz patří mezi tzv. **příkazy větvení**
- příkazy větvení umožňují rozvětvit program na více větví (cest, kterými se program při vykonávání příkazů může vydat)
- příkazy větvení umožňují **interaktivitu** programů
- program se dle zadaných kritérií a aktuální situace rozhodne, kterou větví se vydá (jakou posloupnost příkazů provede)
- měli bychom tedy předem promyslet všechny situace, které použitím příkazu větvení vytváříme
- **podmíněný příkaz** je z příkazů větvení ten univerzálnější
- v různých podobách jej nalezneme ve všech programovacích jazycích

# Zápis podmíněného příkazu

- podmíněný příkaz rozvětjuje program na **2 větve**
- pro jeho zápis je potřeba složitější struktura s **pevnými** a **volitelnými** částmi

```
if podmínka then příkaz1 else příkaz2;
```

- volitelné části umožňují programátorovi zakomponovat do programu popis situace, na kterou reagujeme (podmínka) i reakce na tuto podmínku, mezi kterými se program rozhodne (příkaz1 a příkaz2)
- uvedenou strukturu lze chápat takto:  
jestliže je splněna zadaná podmínka, pak proved' příkaz1,  
jinak proved' příkaz2

# Podmínka

- podmínkou v podmíněném příkazu může být cokoliv, co má výsledek **logického typu** (pravda/nepravda)
- protože možné odpovědi jsou jen dvě, má i podmíněný příkaz vždy jen dvě větve; právě tato logická hodnota rozhodne, kterou ze dvou větví se program vydá
- podmínkou tedy může být:
  1. logická **proměnná**
  2. **výraz** s logickým výsledkem (relační, logický, ...)
  3. **příkaz** s logickým výsledkem (odd, ...)

# Příklady podmínek

```
writeln('Zadej číslo');  
readln(x);
```

```
p:=x mod 2=1;  
if p then writeln('Číslo je liché')  
else writeln('Číslo je sudé');
```

**nebo**

```
if x mod 2=1 then writeln('Číslo je liché')  
else writeln('Číslo je sudé');
```

**nebo**

```
if odd(x) then writeln('Číslo je liché')  
else writeln('Číslo je sudé');
```

# Reakce

- reakcí na splnění či nesplnění zadané podmínky je vždy **příkaz** (může být jednoduchý i strukturovaný); to znamená, že v podmíněném příkazu jsou vnořeny celé další příkazy
- jako reakci na podmínku můžeme použít v každé z větví **pouze jeden příkaz**
- pokud bychom chtěli reagovat na nějakou situaci **více příkazy** (a to bude často), musíme si pomoci složeným příkazem (viz 25-11), do nějž tyto příkazy „zabalíme“
- složený příkaz je jeden příkaz, to, že v něm samotném jsou obsaženy další příkazy, už nehraje roli
- pozor, před slůvkem `else` (za první reakcí) v podmíněném příkazu **nesmí být středník** i když je tu konec jiného příkazu

# Zjednodušený podmíněný příkaz

- podmíněný příkaz má ještě jednu variantu zápisu, která zjednodušuje řešení situace, kdy sice chceme zareagovat na splnění nějaké podmínky, ale **na její nesplnění reagovat nechceme**
- to neznamená, že by měl program jen jednu větev, stále se program rozvětňuje na **dvě větve**; v negativní větvi (`else`) je ale formálně jen prázdná reakce

**if** podmínka **then** příkaz;

- vynecháme zde celou větev `else`, příkaz končí středníkem
- proto nesmí být středník před `else` v normálním podmíněném příkazu; byl by považován za zjednodušený a `else` by překladač považoval za další (neexistující) příkaz

# Vnořování podmíněných příkazů

- reakcí na splnění i nesplnění podmínky v podmíněném příkazu může být jakýkoliv příkaz
- může to tedy být i **další podmíněný příkaz**
- pokud takto do jednoho podmíněného příkazu vnoříme jiný, dojde k **dalšímu větvení** zvolené větve
- vnořováním podmíněných příkazů lze tedy program větvit na téměř libovolný počet větví
- je však potřeba důsledně dodržet syntaxi jednotlivých příkazů a nemíchat je dohromady nesprávně
- počet úrovní vnoření je sice omezen, ale prakticky lze dosáhnout tohoto limitu jen obtížně
- mnohonásobné vnoření podmíněných příkazů lze obvykle s výhodou vyřešit příkazem **vícenásobného větvení** (25-14)