



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: Cyklus řízený proměnnou

Autor: Ing. Hodál Jaroslav, Ph.D.

Číslo: VY_32_INOVACE_25–15

Anotace: *Materiál popisuje obecnou problematiku příkazů cyklu se zaměřením především na cyklus řízený proměnnou. Materiál je určen pro 3. a 4. ročník oboru strojírenství a technické lyceum. Vytvořeno: září 2012.*

15. Cyklus řízený proměnnou

- cyklus je strukturovaný příkaz, který slouží k **opakování jiných příkazů**
- obecnou podmínkou pro použití cyklů je, že můžeme opakovat pouze příkaz, který má být vykonán několikrát **ihned po sobě** (bez přerušení jinými příkazy)
- můžeme opakovat i celou sekvenci příkazů, pokud použijeme složený příkaz (viz 25-11)
- v jazyce Pascal rozlišujeme 3 různé cykly
- cyklus řízený proměnnou představuje speciální případ
- lze ho použít jen tehdy, pokud **předem známe počet kroků** (opakování) cyklu

Zápis cyklu řízeného proměnnou

- pro zápis tohoto cyklu je potřeba složitější struktura s **pevnými** a **volitelnými** částmi

```
for řídicí_proměnná := dolní_mez to horní_mez do příkaz;
```

- **řídicí proměnná** musí být výhradně **ordinálního typu**
- **dolní** i **horní mez** hodnot musí být konstanty, proměnné nebo výrazy s výsledkem stejného typu jako má řídicí proměnná
- **příkaz** je příkaz, který chceme opakovat
- uvedený příkaz lze číst takto – pro hodnoty řídicí proměnné od dolní meze do horní meze dělej příkaz

Fungování cyklu řízeného proměnnou

1. do řídicí proměnné je nastavena hodnota dolní meze
2. ověří se, že hodnota řídicí proměnné není větší než horní mez, pokud ano, cyklus končí
3. provede se příkaz určený k opakování
4. hodnota řídicí proměnné se zvýší o 1 a pokračuje se znovu krokem 2.

(v případě nečíselné řídicí proměnné se její hodnota zvyšuje o 1 v pořadovém smyslu hodnot daného typu – je do ní vždy vložena pořadově následující hodnota)

Příklad cyklu řízeného proměnnou

```
for i := 1 to 10 do writeln('Ahoj');
```

- uvedený cyklus vypíše 10x text Ahoj na obrazovku
- jako meze cyklu mohou být použity i proměnné

```
readln(n);
```

```
for i := 1 to n do writeln('Ahoj');
```

- tento cyklus vypíše text tolikrát, kolikrát si určí uživatel
- hodnotu řídicí proměnné můžeme využít i v příkazu
- tím docílíme konání rozdílných činností v krocích cyklu

```
for i := 1 to 10 do writeln(i);
```

- cyklus vypíše čísla od 1 do 10
- v příkazu ale **nesmíme hodnotu řídicí proměnné měnit**

Druhá varianta cyklu

- cyklus řízený proměnnou má ještě jednu variantu, která je určena pro situace, kdy potřebujeme postupovat od větších hodnot řídicí proměnné k menším

```
for řídicí_proměnná := horní_mez downto dolní_mez do příkaz;
```

- slůvko `to` je zde nahrazeno slůvkem `downto`
- jsou také prohozeny meze
- při provádění tohoto cyklu bude hodnota řídicí proměnné snižována o 1 (u nečíselných hodnot o 1 v pořadovém smyslu – předchozí hodnota)