

Obsah

Obsah

1 Řídicí struktury	1
2 Podmínka	1
2.1 Podmínka typu case	2
3 Příkaz skoku	3
4 Cykly	4
4.1 Cyklus s podmínkou na začátku	4
4.2 Cyklus s podmínkou na konci	5
4.3 Příkazy měnící průběh cyklu	5
4.4 Cyklus s podmínkou uprostřed	6
4.5 Cyklus s iterační proměnnou	6

1 Řídicí struktury

Řídicí struktury

Řídicí struktury

- ovládají běh programu,
- umožňují přizpůsobení „výpočtu“ podle kritérií,
- zjednodušují psaní programu,
- šetří čas, paměť a energii počítače i programátora.

2 Podmínka

Podmínky

- Podmínka umožňuje různý průběh programem podle nějakého kritéria.
- Ve vývojovém diagramu jde o rozhodovací blok (kosočtverec).
- Obecný tvar je vždy `když ... potom ... [jinak ...]`.

Podmínka – obecný tvar

```
if logicky_vyraz then
    prikaz
else
    prikaz;
```

Je-li logický výraz za **if** pravdivý, provede se příkaz za **then**.

Není-li logický výraz za **if** pravdivý, provede se příkaz za **else** (existuje-li).

Před příkazem **else**, středník napsat nelze.

Podmínka – příklad 1

```
if diskriminant<0 then
    WriteLn('Kvadraticka rovnice ma komplexni koreny')
else
    WriteLn('Kvadraticka rovnice ma realne koreny');
```

Podmínka – příklad 2

```
if diskriminant<0 then
begin
    WriteLn('Kvadraticka rovnice ma komplexni koreny')
    WriteLn('Program bude ukoncen');
end
else
    WriteLn('Kvadraticka rovnice ma realne koreny');
```

2.1 Podmínka typu case

Složitější podmínky

- V mnoha případech nestačí podmínky typu ANO/NE.
- Zavádí se koncept „přepínače“.

Přepínač – obecný tvar

```
case vyraz of
    pripad1: prikaz;
    pripad2: prikaz;
else
    prikaz;
end;
```

Případ je:

- konstanta
- konstanta..konstanta
- kombinace obou oddělená čárkou

Podmínka – příklad 1

```
case cislo of
  1: WriteLn('Cislo je jedna');
  2: WriteLn('Cislo je dva');
  3: WriteLn('Cislo je tri');
  4: WriteLn('Cislo je ctyri');
end;
```

Podmínka – příklad 2

```
case cislo of
  1: WriteLn('Cislo je jedna');
  2: WriteLn('Cislo je dva');
  3: WriteLn('Cislo je tri');
  4: WriteLn('Cislo je ctyri');
  else
    WriteLn('Tohle cislo neznam');
end;
```

Podmínka – příklad 3

```
case cislo of
  1..10: WriteLn('Cislo je male');
  11..50: WriteLn('Cislo je stredne velke');
  0,51..1000: WriteLn('Cislo je velke nebo nula');
end;
```

3 Příkaz skoku

Příkaz skoku

- *V moderním programování se nepoužívá.*
- *Nechci ho vidět v žádném z Vašich projektů.*

- Dvě nová klíčová slova *label* a *goto*.
- Příkazem *label* se deklarují návěští.
- Příkazem *goto* se provede skok na dané návěští.

Příkaz skoku – příklad

```
label konec;
begin
  ...
  ...
  goto konec;
  ...
  ...
  konec: WriteLn('Konec programu...');
end.
```

4 Cykly

Cykly

- Místo použití skoků se používají jiné konstrukce:
 1. Podmínky a přepínače
 2. Cykly
 - Určitá činnost se vykonává dokud je splněna nějaká podmínka,
 - nebo pro určitý interval celých čísel n
 3. Volání procedur a funkcí (bude uvedeno v některé z dalších přednášek)

4.1 Cyklus s podmínkou na začátku

Cyklus s podmínkou na začátku

- *Dokud je splněna podmínka* dělej něco.
- Postup práce:
 1. Vyhodnot' podmínku.
 2. Je-li podmínka splněna, proved' příkaz v těle cyklu.
 3. Není-li podmínka splněna, přejdi na příkazy za koncem cyklu.
 4. Po provedení těla cyklu se vrať na bod 1.

```
while podmínka do
  příkaz;
```

Cyklus s podmínkou na začátku – příklad

```
while (je_dalsi_cislo) do
begin
    read(i);
    zpracuj_cislo(i);
end;
```

4.2 Cyklus s podmínkou na konci

Cyklus s podmínkou na konci

- Dělej něco *dokud není splněna podmínka*.
- Postup práce:
 1. Proved' příkazy těla cyklu.
 2. Vyhodnoť podmínku.
 3. Není-li podmínka splněna, jdi na bod 1.
 4. Je-li podmínka splněna, proved' příkazy za koncem cyklu.

```
repeat
    prikaz;
    prikaz;
    ...
until podminka;
```

Cyklus s podmínkou na konci – příklad

```
repeat
    zapis_na_disk(neco);
until disk_plny;
```

4.3 Příkazy měnící průběh cyklu

Příkazy měnící průběh cyklu

- *Continue*
 - Okamžitě přejde na nové vyhodnocení podmínky a další iteraci cyklu.
- *Break*
 - Okamžitě ukončí cyklus a přejde na další příkaz za cyklem.

4.4 Cyklus s podmínkou uprostřed

Cyklus s podmínkou uprostřed

- S pomocí příkazů `if` a `break` lze realizovat cyklus s podmínkou uprostřed.
- Vytvoří se „nekonečná smyčka“ a uprostřed se provede podmínka s příkazem `break`.

```
while true do
begin
  ...
  if podminka then
    break;
  ...
end;
```

4.5 Cyklus s iterační proměnnou

Cyklus s iterační proměnnou

- Pro n z 1 až 100 dělej něco.
- Pascal umí pouze pro celá čísla.
- Postup práce:
 1. Na začátku dej do proměnné počáteční hodnotu.
 2. Je-li proměnná menší (větší) nebo rovna konečné hodnotě, pokračuj bodem 4.
 3. Není-li proměnná menší (větší) nebo rovna konečné hodnotě, přejdi na příkazy za cyklem.
 4. Proveď tělo cyklu.
 5. Přičti (odečti) k proměnné jedničku a jdi na bod 2.

```
for n:=1 to 100 do
  příkaz;
```

Cyklus s proměnnou – příklad

```
faktorial:=1;
for n:=1 to maximum do
begin
  faktorial:=faktorial*n;
  WriteLn(n,'! = ',faktorial)
end;
```