Príkazy cyklu

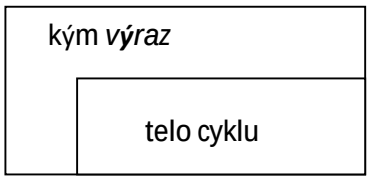
Príkaz cyklu je zložený riadiaci príkaz. Používame ho vtedy, keď potrebujeme, aby sa určitý príkaz alebo príkazy vykonal/vykonali viackrát. Príkaz/príkazy, ktoré sa v cykle opakujú nazývame telo cyklu.

Príkazy cyklu sú

a) podmienený cyklus – používame vtedy, ak nevieme dopredu koľkokrát treba vykonať telo cyklu
   - Ø s podmienkou ukončenia na začiatku – while
   - Ø s podmienkou ukončenia na konci – repeat-until

b) nepodmienený cyklus – používame vtedy, ak vieme koľkokrát máme telo cyklu vykonať – for

Podmienený cyklus s podmienkou ukončenia na začiatku

Značka v štrukturogramoch:	Zápis v Pascale:
	while *výraz* do *telo_cyklu*;

Fungovanie: Najprv sa vyhodnotí pravdivosť logického výrazu. Ak je tento výraz pravdivý, vykoná sa telo cyklu (jeden príkaz alebo blok príkazov), a opäť sa testuje jeho pravdivosť. Ak je opäť výsledok pravdivý, pokračuje sa v cykle, ak nie je, pokračuje sa za telom cyklu. Minimálny počet opakovaní cyklu je 0.

Príklad 1:

Zostavte program, ktorý načíta prirodzené číslo a vypíše jeho faktoriál.

Riešenie:

Skôr než sa pustíme do zápisu zdrojového textu programu tejto úlohy, musíme najprv porozmýšľať ako ju vlastne riešiť. Na hodinách matematiky nás učili, že napr. 5! = 5.4.3.2.1, čo počítame takto: 5.4=20, 20.3=60, 60.2=120 a napokon 120.1=120, takže 5!=120. Ako sme to vlastne riešili? Najprv sme vynásobili 5 so 4 a potom sme 3-krát násobili výsledok postupne číslami 3, 2, 1. Stačilo by nám násobiť výsledok len 2-krát, číslami 3 a 2? Áno, výsledok sa vynásobením číslom 1 nezmení. Skúsme to, na čo sme prišli zovšeobecniť. Ak máme vypočítať n! (n je premenná), tak najprv môžeme do výsledku V uložiť hodnotu n.(n-1), potom zmenšiť hodnotu n o 2. Kým bude hodnota

v premennej n väčšia ako 1 budeme opakovať tieto činnosti (telo cyklu): do premennej V uložíme pôvodnú hodnotu premennej V vynásobenú o hodnotu premennej n; zmenšíme hodnotu premnnej n o 1.

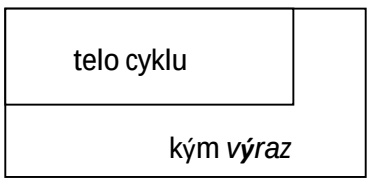
*Zhrňme si dva najpodstatnejšie fakty: čo máme opakovať (v:=v*n; dec(n);) a dokedy to máme opakovať (kým je n>1). Zdrojový text programu podľa popísaného postupu vyzerá takto:*

```
uses Crt;
var v,n:longint;
begin
  clrscr;
  write('Zadaj prirodzene cislo: ');
  readln(n);
  writeln;
  write(n,'! = ');

  v:=n*(n-1);
  n:=n-2;
  while n>1 do
    begin
    v:=v*n;
    dec(n);
    end;
  writeln(v);
  readln;
end.
```

Úlohy:

1. Zostavte program, ktorý bude načítavať reálne čísla dovtedy, kým nezadáme nulu a vypíše ich súčet.
2. Zostavte program, ktorý načíta čas v sekundách a vypíše, koľko je to minút. Delenie realizujte operáciou odčítania.
3. Zostavte program, ktorý bude načítavať celé čísla dovtedy, kým nezadáme nulu a vypíše počet párnych čísel.

Podmienený cyklus s podmienkou ukončenia na konci

Značka v štrukturogramoch:	Zápis v Pascale:
	repeat *telo_cyklu* until *výraz*;

Fungovanie: Najprv sa vykoná telo cyklu (jeden príkaz alebo blok príkazov) a potom sa vyhodnotí pravdivosť logického výrazu. Ak je tento výraz nepravdivý, pokračuje sa v cykle, ak je pravdivý, pokračuje sa za telom cyklu. Minimálny počet opakovaní cyklu je 1.

Príklad 2:

Zostavte program, ktorý načíta prirodzené číslo a vypíše jeho faktoriál.

Riešenie:

Máme riešiť rovnakú úlohu ako v prvom príklade? Prečo? Nuž preto, že vždy môžeme nahradiť cyklus while cyklom repeat-until a naopak, cyklus repeat-until nahradiť cyklom while. Záleží len od nás, čo nám viacej vyhovuje a ktorý z týchto cyklov nám umožní kratší zápis riešenia úlohy. Musíme si však uvedomiť zásadné rozdiely medzi nimi: Cyklus while sa ukončí, ak podmienka cyklu nie je pravdivá (kým platí, POKRAČUJEME) a viacej príkazov tela cyklu musíme uzavrieť do bloku (begin-end). Cyklus repeat-until sa ukončí ak je podmienka cyklu pravdivá (keď platí, KONČÍME)a viacej príkazov tela cyklu nie je potrebné uzatvárať do bloku, lebo kľúčové slová cyklu (repeat, until) uzatvárajú telo cyklu.

*Opäť si zhrňme dva najpodstatnejšie fakty: čo máme opakovať (v:=v*n; dec(n);) a dokedy to máme opakovať (kým je n>1, ale pre cyklus repeat-until zapisujeme podmienku opačne, teda keď platí, končíme cyklus, takže podmienka ukončenia cyklu bude n=1). Zdrojový text programu podľa popísaného postupu vyzerá takto:*

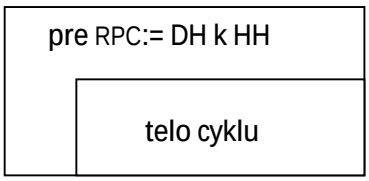
```
uses Crt;
var v,n:longint;
begin
  clrscr;
  write('Zadaj prirodzene cislo: ');
  readln(n);
  writeln;
  write(n,'! = ');

  v:=n*(n-1);
  n:=n-2;
  repeat
    v:=v*n;
    dec(n);
  until n=1;
  writeln(v);
  repeat until keypressed;
end.
```

Prvýkrát sa v tomto programe stretávame so zaujímavým riadkom repeat until keypressed; Čo robí? Je to cyklus, o ktorom hovoríme. Telo cyklu je prázdne, teda v cykle sa nerobí nič. Podmienkou ukončenia cyklu je akási čudná podmienka keypressed. Key je klávesa, pressed je stlačiť. Keypressed je funkcia, ktorá vráti hodnotu true, ak bola na klávesnici stlačená ľubovoľná klávesa. Hodnotu false vráti vtedy, ak klávesa stlačená nebola. Z toho všetkého si vytvárame takýto záver. Program čaká na stlačenie ľubovoľnej klávesy. Po jej stlačení pokračuje ďalej. Zatiaľ budeme tento riadok používať na pozastavenie činnosti programu do okamihu stlačenia ľubovoľnej klávesy.

Úlohy:

1. Zostavte program, ktorý načíta celé číslo a vypíše počet jeho cifier.
2. Zostavte program, ktorý bude načítavať reálne čísla dovtedy, kým nezadáme nulu a vypíše počet záporných čísel.
3. Zostavte program, ktorý bude načítavať celé čísla dovtedy, kým nezadáme nulu a vypíše najväčšie z nich.

Cyklus s pevným počtom opakovaní

Značka v štrukturogramoch:	Zápis v Pascale:
	for *RPC*:= DH to HH do *telo_cyklu*;

Fungovanie: RPC je riadiaca premenná cyklu, je vždy typu integer. DH je dolná hranica, celé číslo. HH je horná hranica, celé číslo. Telo cyklu je jeden príkaz alebo blok príkazov. Na začiatku sa do RPC priradí hodnota DH a testuje sa podmienka cyklu, teda či RPC<=HH. Ak je tento výraz pravdivý, vykoná sa telo cyklu, zväčší sa hodnota RPC o 1 a opäť sa testuje podmienka cyklu. Ak je výsledok opäť pravdivý, pokračuje sa v cykle, ak nie je, pokračuje sa za telom cyklu. Minimálny počet opakovaní cyklu je 0.

Tento cyklus môže mať aj takýto tvar:

```
for RPC:= HH downto DH do        {downto – dole k}
    telo_cyklu;
```

Fungovanie: Na začiatku sa do RPC priradí hodnota HH a testuje sa podmienka cyklu, teda či RPC>=DH. Ak je tento výraz pravdivý, vykoná sa telo cyklu, hodnota RPC sa

zmenší o 1 a opäť sa testuje podmienka cyklu. Ak je výsledok opäť pravdivý, pokračuje sa v cykle, ak nie je, pokračuje sa za telom cyklu. Minimálny počet opakovaní cyklu je 0.

Príklad 3:

Zostavte program, ktorý načíta dve celé čísla a vynásobí ich. Ale pozor! Násobenie realizujte operáciou sčítania.

Riešenie:

Skôr, než sa pustíme do programovania, musíme sa vrátiť do čias, keď sme boli prvákmi na základnej škole. Tam nás učili, že keď máme spočítať počet dielikov mliečnej čokolády, tak najprv musíme spočítať počet dielikov v jednom rade, to nech je napr. 7, spočítať počet radov, to nech je 4. Počet dielikov vypočítame takto: 7+7+7+7=28. Čiže spočítali sme spolu 4 sedmičky. V hlavách sme počítali takto: 0+7=7, 7+7=14, 14 + 7=21, 21+7=28. Vlastne sme 4-krát urobili to isté – k výsledku sme pripočítali 7. Na začiatku bol výsledok rovný 0.

*Už asi tušíme, že na riešenie tejto úlohy by bol cyklus ako stvorený. Lenže ktorý? Uložme počet dielikov čokolády v jednom rade do premennej a, počet radov do premennej b. Keďže dopredu vieme, koľkokrát máme pripočítať číslo a k výsledku (b-krát), tak použijeme cyklus s pevným počtom opakovaní - for. Opäť si musíme uvedomiť dve najdôležitejšie veci: **čo máme opakovať** (v:=v+a;) a dokedy to máme opakovať (keďže použijeme cyklus for, musíme si deklarovať celočíselnú premennú (napr. i), ktorá bude počítadlom cyklu. Počítadlo začne počítať od hodnoty 1 a skončí hodnotou b). Zdrojový text programu podľa popísaného postupu vyzerá takto:*

```
uses Crt;
var a,b,v,i:integer;
begin
  clrscr;
  write('Zadaj 1. cinitela: ');
  readln(a);
  write('Zadaj 2. cinitela: ');
  readln(b);
  writeln;

  v:=0;
  for i:=1 to b do v:=v+a;
  writeln('Sucin zadanych cisel je ',v);
  repeat until keypressed;
end.
```

Počítadlo (premennú i) by sme mohli meniť aj od hodnoty b do hodnoty 1, lenže v tomto prípade by sme museli po každom vykonaní tela cyklu znížiť hodnotu počítadla o 1. Riešenie za týchto podmienok by vyzeralo takto:

```
uses Crt;
var a,b,v,i:integer;
begin
  clrscr;
  write('Zadaj 1. cinitela: ');
  readln(a);
  write('Zadaj 2. cinitela: ');
  readln(b);
  writeln;

  v:=0;
  for i:=b downto 1 do v:=v+a;
  writeln(‘Sucin zadanych cisel je ‘,v);
  repeat until keypressed;
end.
```