



Návestie

Pokiaľ sú v procedúre použité skokové inštrukcie `go to`, je nutné ešte pred kľúčové slovo `begin` uviesť kľúčové slovo `label` a zoznam návestí oddelených čiarkami. V programe sa cieľ skoku označí uvedením návestia, za ktorým nasleduje dvojbodka, napríklad takto:

```
procedure Nič();  
label  
    Skok1, Skok2;  
  
begin  
    (* príkazy *)  
    Skok1:  
    (* príkazy *)  
    Skok2:  
    (* príkazy *)  
end_procedure;
```

Príkaz if

Príkaz `if` slúži k podmienenému vykonávaniu určitých častí programu. Za kľúčovým slovom musí byť uvedený logický výraz a kľúčové slovo `then`. Nasledujúci blok príkazov je vykonaný len v tom prípade, pokiaľ logický výraz bude vyhodnotený ako `true`.

Blok príkazov môže byť ukončený tromi spôsobmi:

- kľúčovým slovom `end`
- kľúčovým slovom `else`. Za `else` nasleduje blok príkazov, ktorý bude vykonaný len v prípade nesplnení logickej podmienky. Blok musí byť ukončený kľúčovým slovom `end`.
- kľúčovým slovom `elsif`. Za týmto slovom nasleduje podmienka a kľúčové slovo `then`

Príklad:

```
if Číslo = 0 then  
    Pamäť = 0
```



```
elseif Číslo = 1 then
    Pamäť = 1
elseif Číslo = 2 then
    Pamäť = 2
else Pamäť = 9
end;
```

Príkaz loop

Príkaz **loop** patrí do skupiny príkazov cyklu. Presne povedané, je to nekonečný cyklus. Pokiaľ nemá byť cyklus nekonečný, musí vo vnútri cyklu existovať algoritmus s kľúčovým slovom **exit**.

Pokiaľ máme v sebe vnorených viacej cyklov **loop**, príkaz **exit** sa vzťahuje vždy na najvnútornejší cyklus. Vo vnútri cyklu je možné použiť kľúčové slovo **continue**, ktoré vráti riadenie na začiatok cyklu.

Príklad_1:

```
loop
    Číslo = Číslo + 1; (* nekonečný cyklus. Pozor na tento
    cyklus, NIKDY NESKONČÍ! *)
end;
```

Príklad_2:

```
loop
    Číslo = Číslo + 1;
    if Číslo = 255 then
        exit; (* cyklus končí po dosiahnutí 255 *)
    end
end;
```

Príklad_3:

```
loop
    Číslo = Číslo + 10;
    Číslo = Číslo - Pamäť;
    if Číslo < 0 then
        continue; (* zvyšok príkazov do konca slučky
preskoč
```



```
end;          (* a vráť sa na prvý príkaz v loop *)  
if Číslo = 255 then  
    exit;  
end;  
end;
```

Príkaz while

Príkaz **while** patrí do skupiny príkazov cyklu. Je to cyklus s podmienkou pri vstupe do slučky. Za kľúčovým slovom **while** nasleduje logický výraz a kľúčové slovo **do**. Pokiaľ je logický výraz vyhodnotený ako **true**, vykoná sa nasledujúci blok príkazov až po kľúčové slovo **end**. Pokiaľ sa v bloku príkazov nevyskytujú príkazy **pause** alebo **wait**, neprebíha v ďalších prechodoch slučkou komunikácia s vonkajším zariadením.

Pokiaľ nie je logický výraz pravdivý pred vstupom do cyklu, cyklus sa neprevedie.

Príklad:

```
Číslo = 1;  
  
while Číslo < Pamäť do  
    Číslo = Číslo + 1;  
end;
```

Príkaz repeat until

Príkaz **repeat until** patrí do skupiny príkazov cyklu. Je to cyklus s podmienkou na konci slučky. Za kľúčovým slovom **repeat** nasleduje blok príkazov, ukončený kľúčovým slovom **until** s logickým výrazom. Ak je logický výraz vyhodnotený ako pravdivý, cyklus končí.

Je teda zrejmé, že blok príkazov sa prevedie vždy minimálne jedenkrát.

Príklad:

```
Číslo = 1;
```



```
repeat  
    Číslo = Číslo + 1;  
until Číslo >= Pamäť
```

Príkaz for

Príkaz **for** patrí do skupiny príkazov cyklu. Je to cyklus s dopredu známym počtom cyklov. Za kľúčovým slovom **for** nasleduje zápis číselnej premennej, znak = (rovná sa) a číselný výraz. Potom nasleduje kľúčové slovo **to** a ďalší číselný výraz nasledovaný kľúčovým slovom **do** a blok príkazov ukončený kľúčovým slovom **end**.

Príklad:

```
for Číslo = 1 to 10 do  
    Číslo = Číslo + 1;  
end;
```

Nie vždy je požadované, aby sa číselná premenná zvyšovala v každom cykle o jednotku. Niekedy je nutné krok zväčšiť. Preto je možné cyklus **for** doplniť o krok. Krok sa zapisuje kľúčovým slovom **by** pred kľúčovým slovom **do**.

Príklad:

```
for Číslo = 1 to 10 by 2 do  
    Číslo = Číslo + 1;  
end;
```

Príklad:

```
for Číslo = 11 to 1 by -2 do  
    Číslo = Číslo + 1;  
end;
```

Príkaz pause

Príkaz **pause** pozastavuje beh programu na určitú dobu. Za kľúčovým slovom nasleduje numerický výraz udávajúci počet sekúnd zadržanie programu.



Príklad:

```
while Číslo < 20.0 do  
    pause 5; (* pozastavenie činnosti programu na 5 sekúnd  
    *)  
end;
```

Príkaz wait

Príkaz **wait** pozastaví činnosť programu do splnenia podmienky za týmto kľúčovým slovom. Ak je hodnota logickej podmienky true, program pokračuje nasledujúcim príkazom v tom istom časovom kroku. Ak je výsledok logickej podmienky false, program poskytne systému čas k prípadnej obsluhu ďalších prístrojov a opäť testuje podmienku.

Príklad:

```
wait Číslo >= 20.0; (* čakanie na splnenie podmienky  
*)
```

Príkaz stop

Príkaz **stop** spôsobí ukončenie procedúry. Ďalšie volanie procedúry spôsobí jej rozbehnutie znovu od začiatku.

Príkaz send

Pomocou kľúčového príkazu **send** voláme iné prístroje v aplikácii. Príkaz aktivuje uvedený prístroj ihneď po skončení práve aktuálneho časového kroku. Pokiaľ chcete, aby prístroj aktivoval sám seba, môžete vypísať jeho meno, alebo použiť kľúčové slovo **self**. Príkaz **send self** znamená: „aktivuj sám seba v najbližšom systémovom časovom kroku“ (používa sa pre komunikáciu s V/V kanálmi).

Príklad:

```
send Vypínač; (* volanie prístroja Vypínač *)  
send Žiarovka; (* volanie prístroja Žiarovka*)  
send self;    (* prístroj volá sám seba    *)
```



Príkaz priradenia

dátový_element = výraz

Dátovým elementom môže byť premenná, prvok alebo pole premenných, výstupný alebo obojsmerný kanál alebo pole výstupných alebo obojsmerných kanálov.

Príklad:

```
x = 1;  
x = x + 1;  
Text = 'Čas';
```

Volanie procedúr

Zápis programu dovoľuje odkazovať v algoritme na iné procedúry, ktoré týmto spúšťame.

Zápis sa prevádza v tvare: meno prístroja, bodka, názov procedúry.

Napríklad:

```
Časové_relé.Proc_1();
```

Ak chceme volať procedúry, ktoré sú v inom module, ako prvý sa uvádza názov modulu.

Napríklad:

```
ModulČasovačov.Časové_relé.Proc_1();
```

V niektorých prípadoch sa v procedúre odvolávame na inú procedúru toho istého prístroja. V takom prípade nemusí byť pre názvom procedúry meno prístroja, ale toto meno môže zastupovať kľúčové slovo self.

Príklad:

```
self.Proc_1();
```



Komentáre

Komentár je sprievodný a doplňujúci text vytvoreného programu. Každý skúsený programátor nešetrí na komentároch. Veď kto si bude po nejakom čase pamätať, prečo práve tú procedúru (alebo premennú...) použil!

Komentár je od zápisu algoritmu oddelený komentárovými zátvorkami (* pre začiatok a *). Všetko, čo bude zapísané vo vnútri, je chápané ako komentár.

Príklad:

```
procedure OnActivate();
begin
  if Vypínač then          (* ak bude stlačený vypínač *)
    Svetlo = not Svetlo;   (* invertuj stav premennej *)
    send Žiarovka;         (* aktivuj aplikáciu Žiarovka *)
  else Svetlo = false; (* pri vypnutí vypínača musí zhasnúť *)
    send Žiarovka;         (* aktivuj aplikáciu Žiarovka *)
end;
end_procedure;
```