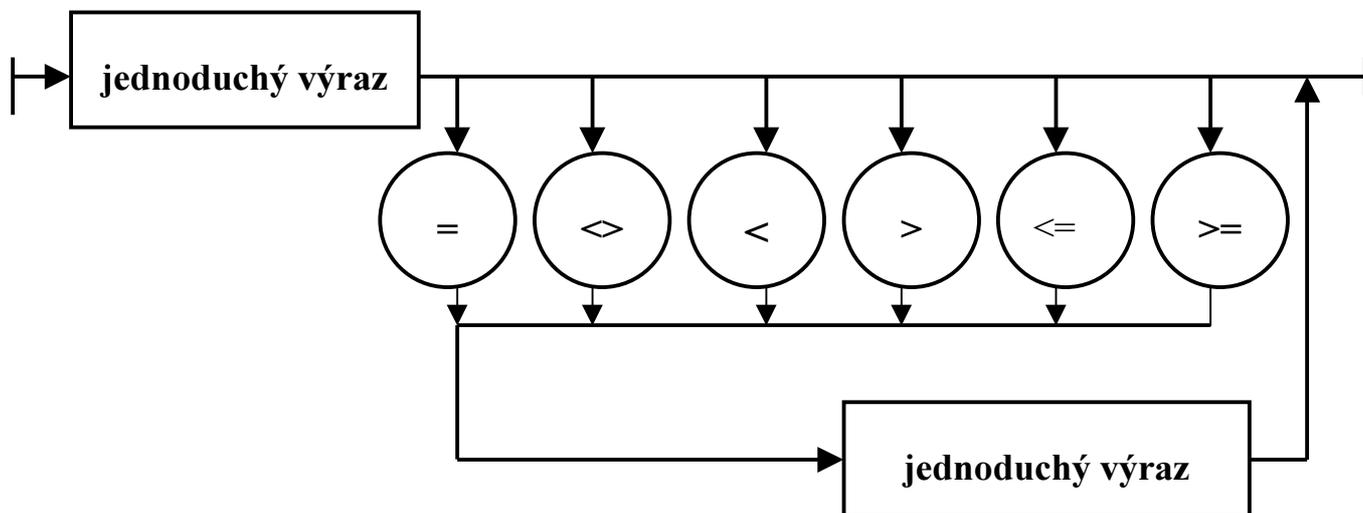
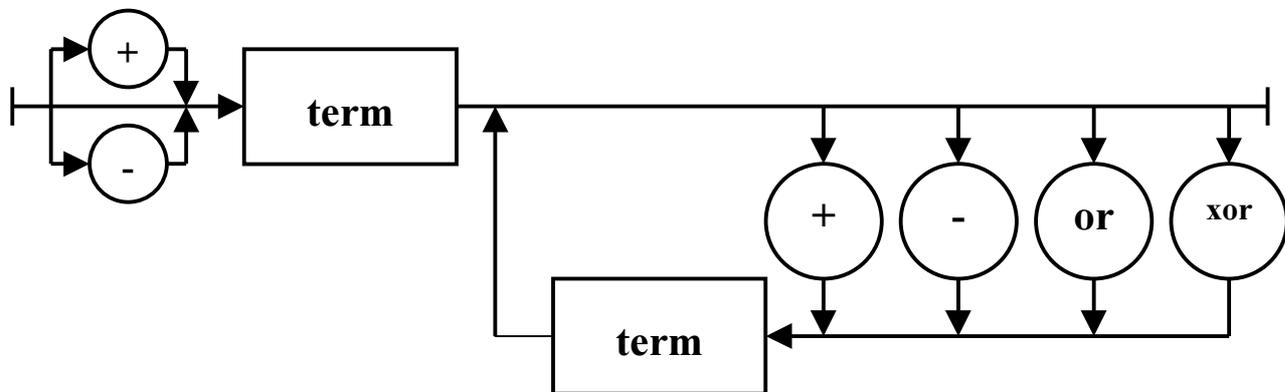


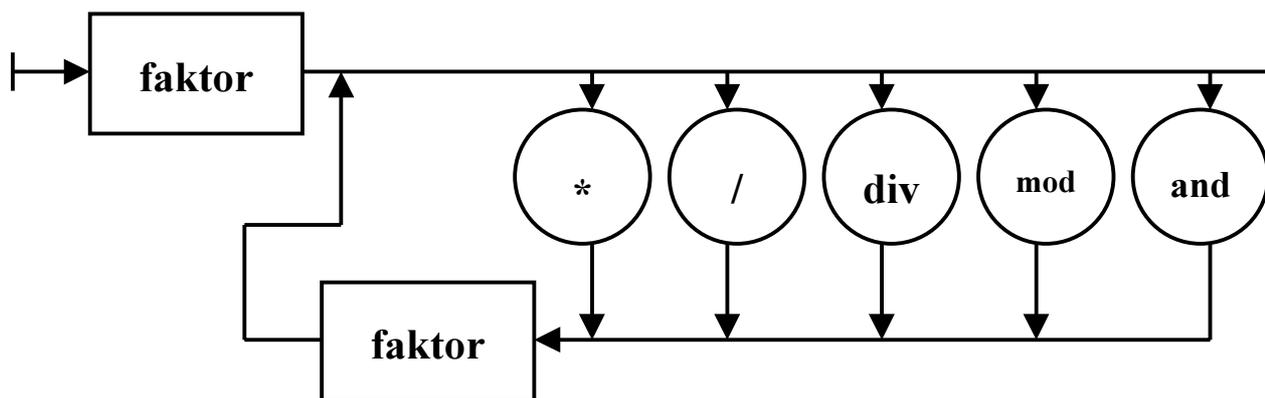
výraz



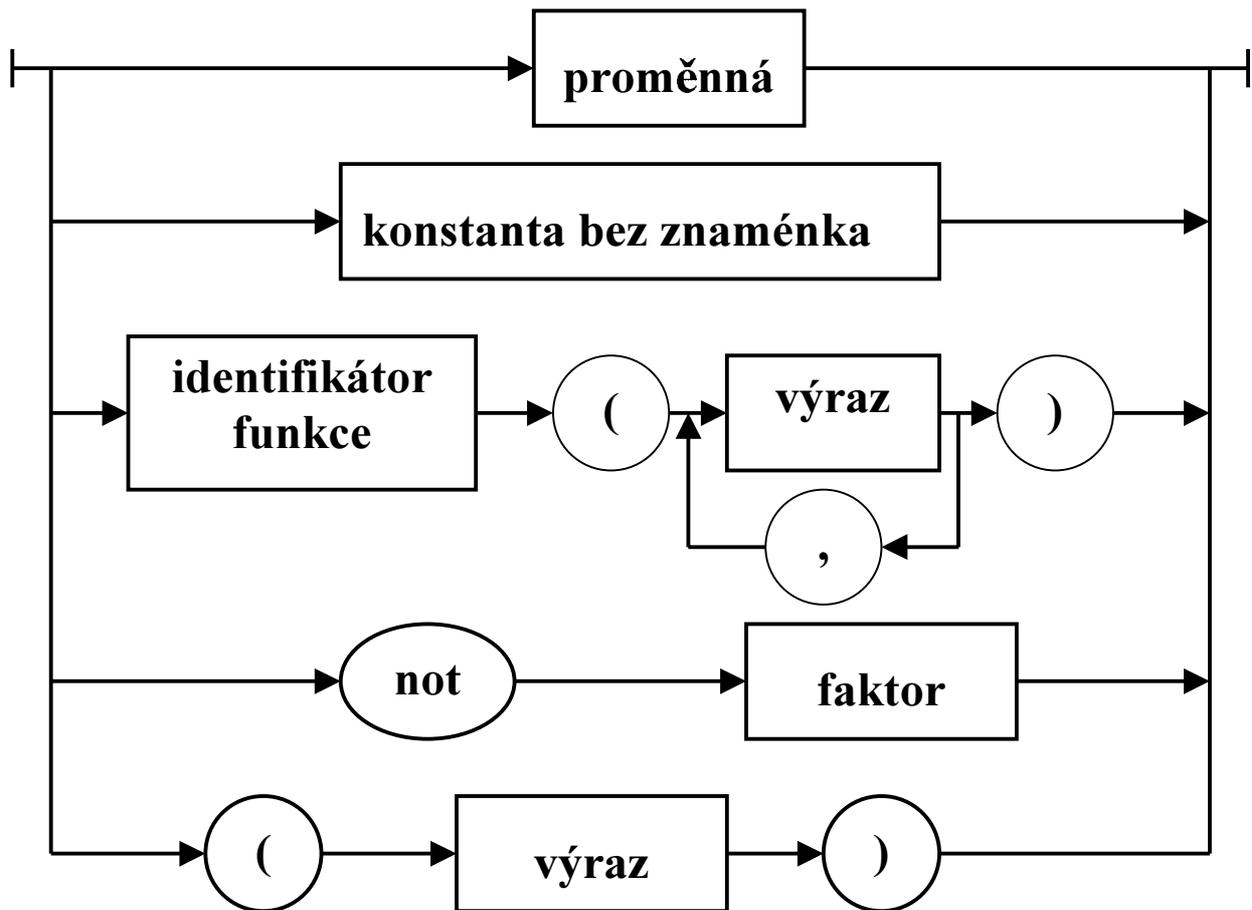
jednoduchý výraz



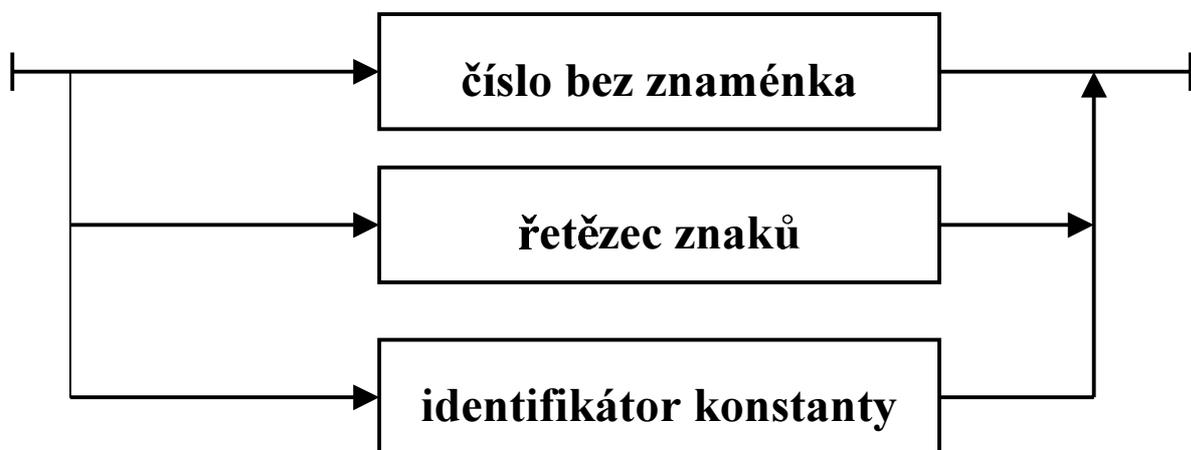
term



faktor



konstanta bez znaménka



Priorita operátorů

not

nejvyšší priorita

*** / div mod and**

+ - or xor

= <> < > <= >=

nejnižší priorita

Výraz je správný, pokud operátory, které se v něm vyskytují mají operandy přípustného typu

| operátor | Typy operandů | Typ výsledku |
|--|--|--|
| + - * | Integer, real | integer pro operandy typu integer, jinak real |
| / | integer, real | real |
| div mod | integer | integer |
| = <> <> <= >= | Oba stejného typu, nebo jeden integer, druhý real | Boolean |

| funkce | Typ argumentu | Typ výsledku |
|---------------------------------------|-------------------------|-------------------------|
| abs sqr | integer real | integer real |
| sin cos arctan ln exp sqrt | integer real | real real |
| odd | integer | Boolean |
| trunc round | real | integer |

trunc (truncation=odseknutí) - výsledkem je celé číslo vzniklé odseknutím desetinné části

$$\mathbf{trunc(7.3) = 7}$$

$$\mathbf{trunc(12.9) = 12}$$

$$\mathbf{trunc(-7.3) = -7}$$

$$\mathbf{trunc(-12.9) = -12}$$

round (roundation = zaokrouhlení)

$$\mathbf{round(x) = \begin{cases} trunc(x+0.5) & \text{pro } x \geq 0 \\ trunc(x-0.5) & \text{pro } x < 0 \end{cases}}$$

$$\mathbf{round(7.3) = 7}$$

$$\mathbf{round(12.9) = 13}$$

$$\mathbf{round(-7.3) = -7}$$

$$\mathbf{round(-12.9) = -12}$$

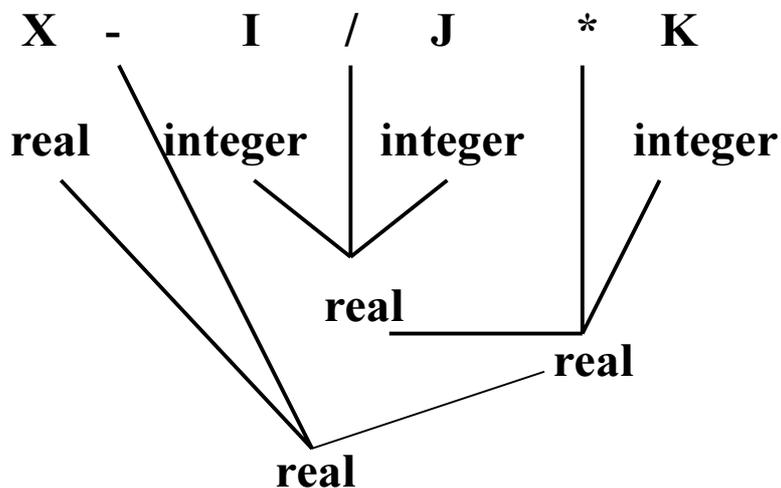
Vyhodnocování výrazů

- zleva doprava s ohledem na prioritu operátorů a na závorky

Typ výrazu - lze určit na základě vyhodnocení výrazu

Př.

var X: real; I,J,K: integer ; P,Q: boolean

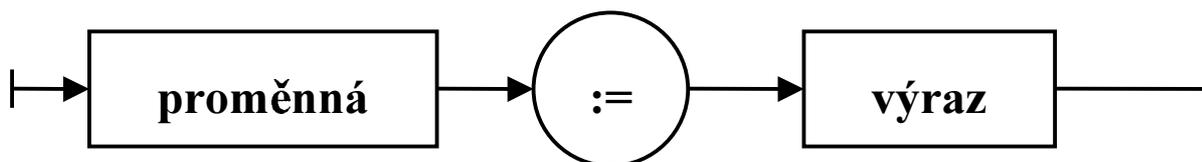


| | |
|----------------------|---------|
| I + trunc(X) div 2 | integer |
| 5 * sin(2*X) | real |
| P or not Q and (A>0) | boolean |
| not odd(round(X)) | boolean |

Příkazy



Přiřazovací příkaz



$P := V$

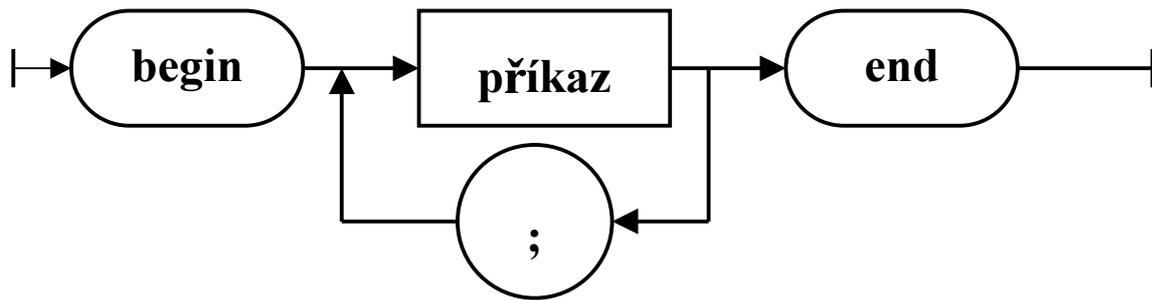
P je proměnná typu **T**

V je výraz s hodnotou **H**

H musí být kompatibilní vzhledem k přiřazení s typem **T**

- hodnota **H** je typu **T**
- hodnota **H** je typu integer a **T** je typu real

složený příkaz

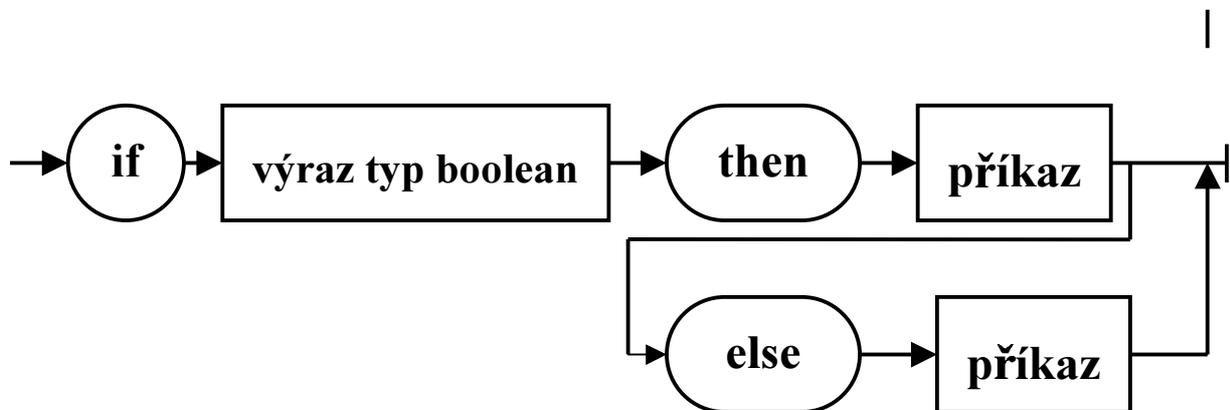


Používá se tehdy, pokud potřebujeme zapsat posloupnost příkazů jako jediný příkaz.

Př.

`begin POM := X; X:=Y; Y:=POM; end`

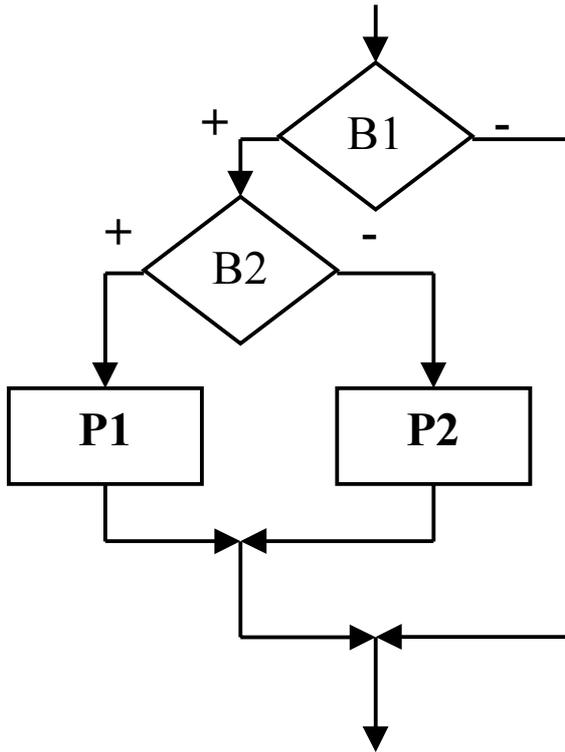
Podmíněný příkaz



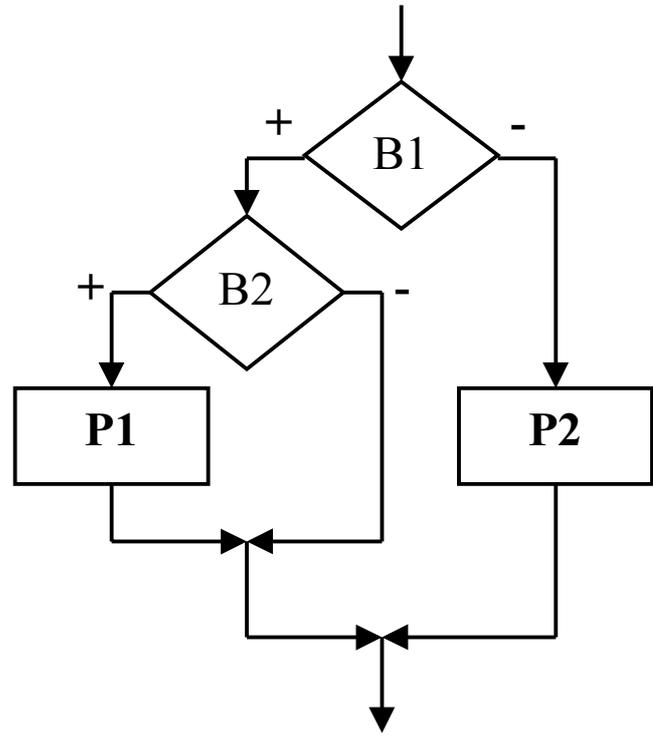
`ifthen else` - úplný podmínovací příkaz

`if ... then` - neúplný podmínovací příkaz

Podmínovací příkazy je možné vnořovat do sebe. Platí zásada, že *else* se vztahuje vždy k nejbližšímu *then*.



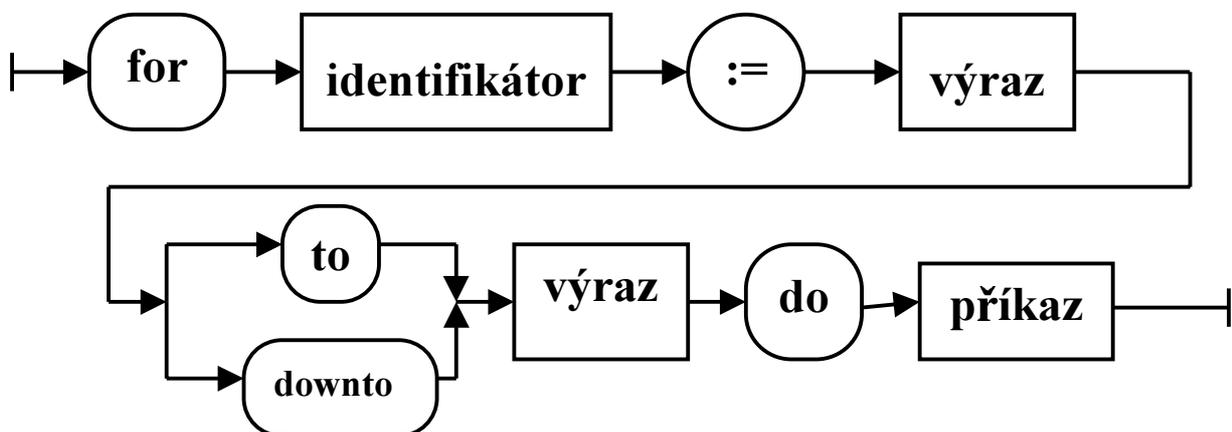
**if B1 then
if B2 then P1 else P2**

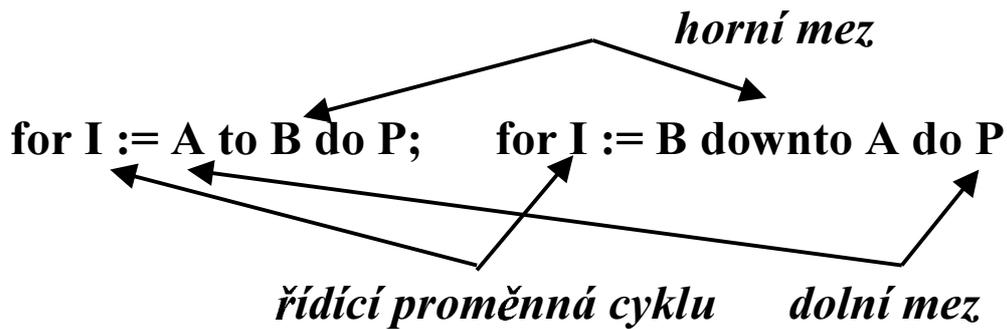


**if B1 then
begin if B2 then P1 end
else P2**

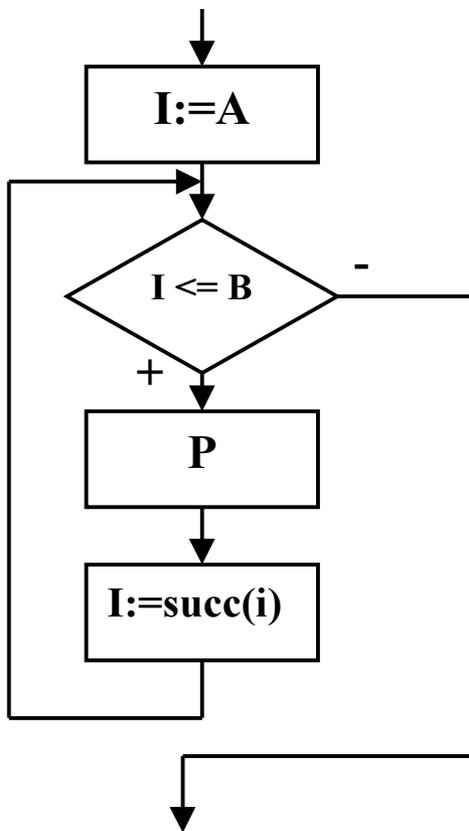
Příkazy cyklu

- cyklus s pevným počtem opakování - je vždy konečný (tj. buď se neprovede vůbec nebo se po konečném počtu opakování ukončí)

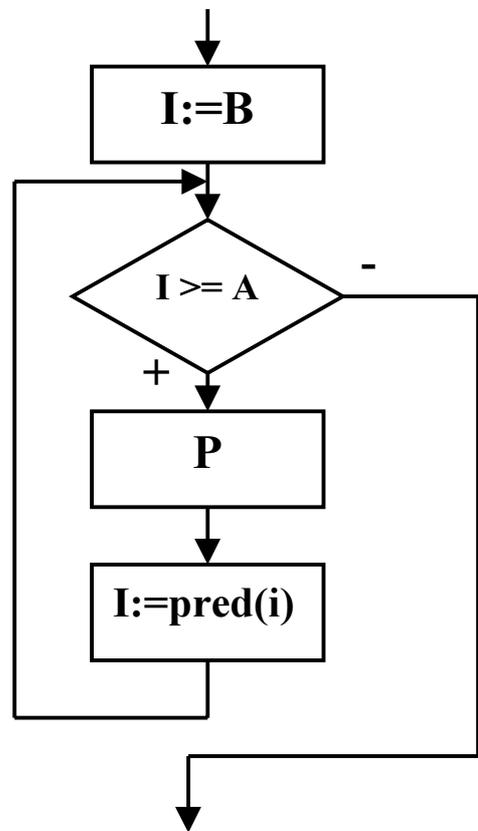




- *řídící proměnná cyklu* – ordinální typ, v těle cyklu by se hodnota řídící proměnné nesmí změnit.
Podle normy není po ukončení cyklu zajištěno, hodnota $i=B$ (pro *to*) nebo $i=A$ (pro *downto*).
- pro $A > B$ (cyklus s *to*), popř. $B < A$ (cyklus s *downto*) se cyklus neprovede ani jednou

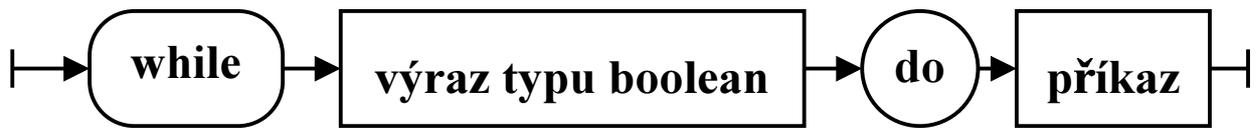


cyklus s *to*



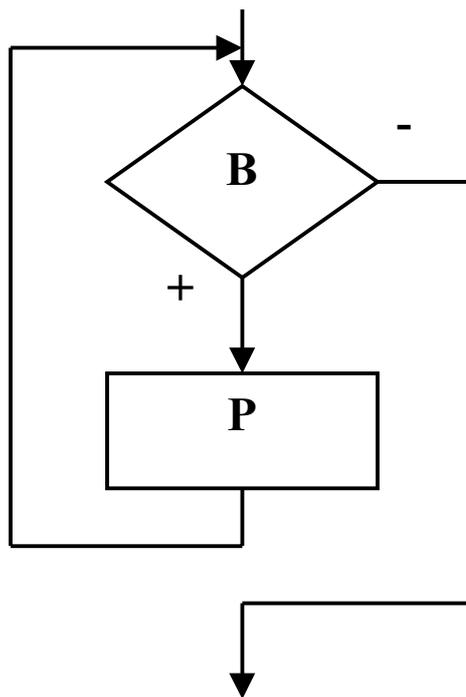
cyklus s *downto*

- cyklus s “testem na začátku”



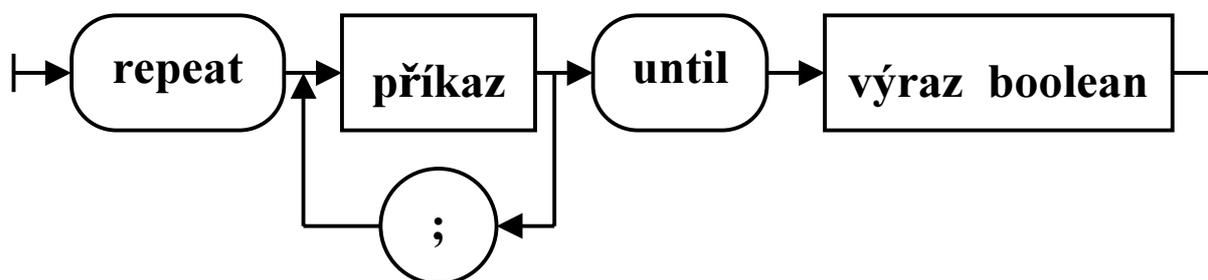
while B do P;

- pokud nabývá B hodnoty *true* provádí se příkaz P, jinak se pokračuje následujícím příkazem → příkaz P se nemusí vykonat ani jednou, popř. se může vykonávat nekonečněkrát !!!!! → P by měl obsahovat příkaz, který mění hodnotu výrazu B.



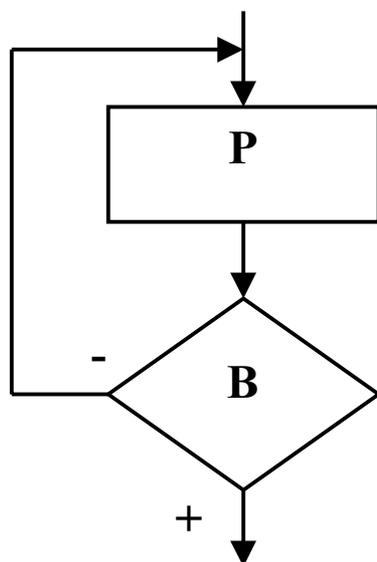
Př: **var X: real ;**
 ...
 X:=1.;
 while X<10. do begin write(X); X:=X+0.5; end;

- cyklus s “testem na konci”



```
repeat  
  P;  
until B;
```

- pokud nabývá B hodnoty *true* opouští se cyklus, jinak se provádí příkaz P → příkaz P se vykoná alespoň jednou, popř. se může vykonávat nekonečněkrát !!!!! → P by měl obsahovat příkaz, který mění hodnotu výrazu B.



Typ boolean

- nabývá hodnot *true* a *false*. Proměnné tohoto typu smí být přiřazena některá z těchto hodnot nebo výraz typu boolean. Platí $false < true$.

Logické operátory: not and or xor

| P | Q | not P | P and Q | P or Q | P xor Q |
|-------|-------|-------|---------|--------|---------|
| true | true | false | true | true | false |
| false | true | True | false | true | true |
| true | false | False | false | true | true |
| false | false | True | false | false | false |

Turbo Pascal – dvojí způsob vyhodnocení boolean výrazů

- 1) úplné $\{B+\}$ - vyhodnotí se kompletní výraz
- 2) zkrácené $\{B-\}$ – výraz se přestává vyhodnocovat, pokud zbývající část nemůže ovlivnit výsledek