

Zápis programů

Při programování (zápisu algoritmů):

- pracujeme s proměnnými různých typů (a s konstantami),
- modifikujeme obsahy proměnných,
- voláme podprocedury,
- porovnáváme obsahy proměnných,
- rozhodujeme se podle toho,
- cyklíme,
- čteme vstup, vypisujeme výstup.

Vzhled programu v Pascalu

Příklad:

```
program nanic;  
const  x=10;  
        text='deset';  
var a,b:integer;
```

Program začíná vždy klíčovým slovem `program`!

Jednotlivé příkazy oddělujeme středníkem!

Následuje sekce definice konstant uvedená slovem `const`.

Konstanty přiřazujeme:

`konstanta = hodnota`

Následuje definice proměnných uvedená slovem `var`.

Definujeme celočíselné proměnné `a` a `b`.

Důležitost indentace:

```
program nanic; const x=10; text='deset'; var
a,b:integer; c:string;
begin write('Napis cislo: '); readln(a);
write('Napis dalsi cislo: '); readln(b);
writeln('Soucet je ',a+b); writeln(x,' je ',text);
end.
```

Proměnné a jejich typy:

Proměnná má stanovený typ. Datové typy (v Pascalu) jsou:

- **byte:** 0 .. 255 (celá čísla),
- **integer:** -32 768 .. 32 768,
- **longint:** -2^{31} .. 2^{31} ,
- **real:** -10^{38} .. 10^{38} (necelá čísla),
- **word:** 0 .. 65 535 (celá čísla),
- **char:** znaky (jeden znak 8-bitového ASCII),
- **string:** řetězec znaků (text) o délce až 255 znaků,
- **boolean:** pravda nebo lež (nabývá hodnot true a false).

Výrazy aritmetické:

- + Sčítání,
- - odčítání,
- * násobení,
- / dělení,
- závorky,
- div celočíselné dělení,
- mod zbytek po dělení.

Pozor na priority!

Pozor, div a mod má prioritu mezi sčítáním a násobením!

Pozor na sčítání stringů!

Příklad:

$$(a + 5) * 17 + (b \text{ mod } c)$$

Přiřazovací příkaz: :=

Příklad: `x := 2*y;`

Relační operátory

- $<$ je menší než (kupř. $a < b$),
- $>$ je větší než,
- \geq je větší nebo rovno,
- \leq je menší nebo rovno,
- $<>$ nerovná se,
- $=$ rovná se (porovnání na rovnost).

Podmínky

Syntax (a sémantika):

- if podmínka then prikaz;
- if podmínka then begin blok prikazu end;
- if podmínka then prikaz else prikaz;
Pozor, před else se středník nepíše!
- if podmínka then begin blok end else begin blok end;

Příklad:

```
if teplota>25 then  
    writeln('Jdu do hostince!');
```

Příklad

```
if teplota>25 then writeln('Jdu do hostince!');
```

```
if teplota>25 then
begin
    writeln('Jdu do hostince!');
end
else
begin
    writeln('Nejdu nikam!');
end;
```


Cykly

- while podmínka do prikaz nebo blok;
Opakuj, dokud je podmínka splněna.
- for i:=1 to 10 do prikaz nebo blok;
Opakuj pro každou hodnotu proměnné od první do druhé meze.
- for i:=100 downto 1 do prikaz nebo blok;
- repeat prikazy; until podmínka;
Opakuj, dokud je podmínka **nesplněna!**

Příklady

```
program dvojkova;  
var a:integer;  
begin  
    readln(a);  
    while a > 0 do  
        begin  
            if a mod 2 = 1 then  
                write(1)  
            else    write(0);  
            a:=a div 2;  
        end;  
end.
```

Příklad vylepšený

Při programování je hlavní **myslet, jinak si často zbytečně přiděláme práci!**

```
program dvojk;  
var a:integer;  
begin  
    readln(a);  
    while a > 0 do  
        begin  
            write(a mod 2);  
            a:=a div 2;  
        end;  
end.
```

Rozklad na prvočinitele

- Nápady
- Naivní algoritmus: Hledej postupně prvočísla až do n a zkoušej jimi dělit.
- Méně naivní algoritmus: Hledej prvočísla až do $\sqrt{n_i}$ (kde n_i je hodnota, kterou ještě zbývá faktorizovat).
- Ještě méně naivní algoritmus: Zkoušej všechna čísla až do $\sqrt{n_i}$.
Proč toto funguje?

Příklad, hledání prvočíselného rozkladu:

```
program rozklad;  
var a,i:integer;  
begin  
    i:=2;  
    readln(a);  
    while i <= a do  
    begin    if (a div i)*i = a then  
            begin  
                write(i);  
                a:=a div i;  
            end  
            else    i:=i+1;  
    end;  
end.
```

Příklad, hledání prvočíselného rozkladu vylepšený:

```
program rozklad;  
var a,i:integer;      opakujeme:boolean;  
begin   i:=2;   opakujeme:=false;  
        readln(a);  
        while i <= a do  
        begin   if (a div i)*i = a then  
                begin   if opakujeme then  
                        write('*')  
                else   opakujeme:=true;  
                        write(i);  
                        a:=a div i;  
                end   else   i:=i+1;  
        end   end;  
end.
```

Komentáře

Komentáře umísťujeme do složených závorek:

Příklad: {Toto je kus komentare}

Komentář může obsahovat kusy kódu, které si nejsme jisti, zda použijeme:

```
{x=sqrt(x); Tak nevim, melo to byt pro odmocninu nebo  
ne... :-}
```