

Obsah

Obsah

1 Programovací jazyk Pascal	1
1.1 Struktura programu v Pascalu	1
2 Proměnné	2
2.1 Vstup a výstup	3
3 Operátory a některé matematické funkce	5

1 Programovací jazyk Pascal

Programovací jazyk Pascal

- Jedná se o jazyk navržený prof. Niklausem Wirthem (TU Curych) na počátku 70. let 20. století.
- Jazyk byl navržen, aby:
 - se dal snadno implementovat na většině tehnedjších počítačů a
 - byl vhodný pro výuku programování.
- Od roku 1981 mezinárodním standardem jako ISO 7185, nyní ISO 7185:1990 a ISO 10206:1990 (celkem více než 300 stran).
- Vznikla (a stále vzniká) řada implementací. Mimo jiné:
 - Turbo Pascal (a Borland Pascal) firmy Borland,
 - Delphi – Borland,
 - Free Pascal – GNU/GPL licence (<http://www.freepascal.org/>).

1.1 Struktura programu v Pascalu

Struktura programu v Pascalu

Program v Pascalu se skládá z vnořených bloku, přičemž každý blok má následující strukturu:

```
typ_bloku jméno[(parametr[, parametr]...)]
  [label x[, y]...;]
  [const x=y;
    [q=r;]...]
  [type x=y;
    [q=r;]...]
  [var x[,y]...:z;
```

```

        [q[,r]...:s;]]
[block]...
begin
    příkaz[; příkaz]...
end[. |;]

```

Toto odpovídá normě ISO, Borland Pascal se v některých detailech odlišuje, je pro nás však použitelnější.

Struktura programu v Pascalu

K předchozímu schématu je potřeba dodat několik poznámek:

- Blok nejvyšší úrovně je vždy typu „program“ a příslušející „end“ končí tečkou.
- Ostatní bloky končí endem se středníkem.
- Jazyk Pascal nerozlišuje velká a malá písmena v identifikátorech ani v klíčových slovech.
- Jak je vidět z předchozího, slouží středník v bloku příkazů k oddělení příkazů (ne ukončení, jak je obvyklé v C-like jazycích).

2 Proměnné

Proměnné v Pascalu

- Data v Pascalu (i jiných programovacích jazycích) jsou ukládána v paměti počítače, která je přístupná vždy pouze pomocí proměnných.
- Proměnné jsou deklarovány direktivou **var**.
- Každá proměnná je určitého datového typu:
 - Znakový typ **char** – pro ukládání jednotlivých znaků (v kódování ASCII); literály se zapisují buď do apostrofů nebo jako #12 (ASCII hodnota).
 - Pravdivostní typ **boolean** – pouze 2 možné hodnoty (konstanty **true** a **false**).
 - Datový typ **string**, který slouží k ukládání řetězců znaků; literály se zapisují do apostrofů.
 - Celočíslné typy – se znaménkem (v doplňkovém kódu) i bez znaménka:
 - * **shortint**, byte (8bitový),
 - * **integer**, word (16bitový),
 - * **longint** (32bitový).

Proměnné v Pascalu – 2

- Každá proměnná je určitého datového typu:
 - Čísla s plovoucí řádovou čárkou (dle IEEE 754):
 - * real (48bitový) – rozsah $2,9 \cdot 10^{-39} - 1,8 \cdot 10^{38}$; 11-12 číslic,
 - * single (32bitový) – rozsah $1,5 \cdot 10^{-45} - 3,4 \cdot 10^{38}$; 7-8 číslic,
 - * double (64bitový) – rozsah $5,0 \cdot 10^{-324} - 1,7 \cdot 10^{308}$; 15-16 číslic,
 - * extend (80bitový) – rozsah $3,4 \cdot 10^{-4932} - 1,1 \cdot 10^{4932}$; 19-20 číslic.

```
var prom1:real;  
    prom2,prom3:string;
```

Příkaz přiřazení „:=“

```
prom1:=12.3;  
prom2:='Ahoj lidi';
```

2.1 Vstup a výstup

Vstup dat z klávesnice

Vstup dat z klávesnice zajišťuje procedura ReadLn, která má proměnný počet parametrů (minimálně 1), které musí být proměnné.

- Procedura načítá data z bufferu klávesnice postupně do proměnných uvedených v parametrech.
- Pro proměnné typu **string**:
 - Čte znaky z bufferu, dokud nenarazí na znak konce řádku (enter).
- Pro proměnné číselných typů:
 - Přeskočí všechny mezery a tabelátory které jsou na začátku bufferu.
 - Není-li následující posloupnost znaků v požadovaném formátu, nastane běhová chyba.
 - Následující čtení začíná na mezeře nebo tabelátoru, který ukončil předchozí číselný řetězec.

```
ReadLn(a);  
ReadLn(a,b);
```

Výstup dat na obrazovku

Výstup dat na obrazovku zajišťují procedury `Write` a `WriteLn`.

- `WriteLn` je vlastně `Write`, který po provedení výstupu přemístí kurzor na nový řádek.
- `Write` i `WriteLn` mají proměnlivý počet parametrů (i nulový počet), jejichž hodnoty jsou vypsané postupně zleva doprava (bez mezer).

```
WriteLn('Ahoj...');  
WriteLn('a=',a);  
Write('a=');  
WriteLn(a);
```

Formátování výstupu

`Write`, `WriteLn` (a některé další procedury) umožňují formátování výstupu číselných i jiných hodnot:

- Za výraz lze zapsat „:“, která udává minimální počet znaků zabraných na výstupu.
 - Je-li nejmenší šířka výstupu větší než požadovaná, není šířka dodržena.
 - Výstup je doplněn mezerami zleva, tzn. je zarovnán vpravo.
- U čísel s pohyblivou řádovou čárkou lze udat i další část (za další dvojtečkou), která určuje počet desetinných míst.
 - Je-li uveden počet desetinných míst, je výstup v desetinném tvaru se znaménkem (nic či -).
 - Bez udání počtu desetinných míst je výstup v exponenciálním tvaru, přičemž vždy se skládá z:
 - * znaménka mantisy (mezera či -),
 - * cifry mantisy a desetinné tečky na maximální možný počet vzhledem k požadované šířce a rozsahu, doplněný nulami zprava,
 - * E pro oddělení mantisy a exponentu,
 - * znaménka exponentu (+ či -) a
 - * 2 (nebo 4) cifer exponentu.

Příklad formátování výstupu

```
WriteLn('Text':10);  
WriteLn(123:10);  
WriteLn(12.345:10);  
WriteLn(12.345:15);
```

```
WriteLn(12.345:10:2);  
WriteLn(12.345);
```

```
          1111111112  
12345678901234567890  
      Text  
      123  
1.235E+01  
1.23450000E+01  
      12.35  
1.2345000000E+01  
          1111111112  
12345678901234567890
```

3 Operátory a některé matematické funkce

Aritmetické operátory

Výsledkem je vždy číslo.

- + – součet,
- – – rozdíl,
- * – součin,
- / – podíl,
- div – celočíselné dělení,
- mod – zbytek po dělení (dělení modulo).

Relační operátory

Výsledkem je vždy typ boolean.

- = – rovnost *pozor na porovnání čísel v plovoucí řádové čárce,*
- <> – nerovnost,
- < – menší než,
- > – větší než,
- <= – menší nebo rovno,
- >= – větší nebo rovno.

Logické operátory

Pracují s pravdivostními hodnotami a výsledkem je pravdivostní hodnota (boolean).

- not – unární logická negace,
- and – logický součin,
- or – logický součet,
- xor – nonekvivalence.

Bitové operátory

Pracují s celočíselnými typy a výsledkem je celočíselný typ.

- not – unární bitová negace,
- and – bitový součin,
- or – bitový součet,
- xor – bitová nonekvivalence,
- shl – bitový posun vlevo,
- shr – bitový posun vpravo.

Matematické funkce

Některé matematické funkce:

- $\sin, \cos, \tan(x) = \frac{\sin(x)}{\cos(x)},$
- $\arctan, \arcsin(x) = \arctan\left(\frac{x}{\sqrt{1-(x)^2}}\right), \arccos(x) = \arctan\left(\sqrt{1-(x)^2}\right),$
- $\exp, a^b = c^{b \log_c a},$
- $\ln, \log_a b = \frac{\log_c b}{\log_c a},$
- sqrt, sqr.