

základné príkazy – PRÍKAZ VÝSTUPU, PRÍKAZ VSTUPU, PRÍKAZ PRIRADENIA

Premenné

Skôr, ako pridáme k základným príkazom, budeme sa venovať premenným. Premenná je určitý symbol, ktorý zastupuje hodnotu. Z programátorského hľadiska premenná reprezentuje určitú oblasť pamäte, v ktorej je uložená hodnota. Programátor neurčuje adresu v pamäti, urobí to Delphi. Vždy pri spustení programu si OS nájde voľný priestor v pamäti a určí, ktorá premenná patrí ku ktorej adrese. Na túto adresu sa programátor jednoducho dostane podľa názvu premennej. V Pascale je vždy presne daného typu, to znamená, že do miesta pre číslo, nemôžeme zadať text a pod. Každú premennú musíme deklarovať, inak kompilátor pri spúšťaní ohlásí chybu.

V Pascale sa stretávame s pojmom **identifikátor**.

Sú to mená premenných, funkcií, tried, vlastných údajových typov, ktoré si vytvára sám programátor

Pravidlá pre tvorbu identifikátorov:

1. identifikátor sa skladá z postupnosti písmen, číslíc a podčiarkovníka (na začiatku nemôže byť číslica)
2. nie je rozdiel medzi malými a VEĽKÝMI písmenami (v Pascale)
3. identifikátor nesmie byť totožný s rezervovaným slovom,
4. identifikátor musí byť jednoznačný v rámci programu alebo jeho časti (procedúra, funkcia, ...) t.j. nemožno označiť rovnakým identifikátorom viacero funkcií, premenných, typov,

Deklarácia premenných

Deklarácia môže byť v rôznych častiach programu, najčastejšie sa však dáva do deklaračnej časti, začína rezervovaným slovom **var**. Slovo **var** je skratka zo slova **variable**, čo znamená – premenná. Deklarácia premennej pozostáva z dvoch častí a to z uvedenia názvu a typu. V Delphi poznáme niekoľko jednoduchých typov, ale aj iné štruktúrované, ktoré sú väčšinou určené len pre špeciálne účely.

Takto vyzerá typická deklarácia:

```
Var cislo,pocet:integer;
```

Pre každú premennú v programe treba deklarovať množinu prípustných hodnôt, to je typ premennej. Podľa typu sa rezervuje priestor v pamäti.

Napríklad nižšie sú ukážky celočíselných a reálnych typov:

Typy celočíselné	Rozsah	Veľkosť v bajtoch
Shortint	-128..127	1
Byte	0..255	1
Smallint	-32768..32767	2
Word	0..65535	2
LongInt Integer	-2147483648..2147483647	4
LongWord	0..4294967295	4
Int64	-(2 ⁶³)..(2 ⁶³)-1	8

Typy reálne	Rozsah	Veľkosť v bajtoch	Presnosť (počet des. miest)
Real	2.9 * (10 ⁻³⁹)..1.7*(10 ³⁸)	6	11-12
Single	1.5 * (10 ⁻⁴⁵)..3.4 * (10 ³⁸)	4	7-8
Double	5.0*(10 ⁻³²⁴)..1.7 * (10 ³⁰⁸)	8	15-16
Extended	3.4*(10 ⁻⁴⁹³²)..1.1*(10 ⁴⁹³²)	10	19-20
Comp	-2 ⁶³ ..2 ⁶³ -1	8	19-20

Celé čísla zapisujeme ako v matematike. Reálne čísla zapisujeme ako desatinné, ale používame desatinnú bodku (123.678) alebo v semilogaritmickom tvare 1.123456E-03, čo znamená 1,123456 · 10⁻³.

Okrem číselných typov máme aj iné, napríklad typ – **String**, ktorý je určený na uchovanie reťazcov, čiže textových hodnôt. Mohli by sme ho použiť napríklad pre meno a priezvisko.

PRÍKAZ VÝSTUPU

Tento príkaz zabezpečuje výstup údajov, napríklad na obrazovku. Ak chceme vypísať viacej hodnôt oddelíme ich čiarkou.

```
write( premenná, výraz, reťazec );
```

– vypíše obsahy premenných, hodnoty uvedených výrazov alebo uvedené reťazce znakov. Po výpise zostáva kurzor na mieste za posledným vypísaným znakom.

```
writeln( premenná, výraz, reťazec );
```

– vypíše obsahy premenných, hodnoty uvedených výrazov alebo uvedené reťazce znakov. Po výpise sa kurzor presunie na začiatok nového riadka.

Upozornenie: Iba pri výpise reťazcov dávame príslušný reťazec medzi apostrofy, tak ako je to v nasledovnom príklade.

Ukážme si malý príklad použitia:

1. <code>Write('Dnes'); Write(' bude '); Write('hodina informatiky'); Writeln;</code>	2. <code>Writeln('Dnes bude hodina informatiky.');</code>
--	--

V oboch prípadoch bude na obrazovke napísaná veta:

Dnes bude hodina informatiky.

Ako bude vyzerat' výstup na obrazovke, ak v príklade 1. Použijeme namiesto Write - Writeln?

3. Vyskúšajte, čo urobí nasledovný príkaz:

```
Writeln(Pi);
```

Program vypíše reálne číslo „škaredým spôsobom“ 1.14159265358979E+0000.

Formátovaný výpis premenných – výpis na pevný počet miest:

napr.: `write(125:5, 123.456:7:2, );`

výstup na obrazovke bude:

```
xx125x123.45  
x znamená medzera
```

napr.: `write(a:5, b:7:2);`

premenná a je typu integer a vypíšeme ju na 5 miest zarovnanú doprava, premenná b je typu real a vypíšeme ju na 7 miest, pričom desatinná bodka bude na 5-tej pozícii a desatinná časť bude mať dĺžku 2 znaky.

Úlohy:

1. Upravte príklad 3 použitím formátovaného výpisu.
2. Vytvorte nasledovnú vizitku so svojimi údajmi:

```
*****  
*                Janko Mrkvicka                *  
*                Gymnazium                      *  
*                Bratislava                     *  
*****
```

PRÍKAZ VSTUPU

Tento príkaz zabezpečuje vstup údajov, napríklad z klávesnice.

```
read( premenné oddelené čiarkou );
```

– prečíta údaje zo vstupu a uloží ho do príslušnej premennej. Znak klávesy Enter, ktorou boli vstupné údaje programu poslané, neprečíta. Tento znak zostáva nespracovaný, bude následne čítaný.

```
readln(premenné oddelené čiarkou );
```

– prečíta každý údaj zo vstupu a uloží ho do príslušnej premennej. Znak klávesy Enter, ktorou boli vstupné údaje programu poslané, prečíta tiež.

Oba príkazy zastavia realizáciu programu, pokiaľ užívateľ nezadá dostatočný počet vstupných hodnôt.

Ukážme malý príklad použitia:

- Napíšte program, ktorý načíta dve celé čísla z klávesnice, vypočíta ich súčet a vypíše výsledok tak, že vypíše zadané čísla, znamienko + medzi nimi, potom = a výsledok.

```
program sucet;  
var a,b:integer;  
begin  
  writeln('zadaj dve cisla');  
  readln(a,b);  
  writeln(a, '+', b, '=', a+b);  
end.
```

Úlohy:

1. Upravte vašu vizitku tak, že vstupný údaj – meno a priezvisko budete zadávať z klávesnice.

PRÍKAZ PRIRADENIA

Tvar príkazu:

premenná := výraz;

Priradenie je jednoduchý príkaz, ktorým sa výsledná hodnota výrazu na pravej strane od dvojice znakov (:=) zapíše do premennej, ktorej identifikátor je uvedený na ľavej strane výrazu.

Napr. `o := 2 * (a + b);`

Tento príkaz priradenia sa vykoná takto: najprv procesor prečíta hodnotu premennej `a`, potom prečíta hodnotu premennej `b`. Prečítané hodnoty spočíta a vynásobí dvomi. Výsledok potom zapíše do premennej `o`.

Ešte zopár postrehov. Na ľavej strane príkazu priradenia je vždy iba jedna premenná – cieľová. Na pravej strane je zvyčajne **aritmetický výraz**, ktorý pozostáva z operandov, medzi ktorými stoja **aritmetické operátory**: `+`, `-`, `*`, `/`, `div`, `mod`. Operandom môže byť konštanta, premenná alebo funkcia. Vo výrazoch môžu byť použité i zátvorky `()`. Výsledkom aritmetického výrazu je vždy číslo.

Samozrejme, že pri vyhodnotení výrazu platí nasledovné poradie:

1. **funkcie** (napr. `ABS(x)`, `SIN(X)` a `pod.`)
2. `()`
3. `*`, `/`, `div`, `mod`
4. `+`, `-`

Teda pri vyhodnocovaní výrazov platia pravidlá:

1. Najskôr sa vyhodnotia funkcie a výrazy v zátvorkách.
2. Ak výraz obsahuje operácie s rôznou prioritou, najskôr sa vyhodnotia tie s vyššou prioritou.
3. Ak výraz obsahuje operácie s rovnakou prioritou, vyhodnocuje sa zľava doprava.

Ešte jedno upozornenie: Pri niektorých aritmetických operáciách môžeme použiť spolu premenné typu `INTEGER` a `REAL`. Nasledujúca tabuľka znázorňuje, pre ktoré aritmetické operácie je to možné a zároveň obsahuje aj typ výsledku príslušnej aritmetickej operácie.

a	b	a+b	a-b	a*b	a/b	a div b	a mod b
integer	integer	integer	integer	integer	real	integer	integer
integer	real	real	real	real	real	-	-
real	integer	real	real	real	real	-	-
real	real	real	real	real	real	-	-

Ukážme si malý príklad použitia:

- Zistíte, čo budú obsahovať jednotlivé premenné po skončení celej postupnosti priradovacích príkazov:
`A:=10;`
`B:=5;`
`C:=A+B;`
`C:=C+1;`
- Zostavme program na sčítanie a vynásobenie dvoch zadaných čísel.

```
program pocy;  
var a,b,sucet,sucin:integer;  
begin  
    writeln('Zadaj dve cisla za sebou pomocou ENTER');  
    readln(a);  
    readln(b);  
    sucet:=a+b;  
    sucin:=a*b;  
    writeln('Sucet ',a:5,'+',b:5,'=',sucet:8);  
    writeln('Sucin ',a:5,'*',b:5,'=',sucin:8);  
    readln;  
end.
```

Úlohy:

1. Doplňte predchádzajúci program, aby program odčítal, delil, počítal celočíselný podiel a zvyšok pri delení dvoch zadaných čísel.
2. Zistíte, ktoré deklarácie sú správne a ktoré nesprávne

```
var  
cislo : real;  
f, m : integer;  
integer : d, o, p;  
6k, 2f : integer;  
g4, v9 : integer;
```

```

end : real;
567 : integer;
podiel_gh : real;
súčetAB : integer;
integer: číslo;
a, A : integer;

```

3. Zostavte program, ktorý načíta celé nezáporné číslo STRANA stranu štvorca a vypočíta obvod a obsah štvorca.
4. Zostavte program, ktorý načíta celé nezáporné čísla STRANA_A, STRANA_B , STRANA_C strany kvádra a vypočíta objem a povrch kvádra.
5. Zostavte program, ktorý načíta reálne číslo R, predstavujúce polomer, a vypočíta obvod kružnice, obsah kruhu, objem gule.
6. Zostavte program, ktorý načíta odvesny A, B pravouhlého trojuholníka a vypočíta veľkosť prepony.
7. Zostavte program, ktorý načíta celé číslo X a vypočíta 3. a 6. mocninu.
8. Zostavte program, ktorý načíta VKLAD výšku vkladu v €, ročnú úrokovú mieru M v percentách a vypočíta sumu na konci roka.
9. Zostavte program, ktorý vymení obsah dvoch premenných X, Y. Existujú dva spôsoby.
10. Zostavte program, ktorý načíta celé nezáporné čísla H, M, S, (predstavujúce čas v hodinách, minútach, sekundách) a vyjadrite tento čas v sekundách.
11. Zostavte program, ktorý načíta celé nezáporné číslo S,(predstavujúce čas v sekundách) a vyjadrite tento čas v hodinách, minútach a sekundách.
12. Zostavte program, ktorý načíta celé nezáporné číslo S predstavujúce sumu na výplatu v €, vypíšte najmenší počet bankoviek potrebných na výplatu uvedenej sumy.
13. Zostavte program, ktorý načíta SPOTREBA – spotrebu auta na 100 km, CENA – za 1l benzínu a VZDIALENOST - vzdialenosť v km, ktorú má auto prejsť a vypočíta, koľko bude stáť benzín potrebný na cestu.
14. Zostavte program, ktorý načíta dve reálne čísla X, Y a bez porovnania vypíše hodnotu menšieho z nich a potom hodnotu väčšieho z nich.