

Lyginamoji Pascal/Delphi ir C/C++ kalbų apžvalga

Kalbos savybės ir elementai	PASCAL (Turbo, Borland, Delphi)	C / C ++ (Turbo, Borland, MS)
Bendra charakteristika	<p>Sukurta 1970-1971 metais, Zurich'e, Šveicarijoje; autorius prof. Niklaus'as Wirth'as.</p> <p>Prototipai: ALGOL-60, -68.</p> <p>Paskirtis: sistemingas programavimo mokymas, algoritminės ir programavimo kalbos savybių derinimas, taikomųjų užduočių programavimas.</p> <p>Pagrindinės savybės: griežtumas, stipri tipizacija, daug apribojimų, programos patikimumas, programos skaidrumas, įvedimo-išvedimo priemonės įeina į kalbą,</p> <p>kalba orientuota į virtualinę PASCAL mašiną ir "slepia" realų kompiuterį nuo programuotojo.</p>	<p>Sukurta 1971-1972 metais, Bell Laboratorijoje; autoriai: Bryan Kernighan, Dennis Ritchie (K&R).</p> <p>Prototipai: BPCL ir B kalbos.</p> <p>Paskirtis: sisteminis programavimas (UNIX OS sukurta C), aukšto ir žemio lygio programavimo kalbų savybių derinimas, profesionalus sudėtingų taikymų programavimas.</p> <p>Pagrindinės savybės: lankstumas, betipinių kalbų tradicija, beveik nėra apribojimų, programos efektyvumas, sunkiai skaitoma programa, įvedimo-išvedimo priemonės nėra kalbos dalis ir įeina tik į standartines bibliotekas; kalba kiek žemesnio lygio ir leidžia patogiai bendrauti su realiu kompiuteriu.</p>
Abėcėlė	<p>Lot. abėcėlės raidės, spec.simboliai: ., ; : - _ + - * / = > < ^ @ () { } [] . ' .</p>	<p>Lot. abėcėlės raidės, spec.simboliai: ., ; : - _ + - * / = > < ^ @ () { } [] . ' . "! ? % & # \ ~</p>
Identifikatoriai (žodžiai)	<p>Apibrėžia konstantas, kintamuosius, žymes, tipus, funkcijas ir procedūras. Prasideda raide arba _ . Susideda iš raidžių, _ , . , skaičių. Ilgis neribotas, reikšminiai 128 simboliai. Didž. ir maž. neskiriamos. Rezervuoti, standartiniai, vartotojo.</p>	<p>Apibrėžia konstantas, kintamuosius, tipus ir funkcijas. Prasideda raide arba _ . Susideda iš raidžių, _ , . , skaičių. Ilgis neribotas, reikšminiai 64 simboliai. Didž. ir maž. raidės skiriasi. Rezervuoti, standartiniai, vartotojo.</p>

Paprastieji duomenų tipai	integer (byte, shortint, word, longint), real (double, extended, comp), char, boolean, programuotojo tipai: .. , (, , ,) 15, 128, 2000 \$15 (=21 ₁₀) 'a', 'aaa'	unsigned int, short int, int , unsigned long, long, enum, float , double, long double, char , unsigned char, ASCII bool (tik C++) {false, true} C – {0, 1} 15, 123, 2000 015 (=13 ₁₀), 0123 (=83 ₁₀) 0x15 (=21 ₁₀), 0x12L (=18 ₁₀) 'a' (\ddd arba \xhhh), "aaa"
Aritmetinės	+, -, *, /, div, mod, Inc(), Dec()	+, -, *, /, %, x = prefiksinė operacija, kur x gali būti +, -, /, %, ++, --
Loginės	not, and, or, xor	!, &&,
Lyginimo	=, <>, <, >, >=, <=, in	=, !=, <, >, >=, <=
Bitinės	not, and, or, xor shl, shr	~, &, , ^ <<, >> x = prefiksinė operacija, kur x gali būti &, , ^, >>, <<
Sąlyginė Kitos	sizeof	?: sizeof, [], (), ->, . ,
Operacijų prioritetai	not *, /, div, mod, and, shl, shr +, -, or, xor =, <>, <, >, >= in	(), [], →, . !, ~, ++, --, - iš dešinės į kairę *, &, size of *, /, % +, - <<, >> <, <=, >, >= ==, != & ^ && ?: iš dešinės į kairę =, + =, -=, kt. iš dešinės į kairę ,
Išraiškos, priskyrimas	i:=i+5 x:=x*(y+3)	i=i+5 arba i += 5 x=x*(y+3) arba x *= y+3

<p>Programos struktūra</p> <p>Aprašai (deklaracijos)</p>	<p>Program <i>vardas</i>;</p> <p>Label</p> <p>Const <i>SavDienos</i>=7</p> <p>Type <i>metrai</i>=integer; <i>Nuorsveik</i>=^integer; <i>kompleksiniai</i>= record <i>re, im</i> : real; end;</p> <p>Var <i>atstumas</i> : <i>metrai</i>; <i>in, jn</i> : <i>nuorsveik</i>; <i>z</i> : <i>kompleksiniai</i>; <i>zn</i> : ^<i>kompleksiniai</i>;</p> <p>Function <i>vardas</i> () : <i>tipas</i>; Procedure <i>vardas</i> (); Function <i>vardas</i> () : <i>tipas</i>; Procedure <i>vardas</i> ();</p> <p>BEGIN { <i>Tai komentarai</i> }</p> <p>END.</p>	<pre># include <stdio.h> # include <failas1.h> # define savdienos 7 # define Begin { # define End } main () { typedef int metrai,*nuorsveik; typedef struct {double re,im} kompleksiniai; metrai atstumas; int i; nuorsveik in, jn; kompleksiniai z, *zn extern int j; atstumas=25; printf ("Klaipėda- Palanga=%f\n", atstumas) /* Tai komentarai */ } funkcija1 () { float x=0.0 } failas1.h funkcija2 () { int i, j; static int skaitiklis; }</pre>
<p>Sudėtinis</p> <p>Sąlyginis</p> <p>Variantinis</p> <p>Sakiniai</p> <p>Ciklas su "prieš" sąlyga</p>	<p>Begin <i>sakiniai</i> end</p> <p>If <i>sąlyga</i> then <i>sakinys_1</i> else <i>sakinys_2</i>;</p> <p>Case <i>išraiška</i> of {pažymėti sakiniai} žymė 1: <i>sakinys</i>; žymė N: <i>sakinys</i>; [else] <i>sakinys</i> end; while <i>sąlyga</i> do <i>sakinys</i></p>	<p>{ <i>kintamųjų aprašas</i> <i>sakiniai</i> }</p> <p>if (<i>išraiška</i>) {<i>sakinys_1</i>;...} else <i>sakinys_2</i>;</p> <p>switch (<i>išraiška</i>) { <i>pažymėti sakiniai</i> case 1: <i>sakinys</i>; break; case N: <i>sakinys</i>; break; default: <i>sakinys</i>; } while (<i>išraiška</i>) <i>sakinys</i></p>

<p>Ciklas su “post” sąlyga</p> <p>Ciklas su kartojimu skaičiumi</p>	<p>repeat <i>sakiniai</i> until <i>sąlyga</i></p> <p>for <i>ck:=pradinė to [downto] galutinė</i> do <i>sakinys</i></p>	<p>do <i>sakinys</i> while (<i>išraiška</i>)</p> <p>for (<i>ciklo inicializ; išraiška; reinicializacija</i>) <i>sakinys</i>;</p> <p>arba while ekvivalentas: <i>ciklo inicializ; while</i> (<i>išraiška</i>) { <i>sakinys reinicializacija; </i>}</p>
<p>Paprogramėlės (funkcijos, procedūros)</p>	<p>Yra funkcijos (function), kurios gražina reikšmę su funkcijos vardu ir procedūros (procedure), kurios reikšmės negražina.</p> <p>Programos struktūra hierarchinė – funkcijos ir procedūros gali būti įdėtos į kitas funkcijas ir procedūras</p>	<p>Yra tik funkcijos, kurios gražina reikšmę sakiniu return ().</p> <p>Programos struktūra nehierarchinė.</p>
<p>Masyvai</p>	<p><i>a</i> : array [1..10] of integer; <i>matrica</i> : array [1..10 , 1..15] of real;</p>	<pre>int a[]={1,2,3,4,5,6,7}; int *an; int a[10] = {1,2,3,4,5,6,7}; float matrica[10][15]; int index=0,pirmas,galut, blog; pirmas=a[index]; galut=a[6]; blog=a[index+8] an=a; /* an rodys į a[0] */ an++; /* an rodys į a[1] */ an++; /*dabar an rodys a[2] */ a[i] ekvivalentiškas *(a+i) a+ i ekvivalentiškas &a[i] a ekvivalentiškas &a[0] int elem_skaic; elem_skaic=sizeof(a)/sizeof(int)</pre>

Pavyzdžiai.

1. Sąlyginės operacijos panaudojimas. Palyginkite:

PASCAL:

```
if a > b then max := a
      else max := b;
```

C/C++:

```
if (a>b) max = a;
      else max = b;
```

arba

```
max =(a>b) ? a : b;
```

2. Kintamųjų aprašai ir inicializacija, nuorodos, prefiksinės ir postfiksinės operacijos:

```
int a=0; b=1, c;
```

```
a++; a=a+1; a+=1; ++a;      --a;
```

```
c = a ++    /* dabar c lygus 0, a lygus 1 */
```

```
c = ++ a    /* dabar c lygus 2, a lygus 2 */
```

```
c = ++ b    /* dabar c lygus 2, a lygus 2 */
```

int sveikasis

int* nuoroda į sveikąjį

int*[3] masyvas iš 3 n. į sveik.

int * () funkcija, gražinanti nuorodą į sveikąjį kintamąjį

int (*) () nuoroda į funkciją, gražinančią sveikąjį kintamąjį

3. Programos struktūra, aprašai ir veiksmi, funkcijos, ciklai.

3.1. Skaičių sumos skaičiavimas (ciklas **for**):

```
#include <stdio.h>
#include <stdlib.h>

int k, sum;
int main()
{
    sum=0;
    for (k=1; k<=10; k++) {
        sum=sum+k;
    }
    printf("Suma= %d\n", sum);
    system("PAUSE");
    return 0;
}
```



3.2. Kėlimas laipsniu, x^i , kur x ir i – sveikieji skaičiai:

```
1  #include <stdio.h>
2
3  laipsnis(int x, int n) /* calculate x power n, where n>0 */
4  {
5      int i,p;
6      p=1;
7      for (i=1; i<=n; ++i)
8          p=p*x;
9      return(p);
10 }
11
12 main()
13 {
14     int i;
15     for (i=0; i<10; ++i)
16         printf("%d \t %d \t %d \n", i, laipsnis(2,i), laipsnis(-3,i));
17 }
18
19
20
```

```
*G:\Paskaitos\Programavimas\2018\Lesson_2\Daniil Ivanov Funkcija.exe*
0      1      1
1      2      -3
2      4      9
3      8     -27
4     16     81
5     32    -243
6     64     729
7    128   -2187
8    256    6561
9    512   -19683

Process returned 0 (0x0)   execution time : 0.172 s
Press any key to continue.
```