

4-5 PASKAITOS

Turinys: Aritmetiniai reiškiniai. Prieskyra. Kai kurios bibliotekinės matematinės funkcijos. Santykio ir loginiai reiškiniai. Operacijų vyresnumo lentelė.

ARITMETINIAI REIŠKINIAI

- *operandas1* *veiksmo operatorius* *operandas2*

Veiksmo operatorius: +, -, *, /, % .

Operandas: matematinė funkcija, kintamasis, konstanta. Galimi ().

Vyresnumas: 1 lygis – (); 2 lygis – *, /, %; 3 lygis – +, -.

% operacija taikoma tik sveikiesiems duomenims ir grąžina sveikaskaitinės dalybos likutį: $i1 \% i2 = [k * i2] + il$. Neigiamiems operandams: operacija priklauso nuo kompiliatoriaus.

Operandų tipai. Operandai reiškinyje gali būti skirtingų tipų. Reiškinio rezultato tipas priklauso nuo operandų tipų: automatiškai jis yra bendresniojo operando tipo. Pavyzdžiui, $5/2$ ir $5./2$ rezultatai yra skirtingų tipų ir reikšmių. Faktiškai skaičiuojant reiškinį mažiau bendro tipo operandas automatiškai pertvarkomas į bendresnįjį tipą:

```
...
int a = 1;
float b = 2.f;
double c = a * b; // a -> 1.f, po to a*b = 2.f -> 2.
...
```

Galima tipus pertvarkyti ir išreikštiniu būdu:

...(reikiamas tipas) kintamasis;

Rekomenduojamas pertvarkymo būdas:

... *static_cast* <reikiamas tipas> (kintamasis);

Pavyzdys:

```
#include <iostream>
using namespace std;
int main ( ){
    int var = 30000000; //30 000 000
    var = (static_cast<double>(var) * 100) / 100; // gerai
    var = (var * 100) / 100; //vykdymo laiko klaida!
    return 0;
}
```

PRIESKYROS OPERATORIAI

kintamasis = reiškiny;

reiškiny gali būti supaprastėjęs iki kintamojo arba konstantos.

Pavyzdys:

```
...  
int i = 1;  
i = i + 1;
```

...
Sutrumpintoji prieskyros forma:

kintamasis operacija = reiškiny;

Pavyzdžiai:

```
...  
int i = 1;  
i += 1;  
...  
i = 27;  
i /= 3;  
i %= 5;    // =4  
...
```

Inkrementas/dekrementas, pre- ir postformas.

i = i + 1; ≡ ++i; //preforma ≡ i++; //postforma

Atskirame operatoriuje abi formas ekvivalentiškos. Esant kelioms operacijoms – ne:

```
#include <iostream>  
using namespace std;  
int main ( ){  
    int i = 10;  
    cout<<"i = "<<i<<endl;           //bus spausdinama 10  
    cout<<"i = "<<++i<<endl;         //11  
    cout<<"i = "<<i<<endl;           //11  
    cout<<"i = "<<i++<<endl;         //11  
    cout<<"i = "<<i<<endl;           //12  
    return 0;  
}
```

Dekrementas: *--i;* arba *i--;* .

Kai kurios bibliotekinės matematinės funkcijos. Antraštiniai failai

Informacija apie jas – antraštiniame faile *cmath*, o pačios funkcijos mašinos kodo pavidalu yra bibliotekiniuose failuose **.lib*.

Kelios funkcijos:

$\arccos x$	<i>double acos(double)</i>
$\arcsin x$	<i>double asin(double)</i>
$\arctg x$	<i>double atan(double)</i>
$\cos x$	<i>double cos(double)</i>
$\sin x$	<i>double sin(double)</i>
$\operatorname{tg} x$	<i>double tan(double)</i>
e^x	<i>double exp(double)</i>
$\ln x$	<i>double log(double)</i>
$\lg x$	<i>double log10(double)</i>
x^y	<i>double pow(double x, double y)</i>
$x^{1/2}$	<i>double sqrt(double)</i>
$ x $	<i>double abs(double)</i>

SANTYKIO IR LOGINIAI REIŠKINIAI

Santykio: apjungia aritmetinius operandus, o rezultatas – loginis.

Santykio operatoriai: $<$, $<=$, $>$, $>=$, $==$, $!=$.

Pavyzdžiui, jei $a=1$, $b=2$, tai $a<b$ reikšmė yra *true* (arba 1), o $a==b$ reikšmė – *false* (arba 0).

Loginiai: apjungia loginius operandus, o rezultatas – irgi loginis. Yra vienas unarinis (su vienu operandu) loginis veiksmas: neigimas $!$. Binariniai veiksmai: loginė sudėtis ir loginė daugyba: $||$ ir $\&\&$. Veiksmai atliekami pagal “trumpąją schemą”.

Vyresnumas nuo aukščiausio prioriteto žemyn: $!$, santykiai, $||$, $\&\&$.

Pavyzdžiai:

1. Skaičių ašyje duoti intervalai $[\infty, x1]$, $[x2, x3]$, ir $[x4, \infty]$, $x1 < x2 < x3 < x4$ ir skaičius x .

Parašyti reiškinius, kurių reikšmė būtų *true*:

- a) kai x yra pirmame intervale;
- b) antrame;
- c) pirmame, antrame arba trečiame.

a): $x \leq x1$

b): $x \geq x2 \ \&\& \ x \leq x3$

c): $(x \leq x1) \ || \ (x \geq x2 \ \&\& \ x \leq x3) \ || \ (x \geq x4)$

2.

...

int i1 = 1, i2 = 2;

... *!i1 && i2...* //reikšmė 0

...*2||3...* //reikšmė 1

OPERACIJŲ VYRESNUMO LENTELĖ

1	(), [], ->	vykdymo tvarka: ->
2	!, ++, --, sizeof()	<-
3	*, /, %	->
4	+, -	->
5	<<, >>	->
6	>, >=, <, <=	->
7	==, !=	->
8	&	->
9	^	->
10		->
11	&&	->
12		->
13	?:	<-
14	=, +=, -=, *=, /=, %=	<-
15	^=, =, <<=, >>=	<-
16	,	->