

7 tema. Pagrindinės matematinės operacijos

Temos uždaviniai. Susipažinti su pagrindinėmis matematinėmis operacijomis ir jomis atlikti veiksmus.

7.1 Skiriamos šios pagrindinės matematinės operacijos

```
+ - sudėtis  
- - atimtis  
* - daugyba  
/ - dalyba (sveikosios dalies gavimas)  
% - dalybos modulis (liekanos nustatymas)  
++ - inkrementas  
-- - dekrementas
```

Pavyzdys. Programa į ekraną išveda įvairių aritmetinių veiksmų rezultatus.

```
#include<iostream>  
using namespace std;  
int main(void)  
{  
cout << "5+7 =" << 5+7 << endl;  
cout << "12-7 = " << 12-7 << endl;  
cout << "1.2345 * 2 = " << 1.2345 * 2 << endl;  
cout << "15/3 = " << 15/3 << endl;  
cout << "15 %3 = " << 15%3 << endl;  
return 0;  
}
```

```
5+7 =12  
12-7 = 5  
1.2345 * 2 = 2.469  
15/3 = 5  
15 %3 = 0  
  
Process returned 0 (0x0)   execution time : 0.015 s  
Press any key to continue.
```

Komentaras:

C++ kalboje dalybos operacija (/) rezultate pateikia sveikąją dalį, o dalybos operacija (%) – liekaną. Norint rezultate gauti realaus tipo skaičių (su kableliu), reikia skaičiavimuose naudoti dalybos operaciją (/) ir gautą rezultatą konvertuoti į double tipą

7.2 Didinimo vienetu komanda (++)

Kintamojo reikšmei padidinti vienetu (galima ir kita reikšme) plačiai naudojama komanda (++).

Užrašas `kiekis++` atitinką operaciją `kiekis = kiekis + 1;`

Teoriniai pastebėjimai

- Programoje galima patalpinti didinimo operatorių ++ prieš arba po kintamojo: `++skaicius;` `skaicius++;`
- Kai didinimo operatorius ++ randasi prieš kintamąjį, jis vadinamas priešdidinimo operatoriumi.
- Kai didinimo operatorius ++ randasi po kintamojo, jis vadinamas podidinimo operatoriumi.

C++ kalba juos supranta skirtingai.

Skirtumą išsiaiškinsime pavyzdžių pagalba.

1 pavyzdys

```
B = 10;  
A = B++;  
// A = 10, B = 11;  
Kintamojo B reikšmė priskiriama kintamajam A, tada B reikšmė padidinama
```

2 pavyzdys

```
B = 11;  
A = ++ B;    // A = 12, B = 12  
B reikšmė padidinama, ir gauta reikšmė priskiriama kintamajam A.
```

7.3 Mažinimo vienetu operatorius (--)

Operatorius (--) reiškia mažinimą vienetu. Kaip ir padidinimo operacijose C++ kalba turi priešmažinimo ir pomažinimo operatorių.

7.4 Sudėtinės operacijos

Operatorius		Pavyzdžiai ir paaiškinimai
++	K++	K reikšmė panaudojama, po to didinama vienetu.
	++K	K reikšmė didinama vienetu, po to panaudojama.
--	K--	K reikšmė panaudojama, po to mažinama vienetu.
	--K	K reikšmė mažinama vienetu, po to panaudojama.
+=	S+= K;	S = S + K;
-=	S-= K;	S = S - K;

Pavyzdys. Programa, sudaugina su skaičius ir rezultatą parodo ekrane

```
#include<iostream>
using namespace std;
int main(void)
{
    int pirmas=5, antras=3, rez;
    rez = pirmas*antras;    // rez = pirmas * antras;
    cout << rez;
    return 0; }
```

```
15
Process returned 0 (0x0)   execution time : 0.013 s
Press any key to continue.
■
```

7.5 Lyginimo ir loginės operacijos

Lyginimo operacijos

== - ar lygios dvi reikšmės?

> - ar pirma didesnė už antrą?

< - ar pirma mažesnė už antrą?

>= - ar pirma nemažesnė už antrą?

<= - ar pirma nedidesnė už antrą?

!= - ar nelygios dvi reikšmės?

Loginės operacijos

|| - loginė sudėtis arba (or) ! – loginis neigimas ne (not)

&& - loginė daugyba ir (and)

Pavyzdys. Programa apskaičiuoja loginį reiškinį ir ekrane išveda skaitinę reikšmę 0 (False, 1-reiškia True).

```
#include<iostream>
using namespace std;
int main(void)
{
    bool rez;           // aprašome loginio tipo kintamąjį rez
    rez = ((10 == 7) && (-1 > 2));
    cout << rez;
    return 0;
}
```

```
0
Process returned 0 (0x0)   execution time : 0.005 s
Press any key to continue.
```

Komentaras: Ekrane rodomas 0 reiškia atsakymą FALSE, o 1 – reiškia TRUE reikšmę.

7.6 Kablelio (,) operatorius

Naudojamas atskirti du arba daugiau išraiškų, kai viena išraiška nekinta.

```
d = (c=10, c+1); // c=10, d = 11
```

Pavyzdys. Programa į ekraną išveda skaičius 10,11,12,13,14,15

```
#include<iostream>
using namespace std;
int main(void)
{
    int d, c;
    for (int i=0; i<5; i++)
    {
        d = (c=10, c+i);
        cout <<d<<'\n';
    }
    return 0;
}
```

```
10
11
12
13
14
Process returned 0 (0x0)   execution time : 0.005 s
Press any key to continue.
```

7.7 Operacijų vykdymo tvarka

Kad išvengtų painiavos, C++ kalboje kiekvienai operacijai suteikiamas prioritetas, nustatantis veiksmų eilę. Kadangi C++ vykdo aritmetines operacijas tam tikra tvarka, tai ir jūsų programos atitinkama tvarka vykdys veiksmus.

C++ operacijų vykdymo tvarkos lentelė.

Operacija	Apibūdinimas	Pavyzdys
::	Kintamojo apibrėžimo sritis	class_name::class_member_name::variable_name
::	Globalaus kintamojo apibrėžimas	::variable_name
.	Elemento išrinkimas	object.member_name
->	Elemento išrinkimas	pointer-member_name
[]	Indeksacija	pointer[element]

0	Funkcijos kvietimas	expression(parameters)
0	Reiškinio rezultatas	type(parameters)
sizeof	Objekto dydis	sizeof expression
sizeof	Tipo dydis	sizeof(type)
++	Inkrementavimas po	variable++
++	Inkrementavimas prieš	++variable
--	Dekrementavimas po	variable--
--	Dekrementavimas prieš	--variable
&	Objekto adresas	&variable
*	Pervardinimas	*pointer
New	Sukūrimas (išdėstymas)	new type
Delete	Panaikinimas (ištrynimasis)	delete pointer
delete[]	Masyvo panaikinimas	delete pointer
~	Papildymas	~expression
!	Inversija	!expression
+	Pridėti	+1
-	Atimti	-1
.*	Elemento išrinkimas	object.*pointer
->	Elemento išrinkimas	object->*pointer
*	Daugyba	expression*expression
/	Dalyba	expression/expression
%	Liekana po dalybos	expression%expression
+	Sudėtis	expression+expression
-	Atimtis	expression-expression

Pasipraktikuokite

1. Apskaičiuokite reiškinius:

- a) $2 + 34 / 3 * (4 \% 5)$;
- b) $1 + 59 \% 5 + 7 / 4$;
- c) $5 * (4 / 3 * (5 \% 7 * 2))$;
- d) $1 + 59 \% (5+7) / 4$;
- e) $9 / 4 * 3 + 34 \% 3 * 4 \% 5$;
- f) $1 + 59 \% 5 + 7 / 4$;

2. Parašykite loginius reiškinius:

Pradiniai duomenys – trys skaičiai a, b ir c. Parašykite loginį reiškinį, kurio reikšmė būtų *true* (ekrane rodytų skaičių 1), tada ir tik tada, kai:

- a) visų trijų kintamųjų a, b ir c reikšmės lygios;
- b) visų trijų kintamųjų a, b ir c reikšmės skirtingos;
- c) kurių nors dviejų kintamųjų reikšmės lygios;
- d) visų trijų kintamųjų a, b ir c reikšmės yra lyginiai skaičiai;
- e) visų trijų kintamųjų reikšmės yra teigiamos, bet ne didesnės kaip 100.