
C++ programavimo kalba

Įvadas į OOP, I/O biblioteka

doc.dr. Dalius Mažeika

Dalius.Mazeika@fm.vtu.lt

VGTU SC L318

Programavimo kalbos

- **Procedūrinės (Assembler, ankstyvos, Fortran versijos)**
 - Pagrindą sudaro procedūros. Kiekvienas operatorius, lyg procedūra nurodo kompiuteriui, ką daryti (programa – tai instrukcijų rinkinys).
Programos kodo ilgis iki kelių tūkstančių eilučių.
 - **Struktūrinės (Pascal, C, Fortran, Basic, COBOL)**
 - Pagrindą sudaro blokai (funkcijos, paprogramės, bibliotekos) ir duomenys, priskirti tiems blokams. Bendrai naudojami duomenys apibrėžiami kaip globalūs. Programos kodo ilgis iki 50 tūkstančių eilučių.
 - **OOP (C++, C#, Java, Visual Basic)**
 - Naudojami objektai bei jų savybės: inkapsuliacija, paveldimumas, polimorfizmas.
-

Struktūrinis programavimas

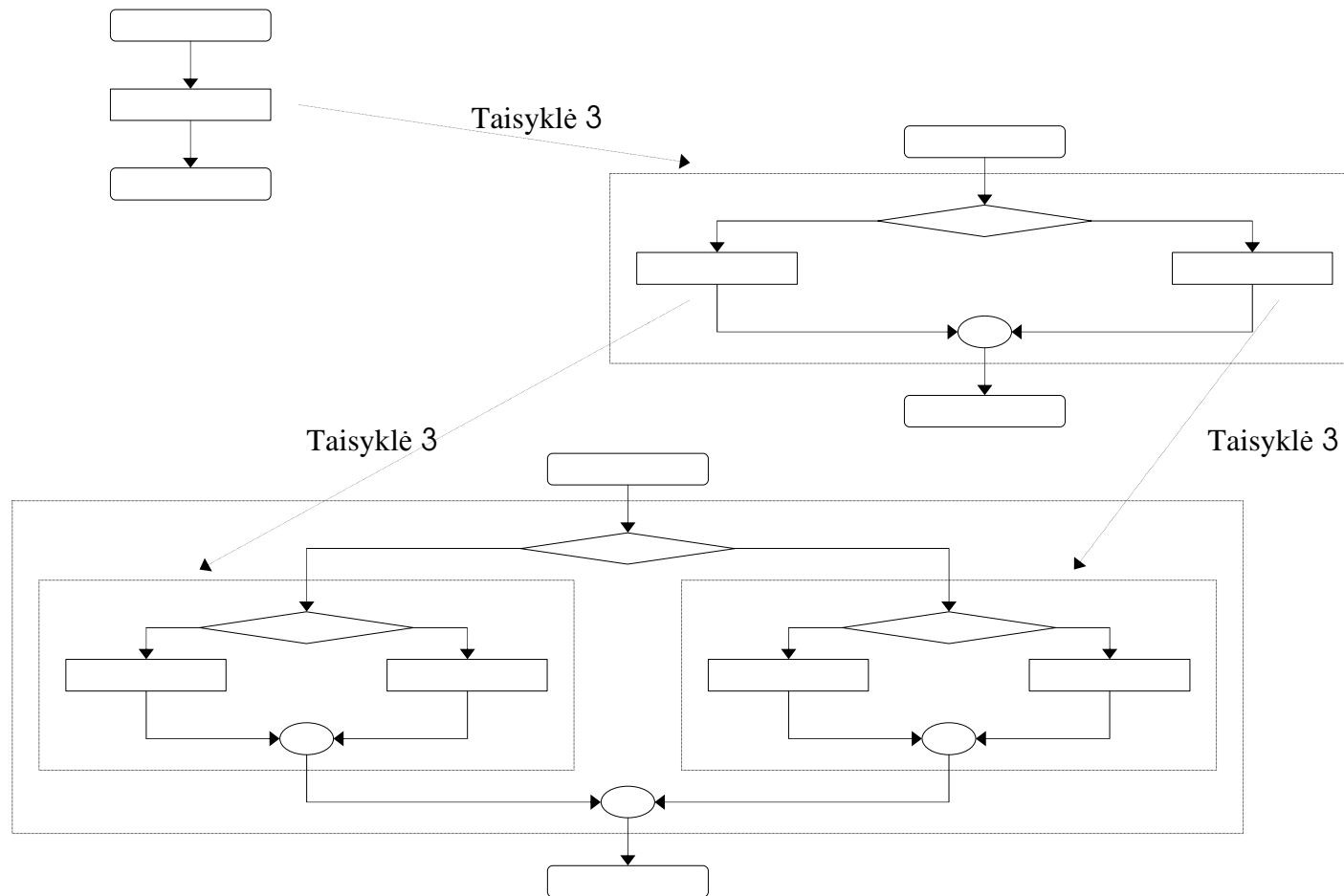
- **Struktūrinio programavimo požymiai:**

- Programos lengvai suprantamos, testuojamos ir modifikuojamos.

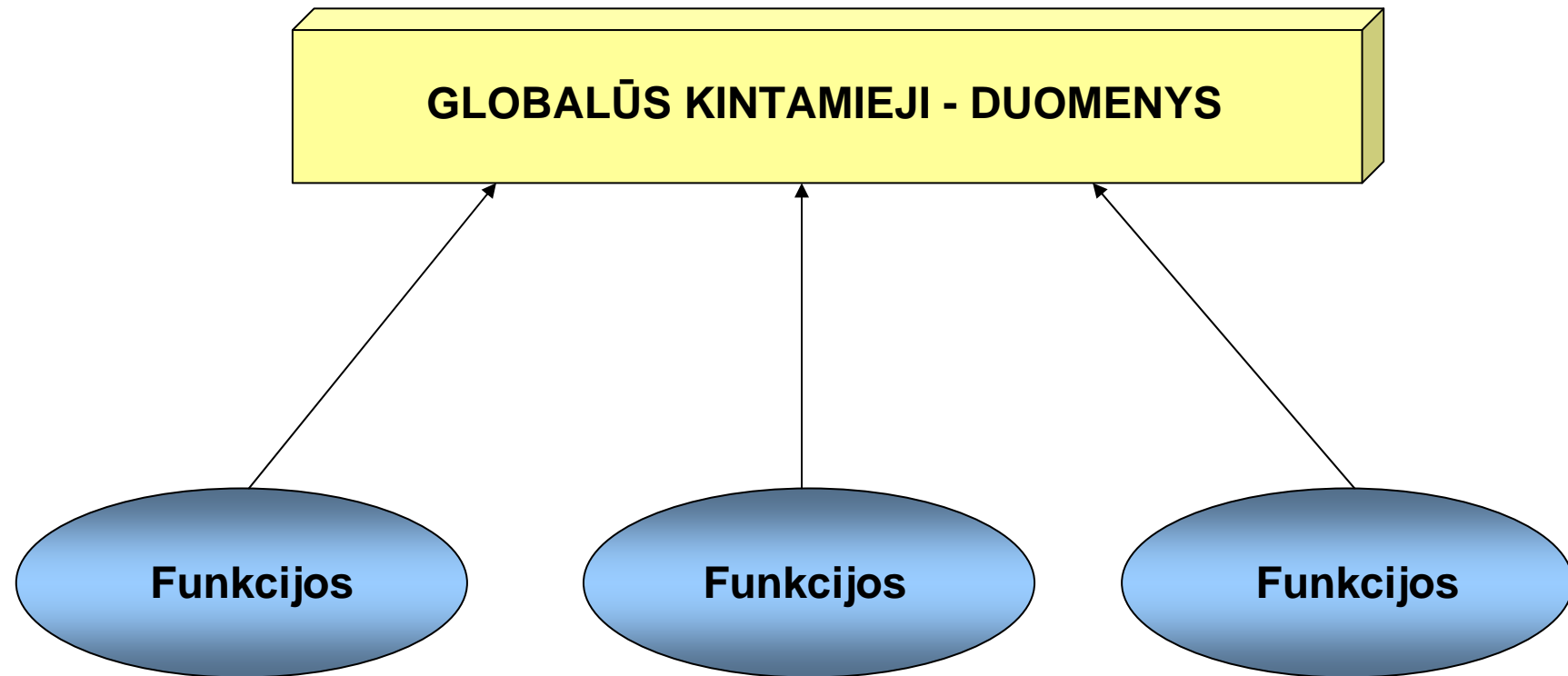
- **Struktūrinio programavimo taisyklės**

- Naudojamos single-entry/single-exit valdymo struktūros;
 - Taisyklės:
 - 1) Pradedama nuo “paprasčiausios” struktūrinės schemas;
 - 2) Bet kuris veiksmas, gali būti pakeistas dviejų veiksmų seka;
 - 3) Bet kuris veiksmas gali būti pakeistas valdymo struktūra (*if, if/else, switch, while, do/while, for*)
 - 4) 2 ir 3 punktas gali kartotis.
-

Struktūrinis programavimas



Struktūrinis programavimas



Objektiškai orientuotas programavimas

- Išradėjai **Ole-Johan Dhal** ir **Kristen Nygaard** iš Oslo Universiteto ir Norvegijos Skaičiavimo centro (Norsk Regnesentral). OOP koncepcija buvo pristatyta 1966.
- OOP esmė - sudaryti realaus pasaulio atspindį programinio kodo struktūroje.
- OOP – tai programavimo paradigma, kai programa modeliuojama, kaip tarpusavyje besirišančių objektų rinkinys.



OOP

```
importantStuff = new Object( );  
importantStuff.money = 500;  
importantStuff.papers = 96;  
importantStuff.annPassport = "er246vjl";  
importantStuff.bobPassport = "kl554mkt";
```

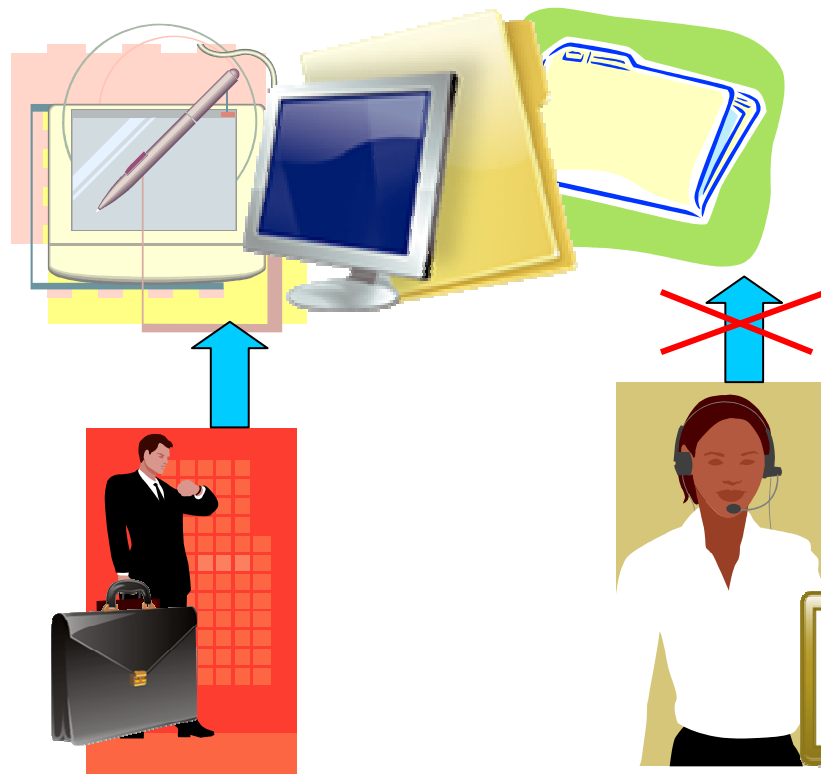
```
importantStuff.jewelryBox = new Object( );  
importantStuff.jewelryBox.necklace1 = "Pearl";  
importantStuff.jewelryBox.necklace2 = "Diamond";
```



Pamąstymui

Kaip suprogramuoti tokią situaciją?

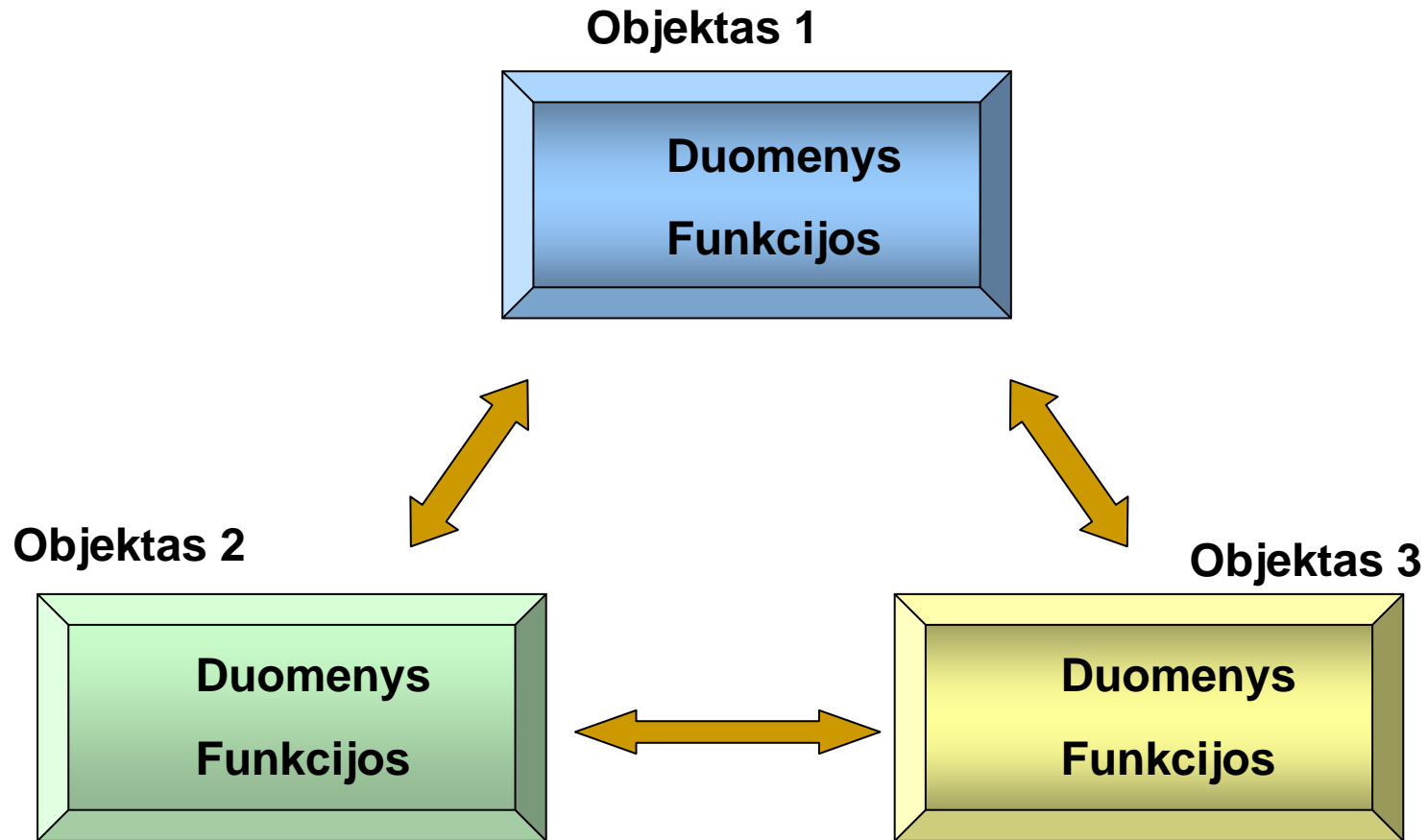
Įmonės buhalteriniai
duomenys



Direktorius

Vadybininkas

OOP



Klasė – tai šablonas, pagal kurį kuriamas objektas

C++

- ANSI/ISO standartizuota 1997m.
 - C++ - tai C kalbos pratęsimas, priskiriant ją OOP kalbų grupei, todėl išlieka C ir C++ suderinamumas.
 - C++ geriau nei C, nes:
 - Išsamesnis klaidų tikrinamas, lengvesnė klaidų paieška;
 - Nuorodų (*reference*) panaudojimas funkcijų argumentuose ir gražinamose reikšmėse yra patogesnis nei rodyklių (*pointer*).
 - Funkcijų perkrovimas (*overloading*) leidžia naudoti tuos pačius funkcijų pavadinimus skirtingoms funkcijoms.
 - Vardų erdvės (*namespace*) leidžia geriau kontroliuoti vardų naudojimą.
-

C++ privalumai prieš C

- ~10 % didesnis programų našumas nei C;
 - Programos (didelės) lengviau taisomos bei modifikuojamos;
 - Dėl klasių pritaikymo lengviau naudojamos bibliotekos;
 - Šablonai (template) automatiškai modifikuoja programinį kodą, taip palengvinami bibliotekų naudojimą;
 - Programinis kodas gali viršyti 50.000 eilučių;
-

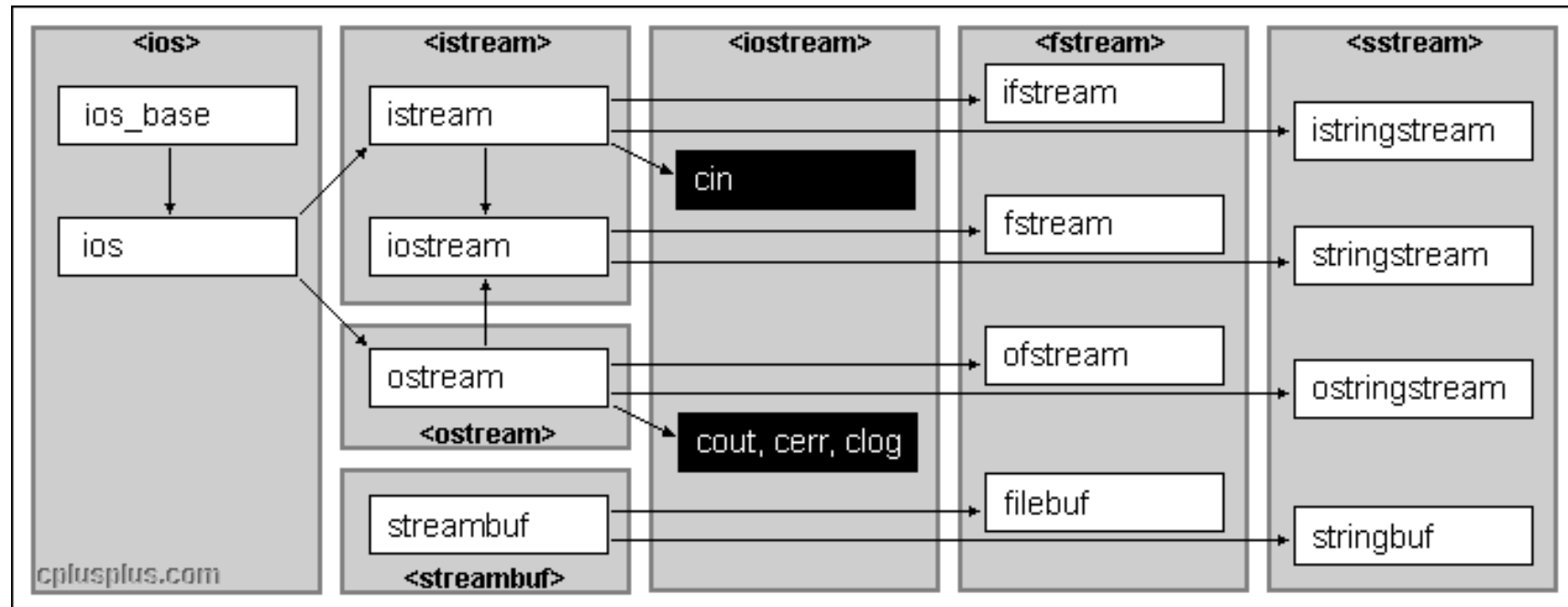
I/O biblioteka

- I/O srautas – tai loginis įrenginys informacijos gavimui iš vartotojo ir informacijos perdavimui vartotojui.
 - I/O srautas siejamas su fiziniais įrenginiais (ekranu, klaviatūra, pele, HDD) per C++ I/O sistemą t.y. I/O biblioteką, vadinamą **iostream**.
 - Tokia I/O sistema užtikrina vieningą tvarką informacijos įvedimui/išvedimui.
 - I/O biblioteka suskirstyta į 4 failus:
 - `iostream.h`
 - `fstream.h`
 - `strstream.h`
 - `iomanip.h`
-

C ir C++ srautų palyginimas

Srautas	C++	C
Įvedimo	cin	stdin
Išvedimo	cout	stdout
Klaidų	cerr	stderr
Registravimo	clog	stderr

Srautų hierarchija



Strautų bibliotekų failai

Antraštės failas	Aprašymas
iostream.h	Defines a hierarchy of classes for low-level (untyped character-level) I/O and high-level (typed) I/O. This includes the definition of the <i>ios</i> , <i>istream</i> , <i>ostream</i> , and <i>iostream</i> classes.
fstream.h	Derives a set of classes from those defined in <i>iostream.h</i> for file I/O. This includes the definition of the <i>ifstream</i> , <i>ofstream</i> , and <i>fstream</i> classes.
strstream.h	Derives a set of classes from those defined in <i>iostream.h</i> for I/O with respect to character arrays. This includes the definition of the <i>istrstream</i> , <i>ostrstream</i> , and <i>strstream</i> classes.
iomanip.h	Defines a set of manipulator which operate on streams to produce useful effects.

iostream bibliotekos panaudojimas

I/O rūšis	Įvedimas	Išvedimas	Įvedimas ir išvedimas
Standard I/O	istream	ostream	iostream
File I/O	ifstream	ofstream	fstream
Array of char I/O	istrstream	ostrstream	strstream

Apibrēžti standartiniai srautai

Srautas (stream)	Tipas	Buferi- zuotas	Aprašymas
cin	istream	Taip	Siejamas su standartiniu įvedimu (klaviatūra)
cout	ostream	Taip	Siejamas su standartiniu išvedimu (monitoriumi)
clog	ostream	Taip	Siejamas su standartiniu log'ų išvedimu (monitorius)
cerr	ostream	Ne	Siejamas su standartiniu klaidų išvedimu (monitorius)

I/O operatoriai

- Įvedimo >>
- Išvedimo <<

Pavyzdys

```
#include <iostream.h>
void main()
{
    float A;
    cout << "Įveskite A reikšmę: ";
    cin >> A;
    cout << "\nA=" << A << endl;
}
```

```
#include <iostream>
using namespace std;
void main()
{
    float A = 18.236;
    cout << "1. A=" << A << endl;
    cout << "2. A=" << A*2.0 <<
    endl;
}
```

Srauto formatavimo manipulatoriai

C++ srautai formatuojami: *manipulatoriais*, *žymėmis*, *funkcijomis*.

Manipulatoriai naudojami norint formatuoti išvedamą/įvedamą informaciją.

- **setw (int n)** n – lauko plotis simboliais išvedimui,
- **setprecision (int n)** n – skaitmenų skaičius, išvedant trupmeninį skaičių.
- **setfill(char c)** c - tarpo užpildymo simbolis.
- **endl** uždedamas naujos eilutės simbolis
- **ends** uždedamas eilutės pabaigos simbolis

Jei naudojami manipulatoriai su argumentais, įtraukiamas **iomanip.h failas**.

Jei *setw()* nurodyto lauko dydžio skaičiui nepakanka, manipulatorius ignoruojamas. *setw()* galioja artimiausiai išvedamai reikšmei, o *setprecision()* – iki naujo nurodymo.

Pavyzdys

```
#include <iostream>
#include <iomanip>
using namespace std;
void main()
{
float A = 18.2345f;
cout << setfill('0');
cout << "1. A=" << setw(9) << A << endl;    // 0018.2345
cout << "2. A=" << setprecision(3) << A << endl; // 18.2
cout << "3. A=" << setw(10) << setprecision(5) << A << endl;
//000018.234

A = 123.45678f;
cout << "4. A=" << A << endl; //123.46
A = 12345678f;
cout << "5. A=" << A << endl; //1.2346e+07
}
```

Srauto formatavimo funkcijos (width)

- Klasės `ostream` metodas `width(int)` skirtas nurodyti išvedamo sraute duomenų laukams. Galioja artimiausiam dydžiui. Tai `setw()` analogas

```
#include <iostream.h>
#include <iomanip.h>
void main() {
    for(int i=2; i<6; i++) {
        cout.width(3);
        cout << "i=";
        cout.width(i);
        cout << i << endl; }
}
```

Srauto formatavimo funkcijos (fill)

Metodas `fill(int)` leidžia pakeisti užpildymo simbolį norimu. Manipulatoriaus `setfill()` analogas.

```
#include <iomanip.h>
#include <iostream.h>
void main() {
    cout.fill('.');
    for(int i=2; i<6; i++) {
        cout << "i=";
        cout.width(i);
        cout << i << endl; }
}
```



Ekrane

```
i=.2
i=..3
i=...4
i=....5
```

Srauto formatavimo funkcijos (precision)

Metodas `precision(int)` leidžia nustatyti, kiek skaičių bus išvesta, išvedant trupmeninį skaičių.

Pagal nutylėjimą išvedami **6** skaičiai.

```
#include <iostream>
#include <cmath>
using namespace std;
void main() {
double x;
cout.precision(4);
cout.fill('0');
```

```
cout << " x  sqrt(x)  x^2\n\n";
for(x = 1.0; x <= 6.0; x++) {
    cout.width(7);
    cout << x << " ";
    cout.width(7);
    cout << sqrt(x) << " ";
    cout.width(7);
    cout << x*x << endl; }
}
```

Atsakymas

x	sqrt(x)	x^2
0000001	0000001	0000001
0000002	001.414	0000004
0000003	001.732	0000009
0000004	0000002	0000016
0000005	002.236	0000025
0000006	002.449	0000036

