7. SĄLYGINĖS VALDYMO STRUKTŪROS

Šiame skyriuje supažindinama su C++ kalboje naudojamais sąryšio (lyginimo) veiksmais, loginėmis operacijomis ir loginių sąlygų sudarymo principais. Taip pat skaitytojas susipažįsta su sąlyginėmis valdymo struktūromis: vienos ir dviejų alternatyvų sąlyginiais sakiniais, įterptaisiais sąlyginiais sakiniais bei daugiavariantinio pasirinkimo sakiniu switch. Skyriuje nagrinėjamos jų panaudojimo taisyklės ir pateikiami įvairūs pavyzdžiai. Skaitytojas išmoksta sudaryti šakotuosius uždavinių sprendimo algoritmus.

➢ **Mokymosi tikslai**

Mokydamasis ir baigęs šį skyrių skaitytojas turi:

- žinoti C++ kalboje naudojamus sąryšio(lyginimo) ir logines operacijas;
- mokėti naudotis sąryšio ir loginėmis operacijomis bei sudaryti logines sąlygas;
- žinoti sąlygines valdymo struktūras ir jų panaudojimo principus;
- mokėti naudotis vienos ir dviejų alternatyvų sąlyginiais sakiniais *if else*;
- mokėti naudotis įterptaisiais sakiniais;
- mokėti naudotis daugiavariantinio pasirinkimo sakiniu *switch*;
- gebėti sudaryti šakotuosius uždavinių sprendimo algoritmus.

7.1 Sąryšio veiksmai

Sąryšio (lyginimo) veiksmai (žr. 7.1 lentelė) naudojami sudarant logines išraiškas, vadinamąsias sąlygas.

Sąryšio veiksmai

7.1 lentelė

Veiksmas	Aprašymas
= =	Lygu
>	Daugiau
<	Mažiau
>=	Daugiau arba lygu
<=	Mažiau arba lygu
!=	Nelygu

Sąryšio veiksmų ženklai rašomi tarp dviejų operandų, sudarytų iš konstantų vardų, kintamųjų vardų arba išraiškų.

1 Pavyzdys

a = = d

b < 100

d >= sqrt(fabs(x))

Išraiškos, sudarytos naudojantis sąryšio veiksmais, palygina dviejų operandų reikšmes. Lyginimo rezultatas gali būti *Tiesa* arba *Netiesa (1 arba 0)*.

7.2 Loginės operacijos ir sąlygos

Išraiškos, skiriamos duomenims palyginti, jungiamos loginėmis operacijomis (žr. 7.1 pav.). Taip sudaromos sudėtingos lyginimo išraiškos, vadinamos loginėmis sąlygomis.

Loginės operacijos

Operacija	aprašymas
&&	ir
||	arba
!	ne

Loginė daugyba (konjunkcija) && ir Loginė sudėtis (disjunkcija) ||

I1	*I2*	*&&*	*||*
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Loginis neigimas (inversija) ! (ne)

I1	!
0	1
1	0

7.1 pav. Loginės operacijos

2 Pavyzdys

((a<b) && (c>d))

((pard > 5000) || (val > 81))

(! (apyvarta < 2000))

(apyvarta >= 250)

((apyvarta < (vid * 2)) && (darb > (10* d_sk)))

1. Patariama *if else* sakiniuose rašyti ne daugiau kaip du lyginimo veiksmus.
2. Nepatariama į *if else* sakinio sąlygą rašyti priskyrimo sakinio. Gali atsitikti taip, kad priskyrimo sakinys nebus atliktas arba iškreips sąlygą.
3. Norint išvengti neapibrėžtumų, patariama naudotis skliausteliais, net jeigu ir be jų, pagal pirmumo lygį, sąlyga būtų patikrinta teisingai.

Aritmetinių, sąryšio veiksmų ir loginių operacijų atlikimo pirmumo lygiai pateikiami 7.2 lentelėje. Pirma atliekami aritmetiniai veiksmai, po to – sąryšio veiksmai ir tik po jų loginės operacijos.

Aritmetinių, sąryšio veiksmų ir loginių operacijų atlikimo pirmumo lygiai

7.2 lentelė

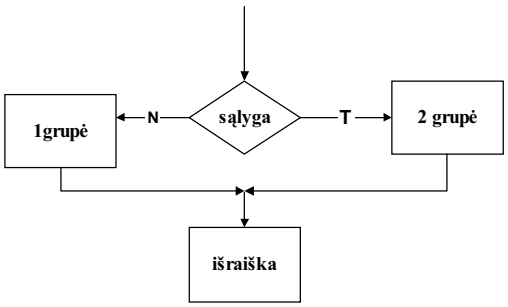
Prioritetas	Operacijos
1	( ),
2	Ženklas (+, -), !
2	Priešdėliniai ++ ir - -
3	*, /, %
4	+, -
5	Priesaginiai ++ ir - -
6	<, <=, >, >=
7	= =, !=
8	&&, ||

7.3 Sąlyginė valdymo struktūra if else

Sąlyginėje valdymo struktūroje *if else* tikrinama sąlyga (loginė išraiška) ir, atsižvelgiant į sąlygos tikrinimo rezultatą, atliekami numatyti veiksmai:

a) Jeigu tikrinama sąlyga tenkinama , atliekama sakinių grupė, parašyta už sąlygos.
b) Jeigu tikrinama sąlyga netenkinama, atliekama sakinių grupė, parašyta šakoje *else*.

Grafinis dviejų alternatyvų sąlyginio sakinio vaizdas pateikiamas 7.2 paveiksle.



7.2 pav. Dviejų alternatyvų sąlyginis sakinys

Dviejų alternatyvų sąlyginės valdymo struktūros if else sintaksė:

```
if (sąlyga) {
        1 sakinių grupė
        }
else  {
        2 sakinių grupė
}
```

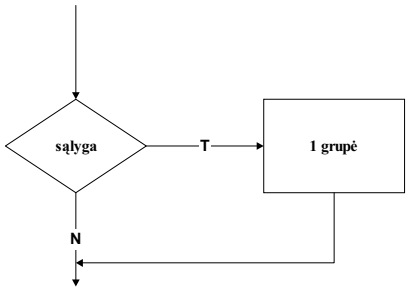
Naudojantis sąlyginėmis valdymo struktūromis patariama laikytis šių taisyklių:

1. Naudojami sudėtiniai operatoriai - sakinių grupės, aprėmintos figūriniais skliausteliais. Jeigu grupė sudaryta iš vieno sakinio, figūrinių skliaustelių rašyti nereikia.
2. Atskira sakinių grupė (blokas) aprėminta figūriniais skliausteliais, pastumiamas į dešinę.
3. Sąlygos pabaigoje kabliataškio rašyti nereikia.

1 Pratimas

```
// Programa įveda skaičių ir paaiškina - teigiamas ar neigiamas skaičius buvo įvestas
#include <iostream.h>
main()
{
        int sk;
        cout << "Įveskite skaičių : ";
        cin >> sk;
        if (sk > 0)
                cout << "\n Daugiau už nulį";
        else
                cout << "\n Mažiau už nulį arba lygu nuliui";
return 0;
}
```

Jeigu sąlyginėje valdymo struktūroje šaka *else* praleidžiama, tokia struktūra vadinama *vienos alternatyvos sąlyginiu sakiniu*. Grafinis vienos alternatyvos sąlyginio sakinio vaizdas pateikiamas 7.3 paveiksle.



7.3 pav. Vienos alternatyvos sąlyginis sakinys

Vienos alternatyvos sąlyginės valdymo struktūros sintaksė:

if (sąlyga)
{
1 sakinių grupė
}

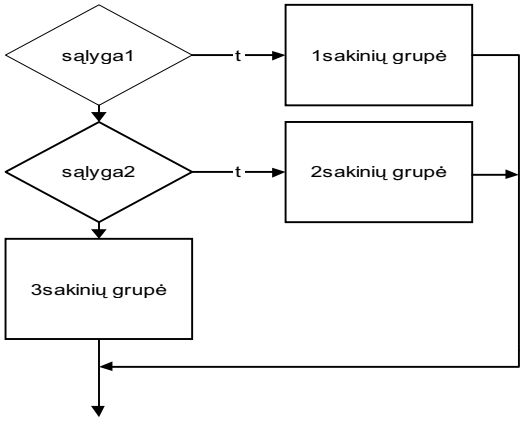
2 Pratimas

/* Programa skaičiuoja darbo užmokestį. Jei darbuotojas dirbo 40 ir mažiau valandų, skaičiuojamas atlyginimas - vienos valandos įkainis dauginamas iš dirbtų valandų skaičiaus. Jeigu dirbo nuo 40 iki 50 valandų - 1,5 karto didesnis valandos įkainis. Jeigu dirbo daugiau nei 50 valandų, skaičuojamas du kartus didesnis atlyginimą už kiekvieną valandą virš 50 */

```
#include <iostream.h>
#include <iomanip.h>
main()
{
        int val;
        float dt, ht, rp, ik,atlyg;
        cout <<"\n\n Kiek valandų dirbta ?"; // įvedami duomenys
        cin >>val;
        cout <<"\n Koks valandos įkainis?";
        cin >>ik;
        if (val> 50) {
                dt=2.0*ik *(val -50); //skaičiuojamas atlyginimas
                ht=1.5*ik *10;          }
        else    dt=0;
        if (val<=40) {
                rp=val*ik;
                ht=0; }
        else    {
                rp=40 *ik;
                ht=(val-40)*1.5*ik; }
atlyg = rp+ht+dt;
cout<<"Atlyginimas "<<setprecision(2)<<setiosflags(ios::fixed)<< atlyg <<' \n';
return 0;}
```

7.4 Įterptieji sąlyginiai sakiniai *if else*

Įterptieji *f else* sakiniai skirti daugiavariantiškai pasirinkti vien ą alternatyvą iš keletos galimų. Grafinis įterptųjų sąlyginių sakinių vaizdas pateikiamas 7.4 paveiksle.



7.4 pav. Įterptieji sąlyginiai sakiniai if else

Įterptųjų sąlyginių sakinių sintaksė:

```
if (sąlyga 1) {
        1 sakinių grupė
        }
else    if (sąlyga 2) {
                2 sakinių grupė
                }
        else    {
                3 sakinių grupė
                }
```

Programuojant daugiavariantinį pasirinkimą, dažniausiai tikrinama sąlygų vienetu mažiau už galimų variantų skaičių.

3 Pratimas

```
// programa įveda aritmetinio veiksmo ženklą ir nurodytą veiksmą atlieka
#include <iostream.h>
        char op;
        double x,y,z;
        cout << "Įveskite  x,  aritmetinį veiksmą  ir  y \n";
        cin >> x >> op >>y;
        if  ( op =='+')  z=x+y;
        else if  ( op == '-') z=x-y;
        else if  (op  == '/' && y!=0) z=x/y;
        else if (op == '*') z=x*y;
cout  <<  '\n'  <<  z;
return 0;
}
```

7.5 Sąlygos veiksmas *? :*

Dviejų alternatyvų sąlyginė valdymo struktūra *if else* gali būti keičiama sąlygos veiksmu. Sąlygos veiksmas yra trinaris – veiksmai atliekami su trimis operandais.

Sąlygos sakinio sintaksė:

sąlyga ? 1 išraiška : 2 išraiška

sąlyga - bet kokia loginė išraiška, įgijanti reikšmę *Tiesa (1)* arba *Netiesa (0).*

1 išraiška - atliekama, jeigu sąlyga tenkinama.

2 išraiška - atliekama, jeigu sąlyga netenkinama.

4 Pavyzdys

a) `( p > 8000 ) ? m=500 : m=0;`

b)
```
if (a>b)                    perrašomas į    ( a>b ) ? (ats = 10 ) : (ats=25);
ats=10;
else     ats = 25;
```

c) `ats = ( a > b ) ? ( 10 ) : ( 25 );`

d) `min = ( var1 < var2)  ?  var1  :  var2 ;`

e) `max = ( var1 > var2)  ?  var1  :  var2 ;`

f) `koks = ( testvar < 0 )  ?  -1  :  1 ;`

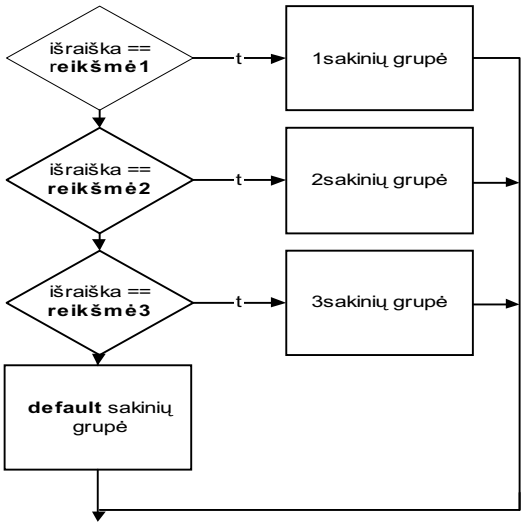
Jeigu reikalinga paprasta sakinio *if else* forma, patogiau naudotis sąlygos veiksmu *? :*

Naudojantis sąlygos veiksmu, patariama laikytis šių taisyklių:

1. Sąlygų suskliausti nebūtina, nors skliausteliai daro programą lengviau skaitomą.
2. Jeigu abi išraiškos kairėje lygybės ženklo pusėje turi tą patį kintamąjį, tai jį galima iškelti prieš sąlygą.

7.6 Daugiavariantinio pasirinkimo struktūra switch

Daugiavariantinių alternatyvų aprašymo struktūra *switch* padaro paprastesniu įterptąjį sąlyginį sakinį *if else*. Sakinys *switch* tinka, kai reikia pasirinkti vieną variantą iš daugelio. *switch* struktūra naudojama tuomet, kai pasirinkimas vykdomas pagal vieno kintamojo, vienos konstantos arba paprastos išraiškos reikšmę. Kad programa dirbtų greičiau, dažniausiai atliekami variantai, surašomi sąrašo pradžioje. Grafinis daugiavariantinių alternatyvų *switch* struktūros aprašo vaizdas pateikiamas 7.6 paveiksle.



7.6 pav. Daugiavariantinių alternatyvų switch struktūros aprašo vaizdas

Daugiavariantinio pasirinkimo struktūros sintaksė:

```
switch (išraiška)
{
        case ( reikšmė 1):
                1 sakinių grupė;
                break;
        case (reikšmė 2):
                2 sakinių grupė;
                break;
        …
        case (reikšmė n):
                n sakinių grupė;
                break;
        default:
                sakinių grupė - alternatyva kitoms reikšmėms;
                break;
}
```

išraiška	- *char* arba *int* tipo konstanta arba kintamasis;
1 išraiška, 2 išraiška	- *int* arba *char* tipo konstantos;
case	- variantų skaičius priklauso nuo sprendžiamo uždavinio;
default	- alternatyvi sakinių grupė. Ši dalis nebūtina. Taip pat nebūtinai rašoma pabaigoje.

Jeigu *išraiškos* rezultatas sutampa su *1 išraiškos* rezultatu, atliekami pirmojo varianto (*case*) sakiniai. Jeigu rezultatas sutampa su 2išraiškos rezultatu , atliekami antrojo varianto sakiniai. Jeigu nė vieno varianto išraiškos rezultatas nesutampa su *switch* išraiškos rezultatau, atliekama *default* sakinių grupė.

Patariama laikytis šių naudojimosi daugiavariantinio pasirinkimo *switch* struktūra taisyklių:

1. *case* išraiškos nebūtinai rašomos skliausteliuose. Tačiau jeigu taip parašyta, programa tampa lengviau skaitoma.
2. Kiekvieno *case* bloko pabaigoje patariama rašyti sakin į *break* tam, kad būtų baigiama tikrinti.

4 Pratimas

```
#include <iostream.h>
main()
{
        int sk, i;
        cout << "Įveskite skaičių :";
        cin >>sk;
        switch (sk)
        {
                case 1:
                        cout << (x/100);
                        break;
                case 2:
                        cout << (x*100);
                        break;
                default:
                for (i=1  i<=sk; i++)
                cout<<'*';
        }
return 0;
}
```

Klausimai

1. Kaip žymimos loginės operacinos *ir, arba, ne*?
2. Nustatykite šių sąlygų rezultatus?
   int i=12, j=10, k=5;
   i && j
   (12 - i) || k
   (j != k) && (i != k)
3. Kodėl sąlygos veiksmas vadinamas trinariu?
4. Perrašykite šį sąlygos veiksmą kaip sąlyginį sakinį?
   ats = (a== b) ? c+2 : c+3;
5. Kokie veiksmai palygina kintamųjų reikšmes?
6. Ar teisingi šie reiškiniai?
   4 = = 4
   165 >= 165
   0 != 25

Užduotys

1. Parašykite programą, kuri prašytų įvesti amžių ir rodytų ekrane įvestą reikšmę. Naudokitės tik vienu *cout* sakiniu, tačiau jame turi būti salyginis veiksmas. Jei gu įvesta reikšmė didesnė už 21, rodoma pastaba "Jūs jau didelis", jei ne - "Jūs dar mažas".
2. Parašykite programą, kurios pradžioje kintamajam *x* priskiriama reikšmė. Klaviatūra įvedami penki skaičiai, kurių reikšmės saugomos skirtinguose kintamuosiuose. Naudokitės vienu sakiniu *cin*. Tikrinkite – kuris iš įvestų skaičių sutampa su pirmuoju. Jei gu sutampa, ekrane rodoma pastaba.
3. Parašykite programą, kuri skaičiuotų mokesčius. Šeima nemoka mokesčių, jei jos pajamos neviršija 1000Lt. Jei šeimos pajamos tarp 1000Lt ir 1200Lt, šeima moka 10% mokesčius. Jei pajamos tarp 1200Lt ir 1500Lt, šeima moka 20 % mokesčius. Jei pajamos viršija 1500Lt, šeima moka 30 % mokesčius.
4. Parašykite programą, kuri įvestų du skaičius ir spausdintų komentarą, koks yra santykis tarp dviejų skaičių (pvz., penki mažiau už septynis). Išnagrinėkite kuo daugiau variantų.

Santrauka

Šiame skyriuje susipažinote su C++ kalboje naudojamais sąryšio (lyginimo) veiksmais, loginėmis operacijomis ir loginių sąlygų sudarymo principais. Taip pat susipažinote su sąlyginėmis valdymo struktūromis: vienos ir dviejų alternatyvų sąlyginiais sakiniais, įterptaisiais sąlyginiais sakiniais bei daugiavariantinio pasirinkimo sakiniu *switch*. Skyriuje išagrinėjote jų panaudojimo taisykles ir įvairius jų panaudojimo pavyzdžius. Be to, išmokote sudaryti šakotuosius uždavinių sprendimo algoritmus.

Informacijos šaltiniai