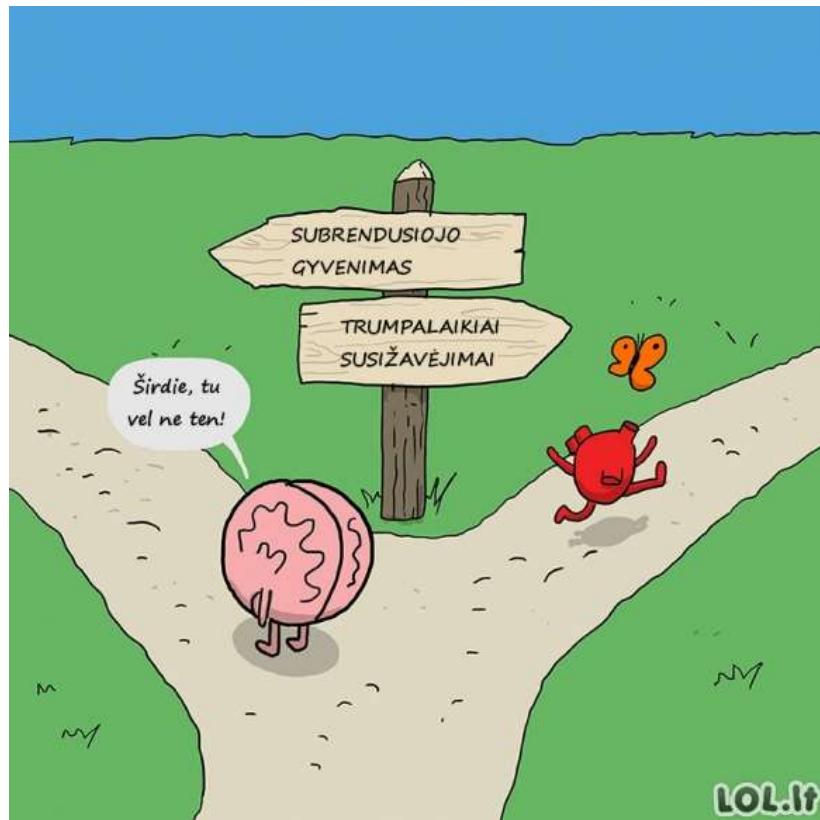


# Sąlygos sakiny

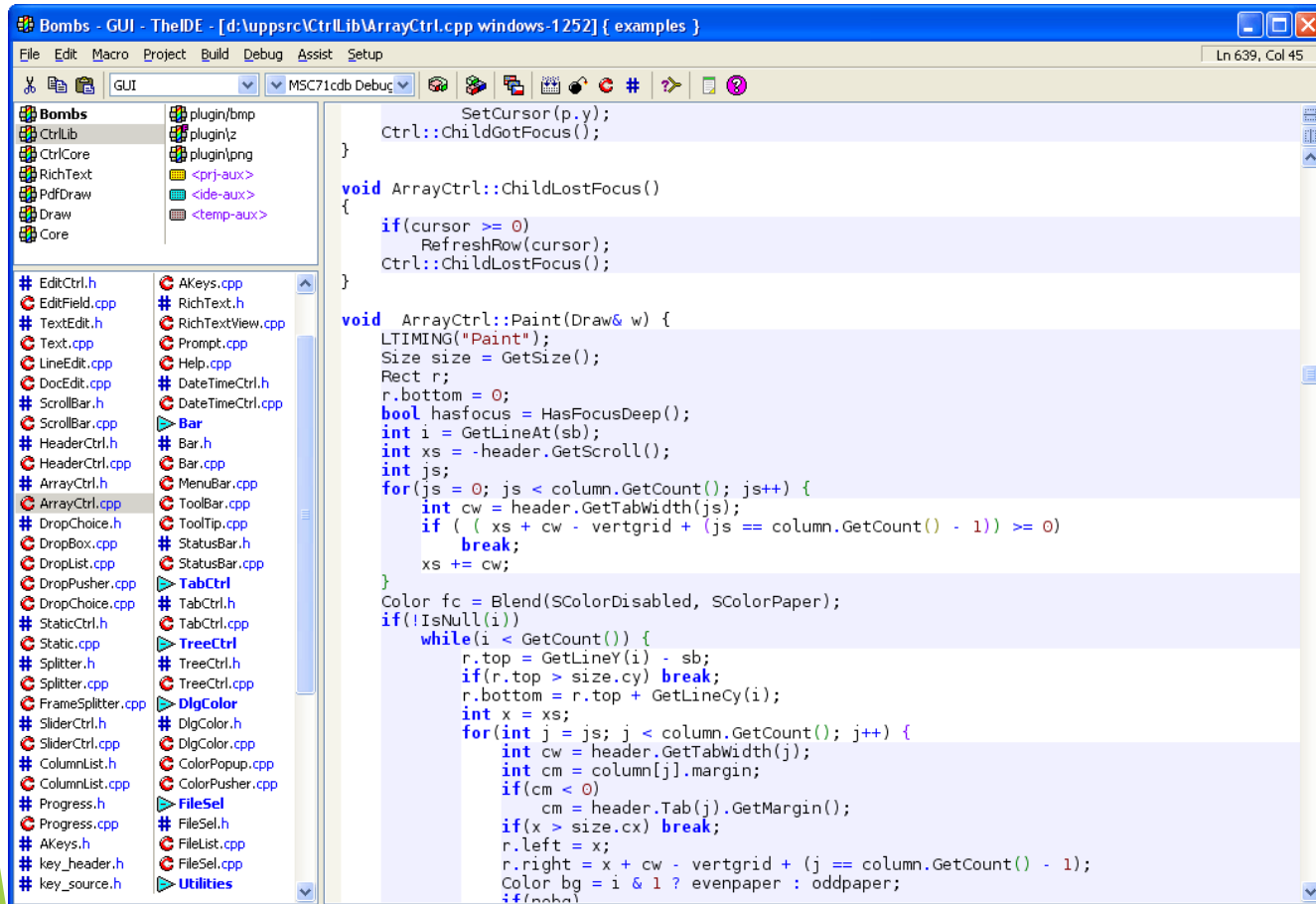
Parengė ITMM Artūras Šakalys

Gyvenime dažnai tenka rinktis iš kelių variantų

Kartais variantai būna aiškūs, o kartais ir ne



# Programavime visos didelės programos turi sąlygos sakinių



The screenshot shows a code editor window titled "Bombs - GUI - TheIDE - [d:\uppsrc\CtrlLib\ArrayCtrl.cpp windows-1252] { examples }". The code is in C++ and shows several methods. The `ArrayCtrl::ChildLostFocus()` method contains an `if` statement that is highlighted in blue. The `ArrayCtrl::Paint()` method also contains several `if` statements, including one that is also highlighted in blue. The code is as follows:

```
SetCursor(p.y);
Ctrl::ChildGotFocus();
}

void ArrayCtrl::ChildLostFocus()
{
    if(cursor >= 0)
        RefreshRow(cursor);
    Ctrl::ChildLostFocus();
}

void ArrayCtrl::Paint(Draw& w) {
    LTIMING("Paint");
    Size size = GetSize();
    Rect r;
    r.bottom = 0;
    bool hasfocus = HasFocusDeep();
    int i = GetLineAt(sb);
    int xs = -header.GetScroll();
    int js;
    for(js = 0; js < column.GetCount(); js++) {
        int cw = header.GetTabwidth(js);
        if ( ( xs + cw - vertgrid + (js == column.GetCount() - 1) ) >= 0)
            break;
        xs += cw;
    }
    Color fc = Blend(SColorDisabled, SColorPaper);
    if(!IsNull(i))
        while(i < GetCount()) {
            r.top = GetLineY(i) - sb;
            if(r.top > size.cy) break;
            r.bottom = r.top + GetLineCy(i);
            int x = xs;
            for(int j = js; j < column.GetCount(); j++) {
                int cw = header.GetTabwidth(j);
                int cm = column[j].margin;
                if(cm < 0)
                    cm = header.Tab(j).GetMargin();
                if(x > size.cx) break;
                r.left = x;
                r.right = x + cw - vertgrid + (j == column.GetCount() - 1);
                Color bg = i & 1 ? evenpaper : oddpaper;
                if(noba
```

▶ Jos prasideda žodeliu IF

## C++ kalboje

▶ if (a>b)

▶ {

▶

```
cout<<„teisingai“<<  
endl;
```

▶ } else

▶ {

▶

```
cout<<„Neteisingai“  
<<endl;
```

▶ C++ kalbos sąlygos sakinio struktūra

## Sąlygos sakinio variantas

- ▶ Jei  $2 > 1$  tai True, priešingai False
- ▶ Jei  $a > b$  tai True, priešingai False
- ▶ Galimybės
- ▶ Jei  $10 > 1$  tai Puiku,
- ▶     Jei  $9 > 1$  tai l.gerai
- ▶     Jei  $8 > 1$  tai gerai
- ▶     Jei  $7 > 1$  patenkinamai
- ▶     .....
- ▶     Jei  $2 > 1$  netenkina
- ▶     Kitaip  $1 > 1$  tai False blogai

- ▶ Sąlyga - tai loginis reiškinys; kai jo reikšmė TRUE, sakoma, sąlyga tenkinama, kai False - netenkinama.
- ▶ Sudėtiniai sakiniai su IF

# Operatoriai

## Aritmetiniai operatoriai

Operacija	Išraiška	Rezultatas	Pastabos
Sudėtis	$a + b$	$26 + 17$	43
Atimtis	$a - b$	$26 - 17$	9
Daugyba	$a * b$	$26 \cdot 17$	442
Dalyba	$a / b$	$26 : 17$ $26,0 : 17$	1 $\approx 1,53$ Sveikų skaičių dalybos rezultatas – sveikoji dalmens dalis
Liekana	$a \% b$	$26 \bmod 17$	9 Liekanos operatorius tinka tik sveikojo tipo reikšmėms
<b>Pastaba:</b> sveikų <i>neigiamų</i> skaičių dalybos ir liekanos rezultatai priklauso nuo kompiuterio architektūros			
Neigimas	$-a$	-26	-26
Teigimas	$+a$	+26	26 Nieko neatlieka; pridėtas dėl simetrijos su -a
Inkrementas	$++a$	[a = 26] $a \leftarrow a + 1$ ; a	[a = 27] 27 Operatorius padidina a reikšmę vienetu ir <i>po to</i> grąžina naująją reikšmę
	$a++$	[a = 26] a; $a \leftarrow a + 1$	26 [a = 27] Operatorius grąžina a ir <i>po to</i> padidina a reikšmę vienetu
Dekrementas	$--a$	[a = 26] $a \leftarrow a - 1$ ; a	[a = 25] 25 Operatorius sumažina a reikšmę vienetu ir <i>po to</i> grąžina naująją reikšmę
	$a--$	[a = 26] a; $a \leftarrow a - 1$	26 [a = 25] Operatorius grąžina a ir <i>po to</i> sumažina a reikšmę vienetu
<b>Pastaba:</b> inkremento ir dekremento operatoriai veikia tik su <i>kintamaisiais</i> ; $26++$ ir $--(a+b)$ negalimi			

# Palyginimo ir loginiai operatoriai

Operacija	Išraiška	Rezultatas	Pastabos	
Daugiau	$a > b$	$26 > 17$	1	
Daugiau ar lygu	$a \geq b$	$26 \geq 17$	1	
Mažiau	$a < b$	$26 < 17$	0	
Mažiau ar lygu	$a \leq b$	$26 \leq 17$	0	
Lygu	$a == b$	$26 = 17$	0	
Nelygu	$a != b$	$26 \neq 17$	1	
Loginis NE	$!a$	NOT 26	0	Rezultatas lygus 0, jei a nelygus 0
Loginis IR	$a \&\& b$	26 AND 17	1	Rezultatas lygus 0, jei a arba b lygūs 0
Loginis ARBA	$a    b$	26 OR 17	1	Rezultatas lygus 0, jei a ir b lygūs 0

**Pastaba:** operatoriais  $\&\&$  ir  $||$  sujungtos išraiškos skaitomos iš kairės į dešinę *tik* tol, kol randamas išraiškos rezultatas. Pavyzdžiui, išraiškos  $(a < 5) \&\& (a != b++)$  antroji dalis nebus vykdoma (ir b nebus padidintas), jei  $a \geq 5$ .

# Priskyrimo operatoriai

## Priskyrimo operatoriai

Priskyrimo operatoriai atlieka nurodytus veiksmus ir įrašo rezultatą į operatoriaus kairėje esantį *kintamąjį* ( $5 = a$  ir  $(a \& b) *= c$  negalimi). Šis apribojimas kartais naudojamas siekiant išvengti dažnos rašybos klaidos, kai palyginimo ( $==$ ) vietoje įrašomas priskyrimas ( $=$ ): lyginant rašoma  $5 == a$  ir  $(a \& b) == c$ . Jungtiniams priskyrimo operatoriams ( $+=, \dots$ ) galioja tos pačios taisyklės, kaip ir atskiriems operatoriams. Visų priskyrimo operatorių rezultatas yra toks pats, kaip įrašoma reikšmė: jei  $c = (a = 5)$ , tai  $c$  lygus 5.

Operacija	Išraiška	[a = 26]	Rezultatas	Pastabos
Priskyrimas	$a = b$	$a \leftarrow 17$	[a = 17]	
Sudėtis – priskyrimas	$a += b$	$a \leftarrow a + 17$	[a = 43]	
Atimtis – priskyrimas	$a -= b$	$a \leftarrow a - 17$	[a = 9]	
Daugyba – priskyrimas	$a *= b$	$a \leftarrow a \cdot 17$	[a = 442]	
Dalyba – priskyrimas	$a /= b$	$a \leftarrow a : 17$	[a = 1] [a ≈ 1,53]	Sveikų skaičių dalyba Slankaus kablelio skaičių dalyba
Liekana – priskyrimas	$a \% = b$	$a \leftarrow a \bmod 17$	[a = 9]	Tik sveikojo tipo reikšmėms
Bitinis IR – priskyrimas	$a \& = b$	$a \leftarrow a \cap 17$	[a = 16]	
Bitinis ARBA – priskyrimas	$a  = b$	$a \leftarrow a \cup 17$	[a = 27]	
Bitinis XOR – priskyrimas	$a ^= b$	$a \leftarrow a \times 17$	[a = 5]	
Postūmis kairėn – priskyrimas	$a \ll = b$	$a \leftarrow a \ll 2$	[a = 104]	
Postūmis dešinėn – priskyrimas	$a \gg = b$	$a \leftarrow a \gg 2$	[a = 6]	



# C++ operatoriai

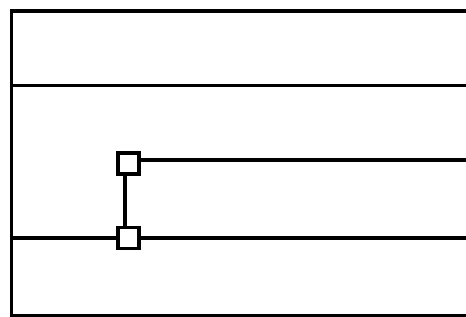
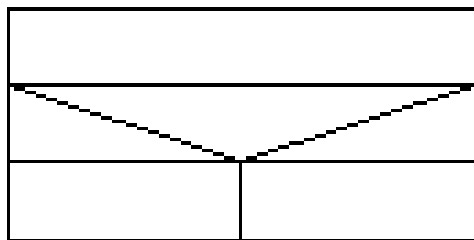
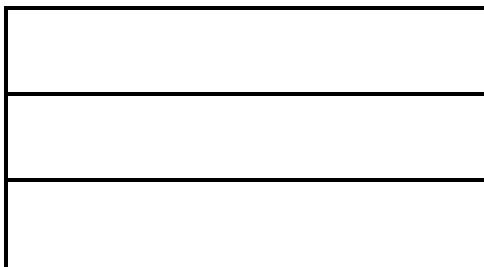
C – operatoriai

## C++ operatoriai

Šie operatoriai C kalboje nenaudojami; jie susiję su programavimo elementais, kurių C nepalaiko.

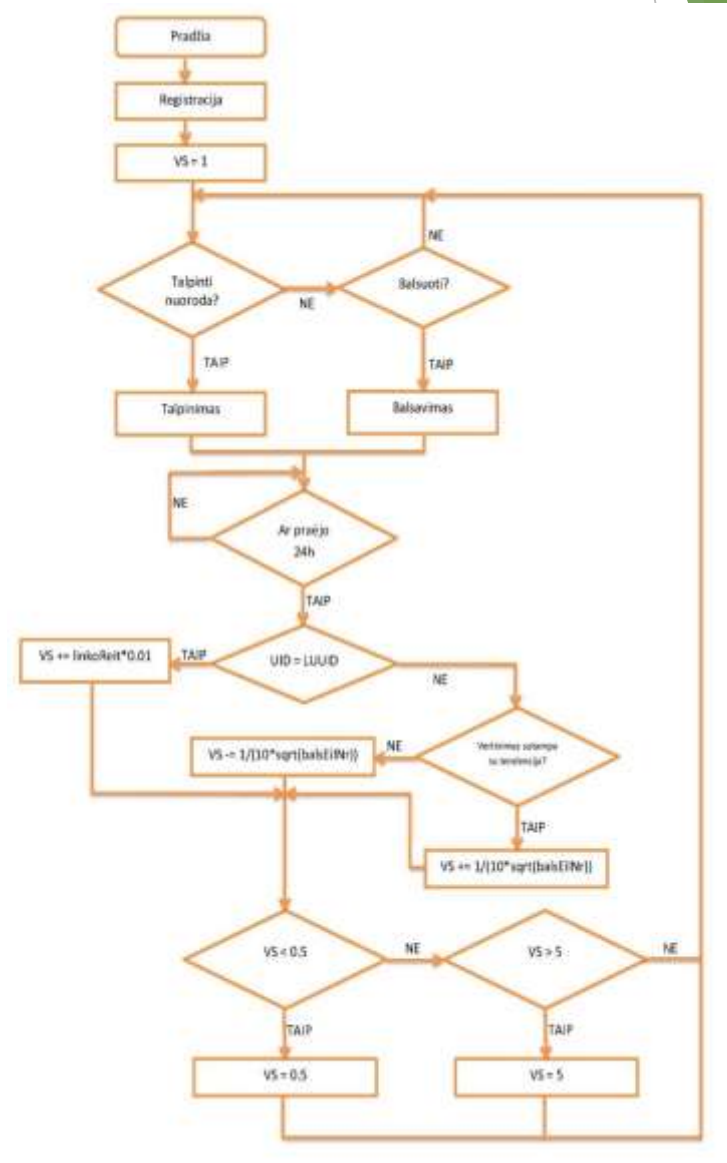
Operacija	Išraiška	Aprašymas
Galiojimo sritis (kontekstas)	<code>a :: b</code>	Nurodo, kuriai sričiai priklauso objekto <code>b</code> vardas. <code>a</code> gali būti klasės ar vardyno pavadinimas.
Nuoroda į objektą	<code>&amp;a</code>	Bet kokie veiksmai su nuoroda į <code>a</code> keis objektą <code>a</code> . <code>&amp;a</code> gali būti funkcijos parametras ir pan.
Atminties išskyrimas	<code>new tipas</code> <code>new tipas [n]</code>	Išskiriamas atminties blokas nurodyto <i>tipo</i> objektui/ <code>n</code> objektų masyvui ir gražinama rodyklė į jį.
Išskirtos atminties panaikinimas	<code>delete a</code> <code>delete [] a</code>	Panaikinamas (anksčiau išskirtas) atminties blokas, į kurį rodo rodyklė <code>a</code> .
Rodyklė į klasės objekto narį	<code>a::*b</code>	<code>b</code> tipas – rodyklė į klasės <code>a</code> objekto narį.
Rodomas klasės/struktūros narys	<code>a.*b</code> <code>a-&gt;*b</code>	Rezultatas – objekto <code>a</code> nario, į kurį rodo rodyklė <code>b</code> , reikšmė. Antruoju atveju <code>a</code> – rodyklė.
Objekto tipas	<code>typeid (a)</code> <code>typeid (tipas)</code>	Objekto <code>a</code> arba klasės <i>tipas</i> tikrasis tipas duotu programos vykdymo momentu.
Tipo pakeitimai	<code>static_cast&lt;tipas&gt;(a)</code> <code>dynamic_cast&lt;tipas&gt;(a)</code> <code>const_cast&lt;tipas&gt;(a)</code> <code>reinterpret_cast&lt;tipas&gt;(a)</code>	Statinis tipo pakeitimas Dinaminis tipo pakeitimas <code>const</code> ir <code>volatile</code> šalinimas Nesuderinamo tipo vertimas

# Struktūrograma



- ▶ Grafinis algoritmu vaizdavimo būdas. Nuo blokinės schemos skiriasi tuo, kad labiau struktūrizuoja algoritmą. Pagrindiniai algoritmų veiksmai išreiškiami tokiomis struktūromis:
  - nuosekli veiksmų seka:
  - veiksmų pasirinkimas (šakojimasis):
  - veiksmų kartojimas

# Struktūrogramas pavyzdys



# Išspręskime uždavinį (1)

- ▶ Visi kartu išspręskime uždavinį.
- ▶ Uždavinio sąlyga.
- ▶ Du draugai Jonas ir Povilas žaidžia šachmatais. Jie surinko skirtingą taškų skaičių: Jonas -  $n$  taškų, Povilas -  $m$  taškų. Parašykite programą, kuri ekrane parodytų rezultata: „Turnyrą laimėjo Jonas“, jei daugiau taškų surinko Jonas, arba „Turnyrą laimėjo Povilas“, jei daugiau taškų surinko Povilas.

## Išspręskime uždavinį (2)

- ▶ *Pasitikrinkite: kai  $n = 3$ ,  $m = 2$ , tuomet ekrane turi būti rodomas pranešimas: Turnyrą laimėjo Jonas; kai  $n = 2$ ,  $m = 3$ , tuomet ekrane turi būti rodomas pranešimas: Turnyrą laimėjo Povilas.*

# Uždavinio sprendimo kodas

```
int main()  
{  
    int n,m;  
    cout << "Kiek taškų surinko Jonas?" << endl;  
    cin >>n;  
    cout << "Kiek taškų surinko Povilas?" << endl;  
    cin >>m;  
    if (n>m) {  
        cout<<"Turnyra laimėjo Jonas"<<endl;  
    }  
    else {  
        cout<<"Turnyra laimėjo Povilas"<<endl;  
    }  
  
    return 0;  
}
```

# Spręsdami uždavinį užrašėme sąlyginį sakinį:

```
▶ if (n>m) {  
▶     cout<<"Turnyrą laimėjo Jonas"<<endl;  
▶ }  
▶ else {  
▶     cout<<"Turnyrą laimėjo Povilas"<<endl;  
▶ }
```

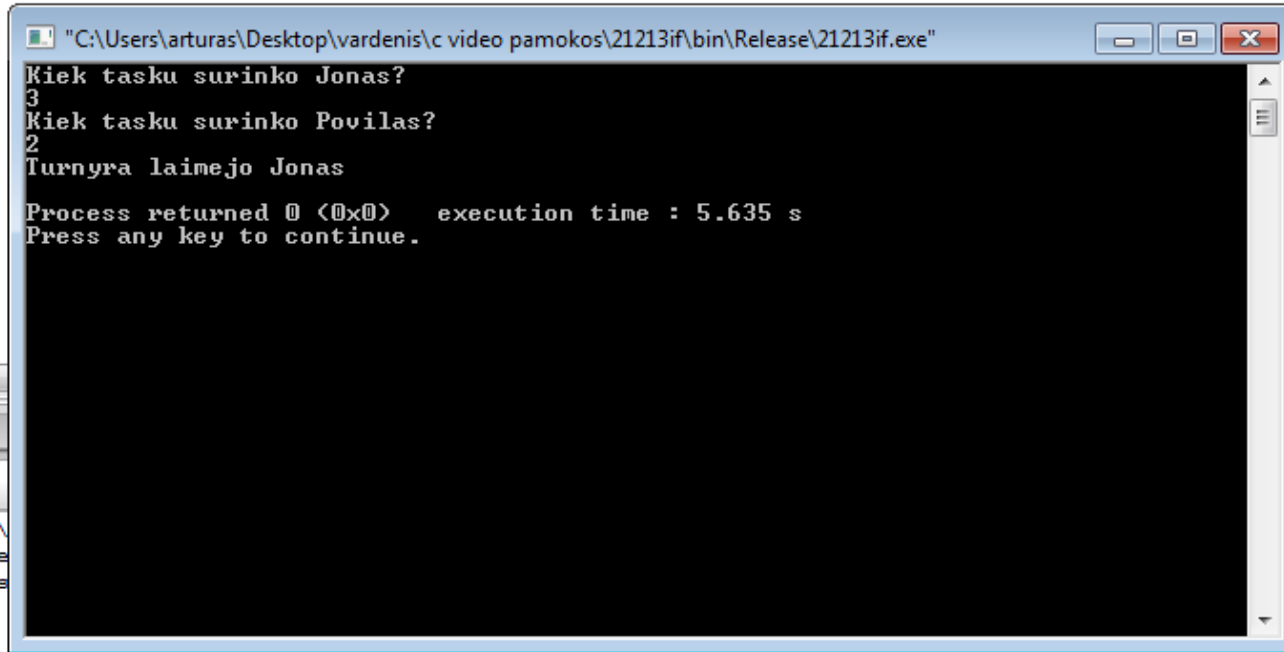
Kaip matome, sąlyginio sakinio sintaksė yra nesudėtinga:

```
▶ if (n>m) {  
▶     cout<<"Pranešimas kai sąlyga teisinga"<<endl;  
▶ }  
▶ else {  
▶     cout<<"Pranešimas kai sąlyga neteisinga"<<endl;  
▶ }
```



# Programos kodas

```
14
15     using namespace std;
16
17     int main()
18     {
19         int n,m;
20         cout << "Kiek tasku surinko Jonas?" << endl;
21         cin >>n;
22         cout << "Kiek tasku surinko Povilas?" << endl;
23         cin >>m;
24         if (n>m) {
25             cout<<"Turnyra laimejo Jonas"<<endl;
26         }
27         else {
28             cout<<"Turnyra laimejo Povilas"<<endl;
29         }
30     }
31     return 0;
32 }
33
34
```



```
"C:\Users\arturas\Desktop\vardenis\c video pamokos\21213if\bin\Release\21213if.exe"
Kiek tasku surinko Jonas?
3
Kiek tasku surinko Povilas?
2
Turnyra laimejo Jonas

Process returned 0 (0x0)   execution time : 5.635 s
Press any key to continue.
```

& others

Code::Blocks Search results

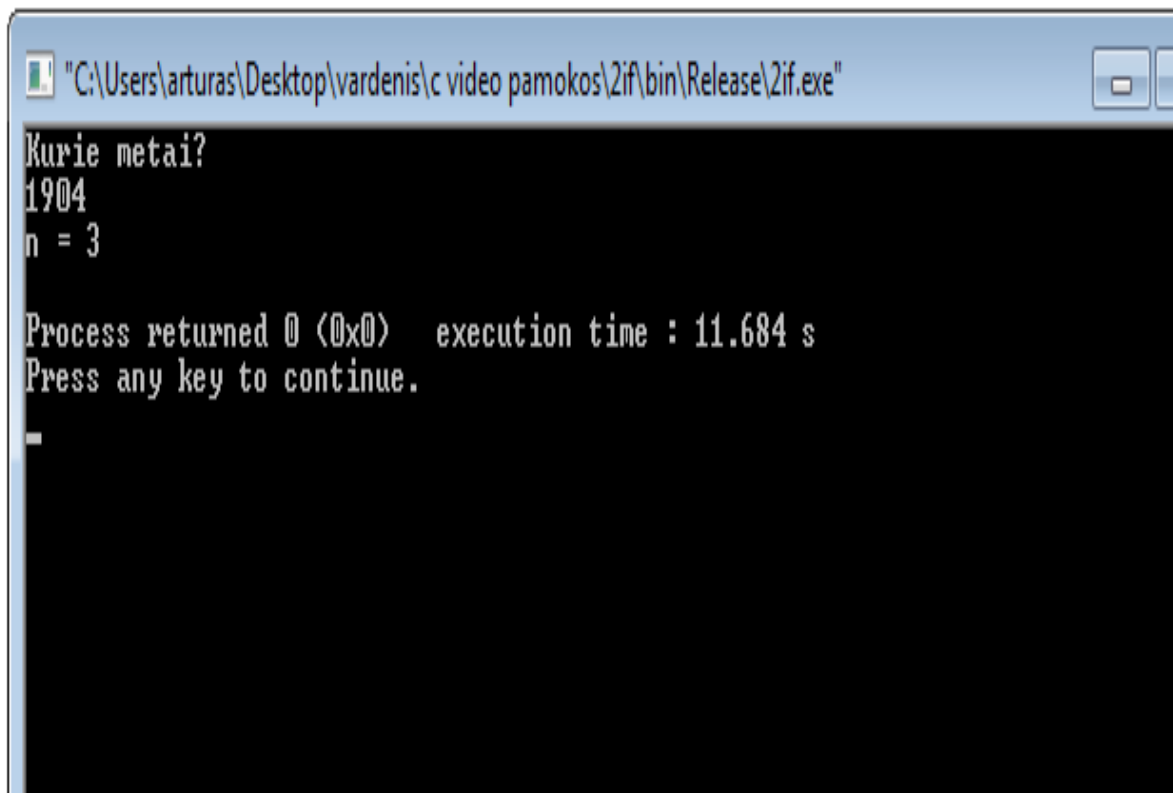
Checking for existence: C:\Users\arturas\Desktop\vardenis\c video pamokos\21213if\bin\Release\21213if.exe  
Executing: "C:\Program Files\Code::Blocks\Code::Blocks\bin\Code::Blocks.exe" -c "C:\Users\arturas\Desktop\vardenis\c video pamokos\21213if\bin\Release\21213if.exe"

## Išspręskime uždavinį (2)

- ▶ Uždavinio sąlyga:
- ▶ Pirmosios olimpinės žaidynės įvyko 1896 metais ir toliau organizuojamos kas ketveri metai. Jei žaidynės neįvyksta, tie metai vis tiek laikomi olimpiniais, o žaidynėms skiriamas eilės numeris. Parašykite programą, kuri surastų  $m$ -ųjų metų olimpinių žaidynių numerį  $n$ . Jei metai neolimpiniai, turi būti spausdinama „Metai neolimpiniai“.
- ▶ *Pasitikrinkite. Kai  $m = 1904$ , ekrane turi būti rodoma:  $n = 3$ . Kai  $m = 2005$ , turi būti rodoma: Metai neolimpiniai.*

# Programos sprendimo kodas

```
11
12 using namespace std;
13
14 int main()
15 {
16     int n,m;
17     cout << "Kurie metai?" << endl;
18     cin >> m;
19     if ((m - 1896) % 4 == 0){
20         n = (m - 1896) / 4 + 1;
21         cout << "n = " << n << endl;
22     }
23     else
24     {
25         cout << "Metai neolimpiniai" << endl;
26     }
27
28     return 0;
29 }
30
```



```
"C:\Users\arturas\Desktop\vardenis\c video pamokos\2if\bin\Release\2if.exe"
Kurie metai?
1904
n = 3

Process returned 0 (0x0)   execution time : 11.684 s
Press any key to continue.
-
```

## Kodas

```
int main()
▶ {
▶   int n,m;
▶   cout << "Kurie metai?"
<< endl;
▶   cin >> m;
▶   if ((m - 1896 ) % 4 ==
0){
▶     n = (m - 1896) / 4 + 1;
```

```
▶   cout << "n = " << n <<
endl;
▶   }
▶   else
▶   {
▶     cout << "Metai
neolimpiniai" << endl;
▶   }
▶   return 0;
▶ }
```

# Sudėtingas sąlyginis sakiny

- ▶ Kai reikia patikrinti, ar tenkinama viena iš kelių sąlygų, rašomas sudėtingas sąlyginis sakiny. Išnagrinėkime pavyzdį:
- ▶ Parenkite programą, kuri praneštų, kada baigiasi pamoka, jei iki pamokos pabaigos liko  $t$  minučių:
  - ▶ jei iki pamokos pabaigos liko daugiau negu 30 minučių, turi būti spausdinamas pranešimas „Liko dar labai daug laiko“,
  - ▶ jei iki pamokos pabaigos liko mažiau negu 30, bet daugiau negu 15 minučių, turi būti spausdinamas pranešimas „Liko dar nemažai laiko“,
  - ▶ jeigu iki pamokos pabaigos liko nuo 15 iki 7 minučių, turi būti spausdinamas pranešimas „Liko nedaug laiko“,
  - ▶ o jei 7 ir mažiau minučių - turi būti spausdinama „Pamoka baigiasi“.
- ▶ Pasitikrinkite: *kai  $t = 5$ , tuomet ekrane turi būti rodoma: „Pamoka baigiasi“.*
- ▶ *Kai  $t = 40$ , tuomet ekrane turi būti rodoma: „Liko dar labai daug laiko“.*
- ▶ *Kai  $t = 20$ , tuomet ekrane turi būti rodoma: „Liko dar nemažai laiko“.*
- ▶ *Kai  $t = 10$ , tuomet ekrane turi būti rodoma: „Liko nedaug laiko“.*

# Uždavinio kodas - analizė

- ▶ `int main()`
- ▶ `{`
- ▶ `int t;`
- ▶ `cout << "Kiek minuciu liko iki pamokos pabaigos" << endl;`
- ▶ `cin>>t;`
- ▶ `if (t<7) {`
- ▶ `cout<<"pamoka baigiasi"<<endl;`
- ▶ `}`
- ▶ `else if(t <= 15){`
- ▶ `cout << "Liko nedaug laiko." << endl;`
- ▶ `}`
- ▶ `else if (t < 30){ cout << "Liko dar nemažai laiko." << endl;`
- ▶ `}`
- ▶ `else {cout << "Liko dar labai daug laiko." << endl;`
- ▶ `}`
- ▶ `return 0;`
- ▶ `}`

```
Kiek minuciu liko iki pamokos pabaigos
6
pamoka baigiasi

Process returned 0 (0x0)   execution time : 2.726 s
Press any key to continue.
```

```
25
26     using namespace std;
27
28     int main()
29     {
30         int t;
31         cout << "Kiek minuciu liko iki pamokos pabaigos" << endl;
32         cin>>t;
33         if (t<7) {
34             cout<<"pamoka baigiasi"<<endl;
35         }
36         else if(t <= 15){
37             cout << "Liko nedaug laiko." << endl;
38         }
39         else if (t < 30){ cout << "Liko dar nemažai laiko." << endl;
40         }
41         else {cout << "Liko dar labai daug laiko." << endl;
42         }
43         return 0;
44     }
45
```

# Kelių sąlygų jungimas loginėmis operacijomis

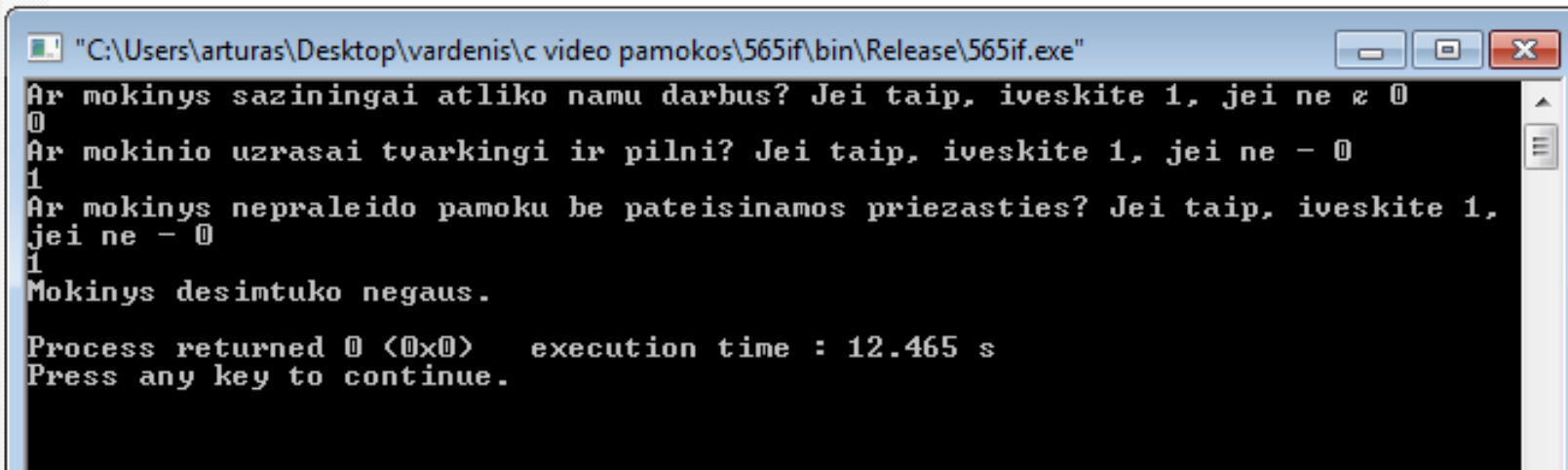
- ▶ Labai dažnai tenka spręsti uždavinius, kai vienu metu turi būti tenkinamos kelios sąlygos, arba bent viena iš kelių sąlygų. Tokiu atveju sąlygos jungiamos loginėmis operacijomis ir arba arba. Kaip tai daroma išsiaiškinsime nagrinėdami pavyzdžius.
- ▶ **1 pavyzdys**
- ▶ Norėdama paskatinti mokinius nuosekliai dirbti, mokytoja nusprendė pusmečio gale parašyti po dešimtuką visiems:
- ▶ kurie sąžiningai sprendė namų darbus ( $n1 = 1$ , kai mokinys sąžiningai sprendė namų darbus,  $n1 = 0$ , kai mokinys atliko ne visus namų darbus arba sprendė juos nesąžiningai),
- ▶ kurių užrašai yra tvarkingi ir pilni ( $n2 = 1$ , kai mokinys veda tvarkingus užrašus ir jie yra pilni,  $n2 = 0$ , kai mokinio užrašai yra netvarkingi arba nepilni),
- ▶ kurie be pateisinamos priežasties nepraleido nė vienos pamokos ( $n3 = 1$ , kai mokinys be pateisinamos priežasties nepraleido nė vienos pamokos,  $n3 = 0$ , kai mokinys praleido pamokas be pateisinamos priežasties).
- ▶ Parašykite programą, kuri kompiuterio ekrane parodytų pranešimą, ar mokinys gaus dešimtuką už gerą pusmečio darbą.
- ▶
- ▶ *Pasitikrinkite: kai  $n1 = 0$ ,  $n2 = 0$ ,  $n3 = 0$ , tuomet ekrane turi būti rodomas pranešimas Mokinys dešimtuko negaus, kai  $n1 = 0$ ,  $n2 = 1$ ,  $n3 = 0$ , tuomet ekrane turi būti rodomas pranešimas Mokinys dešimtuko negaus, kai  $n1 = 1$ ,  $n2 = 1$ ,  $n3 = 1$ , tuomet ekrane turi būti rodomas pranešimas Mokinys dešimtuką gaus.*
- ▶ Norint gauti dešimtuką reikia įvykdyti visas sąlygas. Tokiu atveju sąlygos jungiamos logine operacija ir. Programa būtų tokia:



# Programos kodas

- ▶ `int main()`
- ▶ `{`
- ▶ `int n1, n2, n3;`
- ▶ `cout << "Ar mokinys sąžiningai atliko namų darbus? Jei taip, įveskite 1, jei ne - 0 "<<endl;`
- ▶ `cin >> n1;`
- ▶ `cout << "Ar mokinio užrašai tvarkingi ir pilni? Jei taip, įveskite 1, jei ne - 0 "<<endl;`
- ▶ `cin >> n2;`
- ▶ `cout << "Ar mokinys nepraleido pamokų be pateisinamos priežasties? Jei taip, įveskite 1, jei ne - 0 "<<endl;`
- ▶ `cin >> n3;`
- ▶ `if ((n1 == 1) && (n2 == 1) && (n3 == 1))`
- ▶ `{cout << "Mokinys dešimtuką gaus."<< endl;`
- ▶ `}`
- ▶ `else`
- ▶ `{ cout << "Mokinys dešimtuko negaus." << endl;`
- ▶ `}`
- ▶ `return 0;`
- ▶ `}`

```
4
5 int main()
6 {
7     int n1, n2, n3;
8     cout << "Ar mokinys sazingai atliko namu darbus? Jei taip, iveskite 1, jei ne - 0 "<<endl;
9     cin >> n1;
10    cout << "Ar mokinio uzrasai tvarkingi ir pilni? Jei taip, iveskite 1, jei ne - 0 "<<endl;
11    cin >> n2;
12    cout << "Ar mokinys nepraleido pamoku be pateisinamos priezasties? Jei taip, iveskite 1, jei ne - 0 "<<endl;
13    cin >> n3;
14    if ((n1 == 1) && (n2 == 1) && (n3 == 1))
15    {cout << "Mokinys desimtuka gaus." << endl;
16    }
17    else
18    { cout << "Mokinys desimtuko negaus." << endl;
19    }
20    return 0;
21 }
22
```



```
"C:\Users\arturas\Desktop\vardenis\c video pamokos\565if\bin\Release\565if.exe"
Ar mokinys sazingai atliko namu darbus? Jei taip, iveskite 1, jei ne - 0
0
Ar mokinio uzrasai tvarkingi ir pilni? Jei taip, iveskite 1, jei ne - 0
1
Ar mokinys nepraleido pamoku be pateisinamos priezasties? Jei taip, iveskite 1,
jei ne - 0
1
Mokinys desimtuko negaus.

Process returned 0 (0x0)   execution time : 12.465 s
Press any key to continue.
```