VIDZEMES AUGSTSKOLA

IT nodaļa

Inženierzinātņu fakultāte

2. kurss

Programmas izstrāde C++ vidē

Neto algas aprēķina kalkulators

Izstrādāja: E. M. Maksimova

Pārbaudīja: A. Fjodorovs

Pieņēma: A. Fjodorovs

2018. / 2019. māc. Gads

SATURS

[1. UZDEVUMA NOSTĀDNE 3](#_Toc8648937)

[1.1. Uzdevumu matemātiskais formulējums 3](#_Toc8648938)

[1.2. Matemātiskais pamatojums 3](#_Toc8648939)

[2. PROGRAMMAS APRAKSTS 5](#_Toc8648940)

[2.1. Lietojumsfēra 5](#_Toc8648941)

[2.2. Ievaddati 6](#_Toc8648942)

[2.3. Izvaddati 6](#_Toc8648943)

[3. PROGRAMMAS ATKĻŪDOŠANAS KĻŪDU ANALĪZE 7](#_Toc8648944)

[4. PROGRAMMĀ PIELIETOTĀS BIBLIOTĒKAS APRAKSTS 8](#_Toc8648945)

[5. PROGRAMMAS FUNKCIONĒŠANAS KONTROLES PIEMĒRS 9](#_Toc8648946)

[6. LIETOTĀJA CEĻVEDIS 10](#_Toc8648947)

[7. SECINĀJUMI 11](#_Toc8648948)

[8. PROGRAMMAS PIRMTEKSTS 12](#_Toc8648949)

[8.1. Kalkulators.cpp 12](#_Toc8648950)

[8.2. Kalkulators.h 12](#_Toc8648951)

1. UZDEVUMA NOSTĀDNE
2. Uzdevumu matemātiskais formulējums

Sastādīt programmu, kas no tastatūras ievadītajām vērtībām – stundu skaita, likmes utt. aprēķina strādnieka neto algu, jeb algu, kuru saņem pēc nodokļu nomaksas. Atkarībā no darba vietas, samaksa par darbu var būt dažāda – citiem tā ir fiksētā summa stundā, citiem likme ir atkarīga no dienas – vai tā ir brīvdiena, naktsmaiņa vai svētku diena utt.

1. Matemātiskais pamatojums

Lai aprēķinātu neto darba algu, ir jāzina:

* Darba ņēmēja sociālā nodokļa likme - **11**%;
* Neapliekamais minimums, ko aprēķina atkarībā no algas lieluma – 0-230 eiro;
* Atvieglojums par apgādājamo - 230 eiro par 1 apgādājamo.
* Iedzīvotāju ienākuma nodoklis:

a) līdz 20 004 eiro gadā - **20**%;

b) no 20 004 līdz 62 800 eiro gadā - 23%;

c) virs 62 800 eiro gadā - 31,4%.

Diferencētais neapliekamais minimums (mēnesī), kuru prognozē un aprēķina VID, sūtot paziņojumu EDS sistēmas lietotājiem.

**Neto algas aprēķins.**

1. Sociālais nodoklis = Bruto alga \* Sociālā nodokļa likme
2. Pielieto bruto algai atbilstošo Diferencēto neapliekamo minimumu, kuru aprēķina VID
3. Ar Iedzīvotāju ienākuma nodokli apliekamā summa = Bruto alga - Sociālais nodoklis - Atvieglojumi par apgaidāmajiem - Diferencētais neapliekamais minimums
4. Iedzīvotāju ienākuma nodoklis = Ar iedzīvotāju ienākumu apliekamā summa \* Iedzīvotāju ienākuma nodokļa likme
5. Neto alga = Bruto alga - Sociālais nodoklis - Iedzīvotāju ienākuma nodoklis

Piemērs:

Strādniekam ir bruto alga 800 eiro, viņam ir divi bērni un VID prognozētais diferencētais neapliekamais minimums 82,14 eiro. Aprēķini strādnieka neto algu!

1) Sociālais nodoklis = 800 \* 11% = 88 (eiro)

2) Summa par apgādājamiem (bērniem) = 2 \* 230 = 460 (eiro).

3) Bruto alga ir starp 440 un 1100 eiro, piemēro VID prognozēto diferencēto neapliekamo minimumu 82,14 (eiro)

3) Ar Iedzīvotāju ienākuma nodokli apliekamā summa = 800 - 88 - 460 - 82,14 = 169,86 (eiro)

4) Iedzīvotāju ienākuma likme = 169,86 \* 20% = 33,97 (eiro)

5) Neto alga = 800 - 88 - 33,97 = **678,03** (eiro)

**Likmes.**

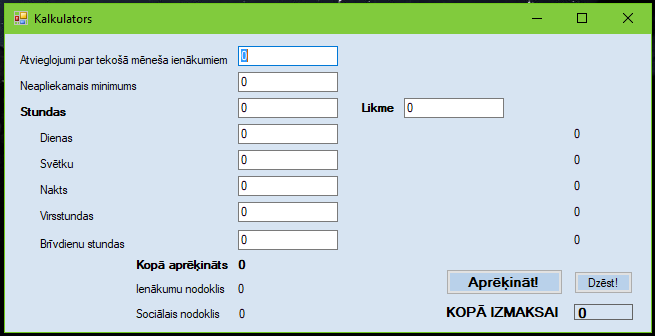
Tā kā bieži nākas saskarties ar dažādām naudas likmēm atkarībā no darba dienas, rēķinot likmi tiek pielietoti dažādi koeficienti. Programmas esošajā versijā tādas opcijas nav, bet nākotnē tiks pievienota iespēja lietotājam ievadīt attiecīgos koeficientus pašam. Patlaban noklusētās vērtības ir 0.40 centu piemaksa par katru brīvdienu (piektdiena sākot no 18:00 un beidzot ar svētdienu 00:00) stundu, 1,5\*likme par naktsmaiņas stundu (ar nosacījumu, ka naktī, pēc 22:00 tiek nostrādās vairāk nekā 2 stundas) un 2\*likme par svētku dienu stundām un virsstundām.

**Nodokļu likmes.**

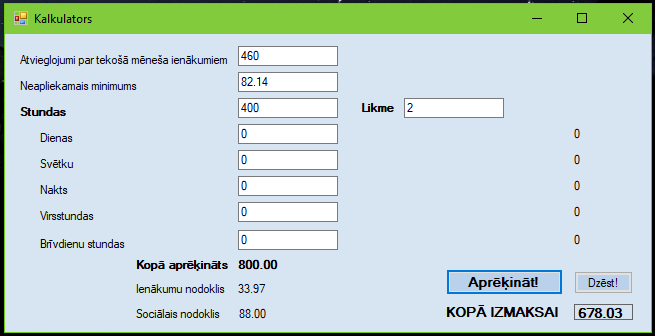
Līdzīgi kā ar likmēm par samaksu stundā, noklusētas ir arī nodokļu likmes: iedzīvotāja ienākuma nodoklis = 0,2 jeb 20% un sociālais nodoklis = 0,11 jeb 11%.

1. PROGRAMMAS APRAKSTS
2. Lietojumsfēra

Programma paredzēta neto algas aprēķināšanai. Paredzēta gan tiem, kuriem stundas samaksa par darbu ir fiksēta, gan tiem, kuriem tā atšķiras.

Programmēšanas valoda - C++. Programma izpildīta ar Intel Core i7 7700HQ ASUS ROG datoru operētājsistēmas Windows 10 vidē. Programmēts tika Microsoft Visual Studio 2017 IDE.

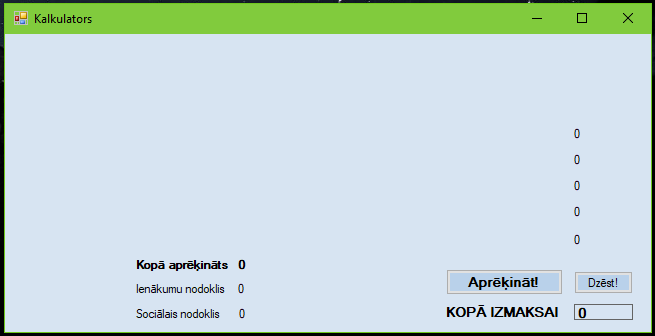
1. attēls. Programma – lodziņu izkārtojums.

1. Ievaddati

2. attēls. Tiek ievadītas vērtības no piemēra, kas aplūkojams matemātiskā pamatojuma piemērā.

Veseli pozitīvi skaitļi un punkts, lai varētu ievadīt daļskaitļa vērtības. Skaitļus ievada lietotājs attiecīgajos lodziņos.

* 1. Izvaddati

Visi izvaddati tiek attēloti nuļļu vietā.

3. attēls. Izvaddati tiek attēloti nuļļu vietā.

1. PROGRAMMAS ATKĻŪDOŠANAS KĻŪDU ANALĪZE

Programmas izveides laikā tika pielaistas dažas sintaktiskās kļūdas:

* izlaists semikols;
* Double vietā izmantots Int;
* Nepareizajā metodē veikta ievadīto skaitļu/simbolu pārbaude.
* Izlaistas figūriekavas;
* Tika pieļauta aprēķināšana ar NULL vērtību, kā arī ar punktu.

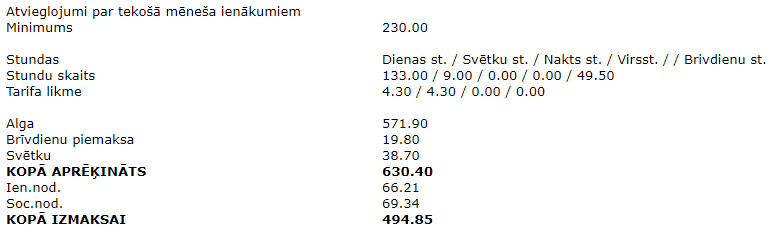
1. PROGRAMMĀ PIELIETOTĀS BIBLIOTĒKAS APRAKSTS

Windows.h bibliotēka pati par sevi iekļauj ļoti daudz mazāku bibliotēku, kuras ir diezgan cieši saistītas. Tā paredzēta lietotņu veidošanai Windows vidē. Windows.h bibliotēkā ir visas Windows API deklarācijas, kā arī daudz papildus funkciju un makro, kurus izmanto liela daļa Windows programmētāju, kurus, savukārt, izmanto daudzās funkcijās un apakšsistēmās. Tajā ir iekļauts ļoti liels daudzums funkciju, kuras atbalsta C programmēšanas valodu.

Windows.h bibliotēkā ir daudz bibliotēku, kas pakļaujas zem tās un tiek automātiski pieskaitītas Windows.h bibliotēkai. Daudzas no šīm mazākajām bibliotēkām nav iespējams lietot atsevišķi, vienas pašas, jo tās ir atkarīgas no citām bibliotēkām.

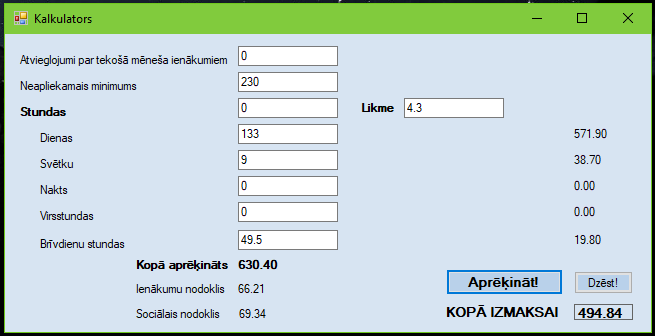
Veidojot programmu, rakstīšanas brīdī jau tika pievienotas, piemēram, ctype.h vai string.h bibliotēkas.

1. PROGRAMMAS FUNKCIONĒŠANAS KONTROLES PIEMĒRS

Tā kā programma tika veidota autores personīgajām vajadzībām, tika ņemts piemērs no jau esošas algas lapiņas. Programmā tika ievadīti galvenie šīs algas lapiņas lielumi.

4. attēls. Algas lapiņas izraksts 2019. gada aprīļa mēnesim.

Esošās algas lapiņas izkārtojums liek maldināt cilvēkus, domājot, ka tiks saņemti 630,40 eiro nevis 494,85 eiro.

5. attēls. Algas aprēķins ar algas lapiņas norādītajām vērtībām.

Viena centa atšķirību varētu izskaidrot ar to, ka algas lapiņā summa tiek noapaļota.

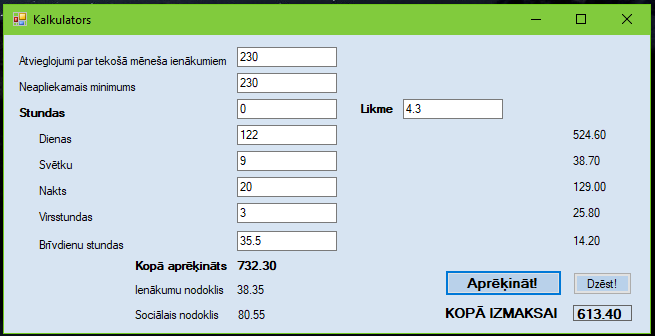
Ievadot . un 0 vērtības, programma ignorē tās un jebkurā gadījumā tukšuma vietā ieliek 0.

1. LIETOTĀJA CEĻVEDIS

Programmai ir izveidota jau strādājoša .exe versija. To iespējams pārsūtīt no ierīces uz ierīci un tai nav nepieciešams līdzi sūtīt arhīvu vai citus failus. Programmas nosaukums satur datumu, kurā pēdējo reizi veiktas kādas izmaiņas, piemēram, Calculator\_rel\_120519 – izmaiņas veiktas 2019. gada 12. maijā.

Pamācība:

1. Atzīmē Calculator\_rel\_120519 programmu un nospiež Enter vai arī veic dubult-klikšķi uz tās, un tā tiek atvērta.
2. Atkarībā no vajadzībām tiek ievadītas nepieciešamās vērtības – minimums, atvieglojumi, stundas un likme.
3. Ja lodziņā nekas netiek ievadīts, automātiski tiek atstāta 0;
4. Lodziņā nevar ievadīt vairāk par 5 simboliem – tiek skaitīts arī punkts, piemēram, 230.4 vai 23.04.
5. Kad visas vērtības ievadītas – tiek spiesta “Aprēķināt!” poga, kas aprēķina sociālo nodokli, iedzīvotāju ienākuma nodokli un, ja ir bijušas dalītās likmes – dienas stundas samaksu, nakts, utt. stundu samaksu.
6. Lodziņā “KOPĀ IZMAKSAI” tiek attēlota summa, kuru lietotājs saņem pēc nodokļu nomaksas.
7. Ja lietotājs vēlas, ir iespējams datus nodzēst, izmantojot “Dzēst!” pogu – visas vērtības ir 0.

Piemērs logam, ko izvada uz ekrāna:

6. attēls. Programmas funkcionēšanas attēlojums.

1. SECINĀJUMI

Darbs bija pietiekami interesants un aizraujošs, jo bieži nācās saskarties ar loģiskajām && un || kļūdām. Interesanti šķita piesaistīt darbam grafisko vidi, kas lietotājam sniedz iespēju labāk saprast ko rakstīt lodziņos. Man šajā darbā visvairāk ieinteresēja iespēja strādāt ar Windows formām. Patiesu prieku guvu tad, kad sapratu, ka esmu izveidojusi pirmo .exe programmu. Lai gan darbs, man, personīgi, šķita piņķerīgs, es labprāt turpinātu darbu pie tā un to brīvajā laikā papildinātu ar papildus funkcijām, piemēram, saglabāt vai drukāt funkcijas. Noderīgi būtu arī info lodziņi.

Domāju, ka manis rakstīto programmu būtu iespējams papildināt info lodziņu un informāciju par šī gada nodokļiem un citu, ar darba samaksu saistītu informāciju, lai lietotājs būtu pēc iespējas vairāk informēts.

1. PROGRAMMAS PIRMTEKSTS

## 8.1. Kalkulators.cpp

#include "Kalkulators.h"

using namespace System;

using namespace System::Windows::Forms;

[STAThreadAttribute]

void Main(array<String^>^ args) {

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

Calculator::Kalkulators form;

Application::Run(%form);

}

## 8.2. Kalkulators.h

#pragma once

namespace Calculator {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Summary for Kalkulators

/// </summary>

public ref class Kalkulators : public System::Windows::Forms::Form

{

public:

Kalkulators(void)

{

InitializeComponent();

}

protected:

/// <summary>

/// Clean up any resources being used.

/// </summary>

~Kalkulators()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::TextBox^ txt\_atvieglojumi;

private: System::Windows::Forms::TextBox^ txt\_minimums;

private: System::Windows::Forms::TextBox^ txt\_stundas;

private: System::Windows::Forms::TextBox^ txt\_dienash;

private: System::Windows::Forms::TextBox^ txt\_svetkuh;

private: System::Windows::Forms::TextBox^ txt\_naktsh;

private: System::Windows::Forms::TextBox^ txt\_virssth;

private: System::Windows::Forms::TextBox^ txt\_brivh;

private: System::Windows::Forms::TextBox^ txt\_likme;

private: System::Windows::Forms::Label^ lbl\_atvieglojumi;

private: System::Windows::Forms::Label^ lbl\_minimums;

private: System::Windows::Forms::Label^ lbl\_stundas;

private: System::Windows::Forms::Label^ lbl\_dienas;

private: System::Windows::Forms::Label^ lbl\_svetku;

private: System::Windows::Forms::Label^ lbl\_nakts;

private: System::Windows::Forms::Label^ lbl\_virsst;

private: System::Windows::Forms::Label^ lbl\_brivst;

private: System::Windows::Forms::Label^ lbl\_likme;

private: System::Windows::Forms::Label^ lbl\_kopaparekinatais;

private: System::Windows::Forms::Label^ lbl\_kopaapr;

private: System::Windows::Forms::Label^ lbl\_dienash;

private: System::Windows::Forms::Label^ lbl\_svetkuh;

private: System::Windows::Forms::Label^ lbl\_naktsh;

private: System::Windows::Forms::Label^ lbl\_virssth;

private: System::Windows::Forms::Label^ lbl\_brivsth;

private: System::Windows::Forms::Label^ lbl\_iennod;

private: System::Windows::Forms::Label^ lbl\_socnod;

private: System::Windows::Forms::Label^ lbl\_ienakumunod;

private: System::Windows::Forms::Label^ lbl\_socialaisnod;

private: System::Windows::Forms::Label^ lbl\_kopaaizmaksai;

private: System::Windows::Forms::Label^ lbl\_izmaksai;

private: System::Windows::Forms::Button^ btn\_aprekinat;

private: System::Windows::Forms::Button^ btn\_dzest;

private:

/// <summary>

/// Required designer variable.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

void InitializeComponent(void)

{

this->txt\_atvieglojumi = (gcnew System::Windows::Forms::TextBox());

this->txt\_minimums = (gcnew System::Windows::Forms::TextBox());

this->txt\_stundas = (gcnew System::Windows::Forms::TextBox());

this->txt\_dienash = (gcnew System::Windows::Forms::TextBox());

this->txt\_svetkuh = (gcnew System::Windows::Forms::TextBox());

this->txt\_naktsh = (gcnew System::Windows::Forms::TextBox());

this->lbl\_atvieglojumi = (gcnew System::Windows::Forms::Label());

this->lbl\_minimums = (gcnew System::Windows::Forms::Label());

this->lbl\_stundas = (gcnew System::Windows::Forms::Label());

this->lbl\_dienas = (gcnew System::Windows::Forms::Label());

this->lbl\_svetku = (gcnew System::Windows::Forms::Label());

this->lbl\_nakts = (gcnew System::Windows::Forms::Label());

this->lbl\_virsst = (gcnew System::Windows::Forms::Label());

this->lbl\_brivst = (gcnew System::Windows::Forms::Label());

this->txt\_virssth = (gcnew System::Windows::Forms::TextBox());

this->txt\_brivh = (gcnew System::Windows::Forms::TextBox());

this->txt\_likme = (gcnew System::Windows::Forms::TextBox());

this->lbl\_likme = (gcnew System::Windows::Forms::Label());

this->lbl\_kopaparekinatais = (gcnew System::Windows::Forms::Label());

this->lbl\_kopaapr = (gcnew System::Windows::Forms::Label());

this->lbl\_dienash = (gcnew System::Windows::Forms::Label());

this->lbl\_svetkuh = (gcnew System::Windows::Forms::Label());

this->lbl\_naktsh = (gcnew System::Windows::Forms::Label());

this->lbl\_virssth = (gcnew System::Windows::Forms::Label());

this->lbl\_brivsth = (gcnew System::Windows::Forms::Label());

this->lbl\_iennod = (gcnew System::Windows::Forms::Label());

this->lbl\_socnod = (gcnew System::Windows::Forms::Label());

this->lbl\_ienakumunod = (gcnew System::Windows::Forms::Label());

this->lbl\_socialaisnod = (gcnew System::Windows::Forms::Label());

this->lbl\_kopaaizmaksai = (gcnew System::Windows::Forms::Label());

this->lbl\_izmaksai = (gcnew System::Windows::Forms::Label());

this->btn\_aprekinat = (gcnew System::Windows::Forms::Button());

this->btn\_dzest = (gcnew System::Windows::Forms::Button());

this->SuspendLayout();

//

// txt\_atvieglojumi

//

this->txt\_atvieglojumi->Location = System::Drawing::Point(233, 12);

this->txt\_atvieglojumi->MaxLength = 3;

this->txt\_atvieglojumi->Name = L"txt\_atvieglojumi";

this->txt\_atvieglojumi->Size = System::Drawing::Size(100, 20);

this->txt\_atvieglojumi->TabIndex = 0;

this->txt\_atvieglojumi->Text = L"0";

this->txt\_atvieglojumi->TextChanged += gcnew System::EventHandler(this, &Kalkulators::txt\_atvieglojumi\_TextChanged);

this->txt\_atvieglojumi->KeyPress += gcnew System::Windows::Forms::KeyPressEventHandler(this, &Kalkulators::txt\_atvieglojumi\_KeyPress);

//

// txt\_minimums

//

this->txt\_minimums->Location = System::Drawing::Point(233, 38);

this->txt\_minimums->MaxLength = 6;

this->txt\_minimums->Name = L"txt\_minimums";

this->txt\_minimums->Size = System::Drawing::Size(100, 20);

this->txt\_minimums->TabIndex = 1;

this->txt\_minimums->Text = L"0";

this->txt\_minimums->TextChanged += gcnew System::EventHandler(this, &Kalkulators::txt\_minimums\_TextChanged);

this->txt\_minimums->KeyPress += gcnew System::Windows::Forms::KeyPressEventHandler(this, &Kalkulators::txt\_minimums\_KeyPress);

//

// txt\_stundas

//

this->txt\_stundas->Location = System::Drawing::Point(233, 64);

this->txt\_stundas->MaxLength = 5;

this->txt\_stundas->Name = L"txt\_stundas";

this->txt\_stundas->Size = System::Drawing::Size(100, 20);

this->txt\_stundas->TabIndex = 2;

this->txt\_stundas->Text = L"0";

this->txt\_stundas->TextChanged += gcnew System::EventHandler(this, &Kalkulators::txt\_stundas\_TextChanged);

this->txt\_stundas->KeyPress += gcnew System::Windows::Forms::KeyPressEventHandler(this, &Kalkulators::txt\_stundas\_KeyPress);

//

// txt\_dienash

//

this->txt\_dienash->Location = System::Drawing::Point(233, 90);

this->txt\_dienash->MaxLength = 5;

this->txt\_dienash->Name = L"txt\_dienash";

this->txt\_dienash->Size = System::Drawing::Size(100, 20);

this->txt\_dienash->TabIndex = 3;

this->txt\_dienash->Text = L"0";

this->txt\_dienash->TextChanged += gcnew System::EventHandler(this, &Kalkulators::txt\_dienash\_TextChanged);

this->txt\_dienash->KeyPress += gcnew System::Windows::Forms::KeyPressEventHandler(this, &Kalkulators::txt\_dienash\_KeyPress);

//

// txt\_svetkuh

//

this->txt\_svetkuh->Location = System::Drawing::Point(233, 116);

this->txt\_svetkuh->MaxLength = 4;

this->txt\_svetkuh->Name = L"txt\_svetkuh";

this->txt\_svetkuh->Size = System::Drawing::Size(100, 20);

this->txt\_svetkuh->TabIndex = 4;

this->txt\_svetkuh->Text = L"0";

this->txt\_svetkuh->TextChanged += gcnew System::EventHandler(this, &Kalkulators::txt\_svetkuh\_TextChanged);

this->txt\_svetkuh->KeyPress += gcnew System::Windows::Forms::KeyPressEventHandler(this, &Kalkulators::txt\_svetkuh\_KeyPress);

//

// txt\_naktsh

//

this->txt\_naktsh->Location = System::Drawing::Point(233, 142);

this->txt\_naktsh->MaxLength = 5;

this->txt\_naktsh->Name = L"txt\_naktsh";

this->txt\_naktsh->Size = System::Drawing::Size(100, 20);

this->txt\_naktsh->TabIndex = 5;

this->txt\_naktsh->Text = L"0";

this->txt\_naktsh->TextChanged += gcnew System::EventHandler(this, &Kalkulators::txt\_naktsh\_TextChanged);

this->txt\_naktsh->KeyPress += gcnew System::Windows::Forms::KeyPressEventHandler(this, &Kalkulators::txt\_naktsh\_KeyPress);

//

// lbl\_atvieglojumi

//

this->lbl\_atvieglojumi->AutoSize = true;

this->lbl\_atvieglojumi->Location = System::Drawing::Point(12, 19);

this->lbl\_atvieglojumi->Name = L"lbl\_atvieglojumi";

this->lbl\_atvieglojumi->Size = System::Drawing::Size(215, 13);

this->lbl\_atvieglojumi->TabIndex = 7;

this->lbl\_atvieglojumi->Text = L"Atvieglojumi par tekošā mēneša ienākumiem\r\n";

//

// lbl\_minimums

//

this->lbl\_minimums->AutoSize = true;

this->lbl\_minimums->Location = System::Drawing::Point(12, 45);

this->lbl\_minimums->Name = L"lbl\_minimums";

this->lbl\_minimums->Size = System::Drawing::Size(124, 13);

this->lbl\_minimums->TabIndex = 8;

this->lbl\_minimums->Text = L"Neapliekamais minimums";

//

// lbl\_stundas

//

this->lbl\_stundas->AutoSize = true;

this->lbl\_stundas->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(186)));

this->lbl\_stundas->Location = System::Drawing::Point(12, 71);

this->lbl\_stundas->Name = L"lbl\_stundas";

this->lbl\_stundas->Size = System::Drawing::Size(53, 13);

this->lbl\_stundas->TabIndex = 9;

this->lbl\_stundas->Text = L"Stundas";

//

// lbl\_dienas

//

this->lbl\_dienas->AutoSize = true;

this->lbl\_dienas->Location = System::Drawing::Point(32, 97);

this->lbl\_dienas->Name = L"lbl\_dienas";

this->lbl\_dienas->Size = System::Drawing::Size(40, 13);

this->lbl\_dienas->TabIndex = 10;

this->lbl\_dienas->Text = L"Dienas";

//

// lbl\_svetku

//

this->lbl\_svetku->AutoSize = true;

this->lbl\_svetku->Location = System::Drawing::Point(32, 123);

this->lbl\_svetku->Name = L"lbl\_svetku";

this->lbl\_svetku->Size = System::Drawing::Size(41, 13);

this->lbl\_svetku->TabIndex = 11;

this->lbl\_svetku->Text = L"Svētku";

//

// lbl\_nakts

//

this->lbl\_nakts->AutoSize = true;

this->lbl\_nakts->Location = System::Drawing::Point(32, 149);

this->lbl\_nakts->Name = L"lbl\_nakts";

this->lbl\_nakts->Size = System::Drawing::Size(35, 13);

this->lbl\_nakts->TabIndex = 12;

this->lbl\_nakts->Text = L"Nakts";

//

// lbl\_virsst

//

this->lbl\_virsst->AutoSize = true;

this->lbl\_virsst->Location = System::Drawing::Point(32, 175);

this->lbl\_virsst->Name = L"lbl\_virsst";

this->lbl\_virsst->Size = System::Drawing::Size(61, 13);

this->lbl\_virsst->TabIndex = 13;

this->lbl\_virsst->Text = L"Virsstundas";

//

// lbl\_brivst

//

this->lbl\_brivst->AutoSize = true;

this->lbl\_brivst->Location = System::Drawing::Point(32, 203);

this->lbl\_brivst->Name = L"lbl\_brivst";

this->lbl\_brivst->Size = System::Drawing::Size(92, 13);

this->lbl\_brivst->TabIndex = 14;

this->lbl\_brivst->Text = L"Brīvdienu stundas";

//

// txt\_virssth

//

this->txt\_virssth->Location = System::Drawing::Point(233, 168);

this->txt\_virssth->MaxLength = 4;

this->txt\_virssth->Name = L"txt\_virssth";

this->txt\_virssth->Size = System::Drawing::Size(100, 20);

this->txt\_virssth->TabIndex = 16;

this->txt\_virssth->Text = L"0";

this->txt\_virssth->TextChanged += gcnew System::EventHandler(this, &Kalkulators::txt\_virssth\_TextChanged);

this->txt\_virssth->KeyPress += gcnew System::Windows::Forms::KeyPressEventHandler(this, &Kalkulators::txt\_virssth\_KeyPress);

//

// txt\_brivh

//

this->txt\_brivh->Location = System::Drawing::Point(233, 196);

this->txt\_brivh->MaxLength = 4;

this->txt\_brivh->Name = L"txt\_brivh";

this->txt\_brivh->Size = System::Drawing::Size(100, 20);

this->txt\_brivh->TabIndex = 17;

this->txt\_brivh->Text = L"0";

this->txt\_brivh->TextChanged += gcnew System::EventHandler(this, &Kalkulators::txt\_brivh\_TextChanged);

this->txt\_brivh->KeyPress += gcnew System::Windows::Forms::KeyPressEventHandler(this, &Kalkulators::txt\_brivh\_KeyPress);

//

// txt\_likme

//

this->txt\_likme->Location = System::Drawing::Point(399, 64);

this->txt\_likme->MaxLength = 5;

this->txt\_likme->Name = L"txt\_likme";

this->txt\_likme->Size = System::Drawing::Size(100, 20);

this->txt\_likme->TabIndex = 19;

this->txt\_likme->Text = L"0";

this->txt\_likme->TextChanged += gcnew System::EventHandler(this, &Kalkulators::txt\_likme\_TextChanged);

this->txt\_likme->KeyPress += gcnew System::Windows::Forms::KeyPressEventHandler(this, &Kalkulators::txt\_likme\_KeyPress);

//

// lbl\_likme

//

this->lbl\_likme->AutoSize = true;

this->lbl\_likme->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(186)));

this->lbl\_likme->Location = System::Drawing::Point(353, 67);

this->lbl\_likme->Name = L"lbl\_likme";

this->lbl\_likme->Size = System::Drawing::Size(40, 13);

this->lbl\_likme->TabIndex = 21;

this->lbl\_likme->Text = L"Likme";

//

// lbl\_kopaparekinatais

//

this->lbl\_kopaparekinatais->AutoSize = true;

this->lbl\_kopaparekinatais->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F, System::Drawing::FontStyle::Bold,System::Drawing::GraphicsUnit::Point, static\_cast<System::Byte>(186)));

this->lbl\_kopaparekinatais->Location = System::Drawing::Point(230, 224);

this->lbl\_kopaparekinatais->Name = L"lbl\_kopaparekinatais";

this->lbl\_kopaparekinatais->Size = System::Drawing::Size(14, 13);

this->lbl\_kopaparekinatais->TabIndex = 25;

this->lbl\_kopaparekinatais->Text = L"0";

//

// lbl\_kopaapr

//

this->lbl\_kopaapr->AutoSize = true;

this->lbl\_kopaapr->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(186)));

this->lbl\_kopaapr->Location = System::Drawing::Point(128, 224);

this->lbl\_kopaapr->Name = L"lbl\_kopaapr";

this->lbl\_kopaapr->Size = System::Drawing::Size(99, 13);

this->lbl\_kopaapr->TabIndex = 24;

this->lbl\_kopaapr->Text = L"Kopā aprēķināts";

//

// lbl\_dienash

//

this->lbl\_dienash->Location = System::Drawing::Point(566, 93);

this->lbl\_dienash->Name = L"lbl\_dienash";

this->lbl\_dienash->Size = System::Drawing::Size(41, 13);

this->lbl\_dienash->TabIndex = 27;

this->lbl\_dienash->Text = L"0";

//

// lbl\_svetkuh

//

this->lbl\_svetkuh->Location = System::Drawing::Point(566, 119);

this->lbl\_svetkuh->Name = L"lbl\_svetkuh";

this->lbl\_svetkuh->Size = System::Drawing::Size(41, 13);

this->lbl\_svetkuh->TabIndex = 29;

this->lbl\_svetkuh->Text = L"0";

//

// lbl\_naktsh

//

this->lbl\_naktsh->Location = System::Drawing::Point(566, 145);

this->lbl\_naktsh->Name = L"lbl\_naktsh";

this->lbl\_naktsh->Size = System::Drawing::Size(41, 13);

this->lbl\_naktsh->TabIndex = 31;

this->lbl\_naktsh->Text = L"0";

//

// lbl\_virssth

//

this->lbl\_virssth->Location = System::Drawing::Point(566, 171);

this->lbl\_virssth->Name = L"lbl\_virssth";

this->lbl\_virssth->Size = System::Drawing::Size(41, 13);

this->lbl\_virssth->TabIndex = 33;

this->lbl\_virssth->Text = L"0";

//

// lbl\_brivsth

//

this->lbl\_brivsth->Location = System::Drawing::Point(566, 199);

this->lbl\_brivsth->Name = L"lbl\_brivsth";

this->lbl\_brivsth->Size = System::Drawing::Size(41, 13);

this->lbl\_brivsth->TabIndex = 35;

this->lbl\_brivsth->Text = L"0";

//

// lbl\_iennod

//

this->lbl\_iennod->AutoSize = true;

this->lbl\_iennod->Location = System::Drawing::Point(128, 248);

this->lbl\_iennod->Name = L"lbl\_iennod";

this->lbl\_iennod->Size = System::Drawing::Size(96, 13);

this->lbl\_iennod->TabIndex = 36;

this->lbl\_iennod->Text = L"Ienākumu nodoklis";

//

// lbl\_socnod

//

this->lbl\_socnod->AutoSize = true;

this->lbl\_socnod->Location = System::Drawing::Point(128, 273);

this->lbl\_socnod->Name = L"lbl\_socnod";

this->lbl\_socnod->Size = System::Drawing::Size(91, 13);

this->lbl\_socnod->TabIndex = 37;

this->lbl\_socnod->Text = L"Sociālais nodoklis";

//

// lbl\_ienakumunod

//

this->lbl\_ienakumunod->AutoSize = true;

this->lbl\_ienakumunod->Location = System::Drawing::Point(230, 248);

this->lbl\_ienakumunod->Name = L"lbl\_ienakumunod";

this->lbl\_ienakumunod->Size = System::Drawing::Size(13, 13);

this->lbl\_ienakumunod->TabIndex = 38;

this->lbl\_ienakumunod->Text = L"0";

//

// lbl\_socialaisnod

//

this->lbl\_socialaisnod->AutoSize = true;

this->lbl\_socialaisnod->Location = System::Drawing::Point(231, 273);

this->lbl\_socialaisnod->Name = L"lbl\_socialaisnod";

this->lbl\_socialaisnod->Size = System::Drawing::Size(13, 13);

this->lbl\_socialaisnod->TabIndex = 39;

this->lbl\_socialaisnod->Text = L"0";

//

// lbl\_kopaaizmaksai

//

this->lbl\_kopaaizmaksai->AutoSize = true;

this->lbl\_kopaaizmaksai->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 9.75F, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(186)));

this->lbl\_kopaaizmaksai->Location = System::Drawing::Point(438, 270);

this->lbl\_kopaaizmaksai->Name = L"lbl\_kopaaizmaksai";

this->lbl\_kopaaizmaksai->Size = System::Drawing::Size(120, 16);

this->lbl\_kopaaizmaksai->TabIndex = 40;

this->lbl\_kopaaizmaksai->Text = L"KOPĀ IZMAKSAI";

//

// lbl\_izmaksai

//

this->lbl\_izmaksai->BorderStyle = System::Windows::Forms::BorderStyle::FixedSingle;

this->lbl\_izmaksai->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 9.75F, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(186)));

this->lbl\_izmaksai->Location = System::Drawing::Point(569, 270);

this->lbl\_izmaksai->Name = L"lbl\_izmaksai";

this->lbl\_izmaksai->Size = System::Drawing::Size(59, 16);

this->lbl\_izmaksai->TabIndex = 41;

this->lbl\_izmaksai->Text = L"0";

//

// btn\_aprekinat

//

this->btn\_aprekinat->BackColor = System::Drawing::SystemColors::GradientActiveCaption;

this->btn\_aprekinat->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 9.75F, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(186)));

this->btn\_aprekinat->Location = System::Drawing::Point(441, 235);

this->btn\_aprekinat->Name = L"btn\_aprekinat";

this->btn\_aprekinat->Size = System::Drawing::Size(117, 26);

this->btn\_aprekinat->TabIndex = 42;

this->btn\_aprekinat->Text = L"Aprēķināt!";

this->btn\_aprekinat->UseVisualStyleBackColor = false;

this->btn\_aprekinat->Click += gcnew System::EventHandler(this, &Kalkulators::btn\_aprekinat\_Click);

//

// btn\_dzest

//

this->btn\_dzest->BackColor = System::Drawing::SystemColors::GradientActiveCaption;

this->btn\_dzest->Cursor = System::Windows::Forms::Cursors::Default;

this->btn\_dzest->Location = System::Drawing::Point(569, 237);

this->btn\_dzest->Name = L"btn\_dzest";

this->btn\_dzest->Size = System::Drawing::Size(59, 23);

this->btn\_dzest->TabIndex = 43;

this->btn\_dzest->Text = L"Dzēst!";

this->btn\_dzest->UseVisualStyleBackColor = false;

this->btn\_dzest->Click += gcnew System::EventHandler(this, &Kalkulators::button1\_Click);

//

// Kalkulators

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->BackColor = System::Drawing::SystemColors::GradientInactiveCaption;

this->ClientSize = System::Drawing::Size(646, 298);

this->Controls->Add(this->btn\_dzest);

this->Controls->Add(this->btn\_aprekinat);

this->Controls->Add(this->lbl\_izmaksai);

this->Controls->Add(this->lbl\_kopaaizmaksai);

this->Controls->Add(this->lbl\_socialaisnod);

this->Controls->Add(this->lbl\_ienakumunod);

this->Controls->Add(this->lbl\_socnod);

this->Controls->Add(this->lbl\_iennod);

this->Controls->Add(this->lbl\_brivsth);

this->Controls->Add(this->lbl\_virssth);

this->Controls->Add(this->lbl\_naktsh);

this->Controls->Add(this->lbl\_svetkuh);

this->Controls->Add(this->lbl\_dienash);

this->Controls->Add(this->lbl\_kopaparekinatais);

this->Controls->Add(this->lbl\_kopaapr);

this->Controls->Add(this->lbl\_likme);

this->Controls->Add(this->txt\_likme);

this->Controls->Add(this->txt\_brivh);

this->Controls->Add(this->txt\_virssth);

this->Controls->Add(this->lbl\_brivst);

this->Controls->Add(this->lbl\_virsst);

this->Controls->Add(this->lbl\_nakts);

this->Controls->Add(this->lbl\_svetku);

this->Controls->Add(this->lbl\_dienas);

this->Controls->Add(this->lbl\_stundas);

this->Controls->Add(this->lbl\_minimums);

this->Controls->Add(this->lbl\_atvieglojumi);

this->Controls->Add(this->txt\_naktsh);

this->Controls->Add(this->txt\_svetkuh);

this->Controls->Add(this->txt\_dienash);

this->Controls->Add(this->txt\_stundas);

this->Controls->Add(this->txt\_minimums);

this->Controls->Add(this->txt\_atvieglojumi);

this->Name = L"Kalkulators";

this->Text = L"Kalkulators";

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private: System::Void btn\_aprekinat\_Click(System::Object^ sender, System::EventArgs^ e)

{

//mainigie

Double atvieglojumi, minimums, stundas,

dienas\_st, svetku\_st, nakts\_st, virs\_st,

brivd\_st, likme, kopa\_aprekinatais, kopa\_izmaksai = 0.00;

Double day, holiday, night, overt, weekend, total = 0.00;

Double ienakumu\_nodoklis, socialais\_nodoklis, ienak\_summa = 0.00;

atvieglojumi = double::Parse(txt\_atvieglojumi->Text);

minimums = double::Parse(txt\_minimums->Text);

stundas = double::Parse(txt\_stundas->Text);

dienas\_st = double::Parse(txt\_dienash->Text);

svetku\_st = double::Parse(txt\_svetkuh->Text);

nakts\_st = double::Parse(txt\_naktsh->Text);

virs\_st = double::Parse(txt\_virssth->Text);

brivd\_st = double::Parse(txt\_brivh->Text);

likme = double::Parse(txt\_likme->Text);

//Aprekina bruto algu, ja likmes stundam nemainas

if (stundas > 0)

{

kopa\_aprekinatais = stundas \* likme;

lbl\_kopaparekinatais->Text = kopa\_aprekinatais.ToString("F");

//Sociala nodokla aprekins

socialais\_nodoklis = kopa\_aprekinatais \* (11 \* 0.01);

lbl\_socialaisnod->Text = socialais\_nodoklis.ToString("F");

//Ar iedzivotaju nodokli apliekama summa

ienak\_summa = kopa\_aprekinatais - socialais\_nodoklis - atvieglojumi - minimums;

ienakumu\_nodoklis = ienak\_summa \* (20 \* 0.01);

lbl\_ienakumunod->Text = ienakumu\_nodoklis.ToString("F");

//Neto algas aprekins

kopa\_izmaksai = kopa\_aprekinatais - socialais\_nodoklis - ienakumu\_nodoklis;

lbl\_izmaksai->Text = kopa\_izmaksai.ToString("F");

}

else

{

day = dienas\_st \* likme;

lbl\_dienash->Text = day.ToString("F");

holiday = svetku\_st \* likme;

lbl\_svetkuh->Text = holiday.ToString("F");

night = nakts\_st \* (1.5 \* likme);

lbl\_naktsh->Text = night.ToString("F");

overt = virs\_st \* (2 \* likme);

lbl\_virssth->Text = overt.ToString("F");

weekend = brivd\_st \* 0.4;

lbl\_brivsth->Text = weekend.ToString("F");

total = day + holiday + night + overt + weekend;

lbl\_kopaparekinatais->Text = total.ToString("F");

//Sociala nodokla aprekins

socialais\_nodoklis = total \* (11 \* 0.01);

lbl\_socialaisnod->Text = socialais\_nodoklis.ToString("F");

//Ar iedzivotaju nodokli apliekama summa

ienak\_summa = total - socialais\_nodoklis - atvieglojumi - minimums;

ienakumu\_nodoklis = ienak\_summa \* (20 \* 0.01);

lbl\_ienakumunod->Text = ienakumu\_nodoklis.ToString("F");

//Neto algas aprekins

kopa\_izmaksai = total - socialais\_nodoklis - ienakumu\_nodoklis;

lbl\_izmaksai->Text = kopa\_izmaksai.ToString("F");

}

}

private: System::Void lblClose\_Click(System::Object^ sender, System::EventArgs^ e)

{

Close();

}

private: System::Void txt\_atvieglojumi\_KeyPress(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e)

{

char ch = e->KeyChar;

if (ch == 46 && txt\_atvieglojumi->Text->IndexOf('.') != -1)

{

e->Handled = true;

return;

}

if (!Char::IsDigit(ch) && ch != 8 && ch != 46)

{

e->Handled = true;

}

}

private: System::Void txt\_minimums\_KeyPress(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e)

{

char ch = e->KeyChar;

if (ch == 46 && txt\_minimums->Text->IndexOf('.') != -1)

{

e->Handled = true;

return;

}

if (!Char::IsDigit(ch) && ch != 8 && ch != 46)

{

e->Handled = true;

}

}

private: System::Void txt\_stundas\_KeyPress(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e)

{

char ch = e->KeyChar;

if (ch == 46 && txt\_stundas->Text->IndexOf('.') != -1)

{

e->Handled = true;

return;

}

if (!Char::IsDigit(ch) && ch != 8 && ch != 46)

{

e->Handled = true;

}

}

private: System::Void txt\_dienash\_KeyPress(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e)

{

char ch = e->KeyChar;

if (ch == 46 && txt\_dienash->Text->IndexOf('.') != -1)

{

e->Handled = true;

return;

}

if (!Char::IsDigit(ch) && ch != 8 && ch != 46)

{

e->Handled = true;

}

}

private: System::Void txt\_svetkuh\_KeyPress(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e)

{

char ch = e->KeyChar;

if (ch == 46 && txt\_svetkuh->Text->IndexOf('.') != -1)

{

e->Handled = true;

return;

}

if (!Char::IsDigit(ch) && ch != 8 && ch != 46)

{

e->Handled = true;

}

}

private: System::Void txt\_naktsh\_KeyPress(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e)

{

char ch = e->KeyChar;

if (ch == 46 && txt\_naktsh->Text->IndexOf('.') != -1)

{

e->Handled = true;

return;

}

if (!Char::IsDigit(ch) && ch != 8 && ch != 46)

{

e->Handled = true;

}

}

private: System::Void txt\_virssth\_KeyPress(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e)

{

char ch = e->KeyChar;

if (ch == 46 && txt\_virssth->Text->IndexOf('.') != -1)

{

e->Handled = true;

return;

}

if (!Char::IsDigit(ch) && ch != 8 && ch != 46)

{

e->Handled = true;

}

}

private: System::Void txt\_brivh\_KeyPress(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e)

{

char ch = e->KeyChar;

if (ch == 46 && txt\_brivh->Text->IndexOf('.') != -1)

{

e->Handled = true;

return;

}

if (!Char::IsDigit(ch) && ch != 8 && ch != 46)

{

e->Handled = true;

}

}

private: System::Void txt\_likme\_KeyPress(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e)

{

char ch = e->KeyChar;

if (ch == 46 && txt\_likme->Text->IndexOf('.') != -1)

{

e->Handled = true;

return;

}

if (!Char::IsDigit(ch) && ch != 8 && ch != 46)

{

e->Handled = true;

}

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e)

{

txt\_atvieglojumi->Text = "0";

txt\_minimums->Text = "0";

txt\_stundas->Text = "0";

txt\_dienash->Text = "0";

txt\_svetkuh->Text = "0";

txt\_naktsh->Text = "0";

txt\_virssth->Text = "0";

txt\_likme->Text = "0";

txt\_brivh->Text = "0";

lbl\_dienash->Text = "0";

lbl\_svetkuh->Text = "0";

lbl\_virssth->Text = "0";

lbl\_naktsh->Text = "0";

lbl\_brivsth->Text = "0";

lbl\_kopaparekinatais->Text = "0";

lbl\_ienakumunod->Text = "0";

lbl\_socialaisnod->Text = "0";

lbl\_izmaksai->Text = "0";

}

private: System::Void txt\_atvieglojumi\_TextChanged(System::Object^ sender, System::EventArgs^ e)

{

char cc = '.';

if (String::IsNullOrEmpty(txt\_atvieglojumi->Text) || "."==txt\_atvieglojumi->Text->ToString() )

{

txt\_atvieglojumi->Text = "0";

}

}

private: System::Void txt\_minimums\_TextChanged(System::Object^ sender, System::EventArgs^ e)

{

char cc = '.';

if (String::IsNullOrEmpty(txt\_minimums->Text) || "." == txt\_minimums->Text->ToString())

{

txt\_minimums->Text = "0";

}

}

private: System::Void txt\_stundas\_TextChanged(System::Object^ sender, System::EventArgs^ e)

{

if (String::IsNullOrEmpty(txt\_stundas->Text) || "." == txt\_stundas->Text->ToString())

{

txt\_stundas->Text = "0";

}

}

private: System::Void txt\_dienash\_TextChanged(System::Object^ sender, System::EventArgs^ e)

{

if (String::IsNullOrEmpty(txt\_dienash->Text) || "." == txt\_dienash->Text->ToString())

{

txt\_dienash->Text = "0";

}

}

private: System::Void txt\_svetkuh\_TextChanged(System::Object^ sender, System::EventArgs^ e)

{

if (String::IsNullOrEmpty(txt\_svetkuh->Text) || "." == txt\_svetkuh->Text->ToString())

{

txt\_svetkuh->Text = "0";

}

}

private: System::Void txt\_naktsh\_TextChanged(System::Object^ sender, System::EventArgs^ e)

{

if (String::IsNullOrEmpty(txt\_naktsh->Text) || "." == txt\_naktsh->Text->ToString())

{

txt\_naktsh->Text = "0";

}

}

private: System::Void txt\_virssth\_TextChanged(System::Object^ sender, System::EventArgs^ e)

{

if (String::IsNullOrEmpty(txt\_virssth->Text) || "." == txt\_virssth->Text->ToString())

{

txt\_virssth->Text = "0";

}

}

private: System::Void txt\_brivh\_TextChanged(System::Object^ sender, System::EventArgs^ e)

{

if (String::IsNullOrEmpty(txt\_brivh->Text) || "." == txt\_brivh->Text->ToString())

{

txt\_brivh->Text = "0";

}

}

private: System::Void txt\_likme\_TextChanged(System::Object^ sender, System::EventArgs^ e)

{

if (String::IsNullOrEmpty(txt\_likme->Text) || "." == txt\_likme->Text->ToString())

{

txt\_likme->Text = "0";

}

}

};

}