ФГАОУ ВО Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа киберфизических систем и управления

**ОТЧЕТ**

**по лабораторной работе №5**

**по теме «Рисование графиков»**

по дисциплине «Практикум по программированию»

Выполнил

студент гр.23533/2 А.Д. Шурак

Проверил

ассистент В.Э. Ковалевский

Санкт-Петербург

2019

Задание

На основе полученных знаний продемонстрировать графические возможности Windows Forms, построение графиков.

Разработка программы

Создаем форму. Добавляем необходимые элементы: поля для ввода текста, заголовок, и область графика – элемент Chart, кнопка.

Создадим программу, которая будет строить график функции, учитывая шаг и значение верхней границы.

Все функции, выполняемые программой, будем выполнять по нажатию кнопки «Построить график».

Необходимый нам элемент Chart можно найти в пространстве имён System::Drawing;

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

if ((stepp->Text == "") || (right->Text == "")) {

MessageBox::Show(this, "Введите число.", "Error");

}

else

{

double limit2 = 0, step = 0, axy = 0, sol1 = 0, res=0;

double localstep;

step = System::Convert::ToDouble(stepp->Text);

limit2 = System::Convert::ToDouble(right->Text);

if ((step <= limit2) && (step > 0))

{

graph->Series[0]->Points->Clear();

localstep = trunc(limit2 / step);

if (localstep > 70000)

{

localstep = 70000;

step = limit2 / localstep;

for (int i = 0; i < localstep; i++)

{

axy = (double(2)\* sin(sol1));

graph->Series[0]->Points->AddXY(sol1, axy);

sol1 += step;

}

graph->ChartAreas[0]->AxisX->Maximum = limit2;

}

else

{

for (int i = 0; i <= localstep; i++)

{

axy = (double(2)\* sin(sol1));

graph->Series[0]->Points->AddXY(sol1, axy);

sol1 += step;

}

res = (double(2)\* sin(sol1));

graph->Series[0]->Points->AddXY(limit2, res);

}

graph->ChartAreas[0]->AxisX->Maximum = limit2;

//graph->ChartAreas[0]->AxisY->Maximum = ceil(axy);

}

else

MessageBox::Show(this, "Введите допустимое значение шага: не больше правой границы и больше нуля.", "Error");

Далее реализуем защиту от ошибок ввода с помощью KeyPress.

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

if ((e->KeyChar != (char)8) && (e->KeyChar < (char)48 || e->KeyChar >(char)57) && (e->KeyChar != (char)44)) {

{ MessageBox::Show(this, "Введен недопустимый символ. Введите число или запятую.", "Error"); }

e->Handled = true;

}

if (e->KeyChar == (char)44)

{

if (stepp->Text->Contains(","))

{

MessageBox::Show(this, "Слишком много запятых.", "Error");

e->Handled = true;

}

}

}

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

if ((e->KeyChar != (char)8) && (e->KeyChar < (char)48 || e->KeyChar >(char)57) && (e->KeyChar != (char)44)) {

{ MessageBox::Show(this, "Введен недопустимый символ. Введите число или запятую.", "Error"); }

e->Handled = true;

}

if (e->KeyChar == (char)44)

{

if (right->Text->Contains(","))

{

MessageBox::Show(this, "Слишком много запятых.", "Eror");

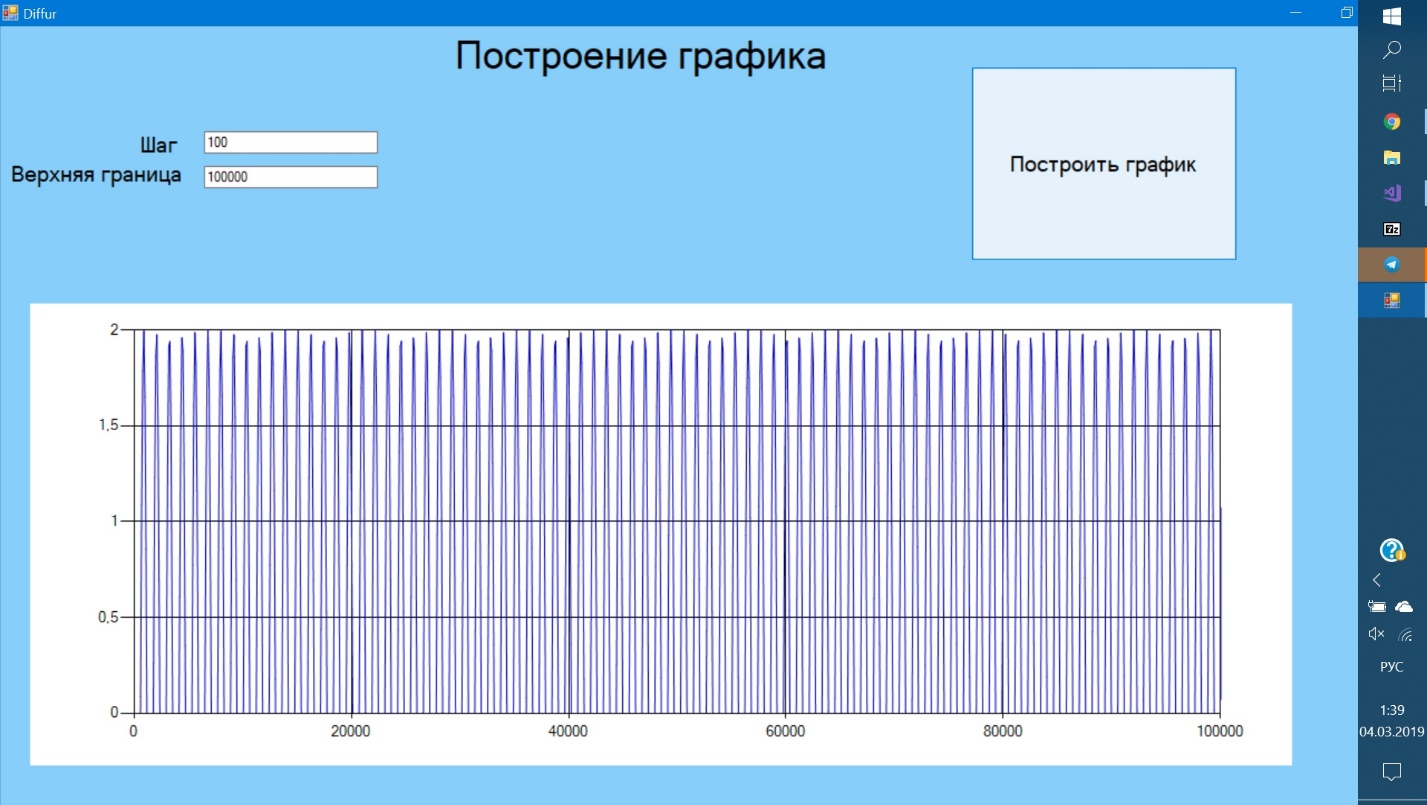
e->Handled = true;

}

}

}

**Демонстрация работы программы**



**Заключение**

В результате выполнения данной лабораторной работы было продемонстрировано построение графиков на формах.

**Приложение**

#pragma once

#include <iostream>

#include <cmath>

#include <string>

namespace Difference {

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>

/// Сводка для Diffur

/// </summary>

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

{

public:

Diffur(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~Diffur()

{

if (components)

{

delete components;

}

}

protected:

private: System::Windows::Forms::DataVisualization::Charting::Chart^ graph;

private: System::Windows::Forms::Button^ graf;

private: System::Windows::Forms::TextBox^ stepp;

private: System::Windows::Forms::TextBox^ right;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Label^ label3;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

System::Windows::Forms::DataVisualization::Charting::ChartArea^ chartArea2 = (gcnew System::Windows::Forms::DataVisualization::Charting::ChartArea());

System::Windows::Forms::DataVisualization::Charting::Series^ series2 = (gcnew System::Windows::Forms::DataVisualization::Charting::Series());

this->graph = (gcnew System::Windows::Forms::DataVisualization::Charting::Chart());

this->graf = (gcnew System::Windows::Forms::Button());

this->stepp = (gcnew System::Windows::Forms::TextBox());

this->right = (gcnew System::Windows::Forms::TextBox());

this->label1 = (gcnew System::Windows::Forms::Label());

this->label2 = (gcnew System::Windows::Forms::Label());

this->label3 = (gcnew System::Windows::Forms::Label());

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->graph))->BeginInit();

this->SuspendLayout();

//

// graph

//

chartArea2->AxisX->Crossing = 0;

chartArea2->AxisX->Minimum = 0;

chartArea2->AxisY->Crossing = 0;

chartArea2->AxisY->Minimum = 0;

chartArea2->Name = L"ChartArea1";

this->graph->ChartAreas->Add(chartArea2);

this->graph->Location = System::Drawing::Point(41, 381);

this->graph->Name = L"graph";

this->graph->Palette = System::Windows::Forms::DataVisualization::Charting::ChartColorPalette::Berry;

series2->ChartArea = L"ChartArea1";

series2->ChartType = System::Windows::Forms::DataVisualization::Charting::SeriesChartType::Line;

series2->Color = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(0)), static\_cast<System::Int32>(static\_cast<System::Byte>(0)),

static\_cast<System::Int32>(static\_cast<System::Byte>(192)));

series2->Name = L"Series1";

series2->XValueType = System::Windows::Forms::DataVisualization::Charting::ChartValueType::Double;

series2->YValueType = System::Windows::Forms::DataVisualization::Charting::ChartValueType::Double;

this->graph->Series->Add(series2);

this->graph->Size = System::Drawing::Size(1698, 636);

this->graph->TabIndex = 4;

//

// graf

//

this->graf->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 14, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->graf->Location = System::Drawing::Point(1306, 56);

this->graf->Name = L"graf";

this->graf->Size = System::Drawing::Size(358, 267);

this->graf->TabIndex = 7;

this->graf->Text = L"Построить график";

this->graf->UseVisualStyleBackColor = true;

this->graf->Click += gcnew System::EventHandler(this, &Diffur::graf\_Click);

//

// stepp

//

this->stepp->Location = System::Drawing::Point(275, 145);

this->stepp->Name = L"stepp";

this->stepp->ShortcutsEnabled = false;

this->stepp->Size = System::Drawing::Size(232, 26);

this->stepp->TabIndex = 8;

this->stepp->KeyPress += gcnew System::Windows::Forms::KeyPressEventHandler(this, &Diffur::stepp\_KeyPress);

//

// right

//

this->right->Location = System::Drawing::Point(275, 192);

this->right->Name = L"right";

this->right->ShortcutsEnabled = false;

this->right->Size = System::Drawing::Size(232, 26);

this->right->TabIndex = 9;

this->right->TextChanged += gcnew System::EventHandler(this, &Diffur::right\_TextChanged);

this->right->KeyPress += gcnew System::Windows::Forms::KeyPressEventHandler(this, &Diffur::right\_KeyPress);

//

// label1

//

this->label1->AutoSize = true;

this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 14, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label1->Location = System::Drawing::Point(183, 145);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(66, 32);

this->label1->TabIndex = 10;

this->label1->Text = L"Шаг";

this->label1->Click += gcnew System::EventHandler(this, &Diffur::label1\_Click);

//

// label2

//

this->label2->AutoSize = true;

this->label2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 14, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label2->Location = System::Drawing::Point(9, 185);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(240, 32);

this->label2->TabIndex = 11;

this->label2->Text = L"Верхняя граница";

this->label2->Click += gcnew System::EventHandler(this, &Diffur::label2\_Click);

//

// label3

//

this->label3->AutoSize = true;

this->label3->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 25, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label3->Location = System::Drawing::Point(601, 9);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(515, 58);

this->label3->TabIndex = 12;

this->label3->Text = L"Построение графика";

//

// Diffur

//

this->AutoScaleDimensions = System::Drawing::SizeF(9, 20);

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

this->BackColor = System::Drawing::Color::LightSkyBlue;

this->ClientSize = System::Drawing::Size(1779, 1047);

this->Controls->Add(this->label3);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->Controls->Add(this->right);

this->Controls->Add(this->stepp);

this->Controls->Add(this->graf);

this->Controls->Add(this->graph);

this->Name = L"Diffur";

this->Text = L"Diffur";

this->Load += gcnew System::EventHandler(this, &Diffur::Diffur\_Load);

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->graph))->EndInit();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

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

if ((stepp->Text == "") || (right->Text == "")) {

MessageBox::Show(this, "Введите число.", "Error");

}

else

{

double limit2 = 0, step = 0, axy = 0, sol1 = 0, res=0;

double localstep;

step = System::Convert::ToDouble(stepp->Text);

limit2 = System::Convert::ToDouble(right->Text);

if ((step <= limit2) && (step > 0))

{

graph->Series[0]->Points->Clear();

localstep = trunc(limit2 / step);

if (localstep > 70000)

{

localstep = 70000;

step = limit2 / localstep;

for (int i = 0; i < localstep; i++)

{

axy = (double(2)\* sin(sol1));

graph->Series[0]->Points->AddXY(sol1, axy);

sol1 += step;

}

graph->ChartAreas[0]->AxisX->Maximum = limit2;

}

else

{

for (int i = 0; i <= localstep; i++)

{

axy = (double(2)\* sin(sol1));

graph->Series[0]->Points->AddXY(sol1, axy);

sol1 += step;

}

res = (double(2)\* sin(sol1));

graph->Series[0]->Points->AddXY(limit2, res);

}

graph->ChartAreas[0]->AxisX->Maximum = limit2;

//graph->ChartAreas[0]->AxisY->Maximum = ceil(axy);

}

else

MessageBox::Show(this, "Введите допустимое значение шага: не больше правой границы и больше нуля.", "Error");

}

}

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

if ((e->KeyChar != (char)8) && (e->KeyChar < (char)48 || e->KeyChar >(char)57) && (e->KeyChar != (char)44)) {

{ MessageBox::Show(this, "Введен недопустимый символ. Введите число или запятую.", "Error"); }

e->Handled = true;

}

if (e->KeyChar == (char)44)

{

if (stepp->Text->Contains(","))

{

MessageBox::Show(this, "Слишком много запятых.", "Error");

e->Handled = true;

}

}

}

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

if ((e->KeyChar != (char)8) && (e->KeyChar < (char)48 || e->KeyChar >(char)57) && (e->KeyChar != (char)44)) {

{ MessageBox::Show(this, "Введен недопустимый символ. Введите число или запятую.", "Error"); }

e->Handled = true;

}

if (e->KeyChar == (char)44)

{

if (right->Text->Contains(","))

{

MessageBox::Show(this, "Слишком много запятых.", "Eror");

e->Handled = true;

}

}

}

private: System::Void dataGrid1\_Navigate(System::Object^ sender, System::Windows::Forms::NavigateEventArgs^ ne) {

}

private: System::Void label1\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void label2\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void right\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void Diffur\_Load(System::Object^ sender, System::EventArgs^ e) {

}

};

}