Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Удмуртский государственный университет»

Институт математики, информационных технологий и физики

Кафедра теоретических основ информатики

ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ

Срок прохождения практики: 13.07.2020 — 26.07.2020

**«Разработка приложения с графическим интерфейсом для вычисления периметра параллелограмма»**

Выполнил студент:

Котов Михаил Валерьевич

(ФИО студента)

Направления подготовки/специальности

Прикладная информатика

группы ОБ-09.03.03.д-11

(наименование группы)

Руководитель практики от кафедры:

Сапаров А.Ю. доцент кафедры ТОИ, к.т.н.

(ФИО, должность, ученое звание, ученая степень)

Итоговая оценка по учебной практике\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка, подпись руководителя)

Ижевск 2020 г.

**Оглавление:**

Введение…………………………………………………………………………………………………………….2Постановка задачи……………………………………………………………………………………………..3 Теоретический обзор………………………………………………………………………………………….4 Описание метода решения…………………………………………………………………………….5-9 Заключение……………………………………………………………………………………………………….10Список литературы……………………………………………………………………………………………11 Приложения…………………………………………………………………………………………………12-20

**Введение**

**Тема** данной работы и по сегодняшний день актуальна, любому современному пользователю удобней видеть перед своими глазами оконное приложение вместо консольного. При разработке приложения с графическим интерфейсом немало внимания уделяется удобству его использования, поэтому правильное умение создавать графические интерфейсы довольно востребовано.

**Целью** работы является выработка навыков разработки приложений с графическим интерфейсом в среде программирования Visual Studio.

**Задачи** которые необходимо решить для достижения поставленной цели:

1)Изучение литературы.

2)Формулировка требований к задаче.

3)Разработка программы.

4)Тестирование программы.

5)Составление отчёта по работе.

**Постановка задачи.**

Требуется разработать простейшее приложение с графическим пользовательским интерфейсом для вычисления периметра параллелограмма для вычисления по заданным: площади, основанию, острому углу.

Требования к интерфейсу:

• Интерфейс должен содержать поля для ввода данных, поле для вывода результата и кнопку для запуска вычисления.

• На форме должен быть чертеж (в виде рисунка PictureBox) с обозначением всех параметров, участвующих в вычислении.

• У формы должен быть соответствующий заголовок и иконка.

• При разворачивании формы на весь экран элементы управления не должны съезжать с места. Допускается растягивать элементы управления по ширине и по высоте.

• При разворачивании формы на весь экран картинка должна разворачиваться пропорционально размерам формы. При этом пропорции самой картинки должны сохраниться.

• При попытке закрытия окна пользователю должен выдаваться вопрос с текстом «Выйти из программы?»

• В полях ввода должна быть возможность ввода только корректных данных (Например, при попытке ввода текста вместо числа, должно выводиться соответствующее предупреждение).

• Поле вывода результата должно быть доступно только для чтения.

Требование к функциональности:

•Достоверность результатов вычисления

**Теоретический обзор:**

В основе графического пользовательского интерфейса лежит понятие формы (Form). Форма представляет окно или диалоговое окно, которое составляет пользовательский интерфейс приложения. Процесс разработки графического интерфейса заключается в создании формы и добавления на ней элементов управления. Элементы управления (Control) из себя представляют примитивные компоненты с визуальным представлением, имеющие стандартный вид и выполняющие стандартные действия. (1)

Каждый элемент управления имеет свой набор свойств, методов и событий.

Свойство элемента — это параметр, описывающий его состояние (внешний вид, координаты расположения на форме, доступность пользователю и т.д.).

Метод — это процедура или функция, определяющая те действия, которые можно выполнить над этим элементом управления либо может выполнить сам элемент.

Событие — это действие, распознаваемое элементом управления (манипуляции пользователя, действия операционной системы). Таким образом, для создания правильного элемента управления требуется задать нужные значения свойств, задать необходимые обработчики событий и корректно использовать методы.

•Для ввода/вывода текстовых или числовых данных принято использовать объект класса TextBox

•Для выполнения какого-либо действия используется объект класса Button

•Для вывода изображения используется объект класса PictureBox (3)

•Для отображения текста используется объект класса Label

**Описание метода решения задачи:**

1 Разработка программы.

Решение основной задачи, а именно разработка программы, выполняется в среде программирования Visual Studio 2017 с использованием проекта на языке «Visual C++» с типом “Среда CLR”, чтобы добавить пустую форму, нужно кликнуть по «исходные файлы» в обозревателе решений правой кнопкой мыши, нажать «добавить» -> «создать элемент», после этого нужно перейти в раздел «Visual C++» -> “UI”и выбрать «форма Windows Forms».

Чтобы всё заработало, нужно в *cpp файл вставить следующий код:*

#include "(название формы).h"

#include <Windows.h>

using namespace (название проекта);

int WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)

{

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

Application::Run(gcnew MyForm);

return 0;

}

и перезапустить Visual Studio. (4)

Теперь наш проект содержит одну пустую форму, требуется добавить на неё все необходимые элементы управления. Будем использовать:

•Ввод и вывод данных выполняется при помощи объектов типа TextBox. Свойства этих объектов содержат:

Anchor(привязывает элемент к граням контейнера)

Font(определяет шрифт и размер шрифта элемента)

ReadOnly(задаёт, можно ли изменять текст в поле редактирования )

Location(координаты элемента)

Size(размер элемента)

События этих объектов:

KeyPress(возникает когда этот элемент находится в фокусе и пользователь нажимает и отпускает клавишу)

•Вывод изображения выполняется при помощи объекта типа PictureBox.

Свойства этого объекта содержат:

Image(изображение отображаемое в PictureBox)

BackColor(фоновый цвет компонента)

Anchor(привязывает элемент к граням контейнера)

Size(размер элемента)

SizeMode(определяет как будет обрабатываться размещение рисунка и его изменение)

• Основное действие выполняется при нажатии на кнопку «Calculate». Описание кнопки:

this->button1->Anchor = System::Windows::Forms::AnchorStyles::Bottom;

this->button1->AutoSize = true;

this->button1->Location = System::Drawing::Point(136, 205);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(75, 23);

this->button1->TabIndex = 0;

this->button1->Text = L"Calculate!";

this->button1->UseVisualStyleBackColor = true;

Действие при нажатии кнопки:

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

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

float s = Convert::ToSingle(textBox1->Text);

float a = Convert::ToSingle(textBox2->Text);

float alpha = Convert::ToSingle(textBox3->Text);

float p = 2 \* a + 2 \* (s / a) / Math::Sin(alpha \* (Math::PI / 180));

textBox4->Text = p.ToString();

}

Для ограничения ввода используются различные проверки:

private: System::Void textBox1\_KeyPress(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e) {

// не даём ввести первым знаком ","

if (textBox1->TextLength == 0 && e->KeyChar == ',') {

e->Handled = true;//не вводим то не число

return;

}

if (Char::IsDigit(e->KeyChar)) { //если в textBox1 жмём цифру то всё ок

return;

}

if (e->KeyChar == (char)Keys::Back) {//если в textBox1 жмём backspace то всё ок

return;

}

if (e->KeyChar == ',') {

return;

}

//если пользователь вводит не число, то

System::Windows::Forms::MessageBox::Show("please, enter the number", "Error");//мы его осведомляем

e->Handled = true;//не вводим тот знак

}

И так для каждого TextBox`a у которого свойство ReadOnly = false.

Для предотвращения ошибок при расчётах используем проверки:

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

//если все или хотя бы один textBox пустой,то вызываем сообщение и пропускаем дальнейшую обработку

if (textBox1->Text == "" || textBox2->Text == "" || textBox3->Text == "") {

System::Windows::Forms::MessageBox::Show("Введены неверные данные", "Ошибка");

return;

}

msclr::interop::marshal\_context context;

//Дальше идут проверки каждого textBox`a на наличие лишних ","

//textBox 1

string std\_string = context.marshal\_as<std::string>(textBox1->Text);

//string str = std\_string;

int count = 0;

for (int i = 0; i < std\_string.size(); i++) {

if (std\_string[i] == ',') {

count++;

}

}

if (count >= 2) {

System::Windows::Forms::MessageBox::Show("Введены неверные данные", "Ошибка");

return;

}

if (std\_string[0] == ',') {

System::Windows::Forms::MessageBox::Show("Введены неверные данные", "Ошибка");

return;

}

//textBox 2

std\_string = context.marshal\_as<std::string>(textBox2->Text);

//string str = std\_string;

count = 0;

for (int i = 0; i < std\_string.size(); i++) {

if (std\_string[i] == ',') {

count++;

}

}

if (count >= 2) {

System::Windows::Forms::MessageBox::Show("Введены неверные данные", "Ошибка");

return;

}

if (std\_string[0] == ',') {

System::Windows::Forms::MessageBox::Show("Введены неверные данные", "Ошибка");

return;

}

//textBox 3

std\_string = context.marshal\_as<std::string>(textBox3->Text);

//string str = std\_string;

count = 0;

for (int i = 0; i < std\_string.size(); i++) {

if (std\_string[i] == ',') {

count++;

}

}

if (count >= 2) {

System::Windows::Forms::MessageBox::Show("Введены неверные данные", "Ошибка");

return;

}

if (std\_string[0] == ',') {

System::Windows::Forms::MessageBox::Show("Введены неверные данные", "Ошибка");

return;

}

float s = Convert::ToSingle(textBox1->Text);

float a = Convert::ToSingle(textBox2->Text);

float alpha = Convert::ToSingle(textBox3->Text);

//проверка на ввод реальных значений

if (alpha >= 90 || alpha == 0) {

System::Windows::Forms::MessageBox::Show("Введены неверные данные", "Ошибка");

return;

}

if (a == 0 || s == 0) {

System::Windows::Forms::MessageBox::Show("Введены неверные данные", "Ошибка");

return;

}

//float h = s / a;

//float radians = (alpha \* ( Math::PI / 180));

//float b = h / Math::Sin(radians);

float p = 2 \* a + 2 \* (s / a) / Math::Sin(alpha \* (Math::PI / 180));

//float p = 2 \* a + 2 \* b;

textBox4->Text = p.ToString();

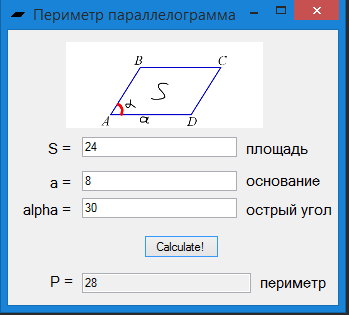
}

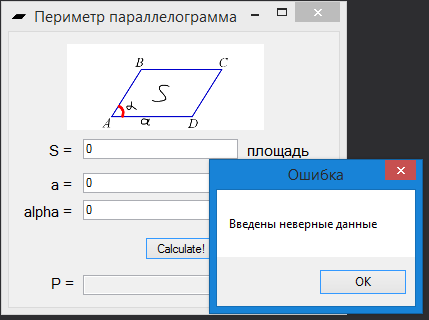
2 Тестирование программы.

Для проверки правильности работы программы выполнено тестирование на входных данных, приведенных в таблице 1. Первая колонка таблицы содержит значение площади, вторая - острого угла, третья - основания, четвёртая — правильное значение периметра параллелограмма, пятая — результат вычисления в разработанной программе.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Площадь | Основание | Острый угол | Ожидаемое значение Р | Результат |
| 15 | 5 | 60 | 16,9282 | 16,9282 |
| 24 | 8 | 45 | 24,48528 | 24,48528 |
| 12 | 2 | 30 | 28 | 28 |
| 12,0 | 2,0 | 30,0 | 28,0 | 28 |
| 16 | 4 | 96 | Введены неверные данные | Введены неверные данные |
| 0 | 1 | ,0 | Введены неверные данные | Введены неверные данные |
|  |  |  | Введены неверные данные | Введены неверные данные |

При всех тестовых входных данных получен результат, совпадающий c ожидаемым результатом. Отсюда следует, что программа работает исправно.





**Заключение**

Во время прохождения практики были выполнены все поставленные задачи, а именно:

Реализовать собственный продукт с графическим интерфейсом на языке программирования С++.

Разработанная программа протестирована, ошибок не выявлено.

Также в ходе практики были получены все базовые и необходимые навыки для создания собственного приложения с использованием классов Windows Forms и развито умение работы в среде Visual Studio.

Самая большая проблема в ходе работы заключалось в том, что нельзя было присвоить переменной типа string значение из TextBox`a из – за несовпадения типов, пример:

string qw;

qw = textBox1->Text;

Пришлось пользоваться различными хитростями. (5)

**Список литературы**

1. Пространство имен System.Windows.Forms [Электронный ресурс] / Microsoft — Электрон. дан. — Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.forms?view=netcore-3.1>

2. Программирование на С, C# и Java[Электронный ресурс] – Режим доступа: <https://vscode.ru/articles/rastyanut-elementyi-na-forme-windows-forms.html>

3. Элемент PictureBox[электронный ресурс] – Режим доступа: <https://metanit.com/sharp/windowsforms/4.16.php>

4. Как создать Windows Forms проект на C++[электронный ресурс] – Режим доступа: <https://programforyou.ru/poleznoe/how-to-create-windows-forms-cpp-project>

5. Как использовать текст textbox в форме c ++?[электронный ресурс] – Режим доступа: <https://web-answers.ru/c/kak-ispolzovat-tekst-textbox-v-forme-c.html>

**Приложения**

**Код модуля формы:**

#pragma once

#include <msclr\marshal\_cppstd.h>

#include <string>

#include <iostream>

using namespace std;

//void paint(TextBox ^a);

namespace pr {

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 MyForm

/// </summary>

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

{

public:

MyForm(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~MyForm()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button1;

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

private: System::Windows::Forms::PictureBox^ pictureBox1;

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

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

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::TextBox^ textBox2;

private: System::Windows::Forms::TextBox^ textBox3;

private: System::Windows::Forms::Label^ label4;

private: System::Windows::Forms::TextBox^ textBox4;

private: System::Windows::Forms::Label^ label5;

private: System::Windows::Forms::Label^ label6;

private: System::Windows::Forms::Label^ label7;

private: System::Windows::Forms::Label^ label8;

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::ComponentModel::ComponentResourceManager^ resources = (gcnew System::ComponentModel::ComponentResourceManager(MyForm::typeid));

this->button1 = (gcnew System::Windows::Forms::Button());

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

this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());

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

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

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

this->textBox3 = (gcnew System::Windows::Forms::TextBox());

this->label4 = (gcnew System::Windows::Forms::Label());

this->textBox4 = (gcnew System::Windows::Forms::TextBox());

this->label5 = (gcnew System::Windows::Forms::Label());

this->label6 = (gcnew System::Windows::Forms::Label());

this->label7 = (gcnew System::Windows::Forms::Label());

this->label8 = (gcnew System::Windows::Forms::Label());

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->BeginInit();

this->SuspendLayout();

//

// button1

//

this->button1->Anchor = System::Windows::Forms::AnchorStyles::Bottom;

this->button1->AutoSize = true;

this->button1->Location = System::Drawing::Point(136, 205);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(75, 23);

this->button1->TabIndex = 0;

this->button1->Text = L"Calculate!";

this->button1->UseVisualStyleBackColor = true;

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

//

// label1

//

this->label1->Anchor = static\_cast<System::Windows::Forms::AnchorStyles>(((System::Windows::Forms::AnchorStyles::Bottom | System::Windows::Forms::AnchorStyles::Left)

| System::Windows::Forms::AnchorStyles::Right));

this->label1->AutoSize = true;

this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

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

this->label1->Location = System::Drawing::Point(37, 109);

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

this->label1->Size = System::Drawing::Size(31, 18);

this->label1->TabIndex = 1;

this->label1->Text = L"S =";

//

// pictureBox1

//

this->pictureBox1->Anchor = static\_cast<System::Windows::Forms::AnchorStyles>((((System::Windows::Forms::AnchorStyles::Top | System::Windows::Forms::AnchorStyles::Bottom)

| System::Windows::Forms::AnchorStyles::Left)

| System::Windows::Forms::AnchorStyles::Right));

this->pictureBox1->BackColor = System::Drawing::SystemColors::ButtonHighlight;

this->pictureBox1->Image = (cli::safe\_cast<System::Drawing::Image^>(resources->GetObject(L"pictureBox1.Image")));

this->pictureBox1->Location = System::Drawing::Point(58, 12);

this->pictureBox1->Name = L"pictureBox1";

this->pictureBox1->Size = System::Drawing::Size(197, 86);

this->pictureBox1->SizeMode = System::Windows::Forms::PictureBoxSizeMode::Zoom;

this->pictureBox1->TabIndex = 2;

this->pictureBox1->TabStop = false;

//

// label2

//

this->label2->Anchor = static\_cast<System::Windows::Forms::AnchorStyles>(((System::Windows::Forms::AnchorStyles::Bottom | System::Windows::Forms::AnchorStyles::Left)

| System::Windows::Forms::AnchorStyles::Right));

this->label2->AutoSize = true;

this->label2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

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

this->label2->Location = System::Drawing::Point(39, 143);

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

this->label2->Size = System::Drawing::Size(29, 18);

this->label2->TabIndex = 3;

this->label2->Text = L"a =";

//

// label3

//

this->label3->Anchor = static\_cast<System::Windows::Forms::AnchorStyles>(((System::Windows::Forms::AnchorStyles::Bottom | System::Windows::Forms::AnchorStyles::Left)

| System::Windows::Forms::AnchorStyles::Right));

this->label3->AutoSize = true;

this->label3->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

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

this->label3->Location = System::Drawing::Point(12, 170);

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

this->label3->Size = System::Drawing::Size(56, 18);

this->label3->TabIndex = 4;

this->label3->Text = L"alpha =";

//

// textBox1

//

this->textBox1->Anchor = static\_cast<System::Windows::Forms::AnchorStyles>(((System::Windows::Forms::AnchorStyles::Bottom | System::Windows::Forms::AnchorStyles::Left)

| System::Windows::Forms::AnchorStyles::Right));

this->textBox1->Location = System::Drawing::Point(74, 107);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(155, 20);

this->textBox1->TabIndex = 5;

this->textBox1->KeyPress += gcnew System::Windows::Forms::KeyPressEventHandler(this, &MyForm::textBox1\_KeyPress);

//

// textBox2

//

this->textBox2->Anchor = static\_cast<System::Windows::Forms::AnchorStyles>(((System::Windows::Forms::AnchorStyles::Bottom | System::Windows::Forms::AnchorStyles::Left)

| System::Windows::Forms::AnchorStyles::Right));

this->textBox2->Location = System::Drawing::Point(74, 141);

this->textBox2->Name = L"textBox2";

this->textBox2->Size = System::Drawing::Size(155, 20);

this->textBox2->TabIndex = 6;

this->textBox2->KeyPress += gcnew System::Windows::Forms::KeyPressEventHandler(this, &MyForm::textBox2\_KeyPress);

//

// textBox3

//

this->textBox3->Anchor = static\_cast<System::Windows::Forms::AnchorStyles>(((System::Windows::Forms::AnchorStyles::Bottom | System::Windows::Forms::AnchorStyles::Left)

| System::Windows::Forms::AnchorStyles::Right));

this->textBox3->Location = System::Drawing::Point(74, 168);

this->textBox3->Name = L"textBox3";

this->textBox3->Size = System::Drawing::Size(155, 20);

this->textBox3->TabIndex = 7;

this->textBox3->KeyPress += gcnew System::Windows::Forms::KeyPressEventHandler(this, &MyForm::textBox3\_KeyPress);

//

// label4

//

this->label4->Anchor = static\_cast<System::Windows::Forms::AnchorStyles>(((System::Windows::Forms::AnchorStyles::Bottom | System::Windows::Forms::AnchorStyles::Left)

| System::Windows::Forms::AnchorStyles::Right));

this->label4->AutoSize = true;

this->label4->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

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

this->label4->Location = System::Drawing::Point(39, 242);

this->label4->Name = L"label4";

this->label4->Size = System::Drawing::Size(31, 18);

this->label4->TabIndex = 8;

this->label4->Text = L"P =";

//

// textBox4

//

this>textBox4>Anchor=static\_cast<System::Windows::Forms::AnchorStyles>(((System::Windows::Forms::AnchorStyles::Bottom | System::Windows::Forms::AnchorStyles::Left)

| System::Windows::Forms::AnchorStyles::Right));

this->textBox4->Location = System::Drawing::Point(74, 243);

this->textBox4->Name = L"textBox4";

this->textBox4->ReadOnly = true;

this->textBox4->Size = System::Drawing::Size(169, 20);

this->textBox4->TabIndex = 9;

//

// label5

//

this->label5->Anchor = static\_cast<System::Windows::Forms::AnchorStyles>((System::Windows::Forms::AnchorStyles::Bottom | System::Windows::Forms::AnchorStyles::Right));

this->label5->AutoSize = true;

this->label5->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

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

this->label5->Location = System::Drawing::Point(235, 109);

this->label5->Name = L"label5";

this->label5->Size = System::Drawing::Size(70, 18);

this->label5->TabIndex = 10;

this->label5->Text = L"площадь";

//

// label6

//

this->label6->Anchor = static\_cast<System::Windows::Forms::AnchorStyles>((System::Windows::Forms::AnchorStyles::Bottom | System::Windows::Forms::AnchorStyles::Right));

this->label6->AutoSize = true;

this->label6->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

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

this->label6->Location = System::Drawing::Point(235, 141);

this->label6->Name = L"label6";

this->label6->Size = System::Drawing::Size(82, 18);

this->label6->TabIndex = 11;

this->label6->Text = L"основание";

//

// label7

//

this->label7->Anchor = static\_cast<System::Windows::Forms::AnchorStyles>((System::Windows::Forms::AnchorStyles::Bottom | System::Windows::Forms::AnchorStyles::Right));

this->label7->AutoSize = true;

this->label7->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

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

this->label7->Location = System::Drawing::Point(235, 170);

this->label7->Name = L"label7";

this->label7->Size = System::Drawing::Size(94, 18);

this->label7->TabIndex = 12;

this->label7->Text = L"острый угол";

//

// label8

//

this->label8->Anchor = static\_cast<System::Windows::Forms::AnchorStyles>((System::Windows::Forms::AnchorStyles::Bottom | System::Windows::Forms::AnchorStyles::Right));

this->label8->AutoSize = true;

this->label8->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

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

this->label8->Location = System::Drawing::Point(249, 243);

this->label8->Name = L"label8";

this->label8->Size = System::Drawing::Size(74, 18);

this->label8->TabIndex = 13;

this->label8->Text = L"периметр";

//

// MyForm

//

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

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

this->ClientSize = System::Drawing::Size(330, 275);

this->Controls->Add(this->label8);

this->Controls->Add(this->label7);

this->Controls->Add(this->label6);

this->Controls->Add(this->label5);

this->Controls->Add(this->textBox4);

this->Controls->Add(this->label4);

this->Controls->Add(this->textBox3);

this->Controls->Add(this->textBox2);

this->Controls->Add(this->textBox1);

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

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

this->Controls->Add(this->pictureBox1);

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

this->Controls->Add(this->button1);

this->Icon = (cli::safe\_cast<System::Drawing::Icon^>(resources->GetObject(L"$this.Icon")));

this->MinimumSize = System::Drawing::Size(346, 314);

this->Name = L"MyForm";

this->Text = L"Периметр параллелограмма";

this->FormClosing += gcnew System::Windows::Forms::FormClosingEventHandler(this, &MyForm::MyForm\_FormClosing);

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pictureBox1))->EndInit();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

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

//при нажатии на кнопку выходит окно с текстом

//System::Windows::Forms::MessageBox::Show("Текст сообщения", "заголовок окна с сообщением");

//если все или хотя бы один textBox пустой,то вызываем сообщение и пропускаем дальнейшую обработку

if (textBox1->Text == "" || textBox2->Text == "" || textBox3->Text == "") {

System::Windows::Forms::MessageBox::Show("Введены неверные данные", "Ошибка");

return;

}

msclr::interop::marshal\_context context;

//Дальше идут проверки каждого textBox`a на наличие лишних ","

//textBox 1

string std\_string = context.marshal\_as<std::string>(textBox1->Text);

//string str = std\_string;

int count = 0;

for (int i = 0; i < std\_string.size(); i++) {

if (std\_string[i] == ',') {

count++;

}

}

if (count >= 2) {

System::Windows::Forms::MessageBox::Show("Введены неверные данные", "Ошибка");

return;

}

if (std\_string[0] == ',') {

System::Windows::Forms::MessageBox::Show("Введены неверные данные", "Ошибка");

return;

}

//textBox 2

std\_string = context.marshal\_as<std::string>(textBox2->Text);

//string str = std\_string;

count = 0;

for (int i = 0; i < std\_string.size(); i++) {

if (std\_string[i] == ',') {

count++;

}

}

if (count >= 2) {

System::Windows::Forms::MessageBox::Show("Введены неверные данные", "Ошибка");

return;

}

if (std\_string[0] == ',') {

System::Windows::Forms::MessageBox::Show("Введены неверные данные", "Ошибка");

return;

}

//textBox 3

std\_string = context.marshal\_as<std::string>(textBox3->Text);

//string str = std\_string;

count = 0;

for (int i = 0; i < std\_string.size(); i++) {

if (std\_string[i] == ',') {

count++;

}

}

if (count >= 2) {

System::Windows::Forms::MessageBox::Show("Введены неверные данные", "Ошибка");

return;

}

if (std\_string[0] == ',') {

System::Windows::Forms::MessageBox::Show("Введены неверные данные", "Ошибка");

return;

}

float s = Convert::ToSingle(textBox1->Text);

float a = Convert::ToSingle(textBox2->Text);

float alpha = Convert::ToSingle(textBox3->Text);

//проверка на ввод реальных значений

if (alpha >= 90 || alpha == 0) {

System::Windows::Forms::MessageBox::Show("Введены неверные данные", "Ошибка");

return;

}

if (a == 0 || s == 0) {

System::Windows::Forms::MessageBox::Show("Введены неверные данные", "Ошибка");

return;

}

//float h = s / a;

//float radians = (alpha \* ( Math::PI / 180));

//float b = h / Math::Sin(radians);

float p = 2 \* a + 2 \* (s / a) / Math::Sin(alpha \* (Math::PI / 180));

//float p = 2 \* a + 2 \* b;

textBox4->Text = p.ToString();

}

private: System::Void textBox1\_KeyPress(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e) {

// не даём ввести первым знаком ","

if (textBox1->TextLength == 0 && e->KeyChar == ',') {

e->Handled = true;//не вводим то не число

return;

}

if (Char::IsDigit(e->KeyChar)) { //если в textBox1 жмём цифру то всё ок

return;

}

if (e->KeyChar == (char)Keys::Back) {//если в textBox1 жмём backspace то всё ок

return;

}

if (e->KeyChar == ',') {

return;

}

//если пользователь вводит не число, то

System::Windows::Forms::MessageBox::Show("Пожалуйста, введите число", "Ошибка");//мы его осведомляем

e->Handled = true;//не вводим тот знак

}

private: System::Void MyForm\_FormClosing(System::Object^ sender, System::Windows::Forms::FormClosingEventArgs^ e) {

System::Windows::Forms::DialogResult status = MessageBox::Show("Вы хотите выйти?", "Периметр параллелограмма", MessageBoxButtons::YesNo);

if (status == System::Windows::Forms::DialogResult::No) {

e->Cancel = true;

}

}

private: System::Void textBox2\_KeyPress(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e) {

// не даём ввести первым знаком ","

if ( textBox2->TextLength == 0 && e->KeyChar == ',') {

e->Handled = true;//не вводим то не число

return;

}

if (Char::IsDigit(e->KeyChar)) { //если в textBox1 жмём цифру то всё ок

return;

}

if (e->KeyChar == (char)Keys::Back) {//если в textBox1 жмём backspace то всё ок

return;

}

if (e->KeyChar == ',') {

return;

}

//если пользователь вводит не число, то

System::Windows::Forms::MessageBox::Show("Пожалуйста, введите число", "Ошибка");//мы его осведомляем

e->Handled = true;//не вводим тот знак

}

private: System::Void textBox3\_KeyPress(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e) {

// не даём ввести первым знаком ","

if (textBox3->TextLength == 0 && e->KeyChar == ',') {

e->Handled = true;//не вводим то не число

return;

}

if (Char::IsDigit(e->KeyChar)) { //если в textBox1 жмём цифру то всё ок

return;

}

if (e->KeyChar == (char)Keys::Back) {//если в textBox1 жмём backspace то всё ок

return;

}

if (e->KeyChar == ',') {

return;

}

//если пользователь вводит не число, то

System::Windows::Forms::MessageBox::Show("Пожалуйста, введите число", "Ошибка");//мы его осведомляем

e->Handled = true;//не вводим тот знак

}

};

}

**Код из «MyForm.cpp»**

#include "MyForm.h"

#include <Windows.h>

using namespace pr;

int WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)

{

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

Application::Run(gcnew MyForm);

return 0;

}