ФГАОУ ВО Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа киберфизических систем и управления

**ОТЧЕТ**

**по лабораторной работе №6**

**по теме «Графика. Рисование на форме»**

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.

Разработка программы

Создаем форму. Добавляем необходимый элемент: поле для рисунка – PictureBox.  
Создадим программу, которая будет рисовать определённый рисунок.

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

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

Создадим главную функцию, которая будет отвечать за рисование примитивной картинки.

void Draw()

{   
Проверка наличия картинки в PictureBox, очистка.

if(MyPictureBox->BackgroundImage == nullptr)

MyPictureBox->BackgroundImage = gcnew Bitmap(MyPictureBox->Width, MyPictureBox->Height);

Graphics^ buffGraphics = Graphics::FromImage(MyPictureBox->BackgroundImage);

buffGraphics->Clear(MyPictureBox->BackColor);

Создание точек и прорисовка рисунка с помощью линий, построение фигур, закраска определённым цветом.

Point point1 = Point(2\*(MyPictureBox->Width/3), MyPictureBox->Top);

Point point2 = Point(MyPictureBox->Right, MyPictureBox->Top);

Point point3 = Point(MyPictureBox->Width/3, MyPictureBox->Bottom);

Point point4 = Point(MyPictureBox->Right, MyPictureBox->Bottom);

array<Point>^ curvePoints = { point1, point2, point4, point3 };

System::Drawing::Drawing2D::GraphicsPath^ gg = gcnew System::Drawing::Drawing2D::GraphicsPath();

gg->AddLines(curvePoints);

buffGraphics->FillPath(Brushes::Black, gg);

buffGraphics->FillEllipse(Brushes::White, MyPictureBox->Width / 2.5, (MyPictureBox->Height / 3), MyPictureBox->Width / 3, MyPictureBox->Width / 3);

Point point1tri = Point((MyPictureBox->Width / 4), MyPictureBox->Height / 1.5);

Point point2tri = Point(3\*(MyPictureBox->Width / 7), MyPictureBox->Height / 3);

Point point3tri = Point(5\*(MyPictureBox->Width / 7), 3\*(MyPictureBox->Height/5));

array<Point>^ TrianglePoints = { point1tri, point2tri, point3tri };

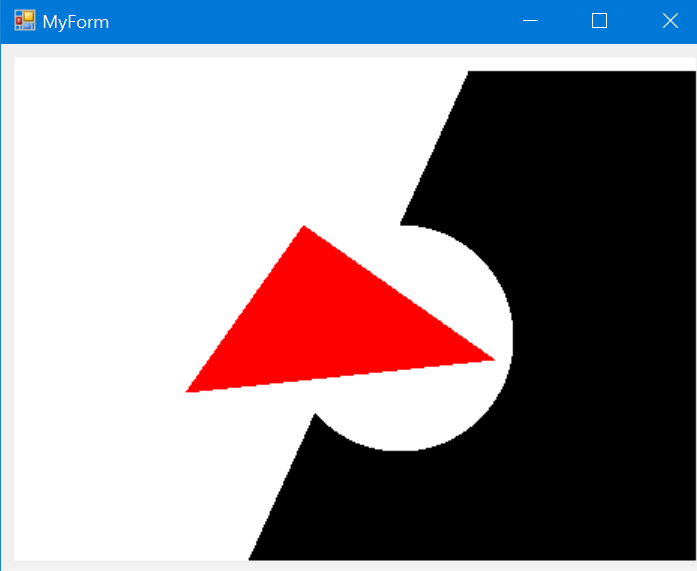
System::Drawing::Drawing2D::GraphicsPath^ PaintedTriangle = gcnew System::Drawing::Drawing2D::GraphicsPath();

PaintedTriangle->AddLines(TrianglePoints);

buffGraphics->FillPath(Brushes::Red, PaintedTriangle);

MyPictureBox->Invalidate();}

**Демонстрация работы программы**



**Заключение**

В результате выполнения данной лабораторной работы было продемонстрировано рисование примитивных рисунков на форме.

**Приложение**

#pragma once

namespace Lab6Drawing {

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();

Draw();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~MyForm()

{

if (components)

{

delete components;

}

}

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

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)

{

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

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

this->SuspendLayout();

//

// MyPictureBox

//

this->MyPictureBox->BackColor = System::Drawing::Color::White;

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

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

this->MyPictureBox->Size = System::Drawing::Size(454, 335);

this->MyPictureBox->TabIndex = 0;

this->MyPictureBox->TabStop = false;

//

// MyForm

//

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

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

this->ClientSize = System::Drawing::Size(469, 351);

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

this->Name = L"MyForm";

this->Text = L"MyForm";

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

this->ResumeLayout(false);

}

void Draw()

{

if(MyPictureBox->BackgroundImage == nullptr)

MyPictureBox->BackgroundImage = gcnew Bitmap(MyPictureBox->Width, MyPictureBox->Height);

Graphics^ buffGraphics = Graphics::FromImage(MyPictureBox->BackgroundImage);

buffGraphics->Clear(MyPictureBox->BackColor);

Point point1 = Point(2\*(MyPictureBox->Width/3), MyPictureBox->Top);

Point point2 = Point(MyPictureBox->Right, MyPictureBox->Top);

Point point3 = Point(MyPictureBox->Width/3, MyPictureBox->Bottom);

Point point4 = Point(MyPictureBox->Right, MyPictureBox->Bottom);

array<Point>^ curvePoints = { point1, point2, point4, point3 };

System::Drawing::Drawing2D::GraphicsPath^ gg = gcnew System::Drawing::Drawing2D::GraphicsPath();

gg->AddLines(curvePoints);

buffGraphics->FillPath(Brushes::Black, gg);

buffGraphics->FillEllipse(Brushes::White, MyPictureBox->Width / 2.5, (MyPictureBox->Height / 3), MyPictureBox->Width / 3, MyPictureBox->Width / 3);

Point point1tri = Point((MyPictureBox->Width / 4), MyPictureBox->Height / 1.5);

Point point2tri = Point(3\*(MyPictureBox->Width / 7), MyPictureBox->Height / 3);

Point point3tri = Point(5\*(MyPictureBox->Width / 7), 3\*(MyPictureBox->Height/5));

array<Point>^ TrianglePoints = { point1tri, point2tri, point3tri };

System::Drawing::Drawing2D::GraphicsPath^ PaintedTriangle = gcnew System::Drawing::Drawing2D::GraphicsPath();

PaintedTriangle->AddLines(TrianglePoints);

buffGraphics->FillPath(Brushes::Red, PaintedTriangle);

MyPictureBox->Invalidate();

}

#pragma endregion

};

}