Министерство образования Республики Беларусь

Учреждение образования

Брестский государственный технический университет

Кафедра ИИТ

Лабораторная работа №7

За 5 семестр

По дисциплине «Современные платформы программирования»

Выполнил: студент 3 курса

Группы ПО-4(2)

Коташевич С.Н.

Проверил: Монтик Н. С.

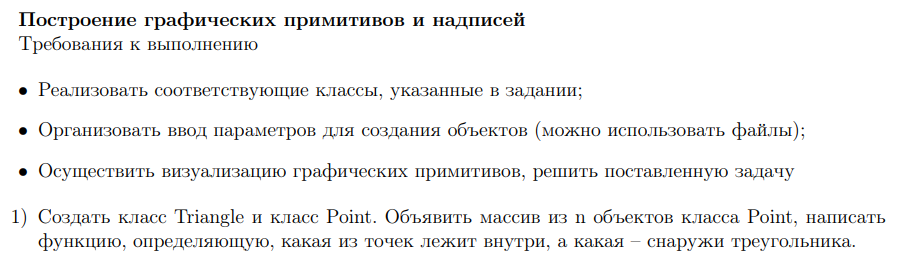
Брест 2021

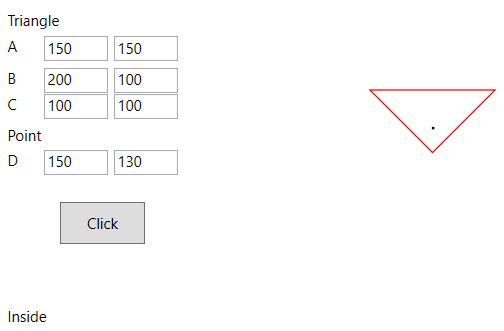
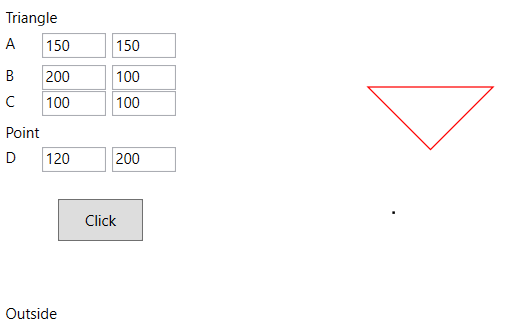
Лабораторная работа №7

Вариант 1

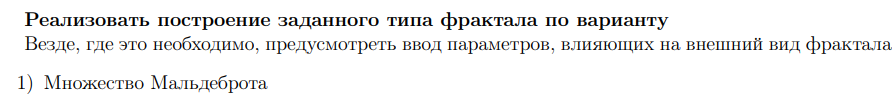
Цель: освоить возможности языка программирования C# в построении графических приложений.

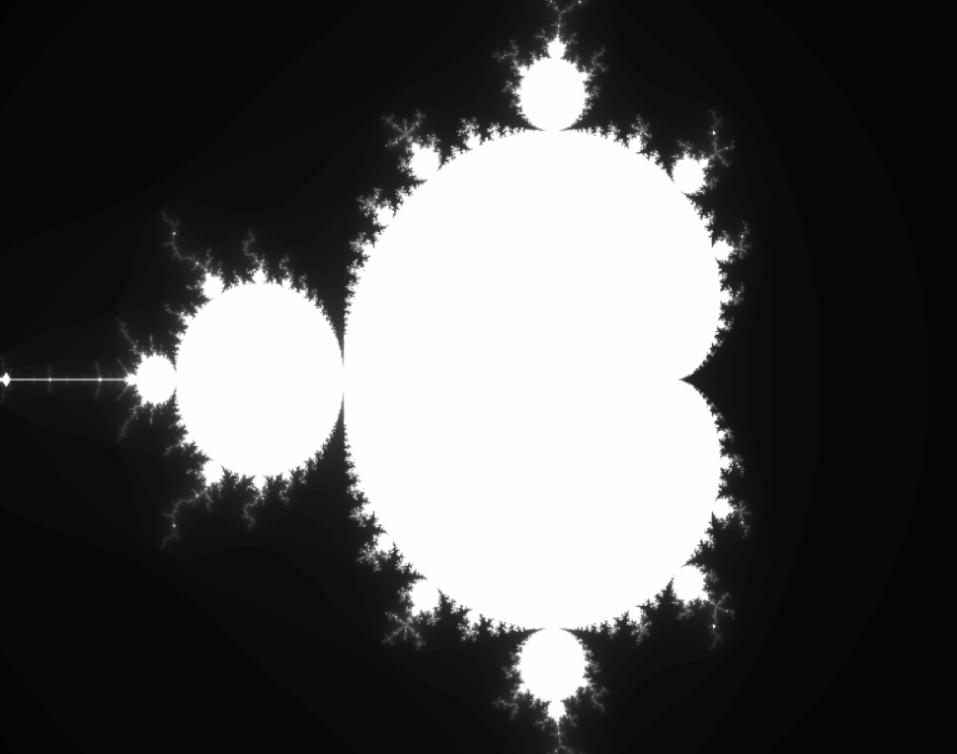
Задание 1





Задание 2





Текст программы:

Задание 1

MainWindow.xaml.cs

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Media;

using System.Windows.Shapes;

namespace lab07SPP

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void ButtonBase\_OnClick(object sender, RoutedEventArgs e)

{

var element = MyGrid.Children

.OfType<Shape>()

.FirstOrDefault(e => e.Name == "triangle");

MyGrid.Children.Remove(element);

element = MyGrid.Children

.OfType<Shape>()

.FirstOrDefault(e => e.Name == "point");

MyGrid.Children.Remove(element);

Polygon trianglePolygon = new Polygon()

{

Name = "triangle"

};

Polygon point = new Polygon()

{

Name = "point"

};

double[] x = {double.Parse(X0.Text), double.Parse(X1.Text), double.Parse(X2.Text), double.Parse(X3.Text)};

double[] y = { double.Parse(Y0.Text), double.Parse(Y1.Text), double.Parse(Y2.Text), double.Parse(Y3.Text) };

Point aPoint = new Point(x[1], y[1]);

Point bPoint = new Point(x[2], y[2]);

Point cPoint = new Point(x[3], y[3]);

Point dPoint = new Point(x[0], y[0]);

Point dPoint1 = new Point(x[0] + 1, y[0]);

Point dPoint2 = new Point(x[0] + 1, y[0] + 1);

Point dPoint3 = new Point(x[0], y[0] + 1);

trianglePolygon.Points.Add(aPoint);

trianglePolygon.Points.Add(bPoint);

trianglePolygon.Points.Add(cPoint);

trianglePolygon.Stroke = Brushes.Red;

point.Points.Add(dPoint);

point.Points.Add(dPoint1);

point.Points.Add(dPoint2);

point.Points.Add(dPoint3);

point.Stroke = Brushes.Black;

MyGrid.Children.Add(trianglePolygon);

MyGrid.Children.Add(point);

Grid.SetColumn(trianglePolygon, 2);

Grid.SetColumn(point, 2);

Result.Text = IsInside(x, y) ? "Inside" : "Outside";

}

private static bool IsInside(double[] x, double[] y)

{

double[] f = new double[3];

f[0] = (x[1] - x[0]) \* (y[2] - y[1]) - (x[2] - x[1]) \* (y[1] - y[0]);

f[1] = (x[2] - x[0]) \* (y[3] - y[2]) - (x[3] - x[2]) \* (y[2] - y[0]);

f[2] = (x[3] - x[0]) \* (y[1] - y[3]) - (x[1] - x[3]) \* (y[3] - y[0]);

if ((f[0] >= 0 && f[1] >= 0 && f[2] >= 0) || (f[0] <= 0 && f[1] <= 0 && f[2] <= 0))

{

return true;

}

return false;

}

}

}

MainWindow.xaml

<Window x:Class="lab07SPP.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Grid x:Name="MyGrid">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="200"></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

</Grid.ColumnDefinitions>

<TextBox Grid.Column="0" x:Name="X1" HorizontalAlignment="Left" Margin="39,57,0,0" VerticalAlignment="Top" Height="20" Width="51"/>

<TextBox Grid.Column="0" x:Name="Y1" HorizontalAlignment="Left" Margin="95,57,0,0" VerticalAlignment="Top" Height="20" Width="51"/>

<TextBox Grid.Column="0" x:Name="X2" HorizontalAlignment="Left" Margin="39,82,0,0" VerticalAlignment="Top" Height="20" Width="51"/>

<TextBox Grid.Column="0" x:Name="Y2" HorizontalAlignment="Left" Margin="95,82,0,0" VerticalAlignment="Top" Height="20" Width="51"/>

<TextBox Grid.Column="0" x:Name="X3" HorizontalAlignment="Left" Margin="39,103,0,0" VerticalAlignment="Top" Height="20" Width="51"/>

<TextBox Grid.Column="0" x:Name="Y3" HorizontalAlignment="Left" Margin="95,103,0,0" VerticalAlignment="Top" Height="20" Width="51"/>

<TextBox Grid.Column="0" x:Name="X0" HorizontalAlignment="Left" Margin="39,148,0,0" VerticalAlignment="Top" Height="20" Width="51"/>

<TextBox Grid.Column="0" x:Name="Y0" HorizontalAlignment="Left" Margin="95,148,0,0" VerticalAlignment="Top" Height="20" Width="51"/>

<TextBlock Grid.Column="0" HorizontalAlignment="Left" Margin="10,57,0,0" Text="A" VerticalAlignment="Top" Width="22" Height="20"/>

<TextBlock Grid.Column="0" HorizontalAlignment="Left" Margin="10,82,0,0" Text="B" VerticalAlignment="Top" Width="22" Height="20"/>

<TextBlock Grid.Column="0" HorizontalAlignment="Left" Margin="10,103,0,0" Text="C" VerticalAlignment="Top" Width="22" Height="20"/>

<TextBlock Grid.Column="0" HorizontalAlignment="Left" Margin="10,148,0,0" Text="D" VerticalAlignment="Top" Width="22" Height="20"/>

<TextBlock Grid.Column="0" HorizontalAlignment="Left" Margin="10,128,0,0" Text="Point" VerticalAlignment="Top" Width="164" Height="15"/>

<TextBlock Grid.Column="0" HorizontalAlignment="Left" Margin="10,36,0,0" Text="Triangle" VerticalAlignment="Top" Width="164" Height="15"/>

<Button Grid.Column="0" Content="Click" Click="ButtonBase\_OnClick" HorizontalAlignment="Left" Margin="52,0,0,0" VerticalAlignment="Center" Width="68" Height="34"/>

<TextBlock Grid.Column="0" x:Name="Result" HorizontalAlignment="Left" Margin="10,273,0,0" VerticalAlignment="Top" Width="164" Height="15"/>

</Grid>

</Window>

Задание 2

Form1.cs

using System;

using System.Drawing;

using System.Windows.Forms;

namespace WindowsFormsApp2

{

public partial class Form1 : Form

{

public double wx = 0, wy = 0;

public double zoom = 2f, speed = 2f;

public int res = 5;

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Q)

{

res -= 1;

}

if (e.KeyCode == Keys.E)

{

res += 1;

}

if (e.KeyCode == Keys.Up)

{

wy -= speed \* (5 - Math.Abs(zoom));

}

if (e.KeyCode == Keys.Down)

{

wy += speed \* (5 - Math.Abs(zoom));

}

if (e.KeyCode == Keys.Left)

{

wx -= speed \* (5 - Math.Abs(zoom));

}

if (e.KeyCode == Keys.Right)

{

wx += speed \* (5 - Math.Abs(zoom));

}

Draw();

}

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

Draw();

}

public void Draw()

{

if (res <= 0)

{

res = 1;

}

Bitmap frame = new Bitmap(Width / res, Height / res);

for (int x = 0; x < Width/res; x++)

{

for (int y = 0; y < Height/res; y++)

{

double a = (double)((x + (wx / res / zoom)) - ((Width / 2d) / res)) / (double)(Width / zoom / res / 1.777f);

double b = (double)((y + (wy / res / zoom)) - ((Height / 2d) / res)) / (double)(Height / zoom / res);

Numbers c = new Numbers(a, b);

Numbers z = new Numbers(0, 0);

int it = 0;

do

{

it++;

z.Sqr();

z.Add(c);

if (z.Magn() > 2.0d)

{

break;

}

} while (it < 100);

frame.SetPixel(x, y, Color.FromArgb((byte)(it \* 2.55f), (byte)(it \* 2.55f), (byte)(it \* 2.55f)));

}

}

pictureBox1.Image = frame;

pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;

}

}

public class Numbers {

public double a;

public double b;

public Numbers(double a, double b)

{

this.a = a;

this.b = b;

}

public void Sqr()

{

double tmp = (a \* a) - (b \* b);

b = 2.0d \* a \* b;

a = tmp;

}

public double Magn()

{

return Math.Sqrt((a \* a) + (b \* b));

}

public void Add(Numbers c)

{

a += c.a;

b += c.b;

}

}

}

Form1.Designer.cs

namespace WindowsFormsApp2

{

partial class Form1

{

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

/// <param name="disposing">истинно, если управляемый ресурс должен быть удален; иначе ложно.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Код, автоматически созданный конструктором форм Windows

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

private void InitializeComponent()

{

this.components = new System.ComponentModel.Container();

this.pictureBox1 = new System.Windows.Forms.PictureBox();

this.timer1 = new System.Windows.Forms.Timer(this.components);

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();

this.SuspendLayout();

//

// pictureBox1

//

this.pictureBox1.Location = new System.Drawing.Point(0, 0);

this.pictureBox1.Margin = new System.Windows.Forms.Padding(4);

this.pictureBox1.Name = "pictureBox1";

this.pictureBox1.Size = new System.Drawing.Size(1268, 755);

this.pictureBox1.TabIndex = 0;

this.pictureBox1.TabStop = false;

//

// Form1

//

this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(1267, 754);

this.Controls.Add(this.pictureBox1);

this.Margin = new System.Windows.Forms.Padding(4);

this.Name = "Form1";

this.Text = "Form1";

this.Load += new System.EventHandler(this.Form1\_Load);

this.KeyDown += new System.Windows.Forms.KeyEventHandler(this.Form1\_KeyDown);

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();

this.ResumeLayout(false);

}

#endregion

private System.Windows.Forms.PictureBox pictureBox1;

private System.Windows.Forms.Timer timer1;

}

}

Вывод: освоил возможности языка программирования C# в построении графических приложений.