Министерство образования и науки Российской

**Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем

и программной инженерии

**Лабораторная работа № 5**

**по дисциплине**

**«Программирование компьютерной графики»**

**По теме: «Инициализация в OpenGL**

Выполнила:

ст. гр. ПРИ-118

Левченко В.П

Принял:

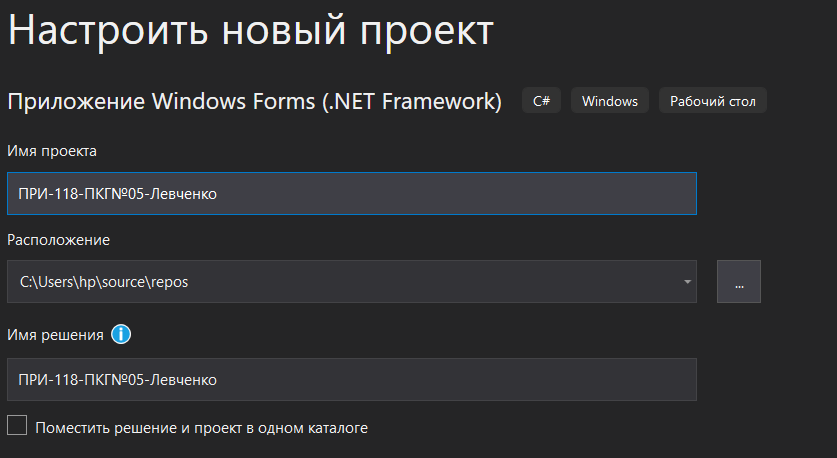
Жигалов И.Е

Владимир, 2021

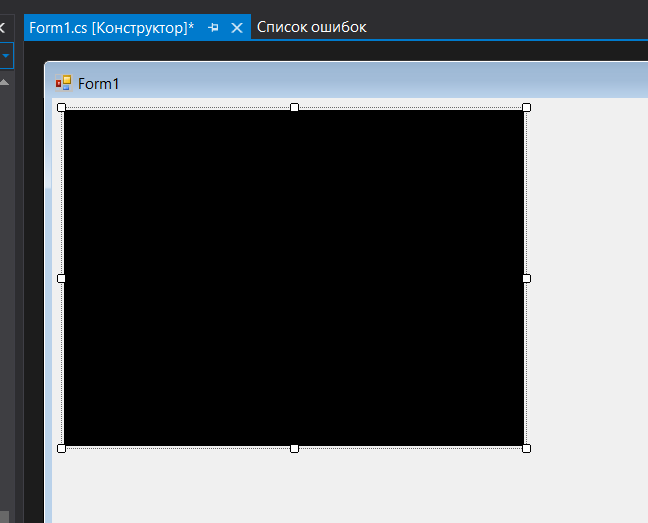
**Цель работы**

Изучение принципов инициализации OpenGL и визуализация объектов при разработке приложений в C#.

**Выполнение работы**

  
Рисунок 1. Создание нового проекта

На созданную форму добавляем элемент simpleOpenGLControl.

  
Рисунок 2. Элемент

Необходимо визуализировать свою фамилию и инициалы, для этого я воспользовалась онлайн-инструментом, чтобы удобнее было определить координаты точек.

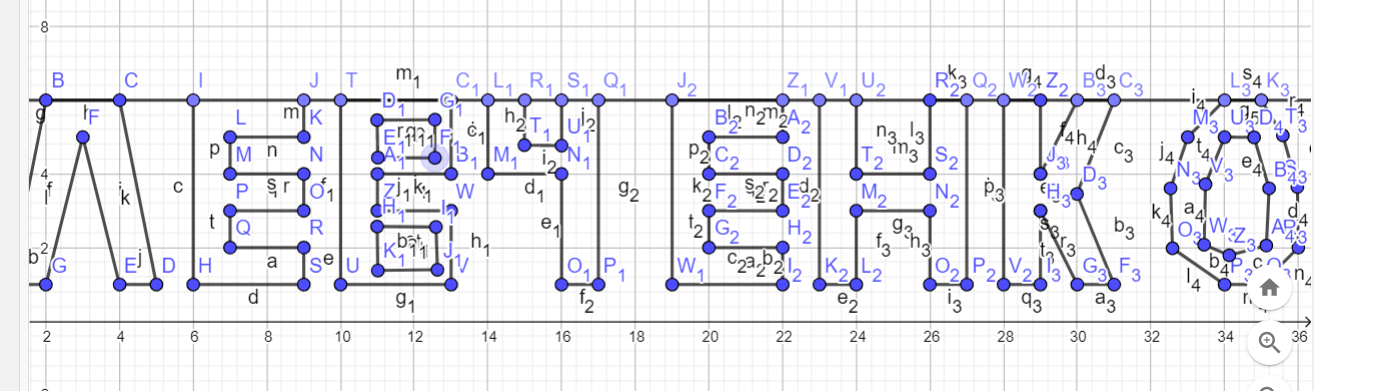
  
Рисунок 3. Координаты для точек

  
Рисунок 4. Вариант задания, вариант №5

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

// для работы с библиотекой OpenGL

using Tao.OpenGl;

using Tao.FreeGlut;

// для работы с библиотекой FreeGLUT

namespace ПРИ\_118\_ПКГ\_05\_Левченко

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

AnT.InitializeContexts();

}

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

{

Glut.glutInit();

Glut.glutInitDisplayMode(Glut.GLUT\_RGB | Glut.GLUT\_DOUBLE | Glut.GLUT\_DEPTH);

Gl.glClearColor(255, 255, 255, 1);

Gl.glViewport(0, 0, AnT.Width, AnT.Height);

Gl.glMatrixMode(Gl.GL\_PROJECTION);

Gl.glLoadIdentity();

if ((float)AnT.Width <= (float)AnT.Height)

{

Glu.gluOrtho2D(0.0, 30.0 \* (float)AnT.Height / (float)AnT.Width, 0.0, 30.0);

}

else

{

Glu.gluOrtho2D(0.0, 30.0 \* (float)AnT.Width / (float)AnT.Height, 0.0, 30.0);

}

Gl.glMatrixMode(Gl.GL\_MODELVIEW);

Gl.glLoadIdentity();

Gl.glEnable(Gl.GL\_DEPTH\_TEST);

Gl.glEnable(Gl.GL\_COLOR\_MATERIAL);

}

private void button1\_Click\_1(object sender, EventArgs e)

{

Gl.glClear(Gl.GL\_COLOR\_BUFFER\_BIT | Gl.GL\_DEPTH\_BUFFER\_BIT);

Gl.glLoadIdentity();

Gl.glColor3f(255, 0, 0);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(8, 7);

Gl.glVertex2d(15, 27);

Gl.glVertex2d(17, 27);

Gl.glVertex2d(23, 7);

Gl.glVertex2d(21, 7);

Gl.glVertex2d(19, 14);

Gl.glVertex2d(12.5, 14);

Gl.glVertex2d(10, 7);

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(18.5, 16);

Gl.glVertex2d(16, 25);

Gl.glVertex2d(13.2, 16);

Gl.glEnd();

Gl.glFlush();

AnT.Invalidate();

}

private void button2\_Click\_1(object sender, EventArgs e)

{

Gl.glClear(Gl.GL\_COLOR\_BUFFER\_BIT | Gl.GL\_DEPTH\_BUFFER\_BIT);

Gl.glLoadIdentity();

Gl.glLineWidth(1);

// л

Gl.glColor3f(0, 0, 255);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(1, 1);

Gl.glVertex2d(2, 6);

Gl.glVertex2d(4, 6);

Gl.glVertex2d(5, 1);

Gl.glVertex2d(4, 1);

Gl.glVertex2d(3, 5);

Gl.glVertex2d(2, 1);

Gl.glVertex2d(1, 1);

Gl.glEnd();

//е

Gl.glColor3f(0, 0, 255);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(6, 1);

Gl.glVertex2d(6, 6);

Gl.glVertex2d(9, 6);

Gl.glVertex2d(9, 5);

Gl.glVertex2d(7, 5);

Gl.glVertex2d(7, 4);

Gl.glVertex2d(9, 4);

Gl.glVertex2d(9, 3);

Gl.glVertex2d(7, 3);

Gl.glVertex2d(7, 2);

Gl.glVertex2d(9, 2);

Gl.glVertex2d(9, 1);

Gl.glVertex2d(6, 1);

Gl.glEnd();

//в

Gl.glColor3f(0, 0, 255);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(10, 1);

Gl.glVertex2d(10, 6);

Gl.glVertex2d(13, 6);

Gl.glVertex2d(13, 4);

Gl.glVertex2d(11, 4);

Gl.glVertex2d(11, 3);

Gl.glVertex2d(13, 3);

Gl.glVertex2d(13, 1);

Gl.glVertex2d(10, 1);

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(11, 4.5);

Gl.glVertex2d(11, 5.5);

Gl.glVertex2d(12.5, 5.5);

Gl.glVertex2d(12.5, 4.5);

Gl.glVertex2d(11, 4.5);

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(11, 1.5);

Gl.glVertex2d(11, 2.5);

Gl.glVertex2d(12.5, 2.5);

Gl.glVertex2d(12.5, 1.5);

Gl.glVertex2d(11, 1.5);

Gl.glEnd();

//ч

Gl.glColor3f(0, 0, 255);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(16, 1);

Gl.glVertex2d(17, 1);

Gl.glVertex2d(17, 6);

Gl.glVertex2d(16, 6);

Gl.glVertex2d(16, 5);

Gl.glVertex2d(15, 5);

Gl.glVertex2d(15, 6);

Gl.glVertex2d(14, 6);

Gl.glVertex2d(14, 4);

Gl.glVertex2d(16, 4);

Gl.glVertex2d(16, 1);

Gl.glEnd();

//е

Gl.glColor3f(0, 0, 255);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(19, 1);

Gl.glVertex2d(19, 6);

Gl.glVertex2d(22, 6);

Gl.glVertex2d(22, 5);

Gl.glVertex2d(20, 5);

Gl.glVertex2d(20, 4);

Gl.glVertex2d(22, 4);

Gl.glVertex2d(22, 3);

Gl.glVertex2d(20, 3);

Gl.glVertex2d(20, 2);

Gl.glVertex2d(22, 2);

Gl.glVertex2d(22, 1);

Gl.glVertex2d(19, 1);

Gl.glEnd();

//н

Gl.glColor3f(0, 0, 255);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(23, 1);

Gl.glVertex2d(23, 6);

Gl.glVertex2d(24, 6);

Gl.glVertex2d(24, 4);

Gl.glVertex2d(26, 4);

Gl.glVertex2d(26, 6);

Gl.glVertex2d(27, 6);

Gl.glVertex2d(27, 1);

Gl.glVertex2d(26, 1);

Gl.glVertex2d(26, 3);

Gl.glVertex2d(24, 3);

Gl.glVertex2d(24, 1);

Gl.glVertex2d(23, 1);

Gl.glEnd();

//к

Gl.glColor3f(0, 0, 255);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(28, 1);

Gl.glVertex2d(28, 6);

Gl.glVertex2d(29, 6);

Gl.glVertex2d(29, 4);

Gl.glVertex2d(30, 6);

Gl.glVertex2d(31, 6);

Gl.glVertex2d(30, 3.5);

Gl.glVertex2d(31, 1);

Gl.glVertex2d(30, 1);

Gl.glVertex2d(29, 3);

Gl.glVertex2d(29, 1);

Gl.glVertex2d(28, 1);

Gl.glEnd();

//о

Gl.glColor3f(0, 0, 255);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(34, 1);

Gl.glVertex2d(33, 2);

Gl.glVertex2d(33, 4);

Gl.glVertex2d(34, 6);

Gl.glVertex2d(35, 6);

Gl.glVertex2d(36, 4);

Gl.glVertex2d(36, 2);

Gl.glVertex2d(35, 1);

Gl.glVertex2d(34, 1);

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(34, 2);

Gl.glVertex2d(33.5, 3.5);

Gl.glVertex2d(34, 4.5);

Gl.glVertex2d(35, 4.5);

Gl.glVertex2d(35.5, 3.5);

Gl.glVertex2d(35, 2);

Gl.glVertex2d(34, 2);

Gl.glEnd();

//в

Gl.glColor3f(0, 0, 255);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(38, 1);

Gl.glVertex2d(38, 6);

Gl.glVertex2d(41, 6);

Gl.glVertex2d(41, 4);

Gl.glVertex2d(39, 4);

Gl.glVertex2d(39, 3);

Gl.glVertex2d(41, 3);

Gl.glVertex2d(41, 1);

Gl.glVertex2d(38, 1);

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(39, 4.5);

Gl.glVertex2d(39, 5.5);

Gl.glVertex2d(40.5, 5.5);

Gl.glVertex2d(40.5, 4.5);

Gl.glVertex2d(39, 4.5);

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(39, 1.5);

Gl.glVertex2d(39, 2.5);

Gl.glVertex2d(40, 2.5);

Gl.glVertex2d(40, 1.5);

Gl.glVertex2d(39, 1.5);

Gl.glEnd();

//п

Gl.glColor3f(0, 0, 255);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(42, 1);

Gl.glVertex2d(42, 6);

Gl.glVertex2d(45, 6);

Gl.glVertex2d(45, 1);

Gl.glVertex2d(44, 1);

Gl.glVertex2d(44, 5);

Gl.glVertex2d(43, 5);

Gl.glVertex2d(43, 1);

Gl.glVertex2d(42, 1);

Gl.glEnd();

AnT.Invalidate();

}

  
Рисунок 5. Результат

Для выполнения следующих двух заданий будет создано ещё две формы, которые будут вызываться из первой:

private void button3\_Click(object sender, EventArgs e)

{

Form2 f2 = new Form2();

f2.Show();

}

private void button4\_Click(object sender, EventArgs e)

{

Form3 f2 = new Form3();

f2.Show();

}

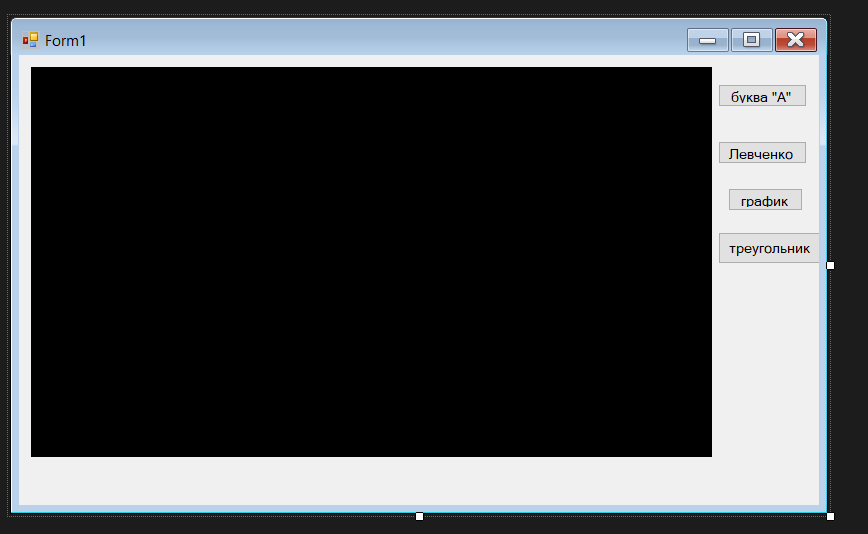
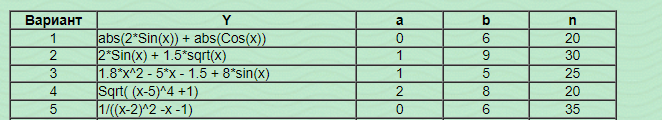
}

Рисунок 6. Вид формы 1

*Задание 2. Визуализация графика функции*

  
Рисунок 7. Вариант задания №5

namespace ПРИ\_118\_ПКГ\_05\_Левченко

{

public partial class Form2 : Form

{

double ScreenW, ScreenH;

private float devX;

private float devY;

private float[,] GrapValuesArray;

private int elements\_count = 0;

private bool not\_calculate = true;

private int pointPosition = 0;

float lineX, lineY;

float Mcoord\_X = 0, Mcoord\_Y = 0;

public Form2()

{

InitializeComponent();

AnT.InitializeContexts();

}

private void Form2\_Load(object sender, EventArgs e)

{

Glut.glutInitDisplayMode(Glut.GLUT\_RGB | Glut.GLUT\_DOUBLE);

Gl.glClearColor(255, 255, 255, 1);

Gl.glViewport(0, 0, AnT.Width, AnT.Height);

Gl.glMatrixMode(Gl.GL\_PROJECTION);

Gl.glLoadIdentity();

if ((float)AnT.Width <= (float)AnT.Height)

{

ScreenW = 30.0;

ScreenH = 30.0 \* (float)AnT.Height / (float)AnT.Width;

Glu.gluOrtho2D(0.0, ScreenW, 0.0, ScreenH);

}

else

{

ScreenW = 30.0 \* (float)AnT.Width / (float)AnT.Height;

ScreenH = 30.0;

Glu.gluOrtho2D(0.0, 30.0 \* (float)AnT.Width / (float)AnT.Height, 0.0, 30.0);

}

devX = (float)ScreenW / (float)AnT.Width;

devY = (float)ScreenH / (float)AnT.Height;

Gl.glMatrixMode(Gl.GL\_MODELVIEW);

PointInGrap.Start();

}

private void PointInGrap\_Tick(object sender, EventArgs e)

{

if (pointPosition == elements\_count - 1) pointPosition = 0;

Draw();

pointPosition++;

}

private void AnT\_MouseMove(object sender, MouseEventArgs e)

{

Mcoord\_X = e.X;

Mcoord\_Y = e.Y;

lineX = devX \* e.X;

lineY = (float)(ScreenH - devY \* e.Y);

}

private void PrintText2D(float x, float y, string text)

{

Gl.glRasterPos2f(x, y);

foreach (char char\_for\_draw in text)

{

Glut.glutBitmapCharacter(Glut.GLUT\_BITMAP\_9\_BY\_15, char\_for\_draw);

}

}

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

{

}

private void functionCalculation()

{

float x = 0, y = 0;

float a = 0, b = 6, n = 35;

GrapValuesArray = new float[300, 2];

elements\_count = 0;

for (x = a; x < b; x += (b - a) / n)

{

// 1/ ((x-2)^2-x-1)

y = (float)1 / ((float)Math.Pow(x-2, 2) -x -1);

GrapValuesArray[elements\_count, 0] = x;

GrapValuesArray[elements\_count, 1] = y;

elements\_count++;

}

not\_calculate = false;

}

private void DrawDiagram()

{

if (not\_calculate)

{

functionCalculation();

}

Gl.glBegin(Gl.GL\_LINE\_STRIP);

Gl.glVertex2d(GrapValuesArray[0, 0], GrapValuesArray[0, 1]);

for (int ax = 1; ax < elements\_count; ax += 2)

{

Gl.glVertex2d(GrapValuesArray[ax, 0], GrapValuesArray[ax, 1]);

}

Gl.glEnd();

Gl.glPointSize(5);

Gl.glColor3f(255, 0, 0);

Gl.glBegin(Gl.GL\_POINTS);

Gl.glVertex2d(GrapValuesArray[pointPosition, 0], GrapValuesArray[pointPosition, 1]);

Gl.glEnd();

Gl.glPointSize(1);

}

private void Draw()

{

Gl.glClear(Gl.GL\_COLOR\_BUFFER\_BIT | Gl.GL\_DEPTH\_BUFFER\_BIT);

Gl.glLoadIdentity();

Gl.glColor3f(0, 0, 0);

Gl.glPushMatrix();

Gl.glTranslated(15, 15, 0);

Gl.glBegin(Gl.GL\_POINTS);

for (int ax = -15; ax < 15; ax++)

{

for (int bx = -15; bx < 15; bx++)

{

Gl.glVertex2d(ax, bx);

}

}

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINES);

Gl.glVertex2d(0, -15);

Gl.glVertex2d(0, 15);

Gl.glVertex2d(-15, 0);

Gl.glVertex2d(15, 0);

Gl.glVertex2d(0, 15);

Gl.glVertex2d(0.1, 14.5);

Gl.glVertex2d(0, 15);

Gl.glVertex2d(-0.1, 14.5);

Gl.glVertex2d(15, 0);

Gl.glVertex2d(14.5, 0.1);

Gl.glVertex2d(15, 0);

Gl.glVertex2d(14.5, -0.1);

Gl.glEnd();

PrintText2D(15.5f, 0, "x");

PrintText2D(0.5f, 14.5f, "y");

DrawDiagram();

Gl.glPopMatrix();

PrintText2D(devX \* Mcoord\_X + 0.2f, (float)ScreenH - devY \* Mcoord\_Y + 0.4f, "[ x: " + (devX \* Mcoord\_X - 15).ToString() + " ; y: " + ((float)ScreenH - devY \* Mcoord\_Y - 15).ToString() + "]");

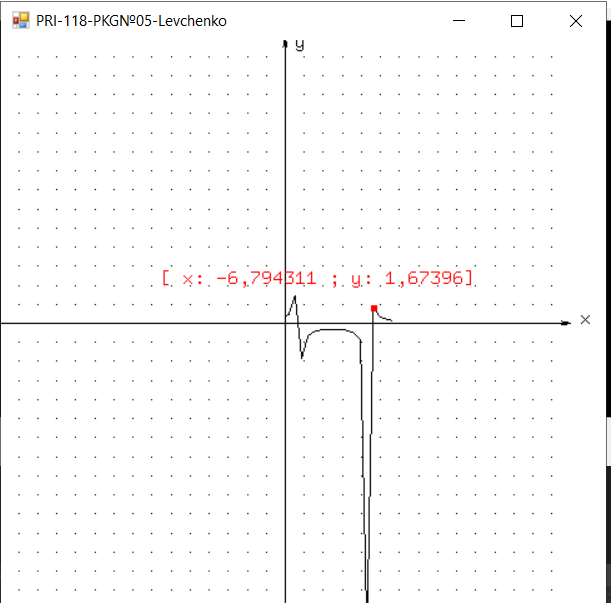
Gl.glFlush();

AnT.Invalidate();

}

}

}

  
Рисунок 8. Результат визуализации

*Задание 3. Вывод 2D цветового треугольника*

namespace ПРИ\_118\_ПКГ\_05\_Левченко

{

public partial class Form3 : Form

{

double a = 0, b = 1, c = 0;

public Form3()

{

InitializeComponent();

AnT.InitializeContexts();

}

private void Form3\_Load(object sender, EventArgs e)

{

Glut.glutInitDisplayMode(Glut.GLUT\_RGB | Glut.GLUT\_DOUBLE | Glut.GLUT\_DEPTH);

Gl.glClearColor(255, 255, 255, 1);

Gl.glViewport(0, 0, AnT.Width, AnT.Height);

Gl.glMatrixMode(Gl.GL\_PROJECTION);

Gl.glLoadIdentity();

if (AnT.Width <= AnT.Height)

Glu.gluOrtho2D(0.0, 30.0, 0.0, 30.0 \* (float)AnT.Height / (float)AnT.Width);

else

Glu.gluOrtho2D(0.0, 30.0 \* (float)AnT.Width / (float)AnT.Height, 0.0, 30.0);

Gl.glMatrixMode(Gl.GL\_MODELVIEW);

}

private void trackBar1\_Scroll(object sender, EventArgs e)

{

a = (double)trackBar1.Value / 1000.0;

label4.Text = a.ToString();

}

private void trackBar2\_Scroll(object sender, EventArgs e)

{

b = (double)trackBar2.Value / 1000.0;

label5.Text = b.ToString();

}

private void trackBar3\_Scroll(object sender, EventArgs e)

{

c = (double)trackBar3.Value / 1000.0;

label6.Text = c.ToString();

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void timer1\_Tick(object sender, EventArgs e)

{

Draw();

}

private void label1\_Click(object sender, EventArgs e)

private void button1\_Click(object sender, EventArgs e)

{

RenderTimer.Start();

}

private void Draw()

{

Gl.glClear(Gl.GL\_COLOR\_BUFFER\_BIT);

Gl.glBegin(Gl.GL\_TRIANGLES);

Gl.glColor3d(0.0 + a, 0.0 + b, 0.0 + c);

Gl.glVertex2d(5.0, 5.0);

Gl.glColor3d(0.0 + c, 0.0 + a, 0.0 + b);

Gl.glVertex2d(25.0, 5.0);

Gl.glColor3d(0.0 + b, 0.0 + c, 0.0 + a);

Gl.glVertex2d(5.0, 25.0);

Gl.glEnd();

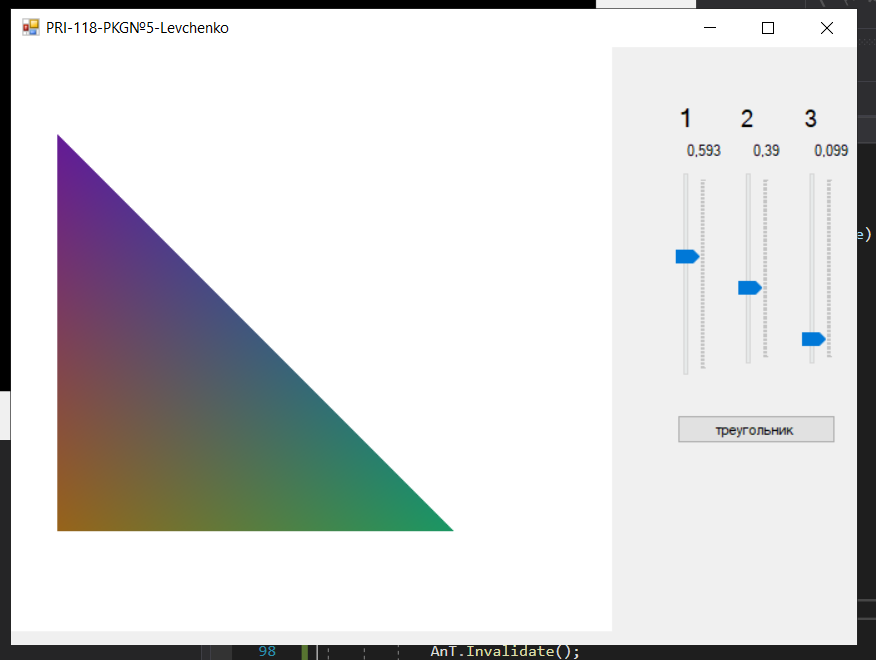
Gl.glFlush();

AnT.Invalidate();

}

}

}

  
Рисунок 9. Результат

**Вывод**

Изучены принципы инициализации OpenGL и визуализация объектов при разработке приложений в C#.