**Индивидуальное задание №3. Задание 5.**

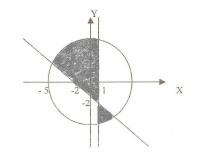
**Выполнил студент Группы ТВИД-124з**

**Вовчок Глеб Иванович**

* 1. **Описание задания**

В задании поставлена цель – Определить, попала ли произвольно

заданная точка с координатами (x,y)в заштрихованную область:



Теперь с графическим отображением в 3D.

В качестве языка для выполнения задания был взят .NET C#. Средой выполнения послужила Visual Studio 2019.

* 1. **Постановка Цели**

Написать программу, способную решить определять результат при введённых координатах.

* 1. **Создание программы.**

Перед тем, как приступить к просмотру процесса создания, прошу обратить внимание, что вся Индивидуальная работа является **цельным проектом**, в выполненной работе нет разделений на каждую задачу, кроме как выбором задания, но при этом каждое задание получило отдельную документацию, чтобы не составлять огромный документ-отчёт.

Программа является **гибридной** (она сочетает в себе функции консольного приложения и приложения на базе WinForms.)

Общий код для всего проекта:

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Здраствуйте! Это задание выполнено студентом ТВИД-124з Вовчок Глебом Ивановичем.");

Start();

}

static public void Start()

{

Console.WriteLine("Введите номер от 1 до 5.");

start:

//Чтобы было безперебойно, была введена проверка фаталок и проверка от неверных значений, не входящих в результат.

try

{

//goto уже устарел, но всё ещё работает, хотя всё можно было чисто через switch... ...case реализовать.

int VarNum = int.Parse(Console.ReadLine());

if (VarNum > 0 && VarNum <= 5)

{

switch (VarNum)

{

case 1:

goto first;

case 2:

goto second;

case 3:

goto third;

case 4:

goto fourth;

case 5:

goto fifth;

}

}

else

{

Console.WriteLine("Такой цифры нет в задании.");

Console.Read();

}

}

catch (System.Exception)

{

Console.WriteLine("Вы ввели неверно. Попробуйте ещё раз");

}

goto start;

first:

//Простое вычисление, не объясняя. Вывод результата.

Console.WriteLine(" Задание 1.\nРешение арифметического уравнеия.");

Num1();

Console.ReadKey();

goto start;

second:

Console.WriteLine(" Задание 2.\nЗадача с графикой.");

Num2();

Console.ReadKey();

goto start;

third:

Console.WriteLine(" Задание 3.\n");

Num3();

Console.ReadKey();

goto start;

fourth:

Console.WriteLine(" Задание 4.\n");

Num4();

Console.ReadKey();

goto start;

fifth:

Console.WriteLine(" Задание 5.\n");

Num5();

Console.ReadKey();

goto start;

}

При запуске выглядит так:

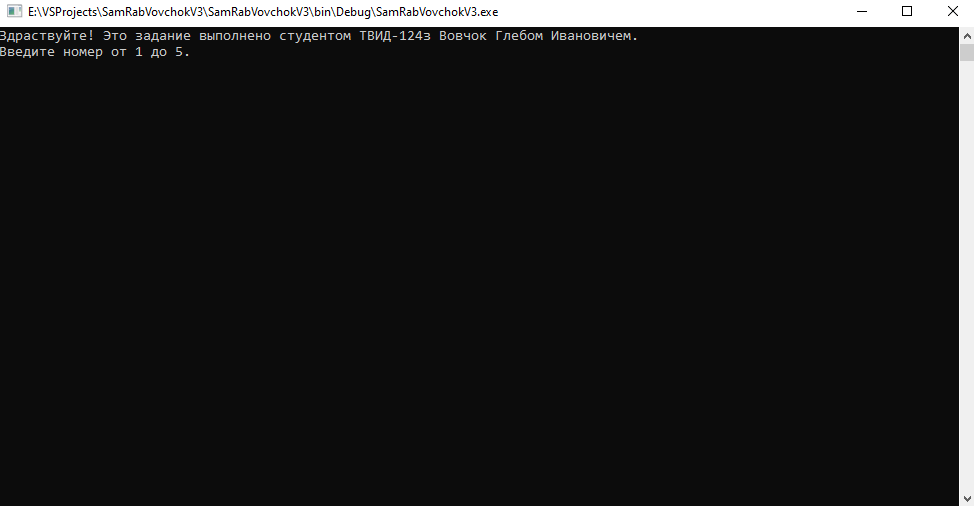


Рис. 1 Общий код индивидуального задания

Чтобы реализовать в 3D, нам необходимо добавить графический рендеринг. Самым быстрым будет OpenGL.:

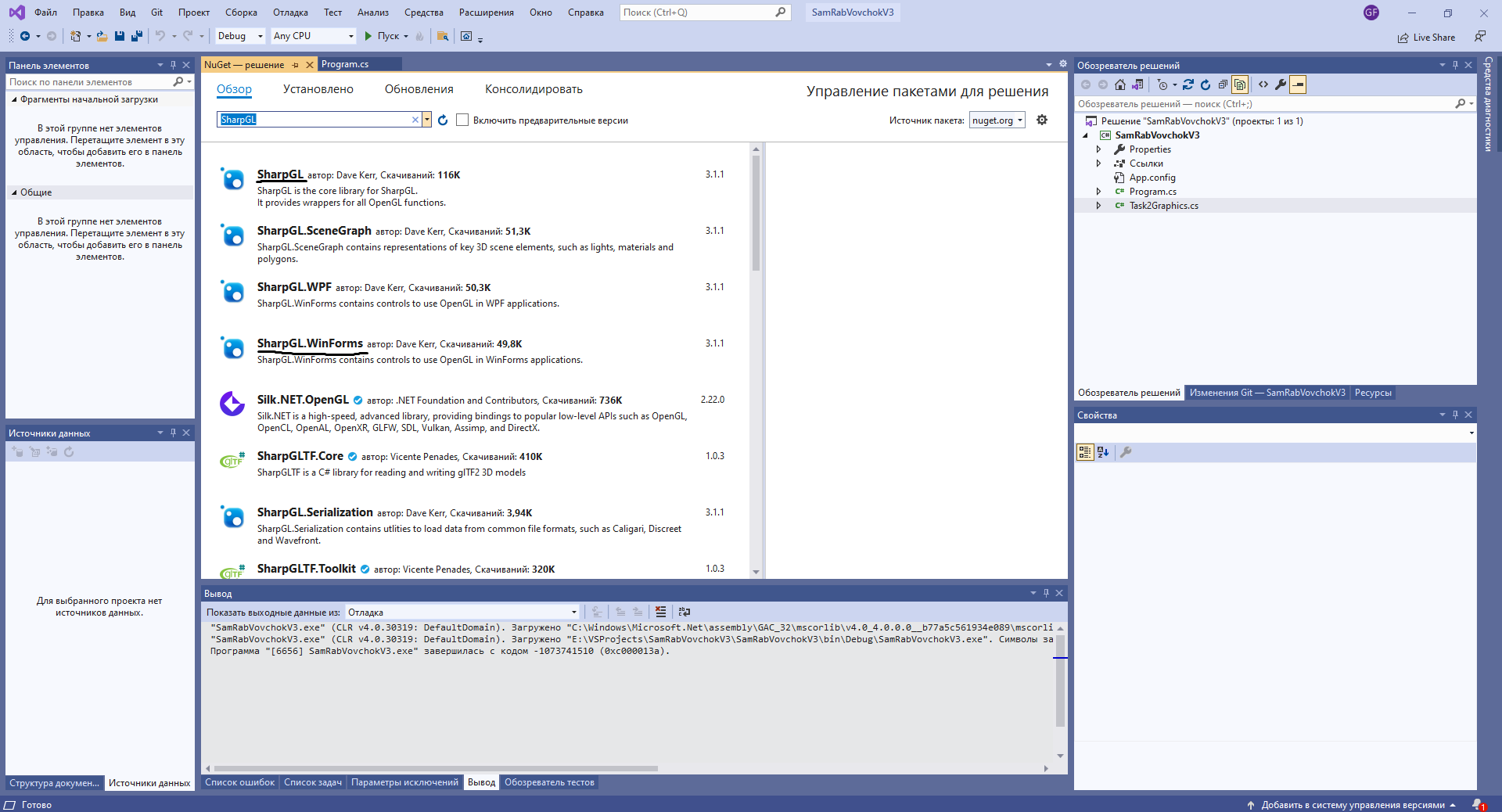


Рис. 2 Установленные модули

Далее был составлен код задания 5, он уже задействует не только основной файл программы, но и имеет свой собственный, в виде окна WinForms:

Код в основном модуле:

static public void Num5()

{

//

//Задействован WinForms.

//Есть отображение.

//Ваши переменные вводятся с проверкой.

//

Console.WriteLine("Задача на поиск точки, которая попадает в закрашенную территорию. (После того, как вы введёте, увидите изображение, куда вы попали.");

Console.Write("Введите X: ");

try

{

X = double.Parse(Console.ReadLine());

}

catch (System.Exception)

{

Console.WriteLine("Произошла критическая ошибка, попробуйте ввести другие значения для выражения");

Num5();

}

Console.Write("Введите Y: ");

try

{

Y = double.Parse(Console.ReadLine());

Task5Graphics task5 = new Task5Graphics(X, Y);

task5.ShowDialog();

Start();

}

catch (System.Exception e)

{

Console.WriteLine(e);

Num5();

}

}

Код в модуле WinForms:

public partial class Task5Graphics : Form

{

private OpenGL gl;

private double x, y;

private double z = 1;

public Task5Graphics(double ValX, double ValY)

{

InitializeComponent();

x = ValX;

y = ValY;

XVal.Text = "" + x;

YVal.Text = "" + y;

openGLControl1.OpenGLInitialized += openGLControl1\_OpenGLInitialized;

openGLControl1.OpenGLDraw += openGLControl1\_OpenGLDraw;

DrawDot();

}

private void openGLControl1\_OpenGLInitialized(object sender, EventArgs e)

{

gl = ((OpenGLControl)sender).OpenGL;

// Настройка OpenGL

gl.ClearColor(255, 255, 255, 1.0f); // Белый фон

gl.Enable(OpenGL.GL\_DEPTH\_TEST); // Включаем тест глубины

}

private void openGLControl1\_OpenGLDraw(object sender, RenderEventArgs args)

{

gl.Clear(OpenGL.GL\_COLOR\_BUFFER\_BIT | OpenGL.GL\_DEPTH\_BUFFER\_BIT);

gl.LoadIdentity();

gl.Translate(0.0f, 0.0f, -5.0f);

// Отрисовка точки

//

gl.Color(1.0f, 0.0f, 0.0f); //Красный для самой точки

gl.PointSize(5.0f);

gl.Begin(OpenGL.GL\_POINTS); //Метод отрисовки - Точки

gl.Vertex(x, y, z); //Позиция

gl.End();

gl.Flush(); //Сама команда отрисовки, сделана принудительной

//Теоретически это должно было создать красную точку на экран, но она является плоской в Трёхмерном пространстве.

//В связи с этим, и полным отсутствием графика, задание можно считать не выполнененным.

}

public void DrawDot()

{

//

//Определяем - попало или нет.

//

if (((x <= 1 && y >= -2) || (x <= -5 && y <= 5)) || ((x >= 1 && y <= -2) || (x >= 2 && y <= -5)))

{

Real.ForeColor = Color.Lime;

Real.Text = "ДА";

}

else

{

Real.ForeColor = Color.Red;

Real.Text = "НЕТ";

}

}

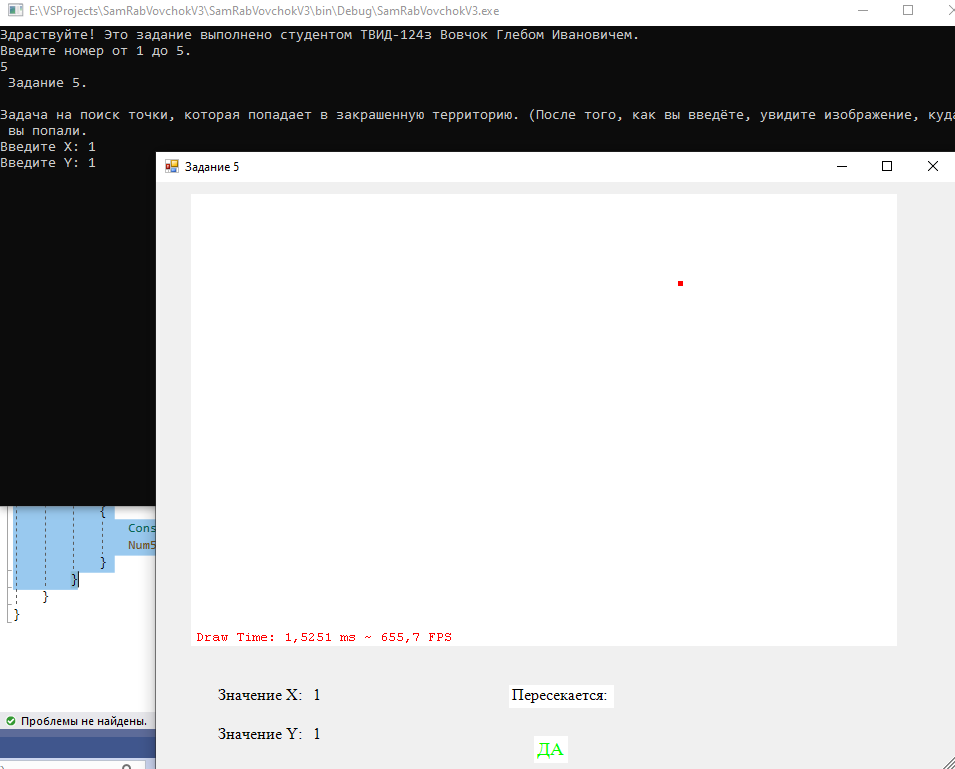
}

Рис. 3 Задание 5 в выполнении

В случае, если пользователь введёт с ошибкой, программа не закроется, а выдаст ошибку, которая сообщит о неверном вводе и предложит повторить попытку:

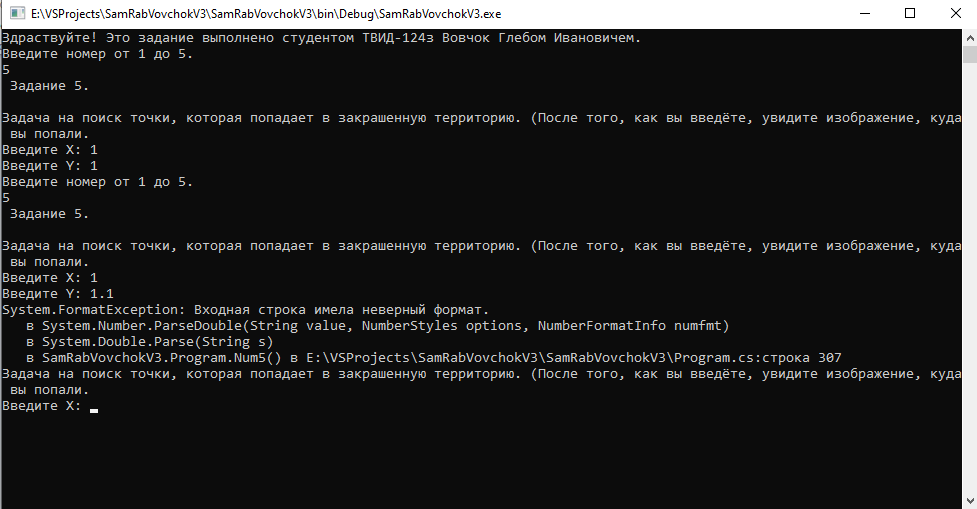


Рис. 4 Случай с ошибкой в задании 5

К сожалению, это не то, что требовалось в задании, я не смог выполнить его, так как знания в OpenGL совсем на начальном уровне.

**ЗАКЛЮЧЕНИЕ**

Поставленная задача не выполнена. Программа высчитывает местоположение и определяет, попал ли пользователь с помощью координат в закрашенную территорию, но отображение трёхмерной графики остаётся плоским, отображение не растянуто на весь экран.