Разветвляющиеся вычислительные процессы **НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **“СИНЕРГИЯ”»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | Программирование |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | Информационные системы и программирование |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | Очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Отчет по практической работе №1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** |  | Использование операторов ветвления | | |
|  |  | (наименование темы) | | |
|  |  |  | | |
| **по дисциплине** | | |  | Разработка программных модулем |
|  | | |  | (наименование дисциплины) |

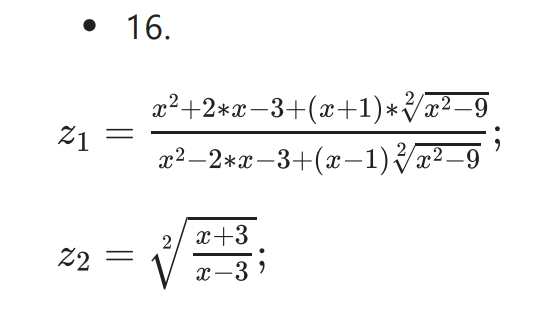
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Стрельцов Егор Михайлович |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | VДКИП-111прог |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Сибирев И. В. |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2024 г.**

Лабораторная работа 1.

Линейные программы. Вариант 16.



Листинг кода:

using System;

namespace ConsoleApplication1

{

class Class1

{

static void Main()

{

Console.WriteLine("Введите x");

double x = Convert.ToDouble(Console.ReadLine());

if (x \* x - 9 >= 0)

{

double chisitel = Math.Pow(x, 2) + 2 \* x - 3 + (x + 1) \* Math.Sqrt(x \* x - 9);

double znamenatel = Math.Pow(x, 2) - 2 \* x - 3 + (x - 1) \* Math.Sqrt(x \* x - 9);

double z1;

if (znamenatel == 0 || Math.Pow(x, 2) - 9 < 0)

{

z1 = double.NaN;

}

else

{

z1 = chisitel / znamenatel;

}

if (!double.IsNaN(z1))

{

Console.WriteLine("z1 = " + z1);

}

else

{

Console.WriteLine("Ошибка: деление на ноль или отрицательное число");

}

double z2;

double result = (x + 3) / (x - 3);

if (x == 3 || result < 0)

{

z2 = double.NaN;

}

else

{

z2 = Math.Sqrt(result);

}

if (!double.IsNaN(z2))

{

Console.WriteLine("z2 = " + z2);

}

else

{

Console.WriteLine("Ошибка: деление на ноль или отрицательное число");

}

Console.ReadKey();

}

}

}

}

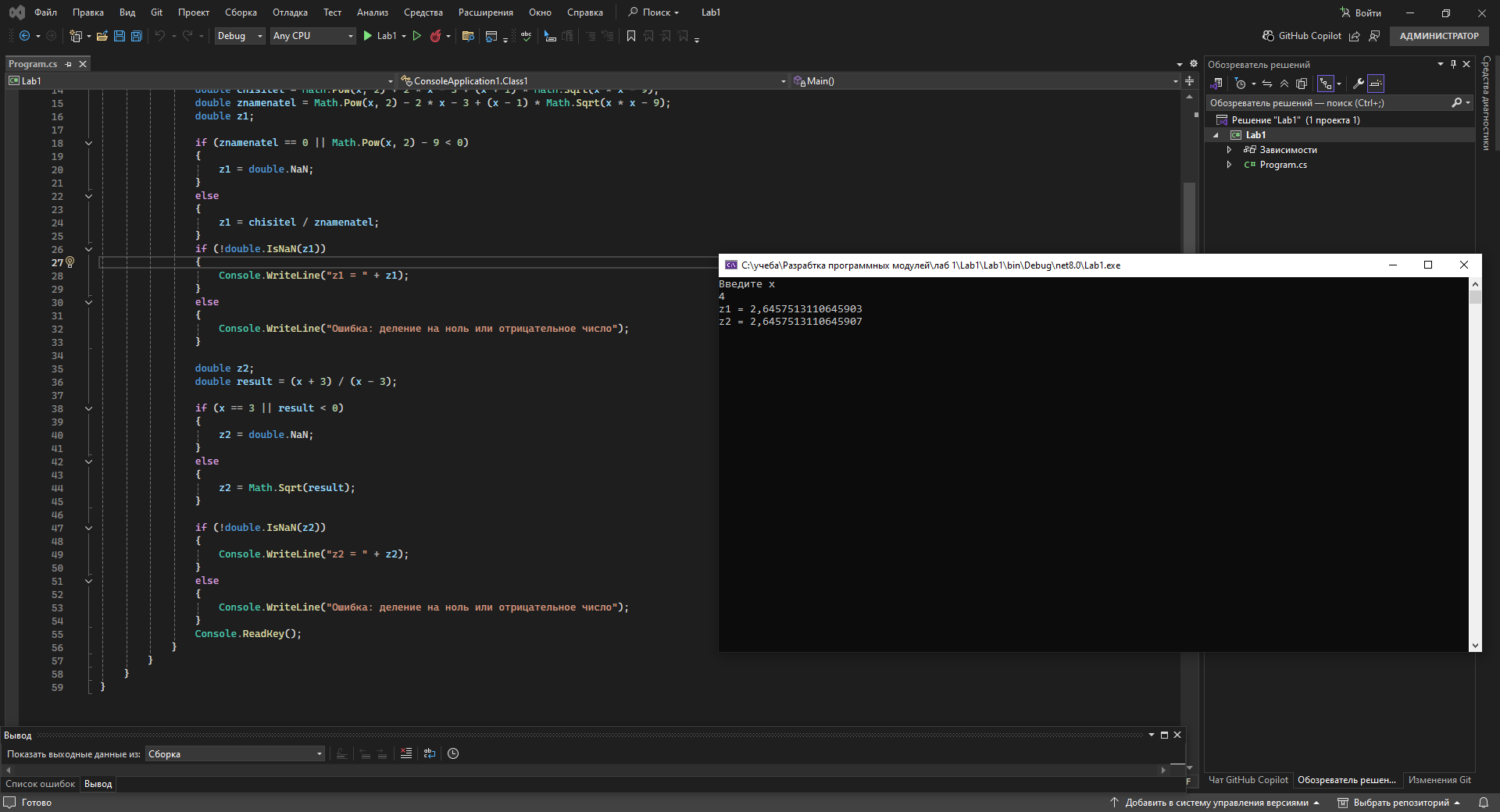
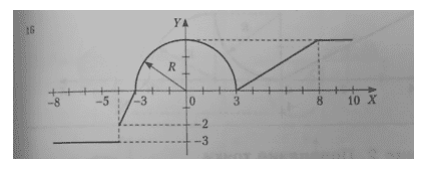


Рис.1 Результат выполнения программы.

Лабораторная работа 2.

Разветвляющиеся вычислительные процессы. Вариант 16.

Задание 1. Вычисление значения функции



Листинг кода:

using System;

using System.Data;

public class FunctionElevator

{

public static double EvaluteY(double x, double R)

{

System.Double y = 0;

if (x >= -8 && x<-4)

{

y = -3;

}

else if (x>=-4 && x<-3)

{

y = -(2 \* x + 6);

}

else if (x >= -3 && x < 3) {

y = Math.Sqrt(R \* R - (x \* x));

}

else if (x>=3 && x<8)

{

y = x - 3;

}

else if (x >= 8)

{

y = R;

}

return y;

}

public static void Main(string[] args)

{

Console.Write("Введите значение x: ");

if (double.TryParse(Console.ReadLine(), out double x))

{

Console.Write("Введите значение радиуса R: ");

if (double.TryParse(Console.ReadLine(), out double R))

{

double y = EvaluteY(x, R);

Console.WriteLine("Значение y: " + y);

}

else

{

Console.WriteLine("Некорректный ввод радиуса. Введите число.");

}

}

else

{

Console.WriteLine("Некорректный ввод x. Введите число.");

}

}

}

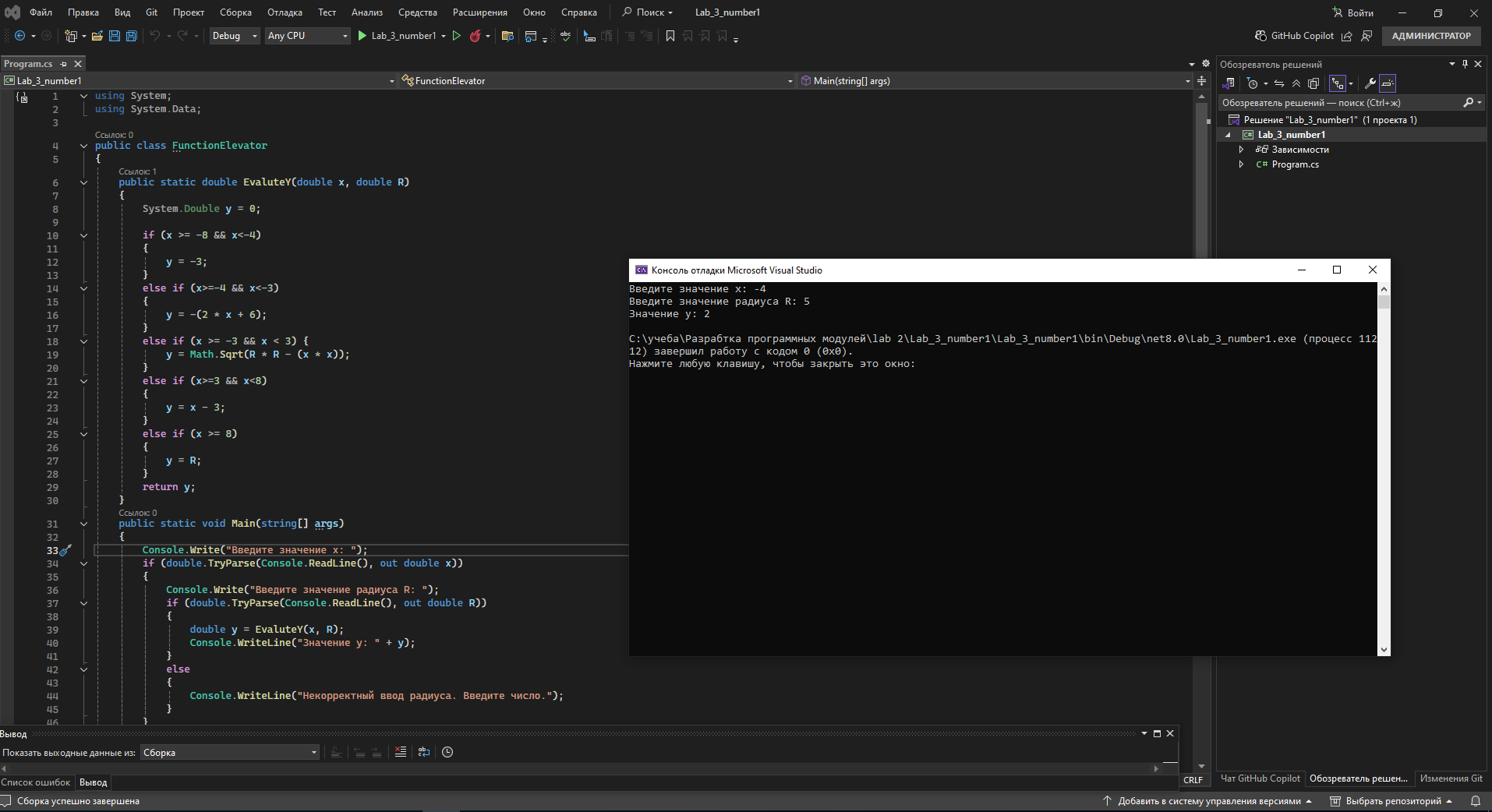
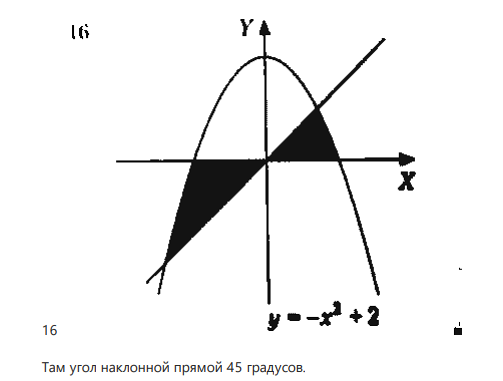


Рис. 2 Результат работы программы.

Задание 2. Попадание точки в заштрихованную область



Листинг кода:

using System;

public class BlackZoneChecker

{

public static void DoSmth(string[] args)

{

Console.Write("Введите значение x: ");

if (double.TryParse(Console.ReadLine(), out double x))

{

Console.Write("Введите значение y: ");

if (double.TryParse(Console.ReadLine(), out double y))

{

// Проверяем, находится ли x в интервале [-sqrt(2), sqrt(2)]

if (x >= -Math.Sqrt(2) && x <= Math.Sqrt(2))

{

// Если x в допустимом диапазоне, проверяем y

if (y >= -2 && y <= 1)

{

// Уравнение параболы: y = -x^2 + 2

double parabolaY = -Math.Pow(x, 2) + 2;

// Уравнение прямой: y = x

double lineY = x;

// Проверяем попадание в черную зону внутри этого интервала

if ((x <= 0 && y >= lineY && y <= parabolaY) || (x >= 0 && y <= lineY && y >= parabolaY))

{

Console.WriteLine("Точка ({0}, {1}) находится в черной зоне.", x, y);

}

else

{

Console.WriteLine("Точка ({0}, {1}) не находится в черной зоне.", x, y);

}

}

else

{

Console.WriteLine("Точка ({0}, {1}) не находится в черной зоне.", x, y);

}

}

else

{

Console.WriteLine("Точка ({0}, {1}) не находится в черной зоне.", x, y);

}

}

else

{

Console.WriteLine("Некорректный ввод y. Введите число."); // Ошибка здесь

}

}

else

{

Console.WriteLine("Некорректный ввод x. Введите число.");

}

}

}

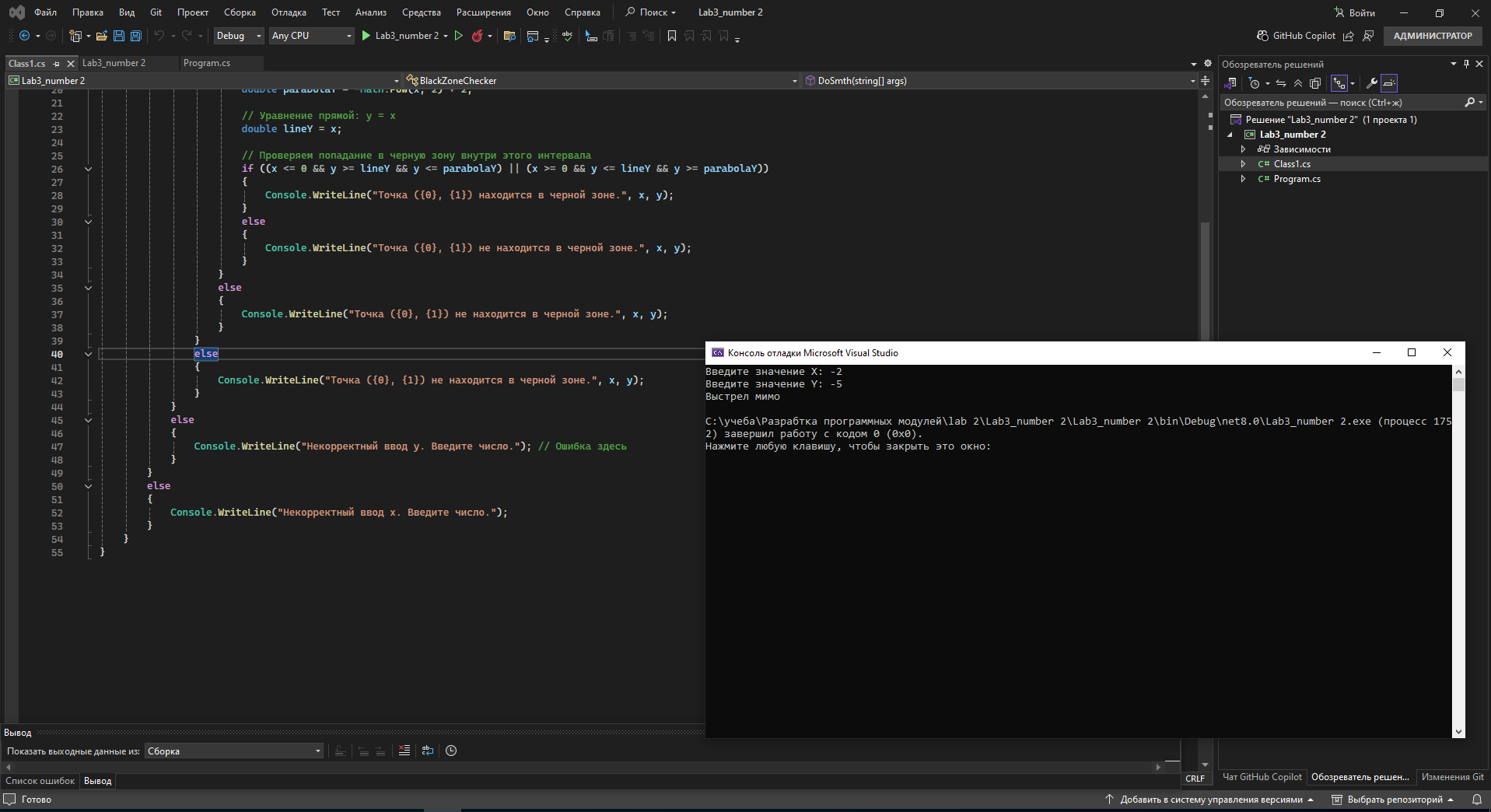


Рис 3. Результат работы программы.