Министерство цифрового развития, связи и массовых коммуникаций

Ордена трудового красного знамени

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Московский технический университет связи и информатики»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра «Информатика»

Лабораторные работы

по дисциплине

«Алгоритмизация и программирование»

Выполнил:

студент гр.БЭИ2102

Ахтямов Б.Р.

Проверил:

доцент кафедры “Информатика”

Волков А.И.

Москва 2022

**Лабораторная работа №1**

**«Программирование арифметических выражений на языке Visual C#»**

1. **Задание на разработку проекта и вариант задания**

Создать приложение для вычисления арифметического выражения:

; (1)

m = l; n =[l],

Где x, y - исходные данные – вещественные (double).

m, l, n – результат вычислений, причем n - целое число (int),

m, l – вещественные (double), а [] целая часть числа.

**2. Запись арифметических выражений, соответствующих заданным математическим формулам, по правилам С#**

l = (3 \* Math.Pow(y,3) - Math.Pow((Math.Pow(3,2) - 6 \* Math.Pow(10,-6) \* y),

1 / 6)) / (2 \* Math.Sin( 3 \* (x - 1)));

m = Math.Floor(l);

n = Convert.ToInt32(Math.Ceiling(l));

**3. Разработка проекта**

3.1. Разработка графического интерфейса пользователя

Анализируя задание на разработку проекта, можно сделать вывод, что для организации ввода и вывода данных можно использовать три элемента управления. Текстовое поле элемента ***textbox*** используется как для ввода исходных данных в программу, так и для отображения результатов, получаемых в этой программе. Элемент ***label*** используется для размещения на форме поясняющих надписей. Кнопка ***button*** используется для запуска и окончания программы и инициализируется щелчком левой кнопки мыши. Таким образом, форма может иметь вид показанный на рисунке 1.

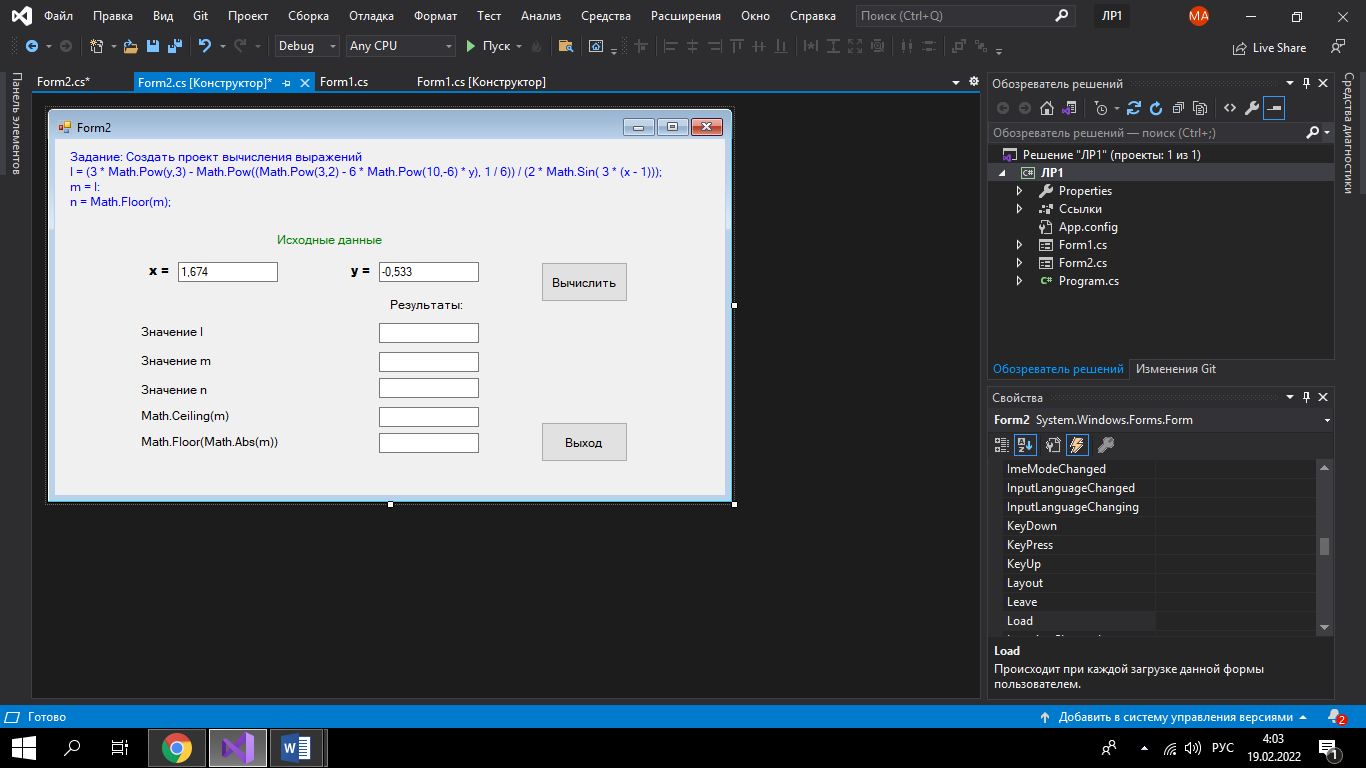


Рисунок 1 – графический интерфейс

3.2. Схема алгоритма решаемой задачи

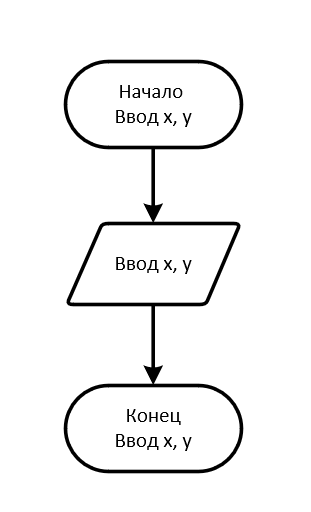


Рисунок 2 – Ввод х, у

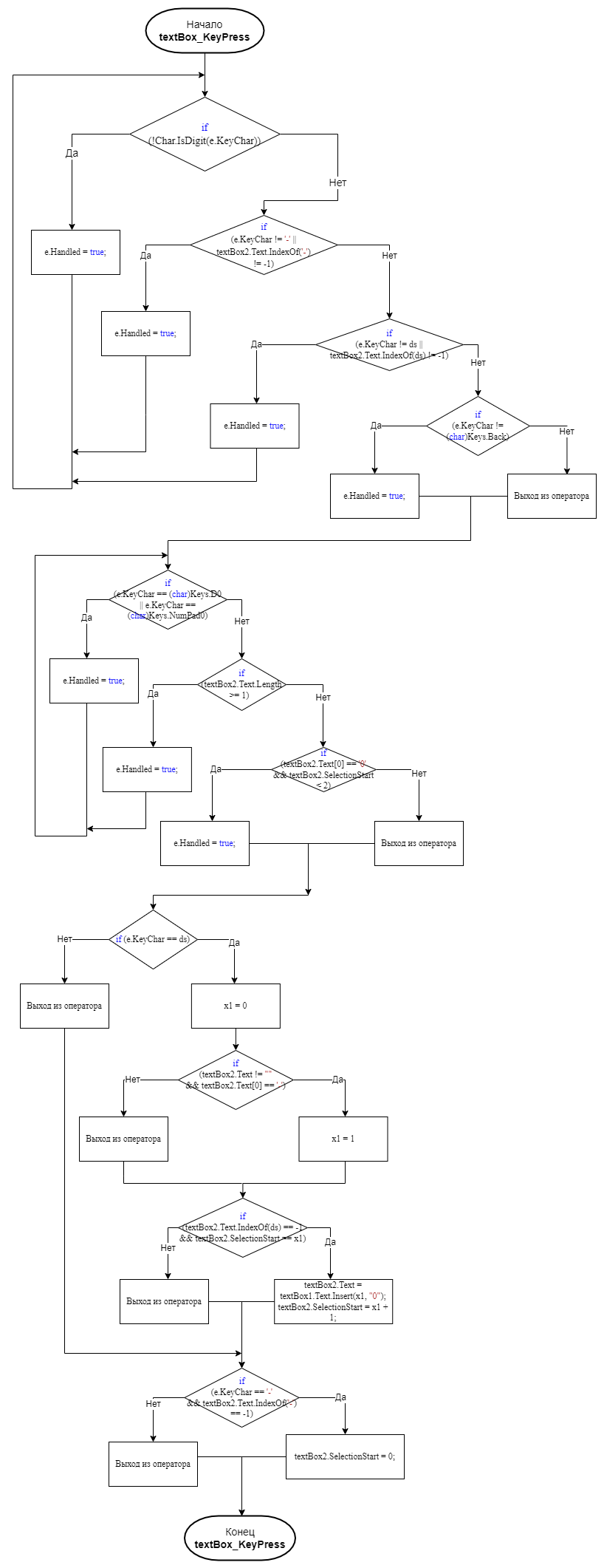


Рисунок 3 – Защита от дурака

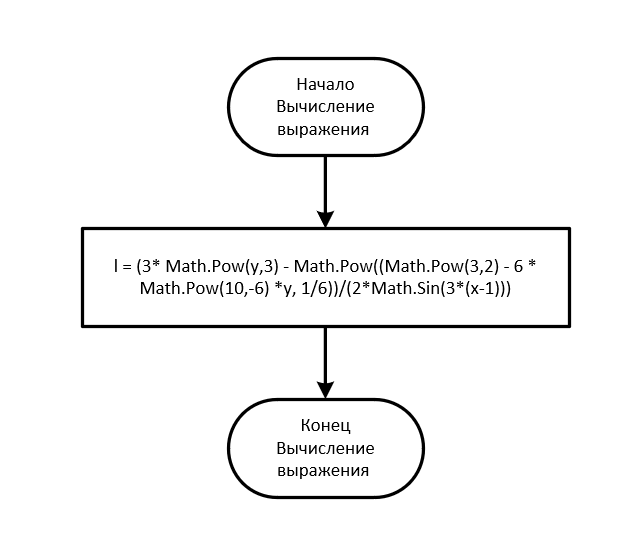


Рисунок 4 – Вычисление выражения

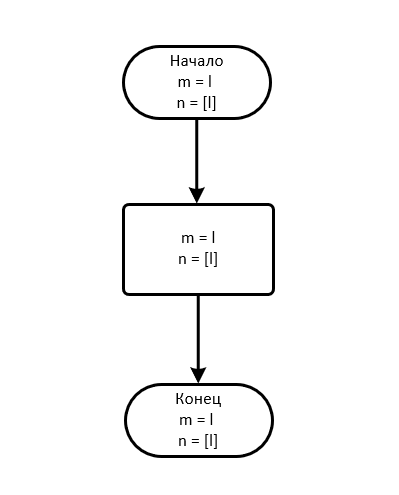


Рисунок 5 – Ввод m и n

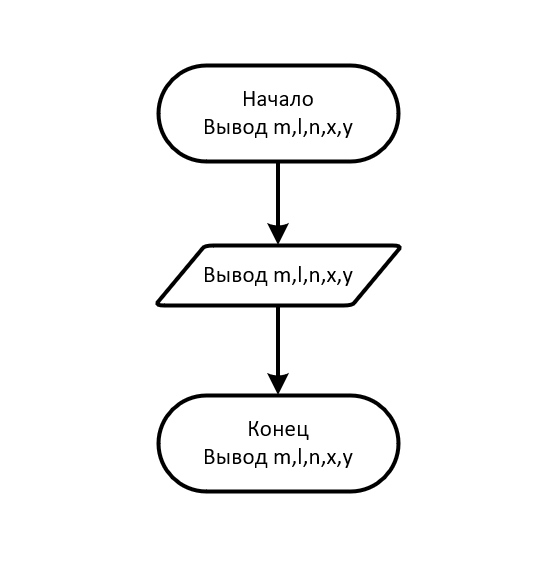


Рисунок 6 – Вывод переменных

3.3. Программный код проекта

Так как данные, введенные в текстовое поле TextBox являются текстовой строкой, то для корректных вычислений необходимо использовать явное преобразование текстового аргумента в числовое значение с помощью функции Convert.ToDouble().

Функция Convert.ToString( ) выполняет обратное преобразование.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Globalization;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AnP\_proj

{

public partial class Form2 : Form

{

int x1;

double x3;

char ds = (char)NumberFormatInfo.CurrentInfo.NumberDecimalSeparator[0];

public Form2()

{

InitializeComponent();

}

double x, y;

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

{

double m, l, n;

x = Convert.ToDouble(textBox1.Text);

y = Convert.ToDouble(textBox2.Text);

if (x == 1)

{

MessageBox.Show("Указанное значение переменной 'x' выходит за пределы области допустимых значений выражения!", "Ошибка!");

return;

}

l = (3 \* Math.Pow(y, 3) - Math.Pow((Math.Pow(3, 2) - 6 \* Math.Pow(10, - 6) \* y), 1 / 6)) / (2 \* Math.Sin(3 \* (x - 1)));

textBox4.Text = Convert.ToString(l);

m = l;

textBox5.Text = Convert.ToString(m);

n = Math.Floor(l);

textBox6.Text = Convert.ToString(n);

textBox7.Text = Convert.ToString(Math.Ceiling(m));

textBox8.Text = Convert.ToString(Math.Floor(Math.Abs(m)));

}

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

{

Application.Exit();

}

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

// Если нажатая клавиша не цифровая.

if (!Char.IsDigit(e.KeyChar))

// Запрет на ввод более одного знака минуса.

if (e.KeyChar != '-' || textBox2.Text.IndexOf('-') != -1)

// Запрет на ввод более одного десятичного разделителя.

if (e.KeyChar != ds || textBox2.Text.IndexOf(ds) != -1)

// Если нажатая клавиша не является клавишей BackSpace.

if (e.KeyChar != (char)Keys.Back)

e.Handled = true; // Запрет ввода

// Запрет ввода повторных нулей в

//начале числа.

//-незначащий ноль

if (Char.IsDigit(e.KeyChar))

if (textBox2.Text.Length == 1 && textBox2.Text[0] == '0')

textBox2.Paste(",");

if (e.KeyChar == (char)Keys.D0 || e.KeyChar == (char)Keys.NumPad0)

if (textBox2.Text.Length >= 1) // Если строка не пустая

if (textBox2.Text[0] == '0' && textBox2.SelectionStart < 2)

e.Handled = true; // Запрет ввода

// Замена десятичного разделителя в начале числа на "0,".

if (e.KeyChar == ds)

{ // Если введен десятичный разделитель

x1 = 0;

if (textBox2.Text != "" && textBox2.Text[0] == '-') { x1 = 1; }

if (textBox2.Text.IndexOf(ds) == -1 && textBox2.SelectionStart == x1)

{

// Если десятичный разделитель уже есть, то 0 не вставлять.

textBox2.Text = textBox1.Text.Insert(x1, "0"); // Вставка 0

textBox2.SelectionStart = x1 + 1; // Текстовый курсор в конец

}

}

// Перенос точки ввода минуса в начало (ввод минуса только спереди)

if (e.KeyChar == '-' && textBox2.Text.IndexOf('-') == -1)

{

textBox2.SelectionStart = 0; // Текстовый курсор в начало

}

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

// Если нажатая клавиша не цифровая.

if (!Char.IsDigit(e.KeyChar))

// Запрет на ввод более одного знака минуса.

if (e.KeyChar != '-' || textBox1.Text.IndexOf('-') != -1)

// Запрет на ввод более одного десятичного разделителя.

if (e.KeyChar != ds || textBox1.Text.IndexOf(ds) != -1)

// Если нажатая клавиша не является клавишей BackSpace.

if (e.KeyChar != (char)Keys.Back)

e.Handled = true; // Запрет ввода

// Запрет ввода повторных нулей в начале числа.

if (e.KeyChar == (char)Keys.D0 || e.KeyChar == (char)Keys.NumPad0)

if (textBox1.Text.Length >= 1) // Если строка не пустая

if (textBox1.Text[0] == '0' && textBox1.SelectionStart < 2)

e.Handled = true; // Запрет ввода

//-незначащий ноль

if (Char.IsDigit(e.KeyChar))

if (textBox1.Text.Length == 1 && textBox1.Text[0] == '0')

textBox1.Paste(",");

// Замена десятичного разделителя в начале числа на "0,".

if (e.KeyChar == ds)

{ // Если введен десятичный разделитель

x1 = 0;

if (textBox1.Text != "" && textBox1.Text[0] == '-') { x1 = 1; }

if (textBox1.Text.IndexOf(ds) == -1 && textBox1.SelectionStart == x1)

{

// Если десятичный разделитель уже есть, то 0 не вставлять.

textBox1.Text = textBox1.Text.Insert(x1, "0"); // Вставка 0

textBox1.SelectionStart = x1 + 1; // Текстовый курсор в конец

}

// Перенос точки ввода минуса в начало (ввод минуса только спереди)

}

if (e.KeyChar == '-' && textBox1.Text.IndexOf('-') == -1)

textBox1.SelectionStart = 0; // Текстовый курсор в начало

//-незначащий ноль

if (Char.IsDigit(e.KeyChar))

if (textBox2.Text.Length == 1 && textBox2.Text[0] == '0')

textBox2.Paste(",");

}

}

}

**4. Результаты выполнения проекта**

Результат выполнения проекта приведён на рисунке 7.

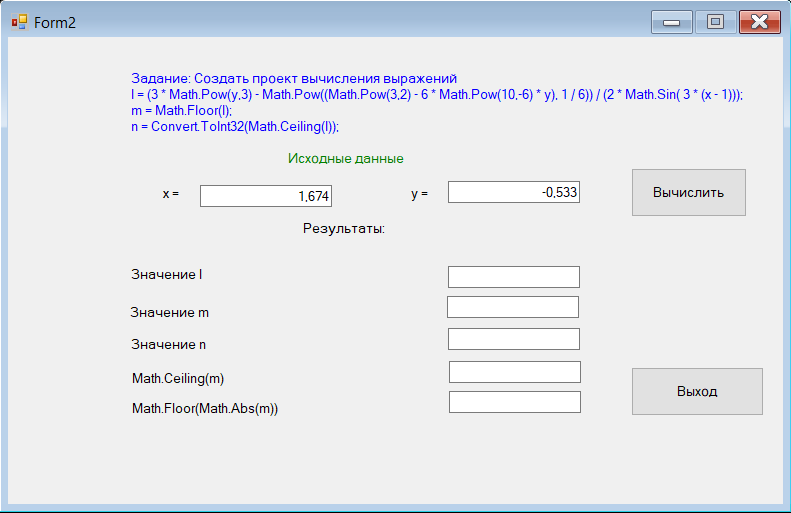


Рисунок 7 – Выполнение проекта

**5. Доказательство правильности работы программы**

Проверка была выполнена с помощью приложения онлайн компилятора Python (рисунок 9), но ответы не совпали, из-за того, что у с# есть погрешность в вычислениях (рисунок 10)

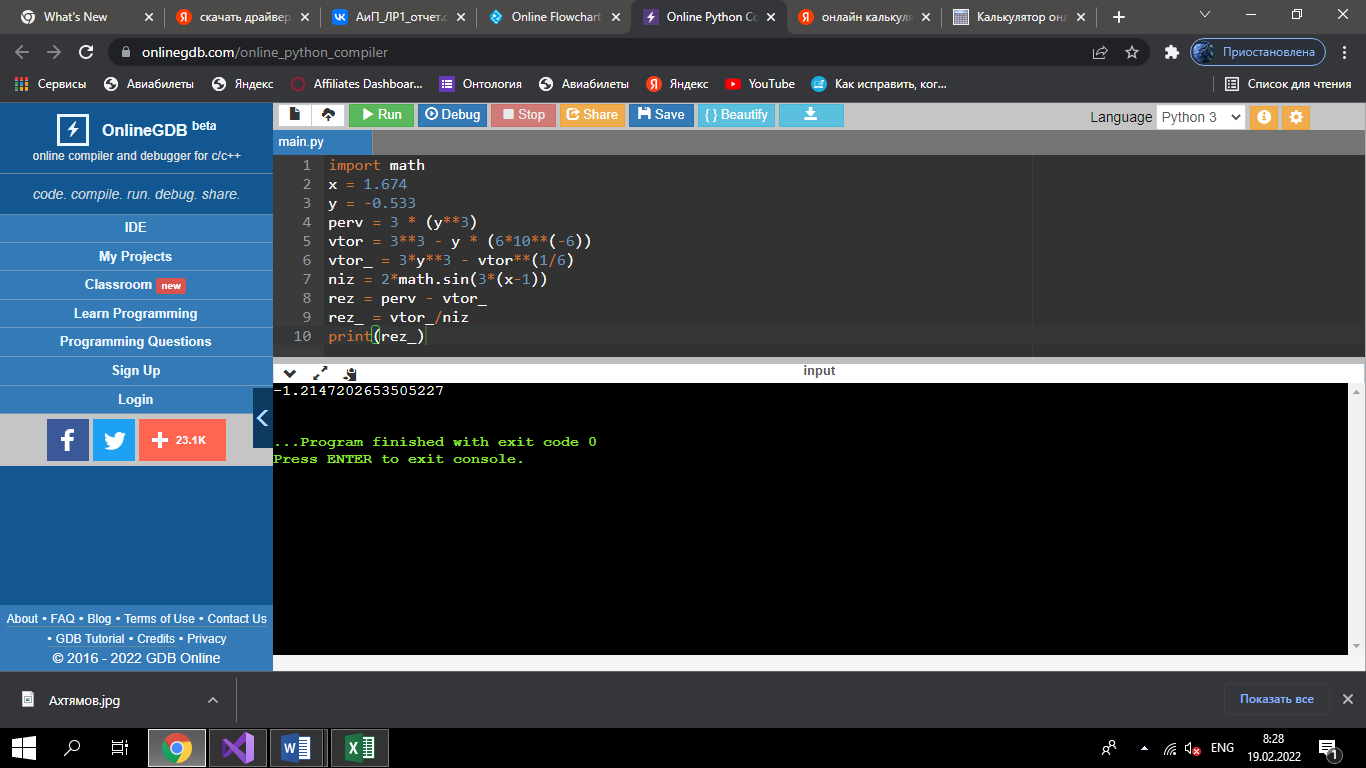


Рисунок 8 – Python

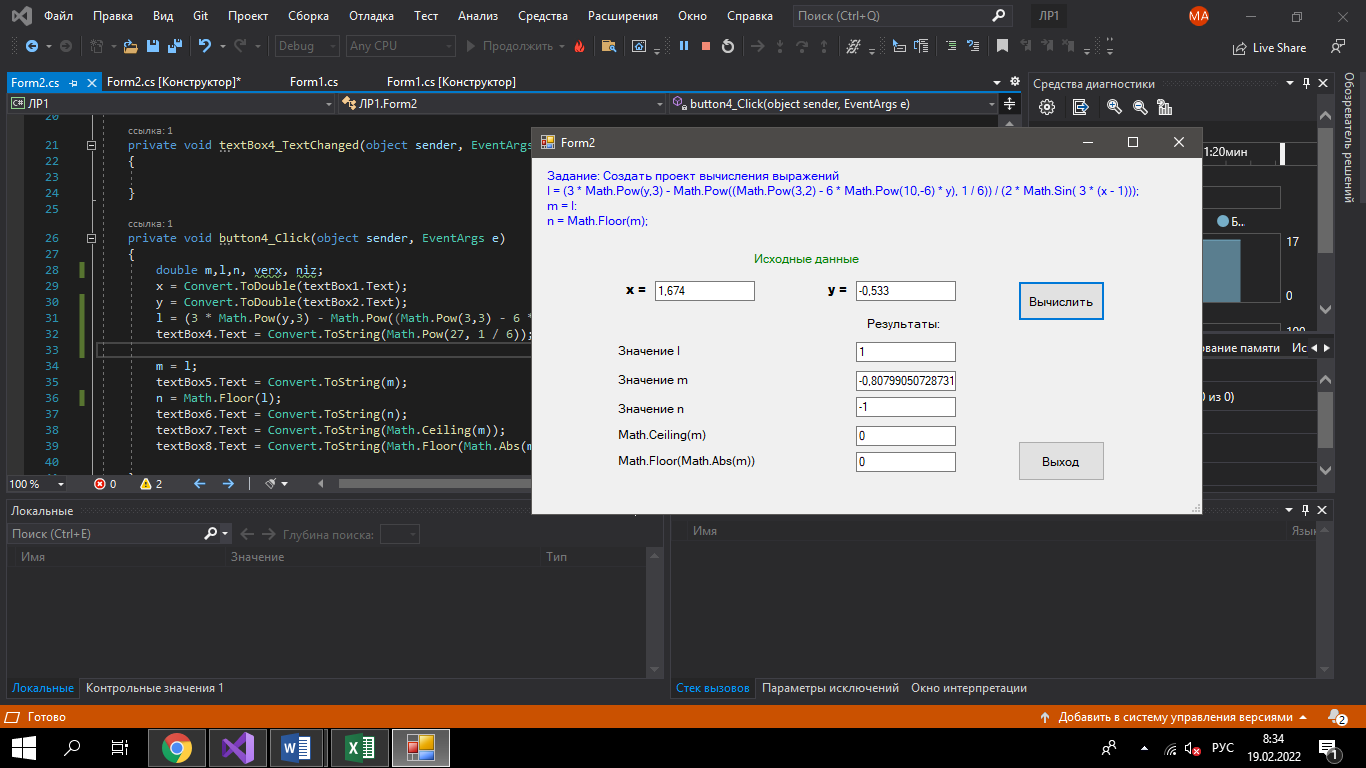


Рисунок 9 - погрешность