Лабораторная работа №2

«Запись арифметических выражений. Использование проверки корректности данных»

**Задание**

Выполнить расчёт следующих арифметических выражений на основе исходных данных. Кроме того, использовать проверку корректности ввода данных, которая заключается в том, что пользователь в поля ввода может вводить только цифры, знак «минус», запятую в качестве разделителя, а также разрешена клавиша «Backspace».

Выражение:

**Запись арифметических выражений на языке программирования.**

Вычисление арифметического выражения и сохранение в переменную представлено следующим образом:

double lab2\_result\_z = ( Math.Sin(x) / Math.Sqrt(1+Math.Pow(m,2) \* Math.Pow(Math.Sin(x),2))) - c \* m \* Math.Log(m \* x);

double lab2\_result\_s = Math.Exp(-a\*x) \* Math.Sqrt(x+1) + Math.Exp(-b\*x) \* Math.Sqrt(x+1.5);

* 1. **Алгоритм решения арифметического выражения.**

Входными данными для алгоритма являются пять значений: "m", "c", "x", "a" и "b". После ввода этих значений, они обрабатываются в соответствии с описанными ранее формулами, и результаты записываются в переменные "s" и "z". Алгоритм решения задачи представлен на рисунке 2.

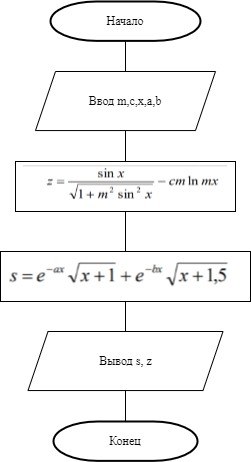


Рисунок 2 – Алгоритм решения арифметического выражения

* 1. **Код на языке С# для решения поставленной задачи**

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;

namespace Лабораторная\_работа\_\_2

{

public partial class CalcLab2 : Form

{

public CalcLab2()

{

InitializeComponent();

}

private void Input\_m\_lab2\_KeyPress(object sender, KeyPressEventArgs e)

{

string inputText = Input\_m\_lab2.Text;

// Разрешить обработку команды Backspace

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

{

return;

}

// Разрешить ввод цифр

if (char.IsDigit(e.KeyChar))

{

return;

}

// Разрешить ввод одного знака минус

if (e.KeyChar == '-' && !inputText.Contains("-") && !inputText.EndsWith(","))

{

return;

}

// Разрешить ввод одного знака запятой

if (e.KeyChar == ',' && !inputText.Contains(",") && !inputText.EndsWith("-"))

{

return;

}

// Запретить ввод, если символ является минусом или запятой и в поле уже присутствует соответствующий символ

if ((e.KeyChar == '-' && (inputText.Contains("-") || inputText.EndsWith(","))) ||

(e.KeyChar == ',' && (inputText.Contains(",") || inputText.EndsWith("-"))))

{

e.Handled = true;

return;

}

e.Handled = true;

}

private void Input\_c\_lab2\_KeyPress(object sender, KeyPressEventArgs e)

{

string inputText = Input\_c\_lab2.Text;

// Разрешить обработку команды Backspace

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

{

return;

}

// Разрешить ввод цифр

if (char.IsDigit(e.KeyChar))

{

return;

}

// Разрешить ввод одного знака минус

if (e.KeyChar == '-' && !inputText.Contains("-") && !inputText.EndsWith(","))

{

return;

}

// Разрешить ввод одного знака запятой

if (e.KeyChar == ',' && !inputText.Contains(",") && !inputText.EndsWith("-"))

{

return;

}

// Запретить ввод, если символ является минусом или запятой и в поле уже присутствует соответствующий символ

if ((e.KeyChar == '-' && (inputText.Contains("-") || inputText.EndsWith(","))) ||

(e.KeyChar == ',' && (inputText.Contains(",") || inputText.EndsWith("-"))))

{

e.Handled = true;

return;

}

e.Handled = true;

}

private void Input\_x\_lab2\_KeyPress(object sender, KeyPressEventArgs e)

{

string inputText = Input\_x\_lab2.Text;

// Разрешить обработку команды Backspace

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

{

return;

}

// Разрешить ввод цифр

if (char.IsDigit(e.KeyChar))

{

return;

}

// Разрешить ввод одного знака минус

if (e.KeyChar == '-' && !inputText.Contains("-") && !inputText.EndsWith(","))

{

return;

}

// Разрешить ввод одного знака запятой

if (e.KeyChar == ',' && !inputText.Contains(",") && !inputText.EndsWith("-"))

{

return;

}

// Запретить ввод, если символ является минусом или запятой и в поле уже присутствует соответствующий символ

if ((e.KeyChar == '-' && (inputText.Contains("-") || inputText.EndsWith(","))) ||

(e.KeyChar == ',' && (inputText.Contains(",") || inputText.EndsWith("-"))))

{

e.Handled = true;

return;

}

e.Handled = true;

}

private void Input\_a\_lab2\_KeyPress(object sender, KeyPressEventArgs e)

{

string inputText = Input\_a\_lab2.Text;

// Разрешить обработку команды Backspace

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

{

return;

}

// Разрешить ввод цифр

if (char.IsDigit(e.KeyChar))

{

return;

}

// Разрешить ввод одного знака минус

if (e.KeyChar == '-' && !inputText.Contains("-") && !inputText.EndsWith(","))

{

return;

}

// Разрешить ввод одного знака запятой

if (e.KeyChar == ',' && !inputText.Contains(",") && !inputText.EndsWith("-"))

{

return;

}

// Запретить ввод, если символ является минусом или запятой и в поле уже присутствует соответствующий символ

if ((e.KeyChar == '-' && (inputText.Contains("-") || inputText.EndsWith(","))) ||

(e.KeyChar == ',' && (inputText.Contains(",") || inputText.EndsWith("-"))))

{

e.Handled = true;

return;

}

e.Handled = true;

}

private void Input\_b\_lab2\_KeyPress(object sender, KeyPressEventArgs e)

{

string inputText = Input\_b\_lab2.Text;

// Разрешить обработку команды Backspace

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

{

return;

}

// Разрешить ввод цифр

if (char.IsDigit(e.KeyChar))

{

return;

}

// Разрешить ввод одного знака минус

if (e.KeyChar == '-' && !inputText.Contains("-") && !inputText.EndsWith(","))

{

return;

}

// Разрешить ввод одного знака запятой

if (e.KeyChar == ',' && !inputText.Contains(",") && !inputText.EndsWith("-"))

{

return;

}

// Запретить ввод, если символ является минусом или запятой и в поле уже присутствует соответствующий символ

if ((e.KeyChar == '-' && (inputText.Contains("-") || inputText.EndsWith(","))) ||

(e.KeyChar == ',' && (inputText.Contains(",") || inputText.EndsWith("-"))))

{

e.Handled = true;

return;

}

e.Handled = true;

}

private void Make\_result\_lab2\_Click(object sender, EventArgs e)

{

// Получаем значения из полей ввода

string input\_m = Input\_m\_lab2.Text;

string input\_c = Input\_c\_lab2.Text;

string input\_x = Input\_x\_lab2.Text;

string input\_a = Input\_a\_lab2.Text;

string input\_b = Input\_b\_lab2.Text;

// Переменная для проверки успешности ввода

bool validInput = true;

// Проверка и парсинг значения m

if (!double.TryParse(input\_m, out double m))

{

validInput = false;

MessageBox.Show("Неверный формат данных для m.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

// Проверка и парсинг значения c

if (!double.TryParse(input\_c, out double c))

{

validInput = false;

MessageBox.Show("Неверный формат данных для c.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

// Проверка и парсинг значения x

if (!double.TryParse(input\_x, out double x))

{

validInput = false;

MessageBox.Show("Неверный формат данных для x.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

// Проверка и парсинг значения a

if (!double.TryParse(input\_a, out double a))

{

validInput = false;

MessageBox.Show("Неверный формат данных для a.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

// Проверка и парсинг значения b

if (!double.TryParse(input\_b, out double b))

{

validInput = false;

MessageBox.Show("Неверный формат данных для b.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

// Выполнение вычислений, если ввод данных был успешным //

if (validInput)

{

//Вычесление z

double lab2\_result\_z = ( Math.Sin(x) / Math.Sqrt(1+Math.Pow(m,2) \* Math.Pow(Math.Sin(x),2))) - c \* m \* Math.Log(m \* x);

lab2\_Textbox\_result\_z.Text = lab2\_result\_z.ToString();

//Вычесление s

double lab2\_result\_s = Math.Exp(-a\*x) \* Math.Sqrt(x+1) + Math.Exp(-b\*x) \* Math.Sqrt(x+1.5);

lab2\_Textbox\_result\_s.Text = lab2\_result\_s.ToString();

}

}

private void Lab2\_exit\_Click(object sender, EventArgs e)

{

this.Close(); // Закрывает последнюю страничку

}

}

}

* 1. **Результаты выполнения программы**

Вся необходимая информация об учащемся и проверяющем указана на титульной форме (см. рисунок 3)

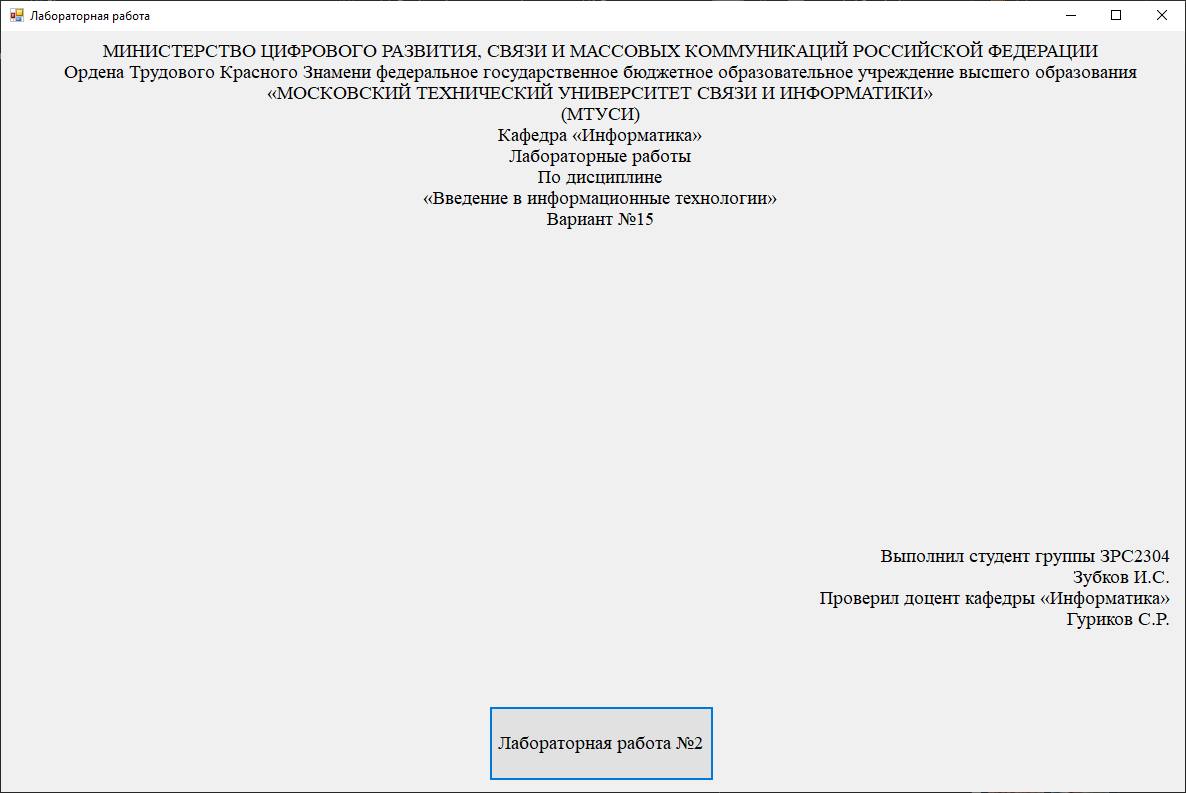


Рисунок 3 – Титульное окно

На второй форме предоставлены краткая информация о работе и формулы. (см. рисунок 4)

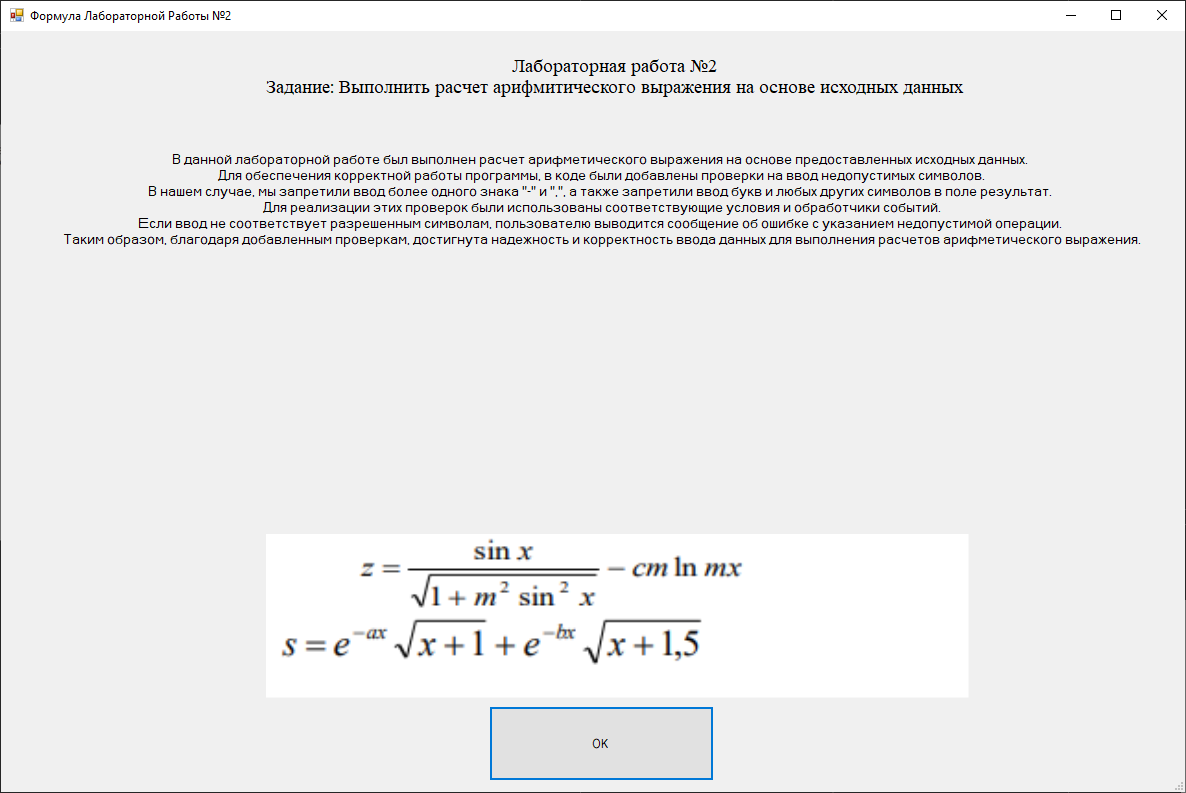


Рисунок 4 – Формула и краткая информация

Визуальное представление решения задачи с указанными данными представлено на рисунке 5.

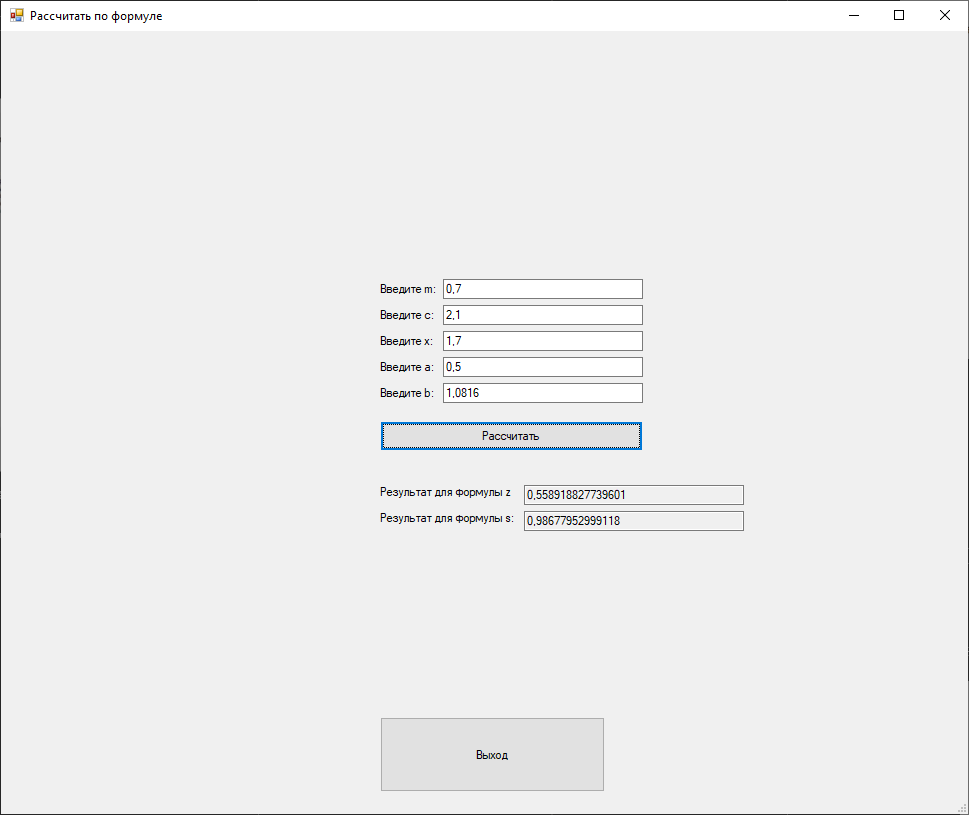


Рисунок 5 – Решение задачи

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гуриков, С. Р. Введение в программирование на языке Visual C#: учебное пособие / С. Р. Гуриков. — Москва: ФОРУМ: ИНФРА-М, 2019. — 447 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-540-0. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1012397> (дата обращения: 10.10.2023).

2. ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления" (введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст) (дата обращений 10.10.2023).

3. ГОСТ 19.701-90 (ИСО 5807-85) “Единая система программной документации СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ, ДАННЫХ И СИСТЕМ Обозначения условные и правила выполнения”. (дата обращения 10.10.2023).

Лабораторная работа №3

«Вычисление значений выражения с использованием метода пошаговой детализации»

**Задание:**

Создать приложение для вычисления данного арифметического выражения.

**Вычислить** координаты точки пересечения двух прямых:

**a1 x + b1 y = c1**

**a2 x + b2 y = c2,**

вычисляемые по формулам: **; ,** где

;

**Назначение метода:** Вычисление определителя второго порядка.

Для данной лабораторной были созданы следующие методы:

public static double Input\_user (TextBox textbox) – Для считывания ввода данных пользователя и последующая конвертации в формат double.

public static double Lab3\_Matrix(double a1, double a2, double b1, double b2) – Для высчитывания определителя второго порядка.

public static void Lab3\_All\_Out(double a1, double a2, double b1, double b2, double c1, double c2, out double Lab3\_Res\_X, out double Lab3\_Res\_Y) – Для получения результата с помощью вызова метода Lab3\_Matrix. Возврат значений происходит методом out.

public static void Out\_info(TextBox usr\_inp, double test) – Для вывода полученных данных.

Все эти методы находятся в DLL библиотеки.

Список блок-схем, использующихся в лабораторной работе:

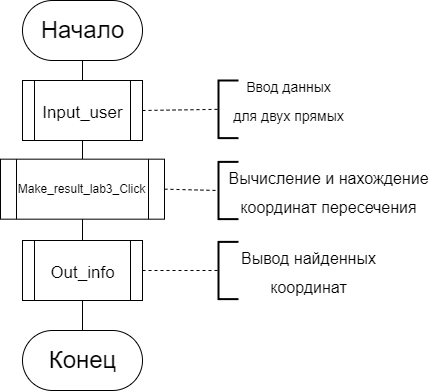


Схема 1 – Алгоритм событийной процедуры.

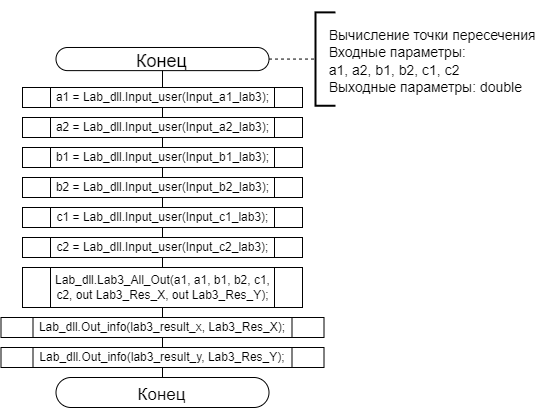


Схема 2 – Схема алгоритма решения задачи.

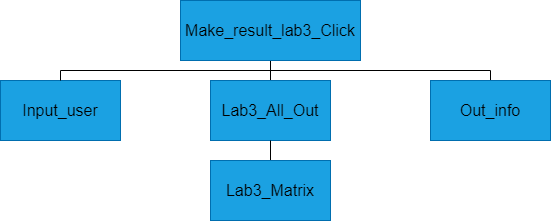


Схема 3 – Иерархия методов для решения задачи.

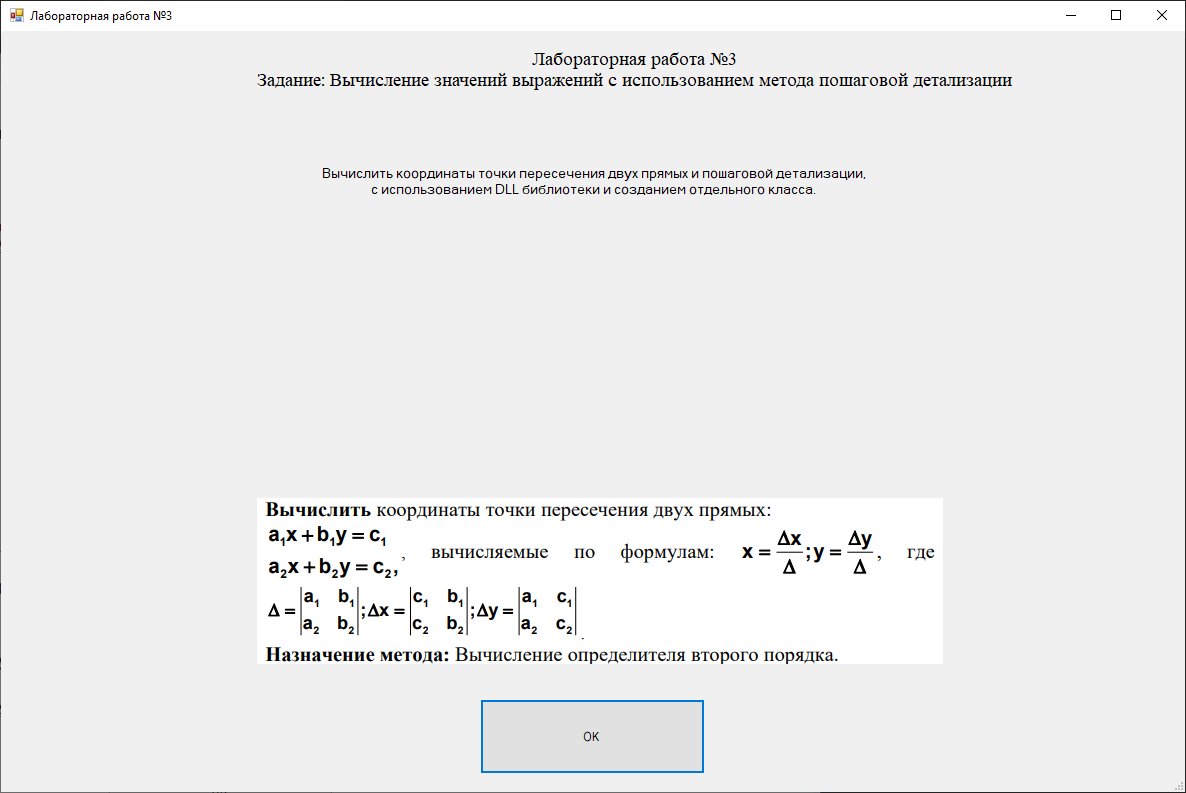


Рисунок 1 – Информация о задаче.

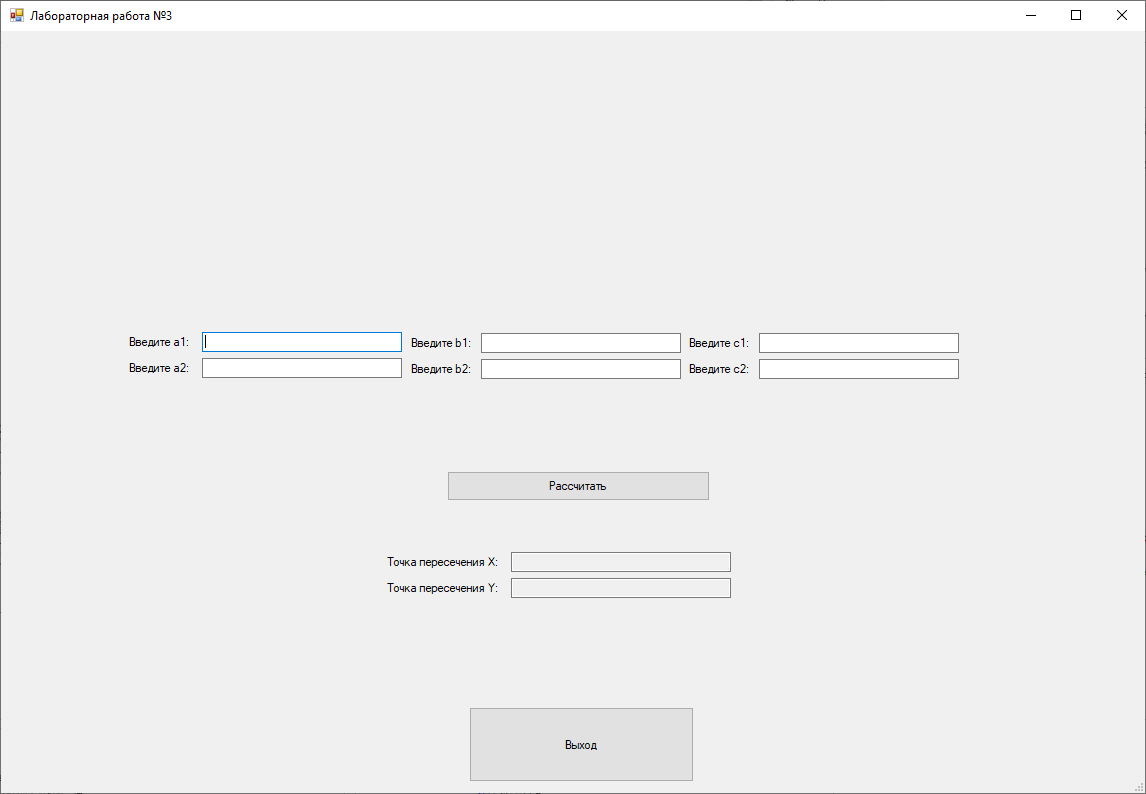


Рисунок 2 – Поле ввода значений.

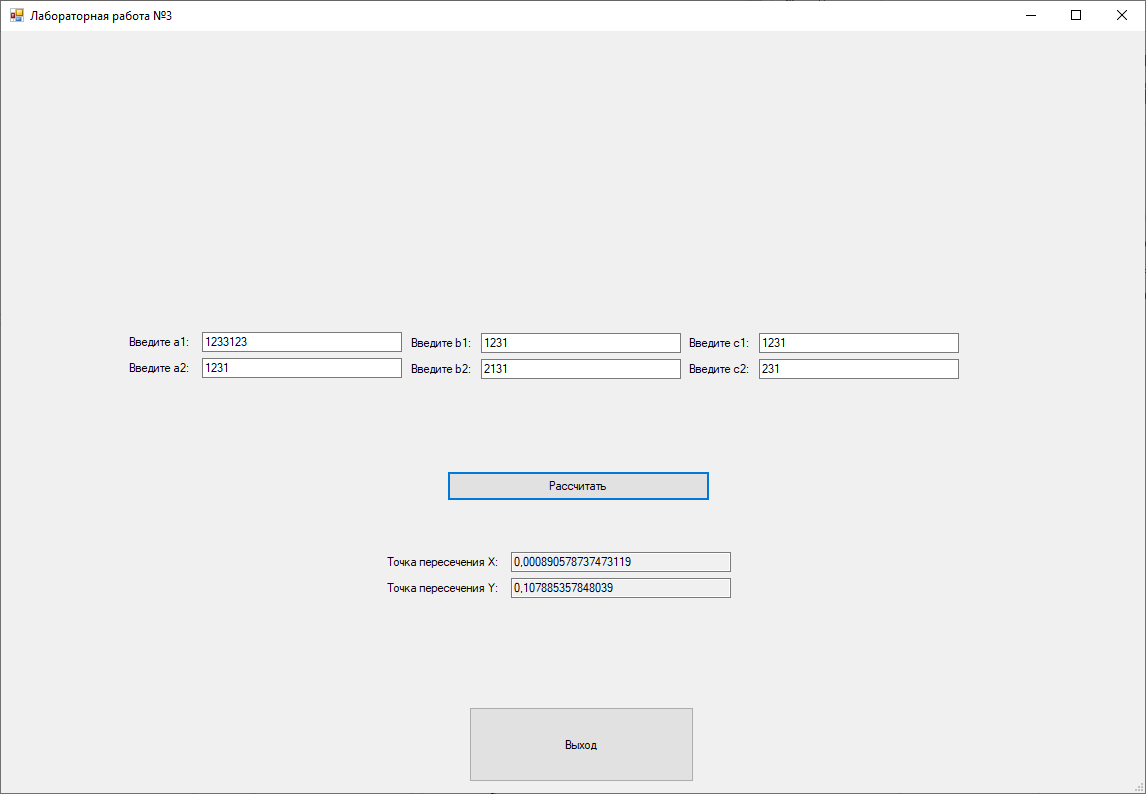


Рисунок 3 – Результат работы программы.

Программные методы находятся в классе:

using Dll\_lab;

using System;

using System.Collections.Generic;

using System.Security.Cryptography;

using System.Windows.Forms;

namespace Dll\_lab

{

public class Lab\_dll

{

// Создать универсальные методы для ввода и вывода

public static void Input\_lab3\_KeyPress(KeyPressEventArgs e, string inputText)

{

// Разрешить обработку команды Backspace

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

return;

// Разрешить ввод одной запятой

if (e.KeyChar == ',' && !inputText.Contains(",") && (inputText.Length != 0))// && !inputText.EndsWith("-"))

return;

// Разрешить ввод цифр

if (char.IsDigit(e.KeyChar))

return;

// Разрешить ввод одного знака минус только в начале

if (e.KeyChar == '-' && (inputText.Length == 0))

return;

// Запретить ввод, если символ является минусом или запятой и в поле уже присутствует соответствующий символ

if ((e.KeyChar == '-' && (inputText.Contains("-") || inputText.EndsWith(","))) || (e.KeyChar == ',' && (inputText.Contains(",") || inputText.EndsWith("-"))))

{

e.Handled = true;

return;

}

e.Handled = true;

}

public static double Input\_user(TextBox textbox)

{

return Convert.ToDouble(textbox.Text);

}

public static void Out\_info(TextBox usr\_inp, double test)

{

usr\_inp.Text = test.ToString();

}

public static double Lab3\_Matrix(double a1, double a2, double b1, double b2)

{

double result\_delta = (a1 \* b2) - (a2 \* b1);

return result\_delta;

}

public static void Lab3\_All\_Out(double a1, double a2, double b1, double b2, double c1, double c2, out double Lab3\_Res\_X, out double Lab3\_Res\_Y)

{

double delta = Lab3\_Matrix(a1, a2, b1, b2);

double delta\_x = Lab3\_Matrix(c1, c2, b1, b2);

double delta\_y = Lab3\_Matrix(a1, a2, c1, c2);

Lab3\_Res\_X = delta\_x / delta;

Lab3\_Res\_Y = delta\_y / delta;

}

}

}

Основная часть программы:

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;

using Dll\_lab;

namespace Лабораторная\_работа\_\_2

{

public partial class CalcLab3 : Form

{

public CalcLab3()

{

InitializeComponent();

}

private void Make\_result\_lab3\_Click(object sender, EventArgs e)

{

double a1 = Lab\_dll.Input\_user(Input\_a1\_lab3);

double a2 = Lab\_dll.Input\_user(Input\_a2\_lab3);

double b1 = Lab\_dll.Input\_user(Input\_b1\_lab3);

double b2 = Lab\_dll.Input\_user(Input\_b2\_lab3);

double c1 = Lab\_dll.Input\_user(Input\_c1\_lab3);

double c2 = Lab\_dll.Input\_user(Input\_c2\_lab3); // Берем данные

double Lab3\_Res\_X = 0;

double Lab3\_Res\_Y = 0;

Lab\_dll.Lab3\_All\_Out(a1, a1, b1, b2, c1, c2, out Lab3\_Res\_X, out Lab3\_Res\_Y);

Lab\_dll.Out\_info(lab3\_result\_x, Lab3\_Res\_X);

Lab\_dll.Out\_info(lab3\_result\_y, Lab3\_Res\_Y);

}

private void Input\_a1\_lab3\_KeyPress(object sender, KeyPressEventArgs e)

{

Lab\_dll.Input\_lab3\_KeyPress(e, Input\_a1\_lab3.Text);

}

private void Input\_a2\_lab3\_KeyPress(object sender, KeyPressEventArgs e)

{

Lab\_dll.Input\_lab3\_KeyPress(e, Input\_a2\_lab3.Text);

}

private void Input\_b1\_lab3\_KeyPress(object sender, KeyPressEventArgs e)

{

Lab\_dll.Input\_lab3\_KeyPress(e, Input\_b1\_lab3.Text);

}

private void Input\_b2\_lab3\_KeyPress(object sender, KeyPressEventArgs e)

{

Lab\_dll.Input\_lab3\_KeyPress(e, Input\_b2\_lab3.Text);

}

private void Input\_c1\_lab3\_KeyPress(object sender, KeyPressEventArgs e)

{

Lab\_dll.Input\_lab3\_KeyPress(e, Input\_c1\_lab3.Text);

}

private void Input\_c2\_lab3\_KeyPress(object sender, KeyPressEventArgs e)

{

Lab\_dll.Input\_lab3\_KeyPress(e, Input\_c2\_lab3.Text);

}

private void Lab3\_exit\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гуриков, С. Р. Введение в программирование на языке Visual C#: учебное пособие / С. Р. Гуриков. — Москва: ФОРУМ: ИНФРА-М, 2019. — 447 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-540-0. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1012397> (дата обращения: 02.11.2023).

2. ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления" (введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст) (дата обращений 02.11.2023).

Лабораторная работа №4

«Программирование алгоритмов разветвляющихся структур»

**Задание**

Разработать приложение определяющее значение d при помощи многозначных ветвленийпо следующей системе:

Для выполнения поставленного задания необходимо разработать следующие функции:

public static double Input\_user(TextBox textbox) – код принимает в качестве аргумента ввод пользователя и возвращает число.

public static double Lab4\_Make\_result(double a, double x, double y) – принимает 3 переменных и возвращает **d**.

public static void Out\_info(TextBox usr\_inp, double test) – принимает имя textBox куда нужно поместить второй переданный параметр test предварительно преобразовав в строку.

Данные функции требуется поместить в предварительно созданной DLL-библиотеке.

**1.1 Алгоритм решения поставленной задачи.**

Схема алгоритма событийной процедуры решения задачи представлена на рисунке 1.

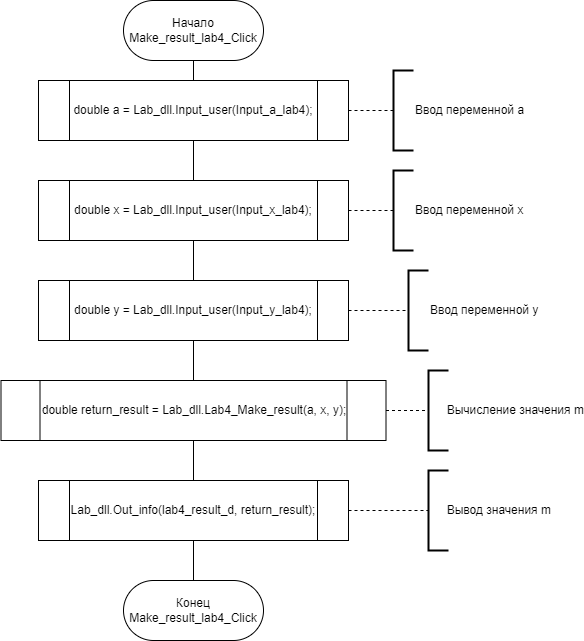


Рисунок 1 – Схема алгоритма событийной процедуры

На рисунке 2 изображена структура алгоритма процедуры, которая выполняет обработку введенных пользователем данных.

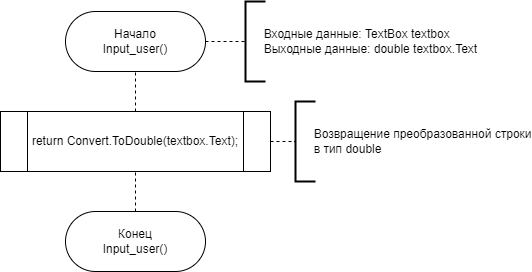


Рисунок 2 – Схема обработки ввода пользователя

На рисунке 3 представлена схема алгоритма процедуры, которая вычисляет значение переменной **d**.

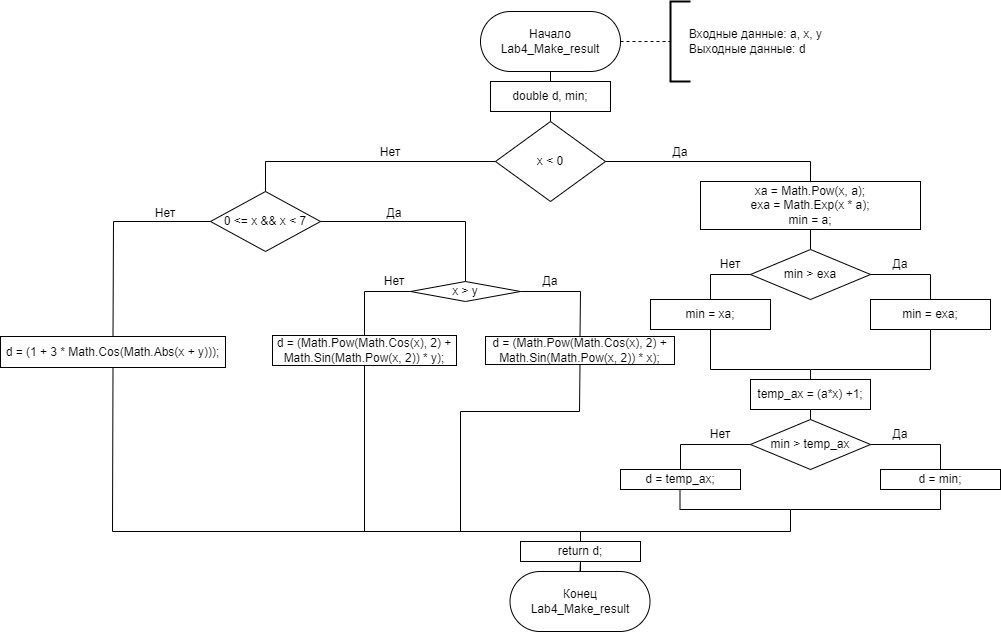


Рисунок 3 – Алгоритм вычислений d.

На рисунке 4 изображена структура алгоритма процедуры, которая осуществляет вывод вычисленных углов.

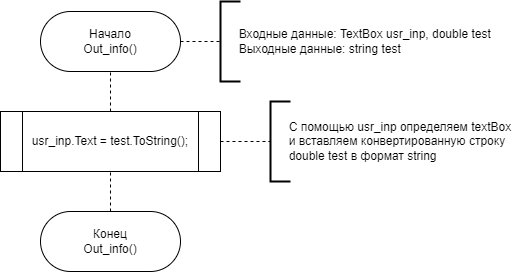


Рисунок 4 – Алгоритм вывода значений

Иерархия всех процедур программы представлена на рисунке 4.

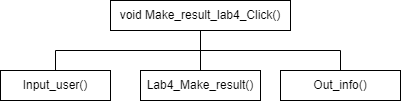


Рисунок 4 – Иерархия программных вызовов

**1.2 Листинг программного кода.**

Программный код в DLL:

using System;

using System.Collections.Generic;

using System.Windows.Forms;

namespace Dll\_lab

{

public class Lab\_dll

{

// Создать универсальные методы для ввода и вывода

public static void Input\_lab3\_KeyPress(KeyPressEventArgs e, string inputText)

{

// Разрешить обработку команды Backspace

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

{

return;

}

// Разрешить ввод цифр

if (char.IsDigit(e.KeyChar))

{

return;

}

// Разрешить ввод одного знака минус

if (e.KeyChar == '-' && !inputText.Contains("-") && !inputText.EndsWith(","))

{

return;

}

// Разрешить ввод одного знака запятой

if (e.KeyChar == ',' && !inputText.Contains(",") && !inputText.EndsWith("-"))

{

return;

}

// Запретить ввод, если символ является минусом или запятой и в поле уже присутствует соответствующий символ

if ((e.KeyChar == '-' && (inputText.Contains("-") || inputText.EndsWith(","))) ||

(e.KeyChar == ',' && (inputText.Contains(",") || inputText.EndsWith("-"))))

{

e.Handled = true;

return;

}

e.Handled = true;

}

public static double Input\_user(TextBox textbox)

{

return Convert.ToDouble(textbox.Text);

}

public static void Out\_info(TextBox usr\_inp, double test)

{

usr\_inp.Text = test.ToString();

}

public static double Lab4\_Make\_result(double a, double x, double y)

{

double d;

if (x < 0)

{

MessageBox.Show("Попали в ветку 1", "Проверка",MessageBoxButtons.OK);

double temp\_x = Math.Pow(x, a);

double temp\_e = Math.Exp(x \* a);

double var\_res;

double temp\_ax = (a \* x) + 1;

// Min

if (a < temp\_x && a < temp\_e)

var\_res = a;

else if (temp\_x < a && temp\_x < temp\_e)

var\_res = temp\_x;

else

var\_res = temp\_e;

// Max

if (var\_res > temp\_ax)

d = var\_res;

else

d = temp\_ax;

}

else if (0 <= x && x < 7)

{

MessageBox.Show("Попали в ветку 2", "Проверка",MessageBoxButtons.OK);

if (x > y)

d = (Math.Pow(Math.Cos(x), 2) + Math.Sin(Math.Pow(x, 2)) \* x);

else

d = (Math.Pow(Math.Cos(x), 2) + Math.Sin(Math.Pow(x, 2)) \* y);

}

else

{

MessageBox.Show("Попали в ветку 3", "Проверка", MessageBoxButtons.OK);

d = (1 + 3 \* Math.Cos(Math.Abs(x + y)));

} return d;

}

}

}

Основной код программы:

using Dll\_lab;

using System;

using System.Collections.Generic;

using System.Collections.Specialized;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Лабораторная\_работа\_\_2

{

public partial class CalcLab4 : Form

{

public CalcLab4()

{

InitializeComponent();

}

private void Make\_result\_lab4\_Click(object sender, EventArgs e)

{

double a = Lab\_dll.Input\_user(Input\_a\_lab4);

double x = Lab\_dll.Input\_user(Input\_x\_lab4);

double y = Lab\_dll.Input\_user(Input\_y\_lab4); // Берем данные

double return\_result = Lab\_dll.Lab4\_Make\_result(a, x, y);

Lab\_dll.Out\_info(lab4\_result\_d, return\_result);

}

private void Lab4\_exit\_Click(object sender, EventArgs e)

{

this.Close();

}

private void Input\_a\_lab4\_KeyPress(object sender, KeyPressEventArgs e)

{

Lab\_dll.Input\_lab3\_KeyPress(e, Input\_a\_lab4.Text);

}

private void Input\_x\_lab4\_KeyPress(object sender, KeyPressEventArgs e)

{

Lab\_dll.Input\_lab3\_KeyPress(e, Input\_x\_lab4.Text);

}

private void Input\_y\_lab4\_KeyPress(object sender, KeyPressEventArgs e)

{

Lab\_dll.Input\_lab3\_KeyPress(e, Input\_y\_lab4.Text);

}

}

}

**1.3 Тестирование программы и результат её работы.**

Задание лабораторной работы представляет Рисунок 6.

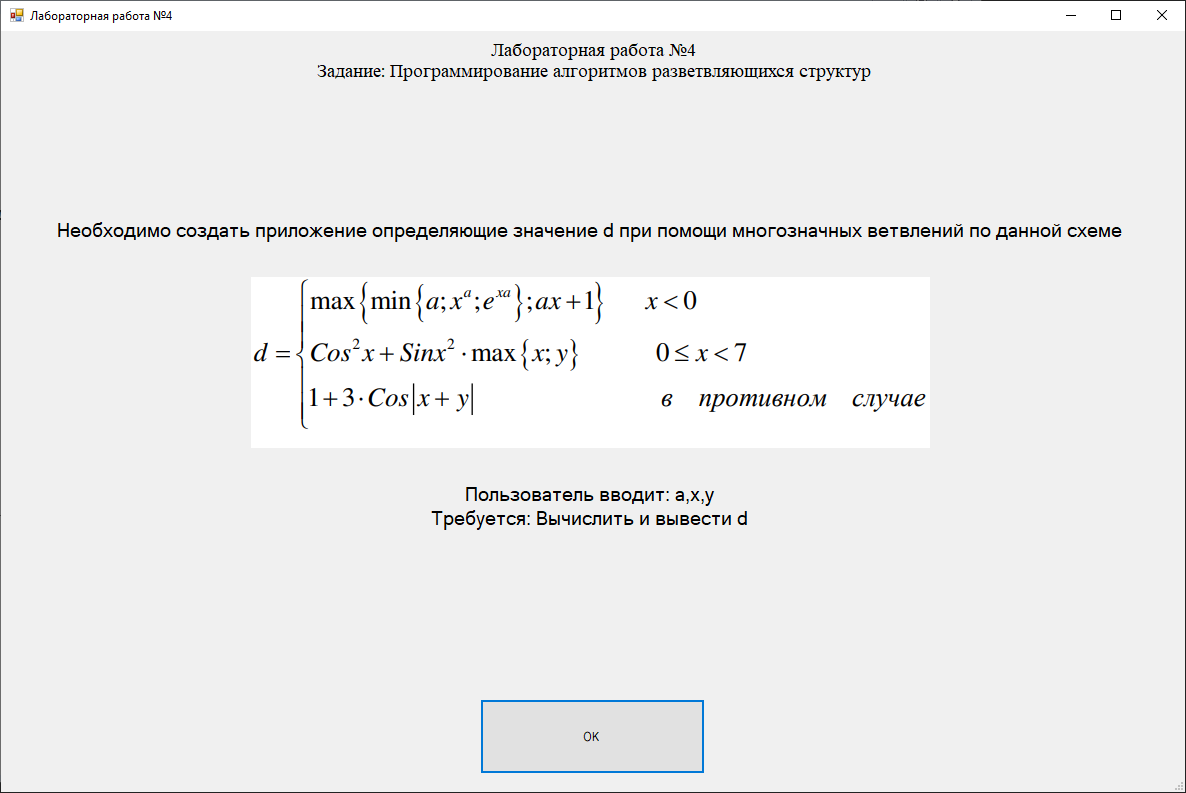


Рисунок 6 – Задание

Введем такие данные чтобы попасть в первую ветку. Обратимся к Рисунку 7.

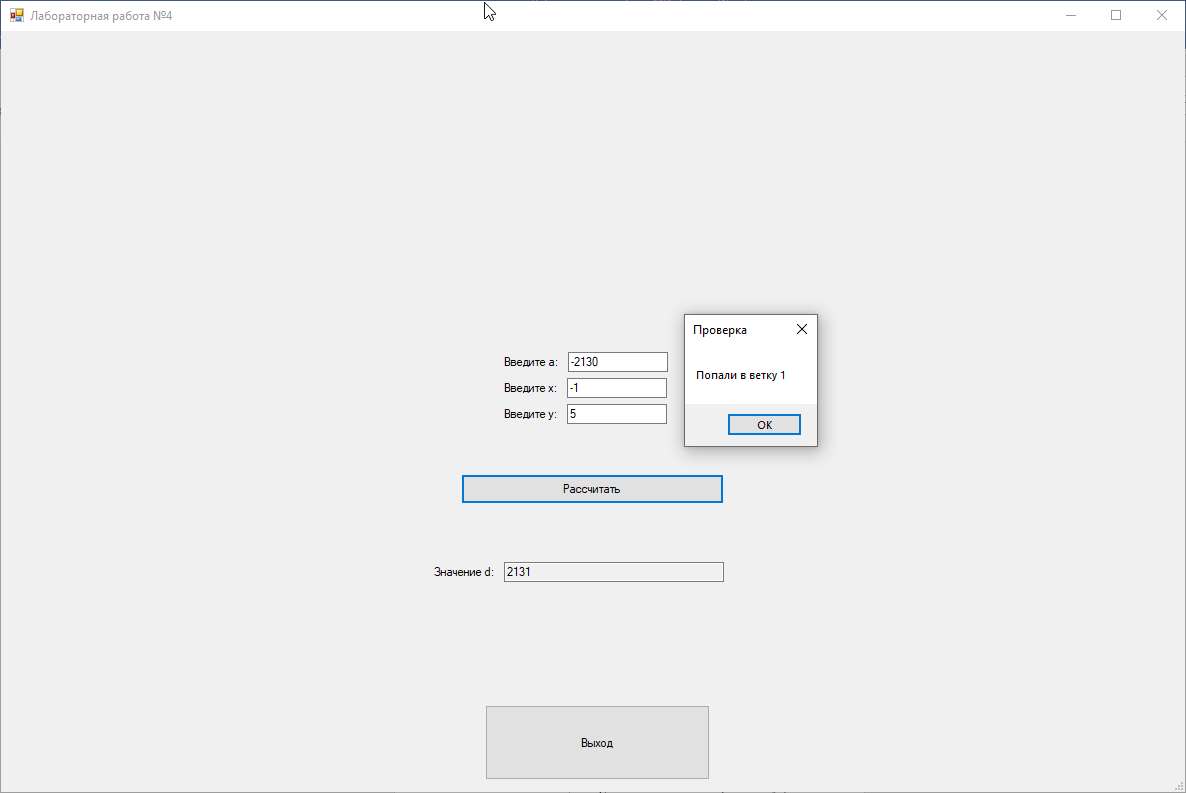


Рисунок 7 – Тестирование первой ветки.

Введем такие данные чтобы попасть во вторую ветку. Обратимся к Рисунку 8.

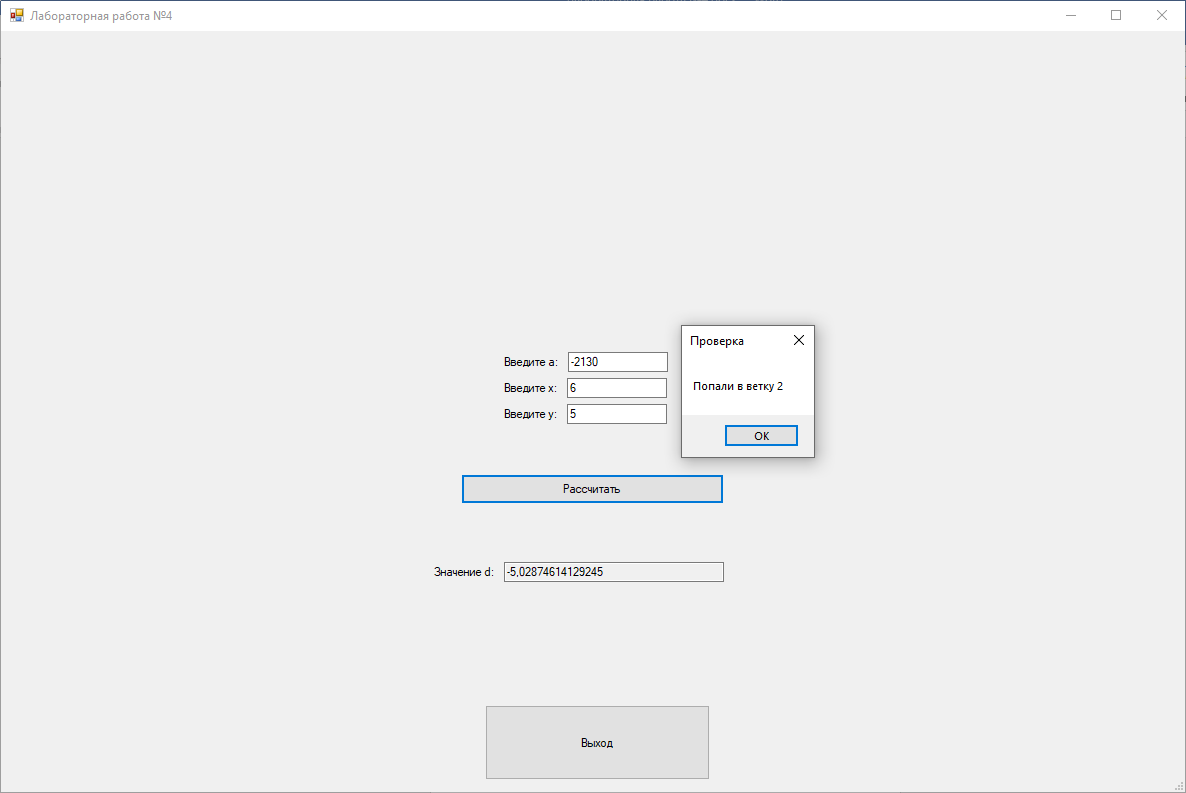


Рисунок 8 – Тестирование второй ветки.

Введем такие данные чтобы попасть в третью ветку. Обратимся к Рисунку 9.

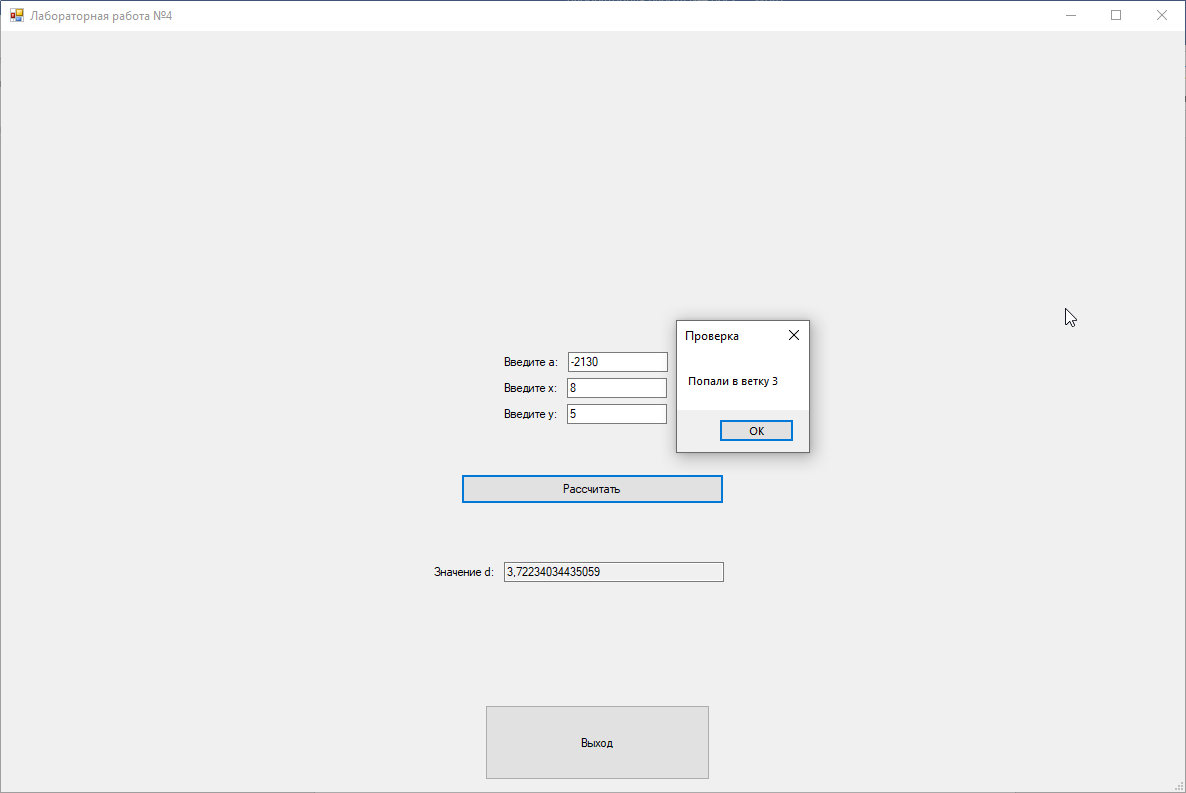


Рисунок 9 – Тестирование третьей ветки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гуриков, С. Р. Введение в программирование на языке Visual C#: учебное пособие / С. Р. Гуриков. — Москва: ФОРУМ: ИНФРА-М, 2019. — 447 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-540-0. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1012397> (дата обращения: 16.11.2023).

2. ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления" (введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст) (дата обращений 16.11.2023).

3. ГОСТ 19.701-90 (ИСО 5807-85) “Единая система программной документации СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ, ДАННЫХ И СИСТЕМ Обозначения условные и правила выполнения”. (дата обращения 16.11.2023).

Лабораторная работа №5

«Запись макросов и использование функции ЕСЛИ в Microsoft Excel»

**Задание:**

Используя функцию «ЕСЛИ» в Microsoft Excel, выполнить вычисление следующего выражения.

Для решения данной задачи требуется создать два макроса: первый для форматирования таблицы и второй для сброса всех параметров форматирования. Также необходимо добавить элементы управления, такие как кнопки, для применения этих макросов.

* 1. **Формула выражения в Microsoft Excel.**

Для удобства написания формулы в Excel каждой ячейке ввода данных было присвоено имя в соответствии с переменной, вводимой в неё. Формула, написанная в Excel, выглядит так:

=ЕСЛИ(x<0; МАКС(МИН(a; x^a; EXP(x\*a)); a\*x+1); ЕСЛИ(И(0<=x; x<7); COS(x)^2 + SIN(x^2)\*МАКС(x; y); 1+3\*(COS(ABS(x+y)))))

* 1. **Результат работы.**

На Рисунке 1 показана таблица без форматирования, которую мы имеем изначально.

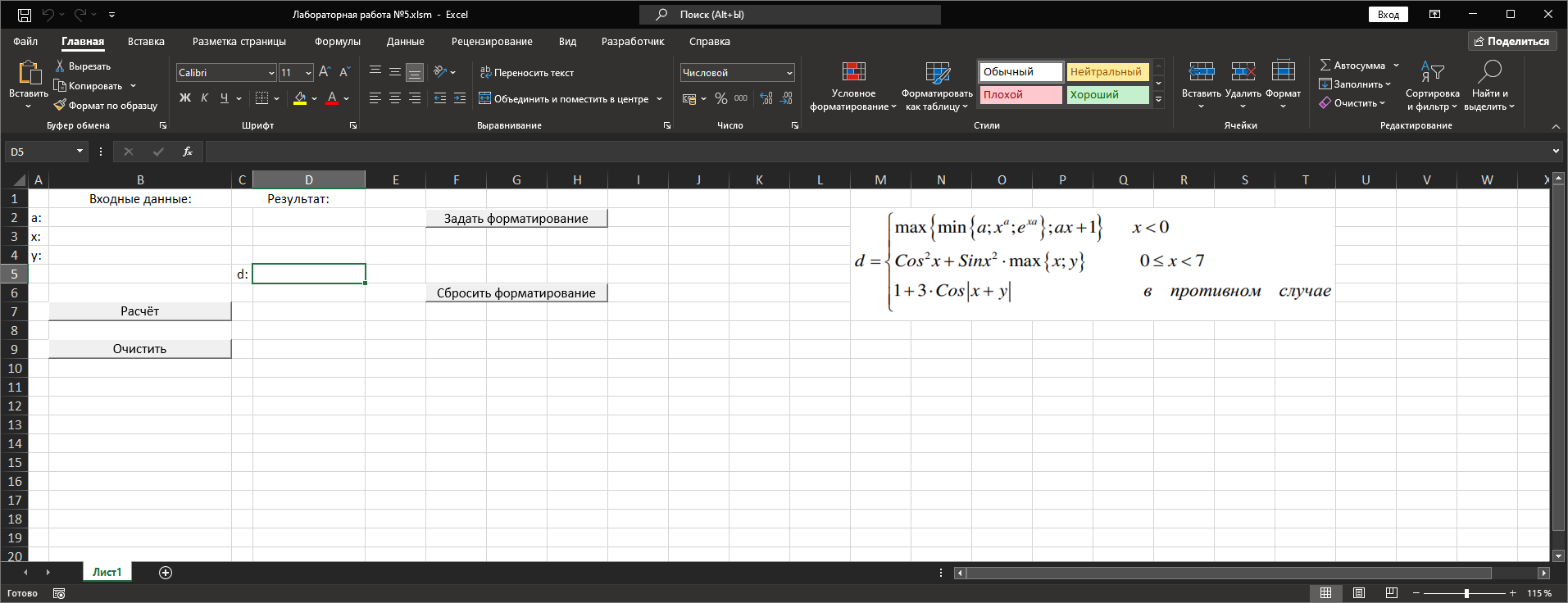


Рисунок 1 – Первоначальная таблица.

На Рисунке 2 показана таблица после применения форматирования с помощью нажатия на кнопку «Задать форматирование».

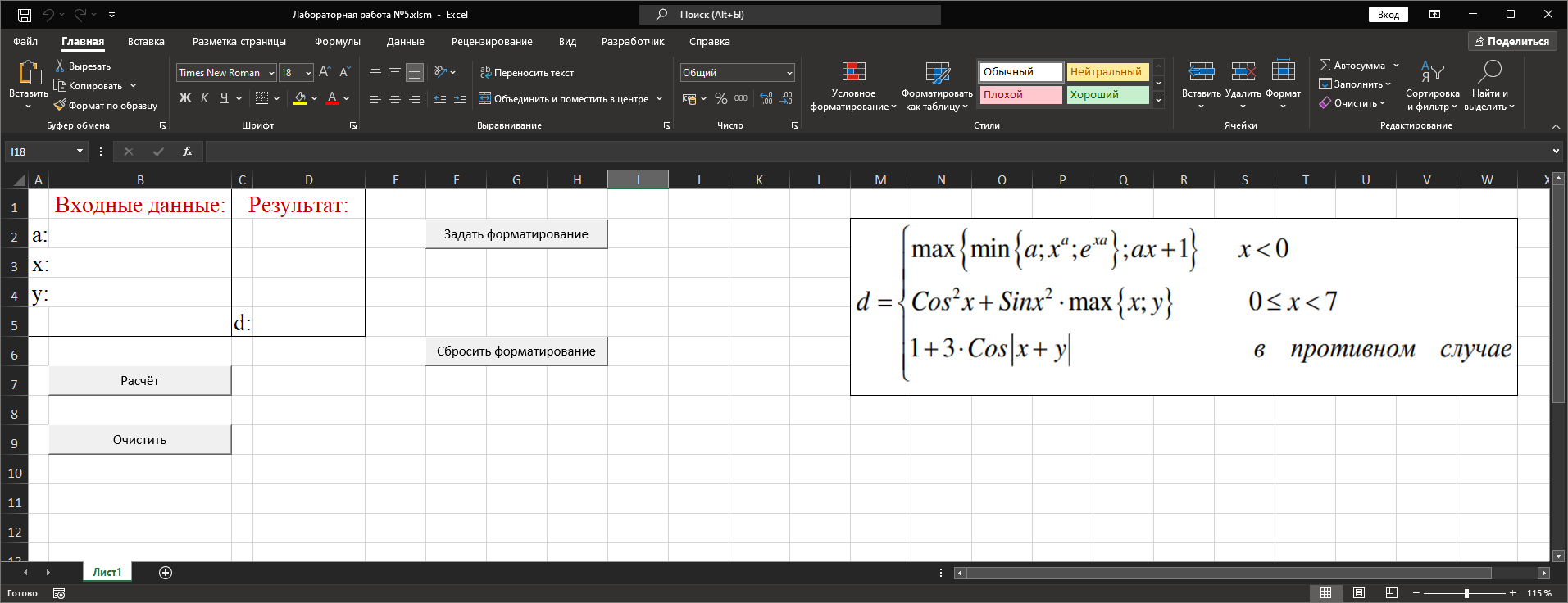


Рисунок 2 – Результат работы кнопки.

В таблице на Рисунке 3 производится вычисление выражения с использованием введенных данных.

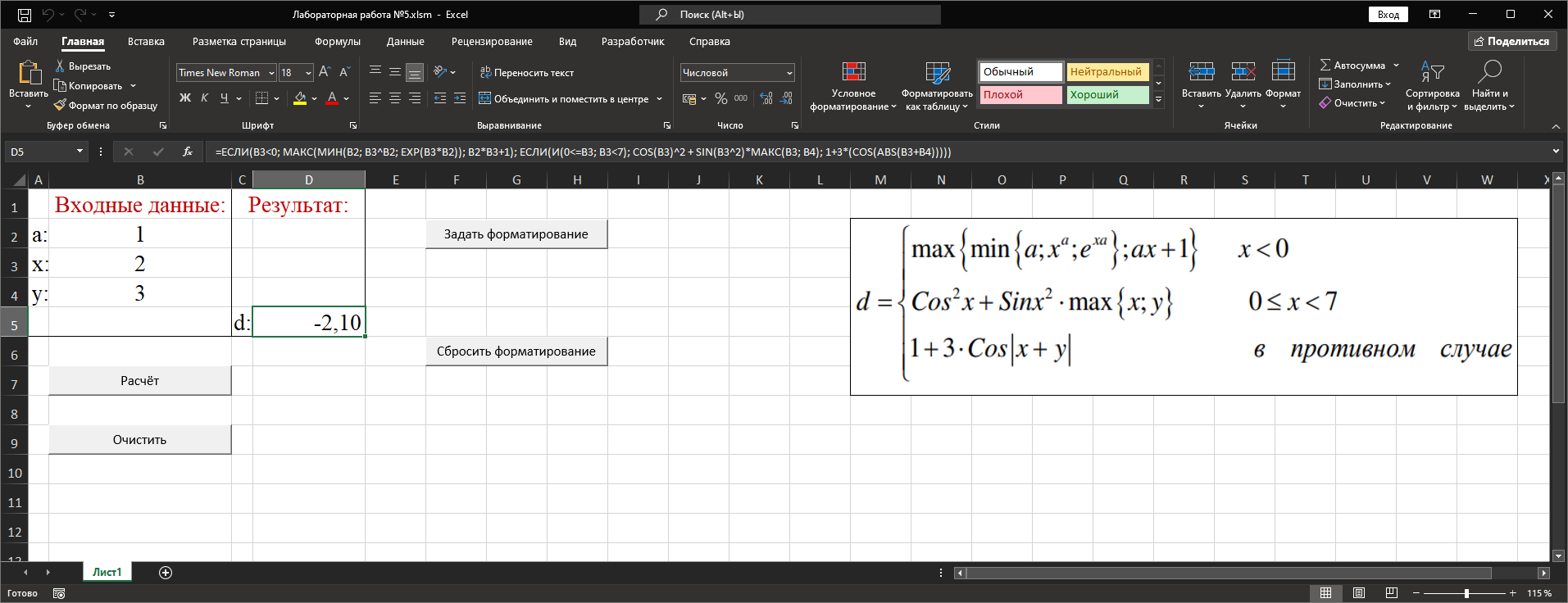


Рисунок 3 – Расчет по кнопке.

**1.3 Сравнение результатов работы.**

Проверим совпадение результатов в Microsoft Excel и в программе Visual Studio. Рассмотрим первую ветку функции, обратимся к Рисунку 4.

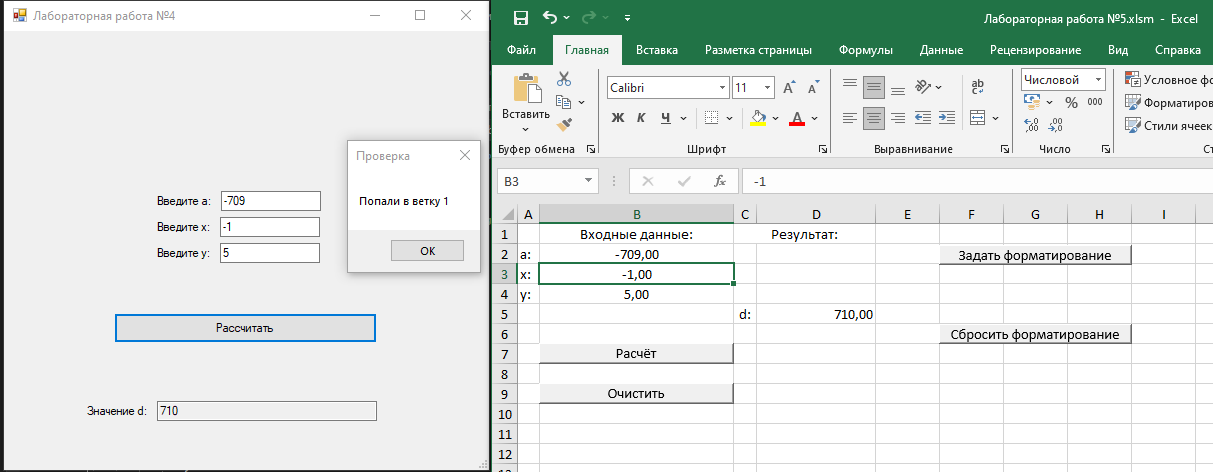


Рисунок 4 – Расчет 1 ветки, сравнение результатов в таблице и в программе

Рассмотрим вторую ветку функции, обратимся к Рисунку 5.

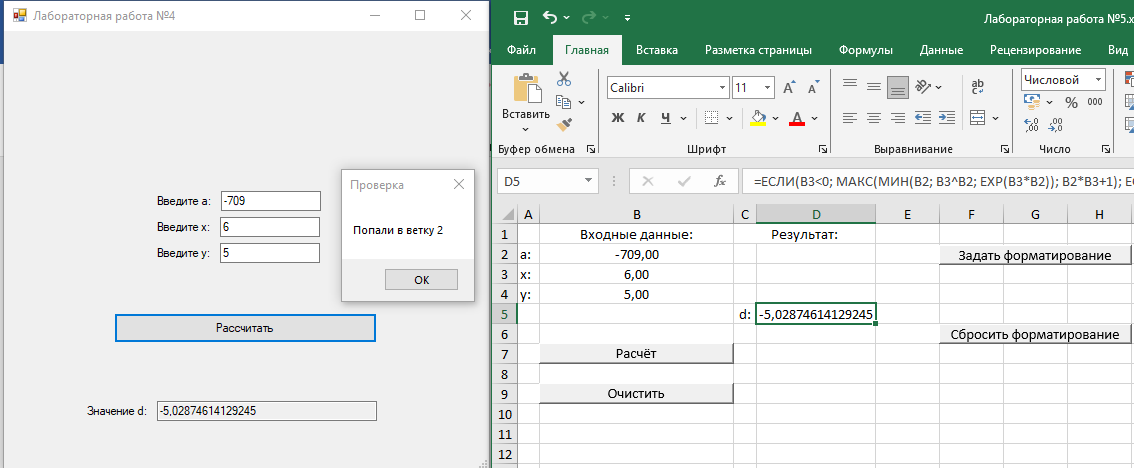


Рисунок 5 - Расчет 2 ветки, сравнение результатов в таблице и в программе

Рассмотрим третью ветку функции, обратимся к Рисунку 6.

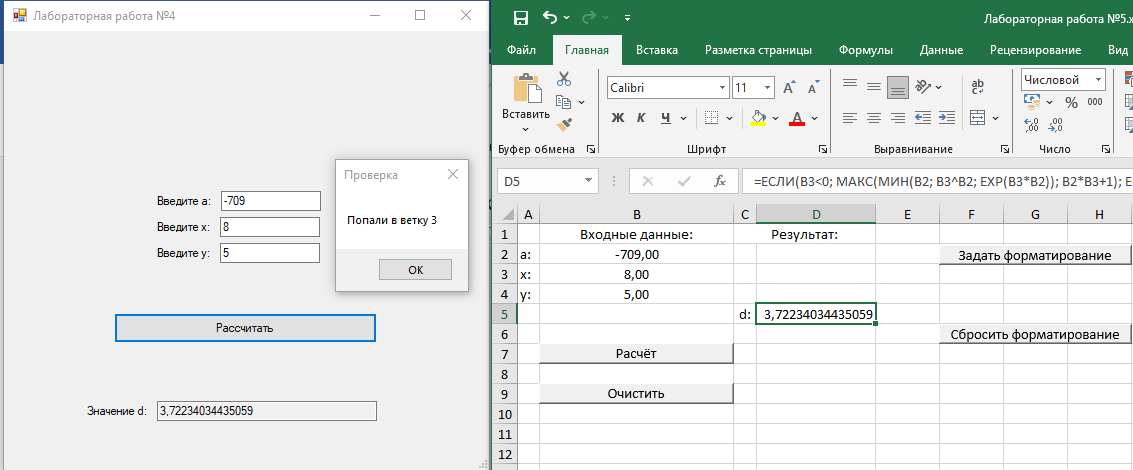


Рисунок 6 - Расчет 3 ветки, сравнение результатов в таблице и в программе

**Вывод:**

В ходе выполнения Лабораторной работы №5 "Запись макросов и использование функции ЕСЛИ в Microsoft Excel" были использованы кнопки для задания форматирования всей таблицы, возврата к исходному состоянию и расчета. Это позволило облегчить процесс форматирования таблицы, а также автоматизировать расчеты на основе введенных данных. При помощи записи макросов и использования функции ЕСЛИ были созданы макросы для выполнения этих операций. Кнопки были добавлены для удобного доступа к этим макросам. Такой подход упрощает работу с таблицами в Excel и повышает эффективность выполнения задач.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гуриков, С. Р. Введение в программирование на языке Visual Basic for Applications (VBA) : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2020. — 317 с. — (Высшее образование: Бакалавриат). — DOI 10.12737/949045. - ISBN 978-5-16-013667-7. - Текст : электронный. - URL: https://znanium.com/catalog/product/949045 (дата обращения: 18.11.2023). – Режим доступа: по подписке.

2. Гуриков С. Р. Введение в программирование на языке Visual Basic for Applications (VBA) : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2024. — 317 с. — (Среднее профессиональное образование). - ISBN 978-5-16-015995-9. - Текст : электронный. - URL: https://znanium.com/catalog/product/1912983 (дата обращения: 18.11.2023). – Режим доступа: по подписке.

3. ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления" (введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст) (дата обращений 18.11.2023).

Лабораторная работа №6

«Табулирование функции»

**Задание**

Создать приложение для построения таблицы значений функции.

d

Программа так же должна вычислять минимальное значение и среднее арифметическое отрицательных значений.

Для выполнения поставленного задания необходимо разработать следующие методы:

public static double Input\_user(TextBox textbox) – принимает значение из поля ввода и возвращает число после его конвертации.

public static void Out\_info(TextBox usr\_inp, double test) – принимает поле вывода и новое числовое значение, затем изменяет текст в переданном поле на новое число, но ничего не возвращает.

public static double Lab4\_Make\_result(double a, double x, double y) –принимает 3 переменные и возвращает значение d в соответствии с формулой.

public static void Add\_Row(double a, double d, DataGridView Lab6\_Table\_Values) – принимает два числовых значения и объект DataGridView, добавляет их в виде новой строки в таблицу. Не возвращает никакого значения.

public static void Lab6\_tabulation(DataGridView Lab6\_Table\_Values, double a, double x, double y, double step, double end\_cycle, out double result\_min\_lab6, out double result\_average\_under\_zero\_lab6)– принимает 6 параметров: DataGridView "Lab6\_Table\_Values", значения "a", "x" и "y", шаг табуляции "step" и конечное значение цикла "end\_cycle". Метод выполняет табуляцию значений и выводит результат в указанную DataGridView.

* 1. **Разработка алгоритма решения задачи.**

На рисунке 1 представлена схема событийной процедуры,описывающаяпоследовательность событий, необходимую для решения задачи.

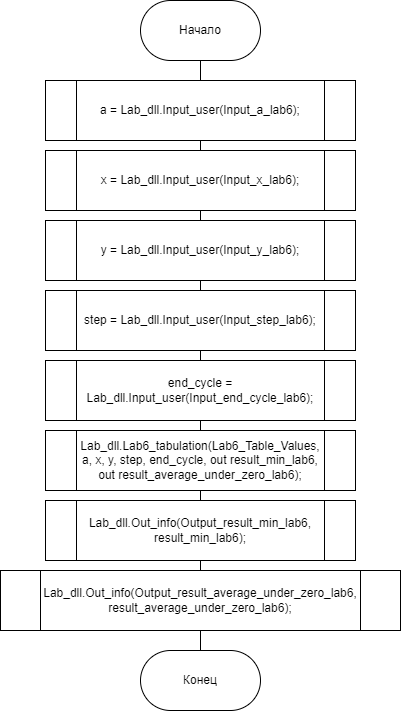
****

Рисунок 1 – Схема событийной процедуры

На рисунке 2 изображена структура алгоритма процедуры, которая выполняет обработку введенных пользователем данных.

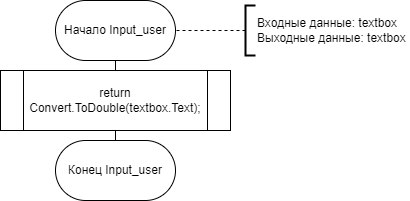


Рисунок 2 – Схема обработки ввода пользователя

На рисунке 3 представлена схема алгоритма процедуры, которая вычисляет значение переменной **d**.

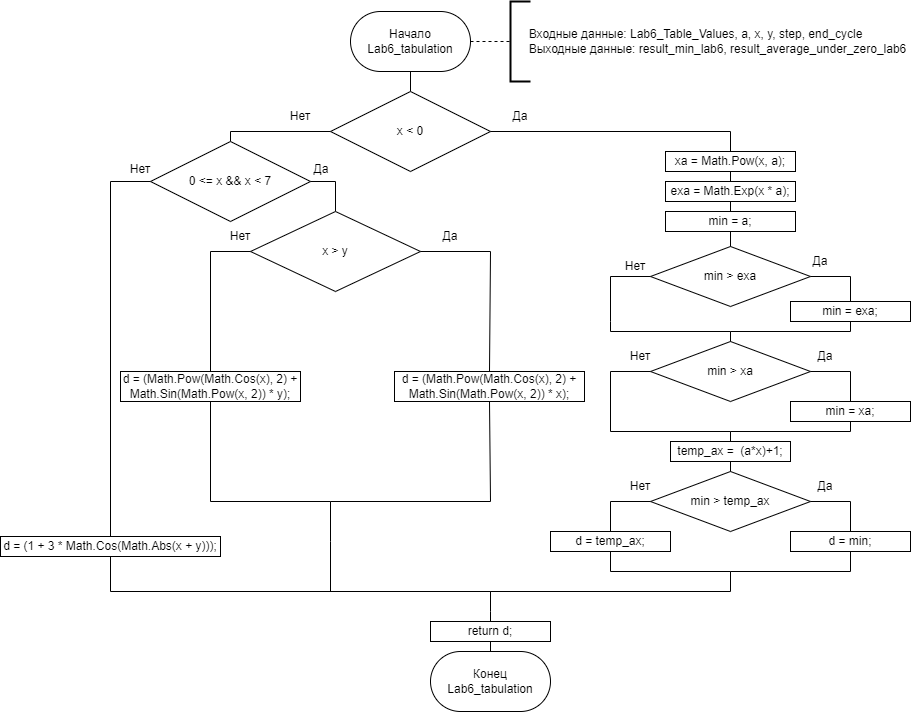


Рисунок 3 – Алгоритм вычислений d.

На рисунке 4 изображена структура алгоритма процедуры, которая осуществляет вывод вычисленных значений.

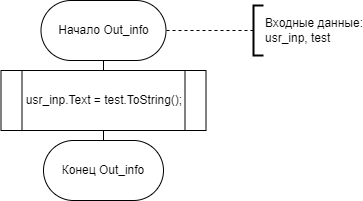


Рисунок 4 – Алгоритм вывода значений

Иерархия всех процедур программы представлена на рисунке 5.

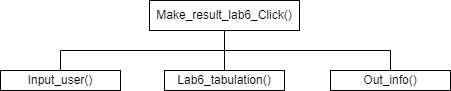


Рисунок 5 – Иерархия программных вызовов

**1.2 Листинг программного кода.**

Программный код в DLL:

using System;

using System.Collections.Generic;

using System.Security.Cryptography;

using System.Windows.Forms;

namespace Dll\_lab

{

public class Lab\_dll

{

// Создать универсальные методы для ввода и вывода

public static void Input\_lab3\_KeyPress(KeyPressEventArgs e, string inputText)

{

// Разрешить обработку команды Backspace

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

return;

// Разрешить ввод одной запятой

if (e.KeyChar == ',' && !inputText.Contains(",") && (inputText.Length != 0))// && !inputText.EndsWith("-"))

return;

// Разрешить ввод цифр

if (char.IsDigit(e.KeyChar))

return;

// Разрешить ввод одного знака минус только в начале

if (e.KeyChar == '-' && (inputText.Length == 0))

return;

// Запретить ввод, если символ является минусом или запятой и в поле уже присутствует соответствующий символ

if ((e.KeyChar == '-' && (inputText.Contains("-") || inputText.EndsWith(","))) || (e.KeyChar == ',' && (inputText.Contains(",") || inputText.EndsWith("-"))))

{

e.Handled = true;

return;

}

e.Handled = true;

}

public static double Input\_user(TextBox textbox)

{

return Convert.ToDouble(textbox.Text);

}

public static void Out\_info(TextBox usr\_inp, double test)

{

usr\_inp.Text = test.ToString();

}

public static double Lab3\_Matrix(double a1, double a2, double b1, double b2)

{

double result\_delta = (a1 \* b2) - (a2 \* b1);

return result\_delta;

}

public static void Lab3\_All\_Out(double a1, double a2, double b1, double b2, double c1, double c2, out double Lab3\_Res\_X, out double Lab3\_Res\_Y)

{

double delta = Lab3\_Matrix(a1, a2, b1, b2);

double delta\_x = Lab3\_Matrix(c1, c2, b1, b2);

double delta\_y = Lab3\_Matrix(a1, a2, c1, c2);

Lab3\_Res\_X = delta\_x / delta;

Lab3\_Res\_Y = delta\_y / delta;

}

public static double Lab4\_Make\_result(double a, double x, double y)

{

double d, min;

if (x < 0)

{

double xa = Math.Pow(x, a);

double exa = Math.Exp(x \* a);

// Min

min = a;

if (min > exa)

min = exa;

if (min > xa)

min = xa;

// Max

double temp\_ax = (a \* x) + 1;

if (min > temp\_ax)

d = min;

else

d = temp\_ax;

}

else

{

if (0 <= x && x < 7)

{

if (x > y)

d = (Math.Pow(Math.Cos(x), 2) + Math.Sin(Math.Pow(x, 2)) \* x);

else

d = (Math.Pow(Math.Cos(x), 2) + Math.Sin(Math.Pow(x, 2)) \* y);

}

else

d = (1 + 3 \* Math.Cos(Math.Abs(x + y)));

}

return d;

}

public static void Add\_Row(double a, double d, DataGridView Lab6\_Table\_Values)

{

Lab6\_Table\_Values.Rows.Add(a.ToString("F1"), d.ToString("F5"));

}

public static void Lab6\_tabulation(DataGridView Lab6\_Table\_Values, double a, double x, double y, double step, double end\_cycle, out double result\_min\_lab6, out double result\_average\_under\_zero\_lab6)

{

double min\_lab6 = 10000000;

double average\_under\_zero\_lab6 = 0;

long end\_for = Convert.ToUInt32(Math.Round(((end\_cycle - x) / step))+1);

DataGridClear(Lab6\_Table\_Values);

for (int i = 1; i <= end\_for; i++)

{

double d = Lab4\_Make\_result(a, x, y);

if (d < min\_lab6)

min\_lab6 = d;

if (d < 0)

average\_under\_zero\_lab6 += d;

Add\_Row(x, d, Lab6\_Table\_Values);

x += step;

}

result\_min\_lab6 = min\_lab6;

result\_average\_under\_zero\_lab6 = average\_under\_zero\_lab6 / end\_for;

}

}

}

Основной код программы:

using Dll\_lab;

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;

namespace Лабораторная\_работа

{

public partial class CalcLab6 : Form

{

public CalcLab6()

{

InitializeComponent();

}

private void Make\_result\_lab6\_Click(object sender, EventArgs e)

{

if (Input\_a\_lab6.Text.Length == 0)

return;

if (Input\_x\_lab6.Text.Length == 0)

return;

if (Input\_y\_lab6.Text.Length == 0)

return;

if (Input\_step\_lab6.Text.Length == 0)

return;

if (Input\_end\_cycle\_lab6.Text.Length == 0)

return;

double a = Lab\_dll.Input\_user(Input\_a\_lab6);

double x = Lab\_dll.Input\_user(Input\_x\_lab6);

double y = Lab\_dll.Input\_user(Input\_y\_lab6);

double step = Lab\_dll.Input\_user(Input\_step\_lab6);

double end\_cycle = Lab\_dll.Input\_user(Input\_end\_cycle\_lab6);

Lab\_dll.Lab6\_tabulation(Lab6\_Table\_Values, a, x, y, step, end\_cycle, out double result\_min\_lab6, out double result\_average\_under\_zero\_lab6);

Lab\_dll.Out\_info(Output\_result\_min\_lab6, result\_min\_lab6);

Lab\_dll.Out\_info(Output\_result\_average\_under\_zero\_lab6, result\_average\_under\_zero\_lab6);

}

private void Input\_a\_lab6\_KeyPress(object sender, KeyPressEventArgs e)

{

Lab\_dll.Input\_lab3\_KeyPress(e, Input\_a\_lab6.Text);

}

private void Input\_x\_lab6\_KeyPress(object sender, KeyPressEventArgs e)

{

Lab\_dll.Input\_lab3\_KeyPress(e, Input\_x\_lab6.Text);

}

private void Input\_y\_lab6\_KeyPress(object sender, KeyPressEventArgs e)

{

Lab\_dll.Input\_lab3\_KeyPress(e, Input\_y\_lab6.Text);

}

private void Input\_step\_lab6\_KeyPress(object sender, KeyPressEventArgs e)

{

Lab\_dll.Input\_lab3\_KeyPress(e, Input\_step\_lab6.Text);

}

private void Input\_end\_cycle\_lab6\_KeyPress(object sender, KeyPressEventArgs e)

{

Lab\_dll.Input\_lab3\_KeyPress(e, Input\_end\_cycle\_lab6.Text);

}

}

}

**1.3 Результаты работы программы.**

Рисунок 6 представляет данное задание:

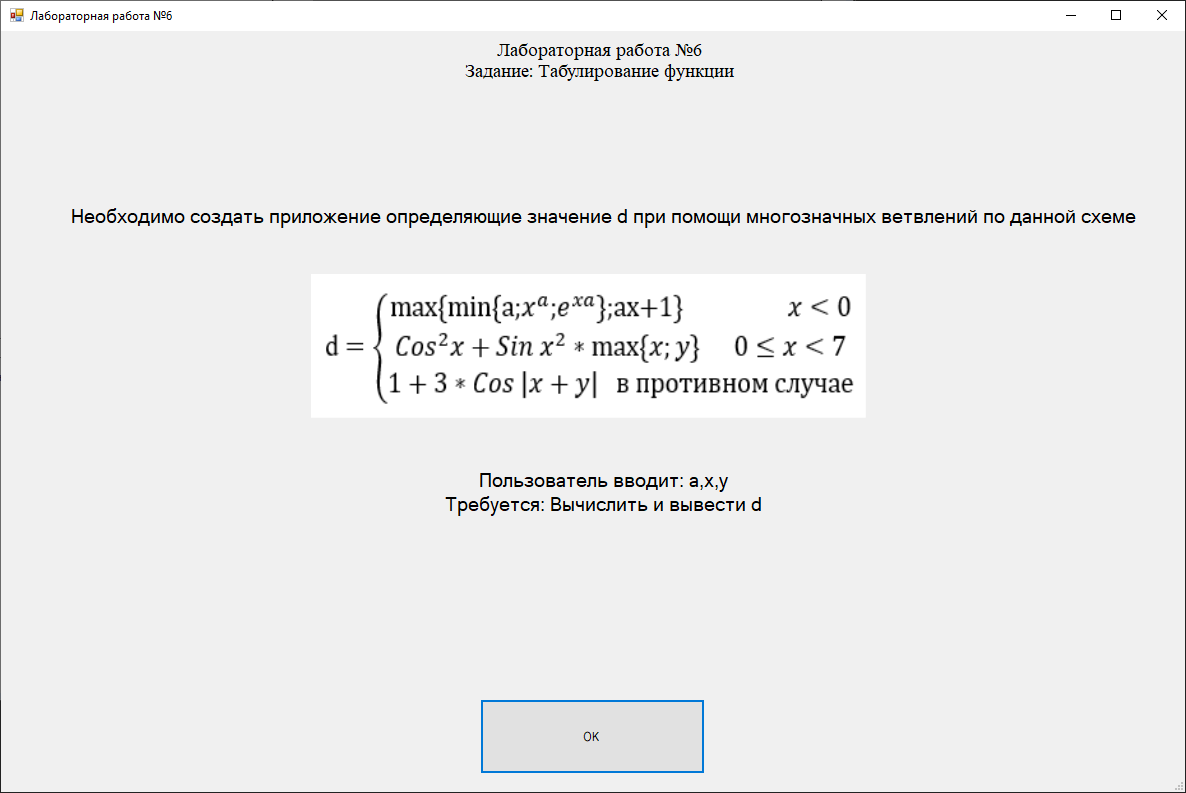


Рисунок 6 – Задание

На рисунке 7 находится демонстрация работы программы.

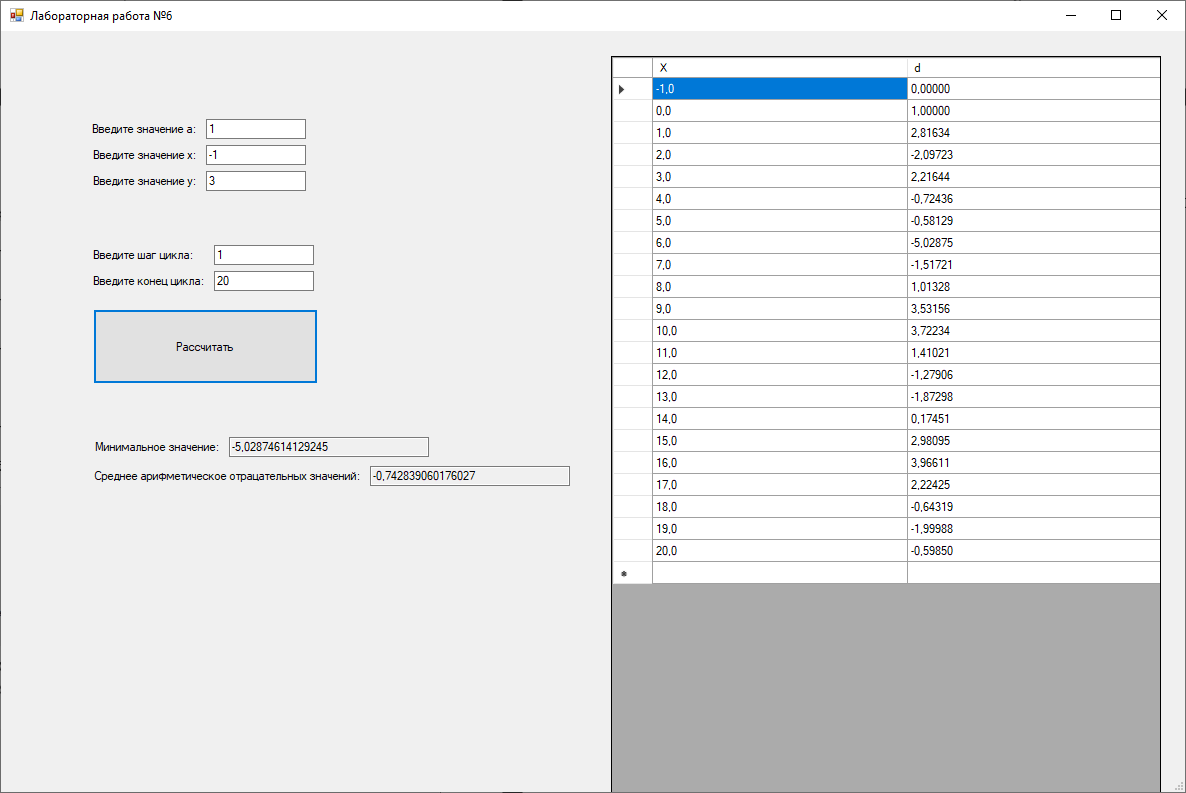


Рисунок 7 – Демонстрация работы программы

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гуриков, С. Р. Введение в программирование на языке Visual C#: учебное пособие / С. Р. Гуриков. — Москва: ФОРУМ: ИНФРА-М, 2019. — 447 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-540-0. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1012397> (дата обращения: 30.11.2023).

2. ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления" (введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст) (дата обращений 30.11.2023).

3. ГОСТ 19.701-90 (ИСО 5807-85) “Единая система программной документации СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ, ДАННЫХ И СИСТЕМ Обозначения условные и правила выполнения”. (дата обращения 30.11.2023).

Лабораторная работа №7

«Решение задачи табулирования функции в Microsoft Excel»

**Задание**

Выполнить табулирование приведенной ниже функции в Microsoft Excel.

Функция:

d

Для решения задачи необходимо записать макросы:

1.Макрос для построения графика

2.Макрос для сброса всех параметров графика

3.Макрос для форматирования таблицы

4.Макрос для сбрасывания всех параметров форматирования.

Также нужно создать элементы управления(кнопки) для применения этих макросов.

**1.1 Написание формулы выражения в Excel**

Формула, написанная в Excel, выглядит так:

=ЕСЛИ(x < 0; МАКС(МИН(a; x^a; EXP(x\*a)); a \*x+1); ЕСЛИ(И(0<=x; x<7); COS(x)^2 + SIN(x^2)\*МАКС(x; y); 1+3\*(COS(ABS(x+y)))))

**1.2 Результат работы.**

На Рисунке 1 показана таблица без форматирования, которую мы имеем изначально.

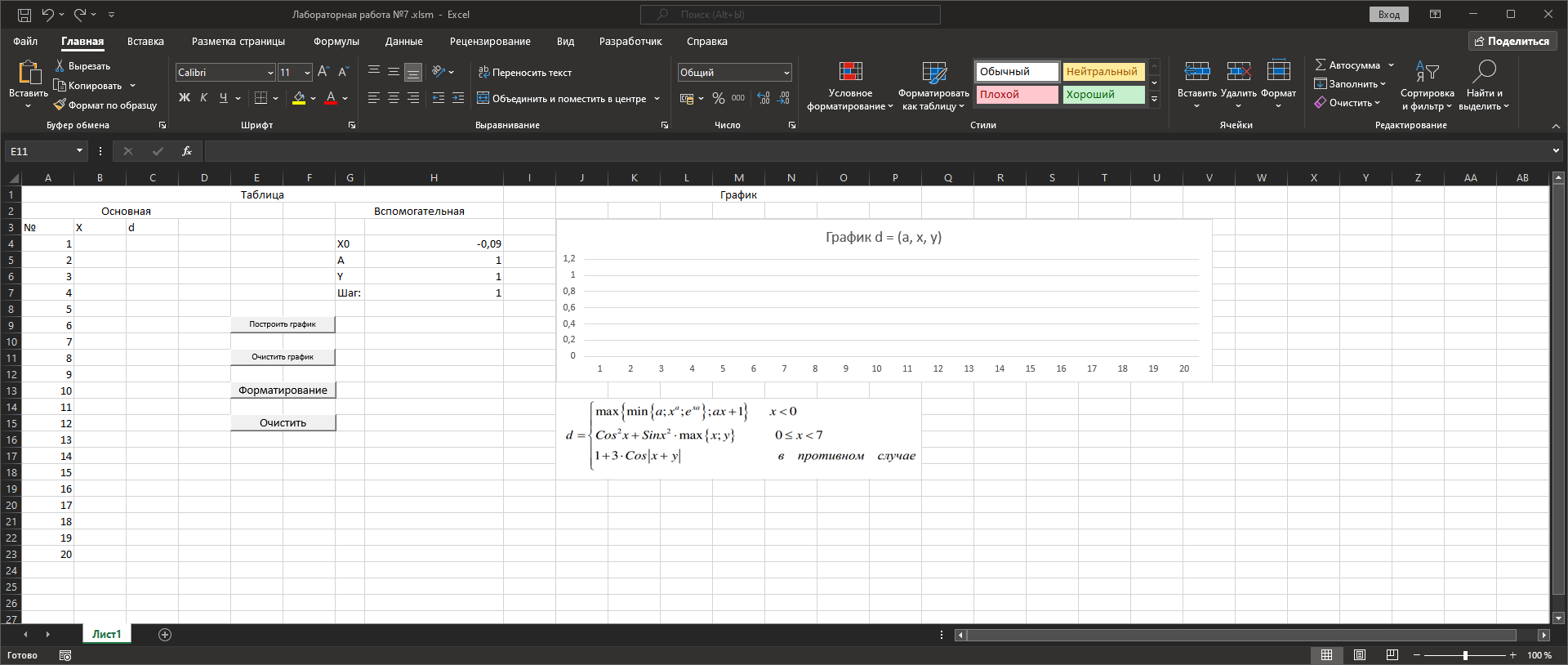


Рисунок 1 – Первоначальная таблица

На Рисунке 2 показана таблица после применения форматирования с помощью нажатия на кнопку «Форматирование».

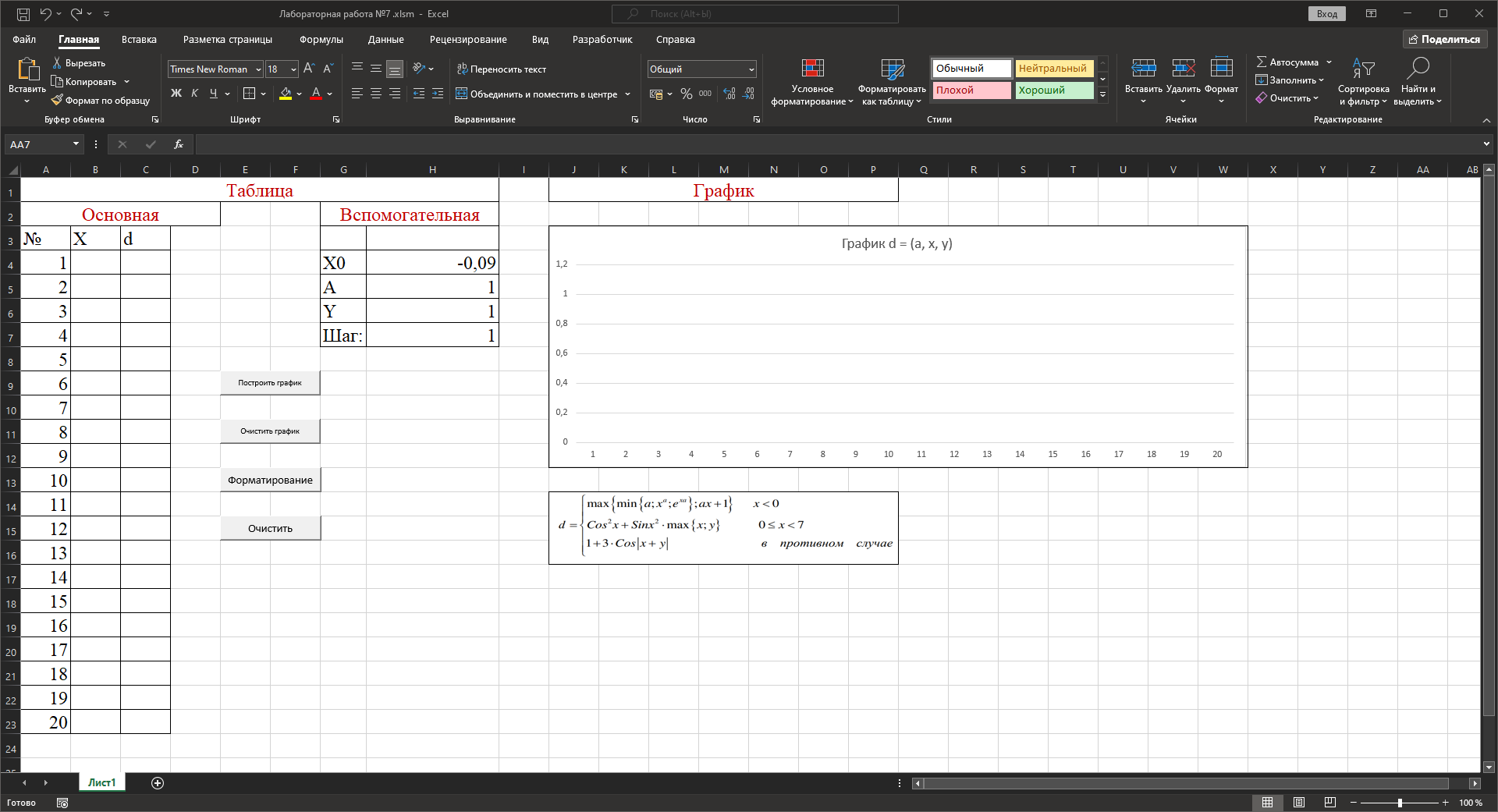


Рисунок 2 – Результат работы кнопки «Форматирование»

В таблице на Рисунке 3 производится вычисление выражения с использованием введенных данных.

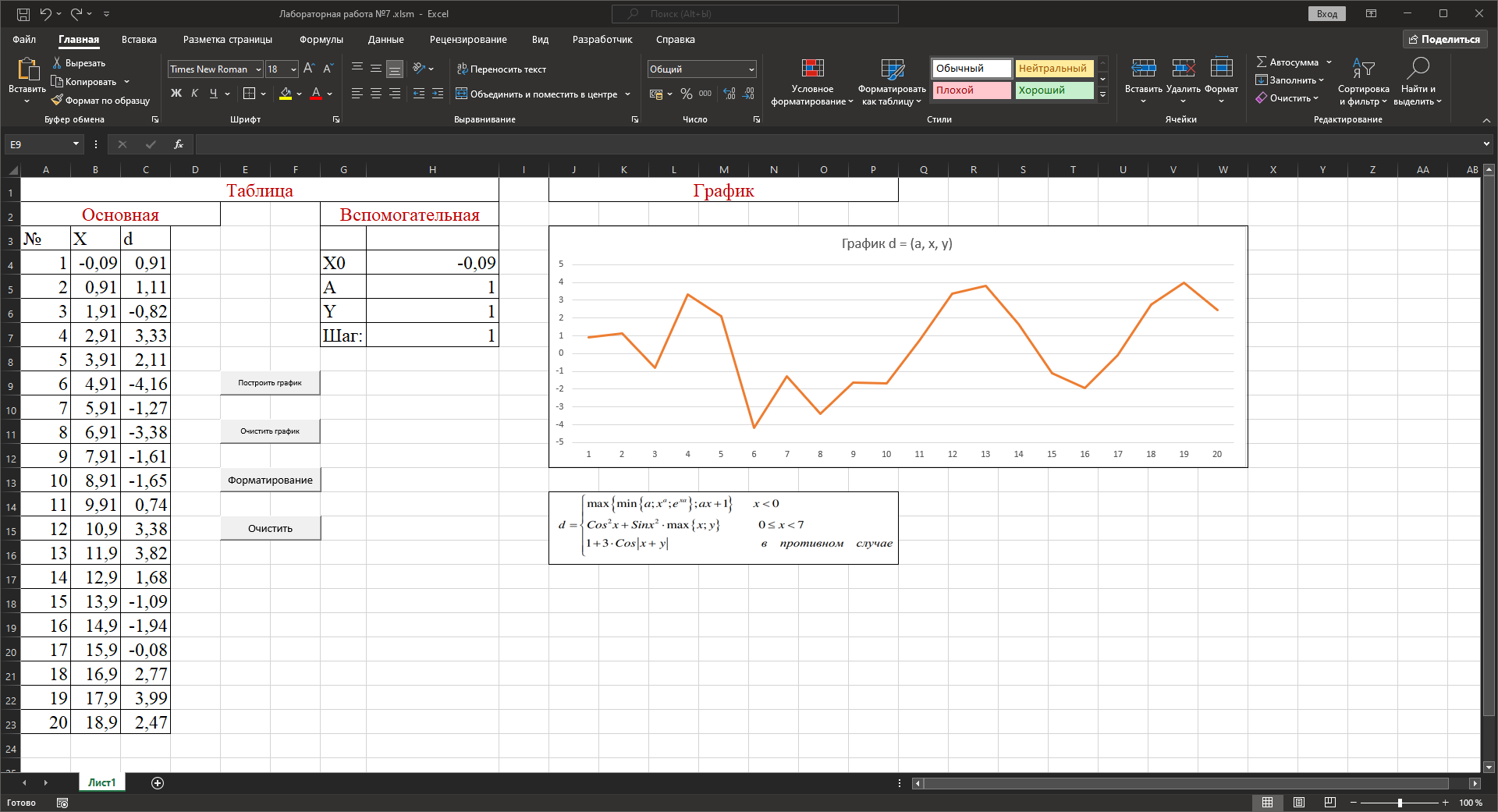


Рисунок 3 – Расчет по кнопке.

**1.4 Сравнение результатов.**

Проверим совпадение результатов в таблице Microsoft Excel и в программе Visual Studio. Обратимся к Рисунку 4.

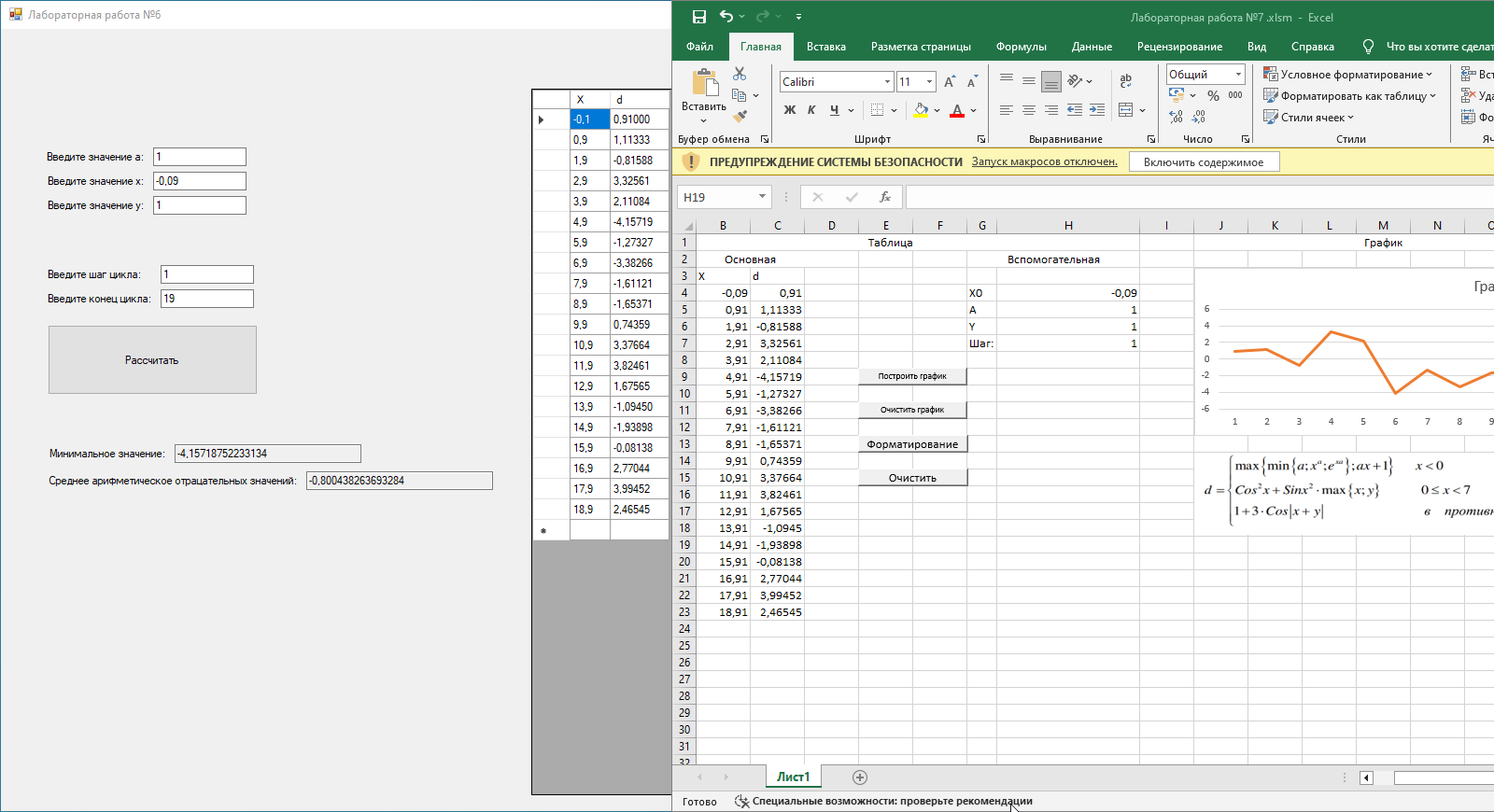


Рисунок 4 – Сравнение результатов.

**Вывод:**

Нажатие кнопки "Построение графика" приводит к правильному табулированию функции. График также отображается корректно, с изменяемыми значениями на оси X. Макросы также функционируют правильно при нажатии на кнопки "Очистить график", "Форматирование" и "Очистить формат".

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гуриков, С. Р. Введение в программирование на языке Visual Basic for Applications (VBA) : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2020. — 317 с. — (Высшее образование: Бакалавриат). — DOI 10.12737/949045. - ISBN 978-5-16-013667-7. - Текст : электронный. - URL: https://znanium.com/catalog/product/949045 (дата обращения: 30.11.2023). – Режим доступа: по подписке.

2. Гуриков С. Р. Введение в программирование на языке Visual Basic for Applications (VBA) : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2024. — 317 с. — (Среднее профессиональное образование). - ISBN 978-5-16-015995-9. - Текст : электронный. - URL: https://znanium.com/catalog/product/1912983 (дата обращения: 30.11.2023). – Режим доступа: по подписке.

3. ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления" (введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст) (дата обращений 30.11.2023).

Лабораторная работа №8

«Итеративный циклический процесс»

**Задание:**

Дано действительное число число x. Получив рекуррентные выражения для вычисления элементов ряда и их сыммы, вычислить с точностью 10 -6

Для выполнения данного задания необходимо разработать следующие методы:

public static long Lab8\_Factorial(long x) – метод вычисления факториала.

public static double Lab8\_MakeResult(double x, int k) – метод, который принимает два аргумента: предыдущее приближение (x) и номер итерации (k). В результате возвращается новое значение приближения.

public static void Lab8\_Result (double x0, double eps, double MaxIter, DataGridView DataGrid) – метод который принимает, x0, eps, MaxIter введенное пользователем, и так же передает DataGridView для дальнейшей работы с таблицей.

Все методы необходимо разместить в DLL-библиотеке.

* 1. Формализация задачи

x0 – начальное значение x;

x – текущее значение;

prev\_x – предыдущее значение x;

k – номер итерации;

Будем считать, что заданная точность обеспечена, если модуль разности между текущим и предыдущим значением меньше допустимой погрешности, то есть .

Для решения поставленной задачи необходимо реализовать метод Lab8\_Result, который в качестве входных параметров получает начальное значение x0, точность , максимальное допустимое число итераций MaxIter и таблицу DataGridView для дальнейшего редактирования таблицы. Результатами работы метода является значение вычисленное по формуле. Кроме того, метод Lab8\_Result должен на каждой итерации выводить в DataGridView номер итерации.

* 1. **Разработка алгоритма решения задачи.**

Схема алгоритма событийной процедуры решения задачи изображена на рисунке 1.

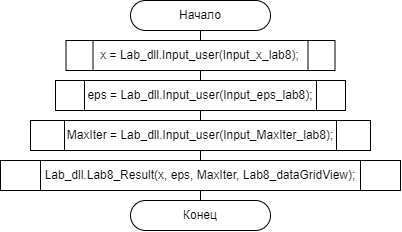


Рисунок 1 – Схема событийной процедуры.

Схема алгоритма вычисления значения по формуле.

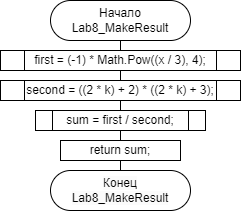


Рисунок 2 – Схема вычисления значения.

Схема алгоритма, выполняющего итерацию функции продемонстрирована на рисунке 3.

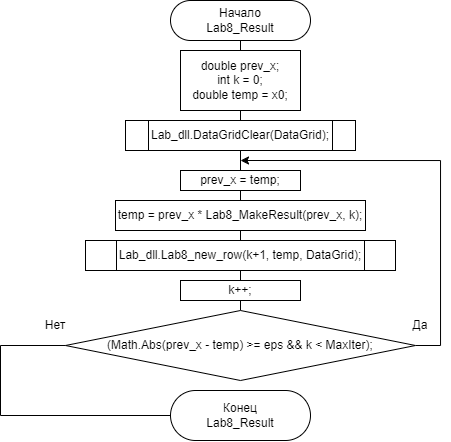


Рисунок 3 - Схема алгоритма, выполняющего итерацию функции

* 1. **Листинг программного кода.**

Основной код программы:

using Dll\_lab;

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;

namespace Лабораторная\_работа

{

public partial class CalcLab8 : Form

{

public CalcLab8()

{

InitializeComponent();

}

private void Lab8\_exit\_Click(object sender, EventArgs e)

{

this.Close();

}

private void Make\_result\_lab8\_Click(object sender, EventArgs e)

{

double x = Lab\_dll.Input\_user(Input\_x\_lab8);

double eps = Lab\_dll.Input\_user(Input\_eps\_lab8);

double MaxIter = Lab\_dll.Input\_user(Input\_MaxIter\_lab8);

Lab\_dll.Lab8\_Result(x, eps, MaxIter, Lab8\_dataGridView);

}

private void Input\_x\_lab8\_KeyPress(object sender, KeyPressEventArgs e)

{

Lab\_dll.Input\_lab3\_KeyPress(e, Input\_x\_lab8.Text);

}

private void Input\_eps\_lab8\_KeyPress(object sender, KeyPressEventArgs e)

{

Lab\_dll.Input\_lab3\_KeyPress(e, Input\_eps\_lab8.Text);

}

private void Input\_MaxIter\_lab8\_KeyPress(object sender, KeyPressEventArgs e)

{

Lab\_dll.Input\_lab3\_KeyPress(e, Input\_MaxIter\_lab8.Text);

}

}

}

Программные методы, которые находятся в DLL-библиотеке DLL:

public static void Lab8\_new\_row(int i, double x, DataGridView DGV)

{

DGV.Rows.Add(i.ToString("F1"), x.ToString("F9")); // ?

}

public static double Lab8\_MakeResult(double x, int k)

{

double first = (-1) \* Math.Pow((x / 3), 4);

double second = ((2 \* k) + 2) \* ((2 \* k) + 3);

double sum = first / second;

return sum;

}

public static void Lab8\_Result(double x0, double eps, double MaxIter, DataGridView DataGrid)//, out double temp)

{

double prev\_x;

int k = 0;

double temp = x0;

Lab\_dll.DataGridClear(DataGrid);

do

{

prev\_x = temp; // Сохраняем прошлое значение

temp = prev\_x \* Lab8\_MakeResult(prev\_x, k); // Рассчитываем

Lab\_dll.Lab8\_new\_row(k+1, temp, DataGrid);

k++;

}

while (Math.Abs(prev\_x - temp) >= eps && k < MaxIter);

}

}

} }

* 1. Результат работы программы.

Задание лабораторной работы представлено на рисунке 4.

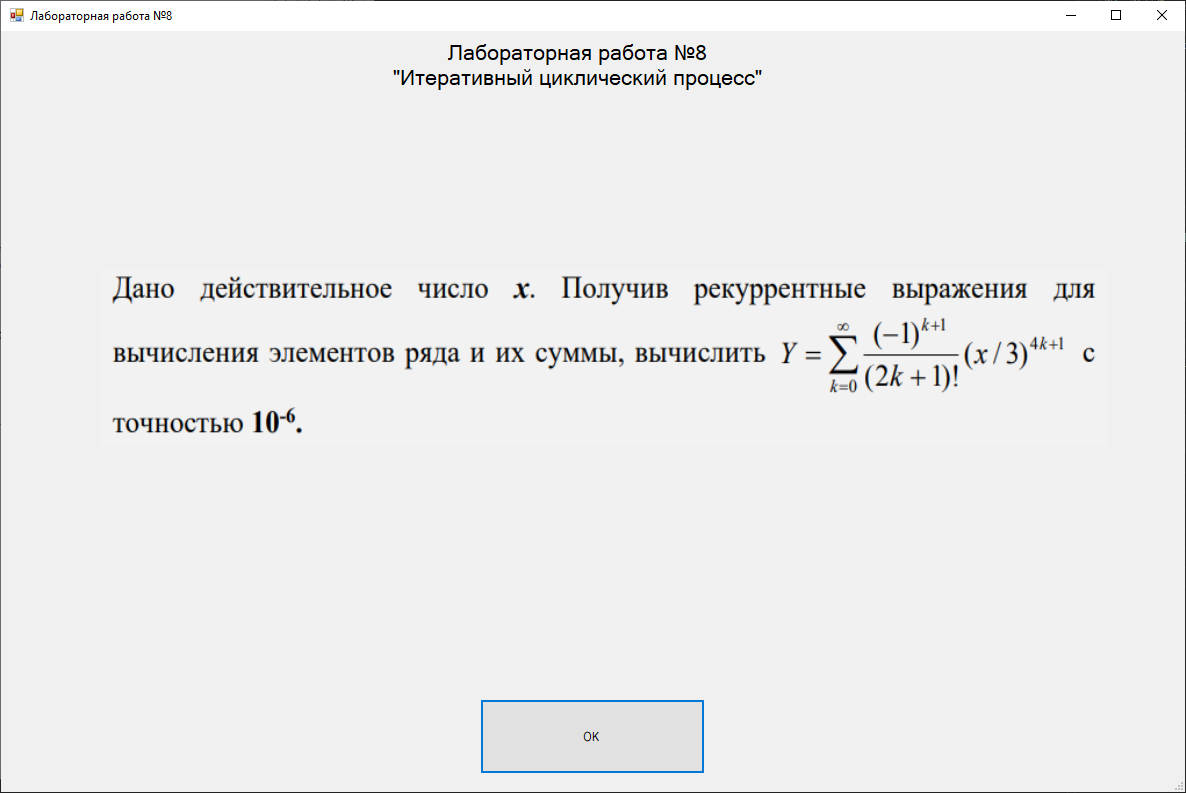


Рисунок 4 – Задание лабораторной работы

Результат работы программы представлен на рисунке 5.

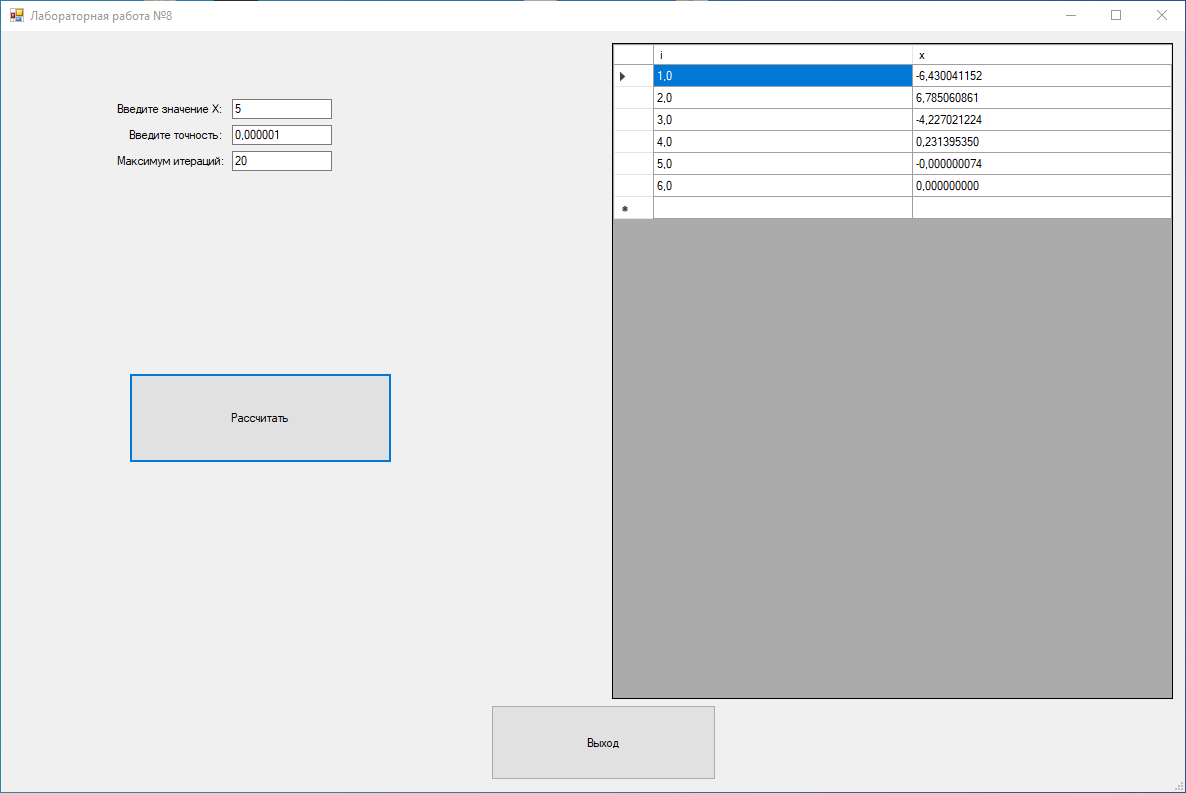


Рисунок 5 – Результат работы

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гуриков, С. Р. Введение в программирование на языке Visual C#: учебное пособие / С. Р. Гуриков. — Москва: ФОРУМ: ИНФРА-М, 2019. — 447 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-540-0. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1012397> (дата обращения: 08.12.2023).

2. ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления" (введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст) (дата обращений 08.12.2023).

3. ГОСТ 19.701-90 (ИСО 5807-85) “Единая система программной документации СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ, ДАННЫХ И СИСТЕМ Обозначения условные и правила выполнения”. (дата обращения 08.12.2023).