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.

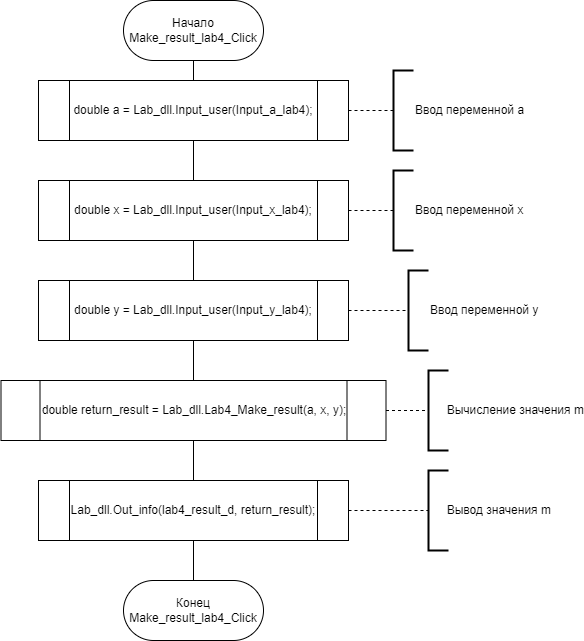


Рисунок 1 – Схема алгоритма событийной процедуры

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

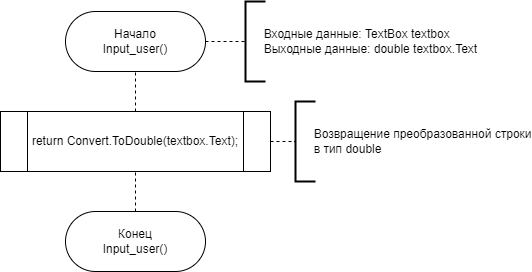


Рисунок 2 – Схема обработки ввода пользователя

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

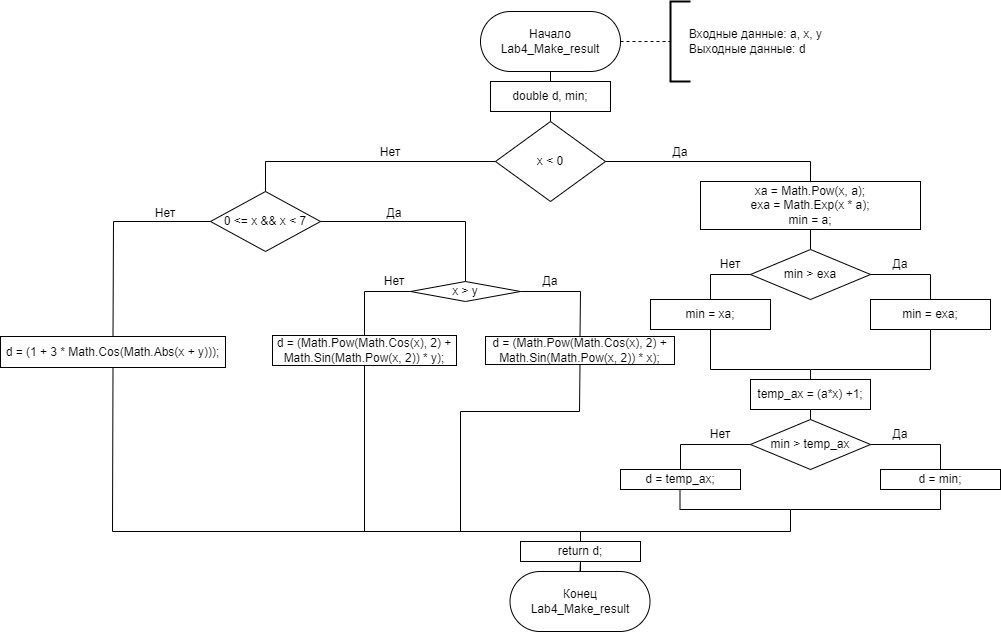


Рисунок 3 – Алгоритм вычислений d.

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

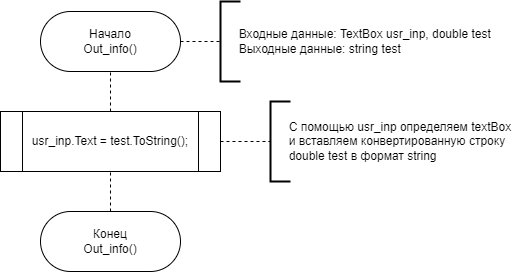


Рисунок 4 – Алгоритм вывода значений

Иерархия всех процедур программы представлена на рисунке 4.

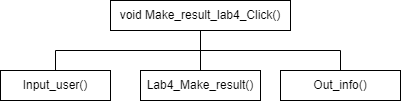


Рисунок 4 – Иерархия программных вызовов

**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.

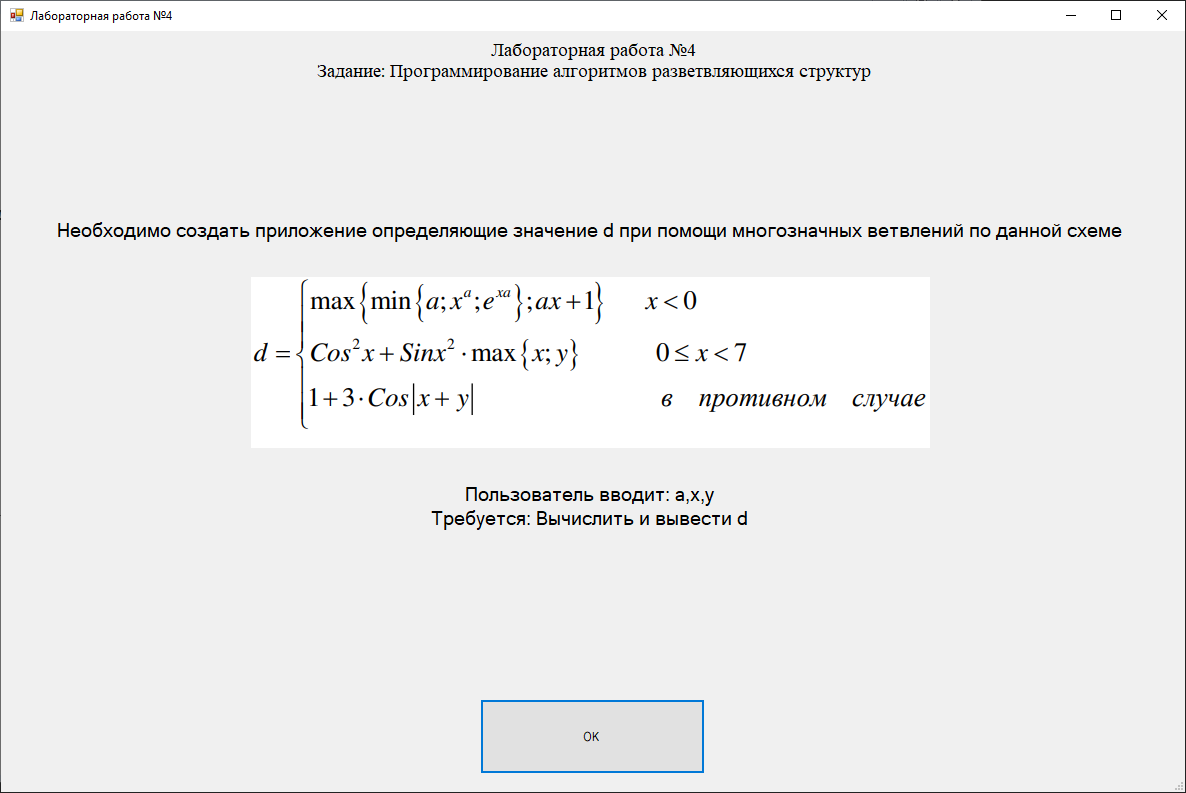


Рисунок 6 – Задание

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

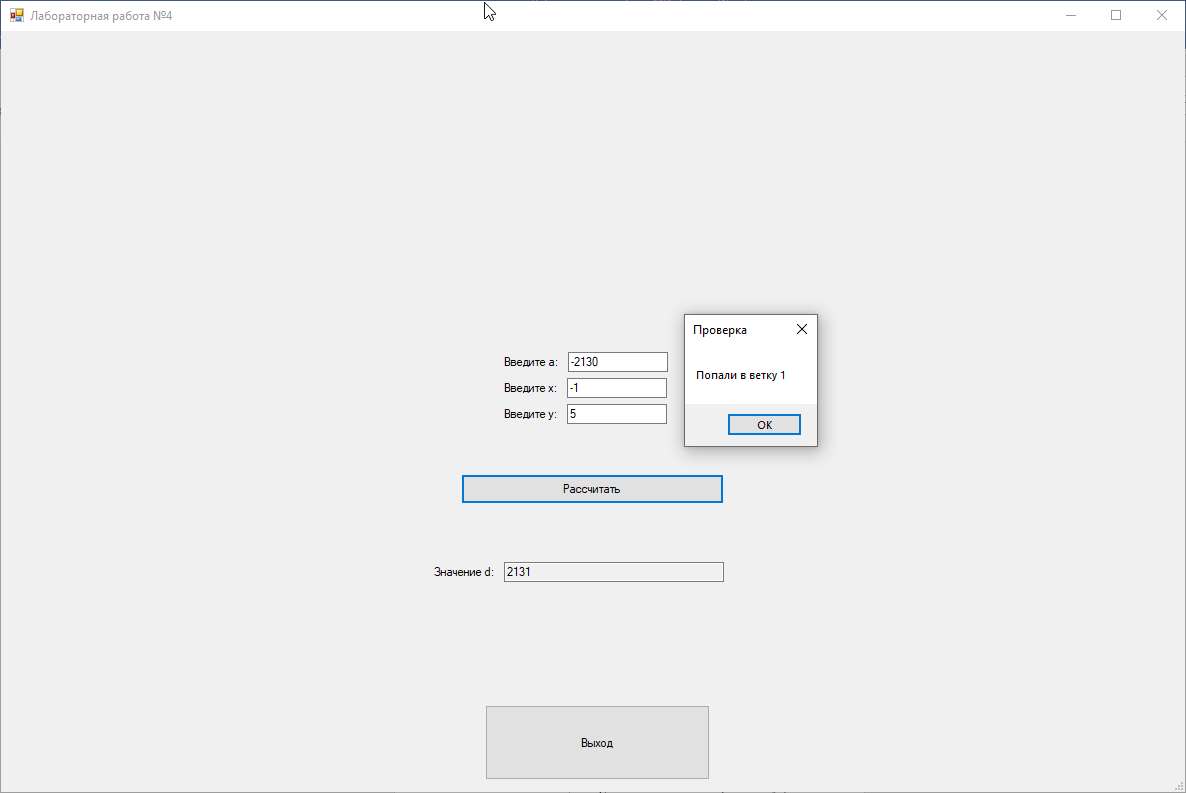


Рисунок 7 – Тестирование первой ветки.

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

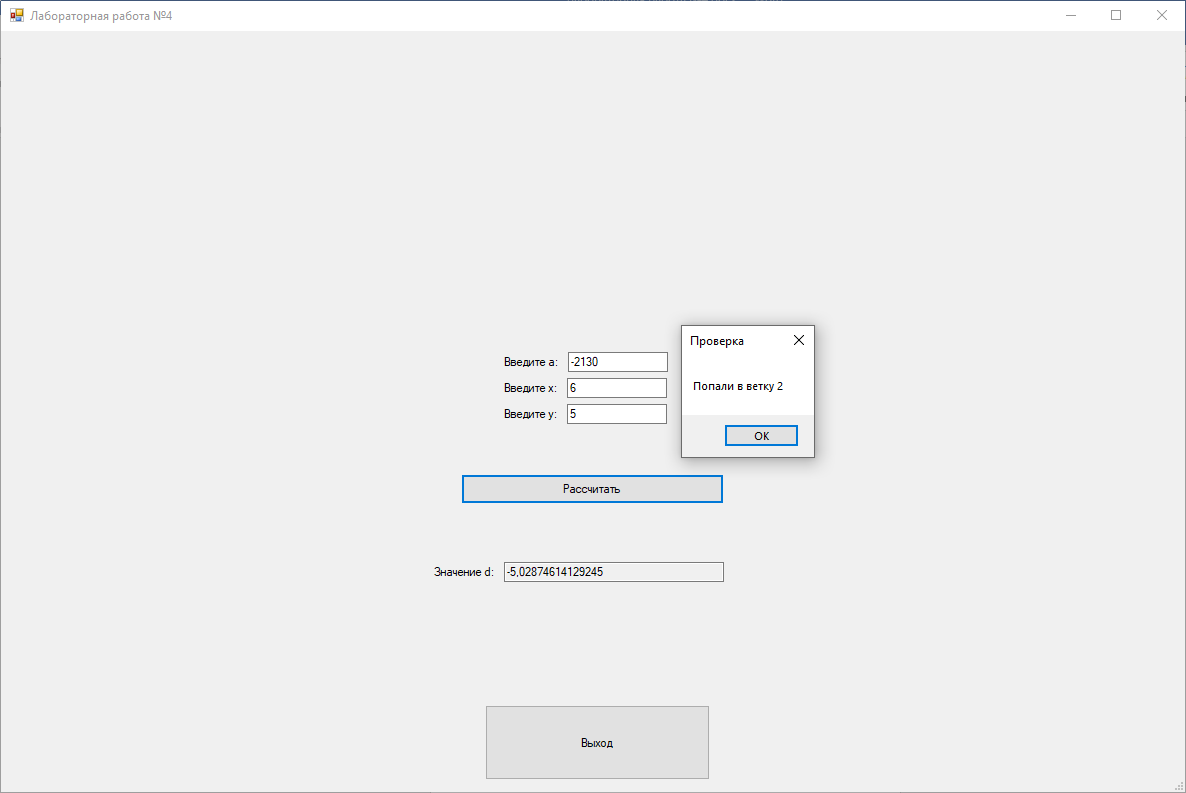


Рисунок 8 – Тестирование второй ветки.

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

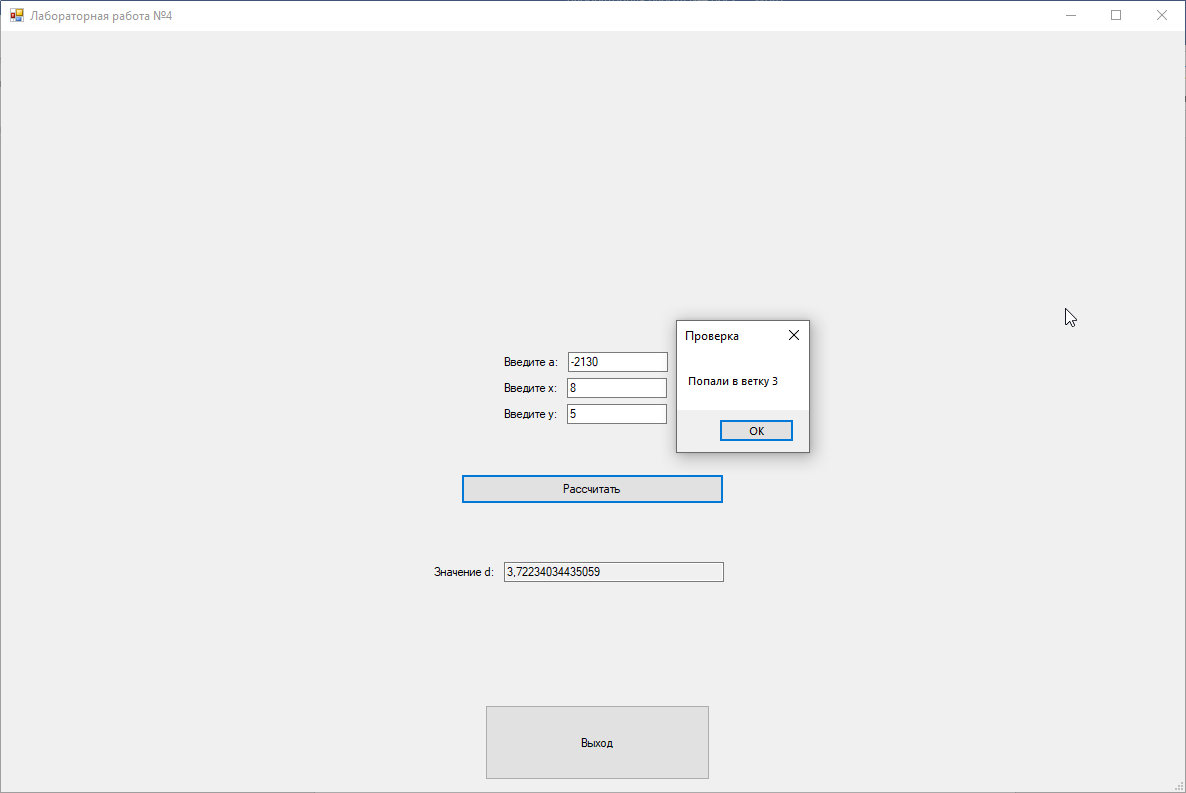


Рисунок 9 – Тестирование третьей ветки.

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).