

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И
МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Ордена Трудового Красного Знамени
федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский технический университет связи и информатики»

Кафедра «Информатика»

ОТЧЕТ

по дисциплине «Алгоритмы и алгоритмические языки»

Лабораторная работа № 2

Программирование алгоритмов разветвляющихся структур

Выполнил: студент группы БФИ №2202

Сидорук Д. В.

Принял: старший преподаватель Загвоздкина А. В.

Москва, 2023 г.

Задание

Разработать программу для расчета функции

$$c = \begin{cases} \min\{\max\{x^2 + y^2; a^x + y^a\}; \sqrt{x} + \sqrt{a^x}\} & x > a; y > 0 \\ \frac{\min\{a+b; x^{a+b}\}}{1+a+b^a} & x \leq a; 0 \leq y \leq 3x \\ \operatorname{tg}^2 |x + y| & \text{в противном случае} \end{cases}$$

Для того, чтобы выполнить задание, необходимо разработать следующие методы:

1. Метод `public static double GetDouble(TextBox t)`, предназначенный для получения текста из текстового поля ввода и преобразования его в значение с плавающей точкой.
2. Метод `public static void OutputDouble(TextBox t, double value)`, предназначенный для конвертации переданного значения в текст и установки его в качестве текста переданного текстового поля.
3. Метод `public static double Evaluate(double a, double b, double x, double y)`, предназначенный для расчета данной функции.

Вышеперечисленные методы должны быть размещены в dll-библиотеке.

Перечень блок-схем

На рисунке ниже приведена блок-схема алгоритма метода `double GetDouble(TextBox t)`, предназначенного для получения текста из текстового поля ввода и преобразования его в значение с плавающей точкой.

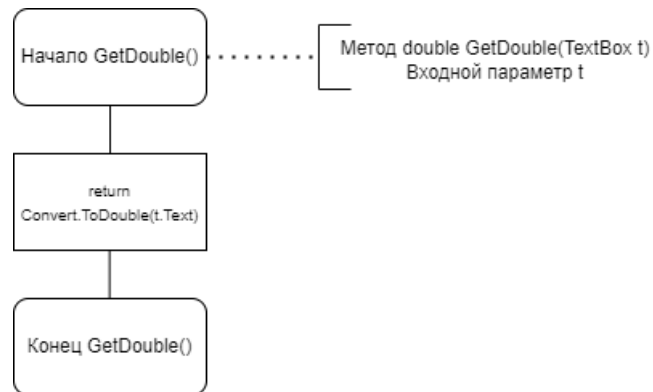


Рисунок 1 — блок-схема алгоритма метода `double GetDouble(TextBox t)`, предназначенного для получения текста из текстового поля ввода и преобразования его в значение с плавающей точкой.

На рисунке ниже приведена блок-схема метода `void OutputDouble(TextBox t, double value)`, предназначенного для конвертации переданного значения в текст и установки его в качестве текста переданного текстового поля.

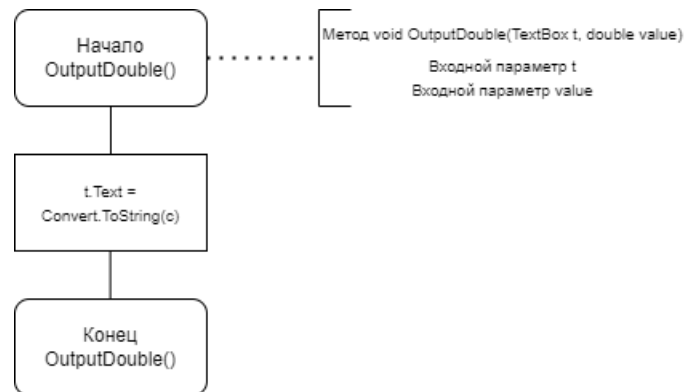


Рисунок 2 — блок-схема алгоритма метода `void OutputDouble(TextBox t, double value)`, предназначенного для конвертации переданного значения в текст и установки его в качестве текста переданного текстового поля.

На рисунке ниже приведена блок-схема алгоритма метода `double Evaluate(double a, double b, double x, double y)` предназначенного для расчета данной функции.

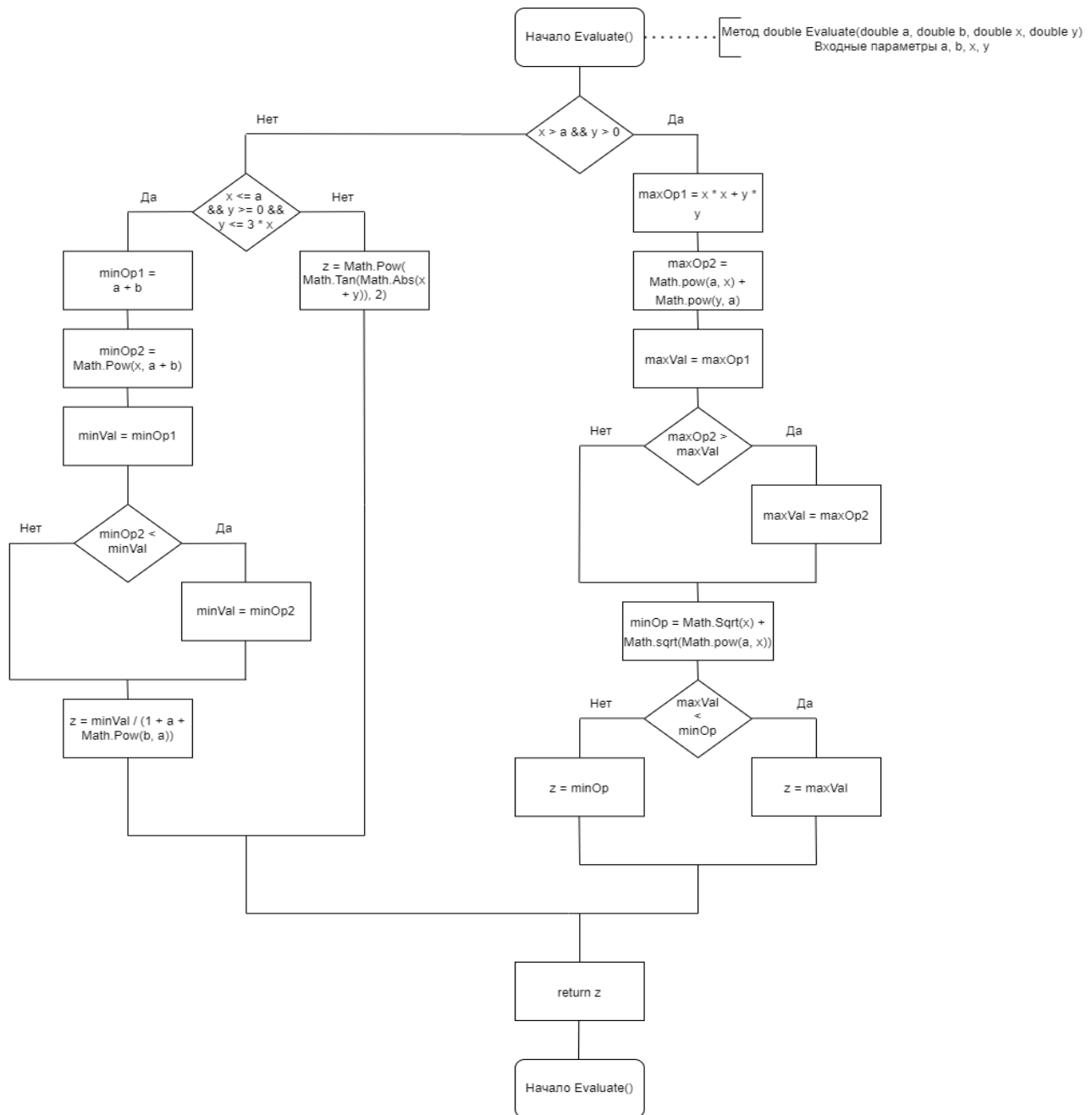


Рисунок 3 — блок-схема алгоритма метода `double Evaluate(double a, double b, double x, double y)`, предназначенного для расчета данной функции.

Алгоритм событийной кнопки

На рисунке ниже приведена блок-схема алгоритма событийной кнопки.

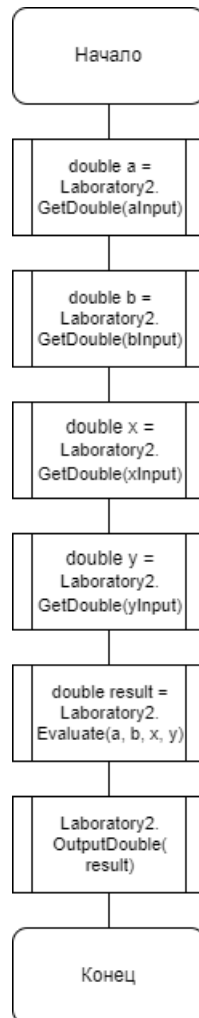


Рисунок 4 — блок-схема алгоритма событийной кнопки.

Содержание DLL-библиотеки

```
using System;
using System.Windows.Forms;

namespace Laboratory2Library
{
    public class Laboratory2
    {
        public static double GetDouble(TextBox t)
        {
            return Convert.ToDouble(t.Text);
        }
        public static void OutputDouble(TextBox t, double value)
        {
            t.Text = Convert.ToString(value);
        }
        public static double Evaluate(double a, double b, double x, double y)
        {
            double z;
            if (x > a && y > 0)
            {
                double maxOp1 = x * x + y * y;
                double maxOp2 = Math.Pow(a, x) + Math.Pow(y, a);

                double maxVal = maxOp1;
                if (maxOp2 > maxVal) maxVal = maxOp2;

                double minOp = Math.Sqrt(x) + Math.Sqrt(Math.Pow(a, x));
                if (maxVal < minOp) z = maxVal;
                else z = minOp;
            } else if (x <= a && y >= 0 && y <= 3 * x)
            {
                double minOp1 = a + b;
                double minOp2 = Math.Pow(x, a + b);

                double minVal = minOp1;
                if (minOp2 < minVal) minVal = minOp2;

                z = minVal / (1 + a + Math.Pow(b, a));
            } else z = Math.Pow(Math.Tan(Math.Abs(x + y)), 2);
            return z;
        }
    }
}
```

Содержание основной части программы

```
using System;
using System.Windows.Forms;

using Laboratory2Library;

namespace LaboratoryWorks
{
    public partial class Laboratory2Form : Form
    {
        public Laboratory2Form()
        {
            InitializeComponent();

            private void evaluateButton_Click(object sender, EventArgs e)
            {
                double a = Laboratory2.GetDouble(aInput);
                double b = Laboratory2.GetDouble(bInput);
                double x = Laboratory2.GetDouble(xInput);
                double y = Laboratory2.GetDouble(yInput);

                double result = Laboratory2.Evaluate(a, b, x, y);
                Laboratory2.OutputDouble(cOutput, result);
            }
        }
    }
}
```


Результаты выполнения работы

На рисунке ниже приведен результат выполнения программы при входных данных $a = 1$; $b = 1$; $x = 2$; $y = 2$, при которых алгоритм вычисляет значения выражения первой ветви

Лабораторная работа №2 - решение

Лабораторная работа №2
Программирование алгоритмов разветвляющихся структур

$$c = \begin{cases} \min\{\max\{x^2 + y^2; a^x + y^a\}; \sqrt{x} + \sqrt{a^x}\} & x > a; y > 0 \\ \frac{\min\{a+b; x^{a+b}\}}{1+a+b^a} & x \leq a; 0 \leq y \leq 3x \\ \operatorname{tg}^2 |x + y| & \text{в противном случае} \end{cases}$$

a = 1 b = 1 x = 2 y = 2 Вычислить

c = 2.41421356237309

К условию Выход

Рисунок 5 — результат выполнения программы при входных данных $a = 1$; $b = 1$; $x = 2$; $y = 2$

$x > a; y > 0 \Leftrightarrow 2 > 1; 2 > 0$, следовательно алгоритм вычисляет значение выражения первой ветви.

Выполним проверку: $x^2 + y^2 = 2^2 + 2^2 = 8$, $a^x + y^a = 1^2 + 2^1 = 3$, максимальное из них — восемь; $\sqrt{x} + \sqrt{a^x} = \sqrt{2} + \sqrt{1} = 2.414\dots$; минимальное значение из 3 и 2.414... — 2.414..., поэтому оно и является результатом выполнения программы.

На рисунке ниже приведен результат выполнения программы при входных данных $a = 2$; $b = 1$; $x = 1$; $y = 2$, при которых алгоритм вычисляет значения выражения второй ветви

Лабораторная работа №2
Программирование алгоритмов разветвляющихся структур

$$c = \begin{cases} \min\{\max\{x^2 + y^2; a^x + y^a\}; \sqrt{x} + \sqrt{a^x}\} & x > a; y > 0 \\ \frac{\min\{a+b; x^{a+b}\}}{1+a+b^a} & x \leq a; 0 \leq y \leq 3x \\ \operatorname{tg}^2 |x + y| & \text{в противном случае} \end{cases}$$

a = b = x = y =

c =

Рисунок 6 — результат выполнения программы при входных данных $a = 2$; $b = 1$; $x = 1$; $y = 2$

$x \leq a; 0 \leq y \leq 3x \Leftrightarrow 1 \leq 2; 0 \leq 2 \leq 3$, следовательно алгоритм вычисляет значение выражения второй ветви.

Выполним проверку: $a+b=2+1=3$, $x^{a+b}=1^3=1$, минимальное из них — 1; $1+a+b^a=1+2+1^2=4$, $\frac{\min\{a+b; x^{a+b}\}}{1+a+b^a}=\frac{1}{4}=0.25$, данное значение является результатом выполнения программы.

На рисунке ниже приведен результат выполнения программы при входных данных $a = 2$; $b = 1$; $x = 1$; $y = 4$, при которых алгоритм вычисляет значения выражения третьей ветви

Лабораторная работа №2 - решение

Лабораторная работа №2

Программирование алгоритмов разветвляющихся структур

$$c = \begin{cases} \min\{\max\{x^2 + y^2; a^x + y^a\}; \sqrt{x} + \sqrt{a^x}\} & x > a; y > 0 \\ \frac{\min\{a+b; x^{a+b}\}}{1+a+b^a} & x \leq a; 0 \leq y \leq 3x \\ \operatorname{tg}^2 |x + y| & \text{в противном случае} \end{cases}$$

a = 2 b = 1 x = 1 y = 4 Вычислить

c = 11.4278817074584

К условию Выход

Рисунок 7 — результат выполнения программы при входных данных $a = 2$; $b = 1$; $x = 1$; $y = 4$

$x > a$; $y > 0$ не справедливо, так как $x = 1, a = 2, y = 4$; так же не справедливо $x \leq a$; $0 \leq y \leq 3x$, следовательно алгоритм вычисляет значение выражения третьей ветви.

$\operatorname{tg} 5 \approx -3.3805, \operatorname{tg}^2 5 \approx 11.4278\dots$, данное значение и является результатом выполнения программы.

Список используемых источников

1. Гуриков С. Р. Введение в программирование на языке Visual C#: учебное пособие / С. Р. Гуриков. — Москва : ФОРУМ : ИНФРА-М, 2020. — 447 с.