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 заданию №3**

**по дисциплине**

**«Учебная (ознакомительная) практика»**

Выполнил: студент гр. БЭИ2202

Кулешов А.С.

Вариант 15.

Проверил: доц. Гуриков С. Р.

Москва, 2023 г.

**1) Индивидуальное задание на разработку проекта**

Разработать программный код функции:

В написанном методе использовать оператор return. Встроенные функции нахождения максимума/минимума не использовать.

**2) Техническое задание**

1. Выбрать вариант задания из таблицы согласно порядковому номеру в журнале группы.

2. Записать заданные или полученные в процессе формализации математические формулы на языке программирования.

3. Разработать алгоритмы решения заданной задачи (количество алгоритмов равно количеству используемых пользовательских методов алгоритм событийной).

4. В ходе решения задачи своего варианта предусмотреть написание только одного метода. Он отвечает за решение системы, которая дана в варианте задания. В написанном методе должен быть только один оператор return. Встроенные функции для нахождения максимального и минимального значений не использовать. Дополнительные методы для нахождения max и min не программировать.

5. Создать проект с DLL-библиотекой, решающий предписанные задачи, разработать метод ввода исходных данных и метод вывода результата. Обмен данными между подпрограммами должен осуществляться через параметры, без использования глобальных переменных. Событийная процедура должна содержать только операторы вызова пользовательских подпрограмм.

6. Организовать использование проверки корректности ввода данных (параграф 2.6 основного источника литературы) относительно объектов Текстовое поле на всех формах проекта.

7. Предусмотреть возможность вывода сообщения об ошибке в виде всплывающей подсказки, используя элемент управления ErrorProvider (стр. 279 основного источника литературы) относительно объектов Текстовое поле на всех формах проекта.

8. Результаты работы программы записать в текстовый файл.

9. Подготовить тесты для контрольного решения задачи. Показать их использование, отобразив в отчете по заданию.

10. Выполнить проект и получить результаты, которые отобразить в виде рисунков при написании отчета по заданию.

11. Выполнить отчет в соответствии с ГОСТ на схемы алгоритмов и согласно ГОСТ оформления отчетов по НИР.

**3) Схемы алгоритмов**

Алгоритмы ввода/вывода представлены на рисунках 1, 2. Ввод реализуется благодаря конвертации строки в вещественное число (метод Convert.ToDouble()), также используется проверка корректности данных для преобразования и выдаётся ошибка в Error Provider. Вывод реализуется благодаря конвертации числа в строку (метод ToString()). Для округления вывода до пятого знака после запятой дополнительно была использована маска “0.00000”.

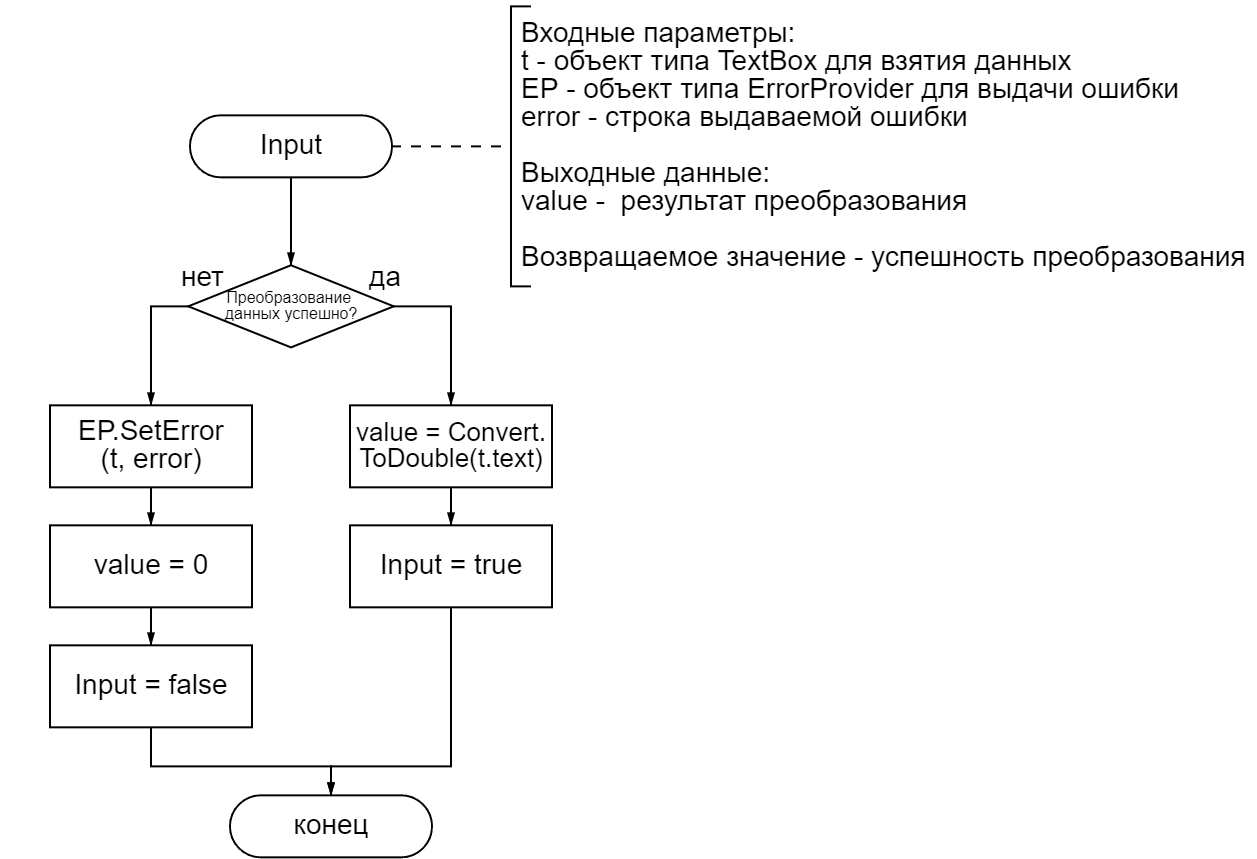


Рисунок 1 – Метод ввода Input

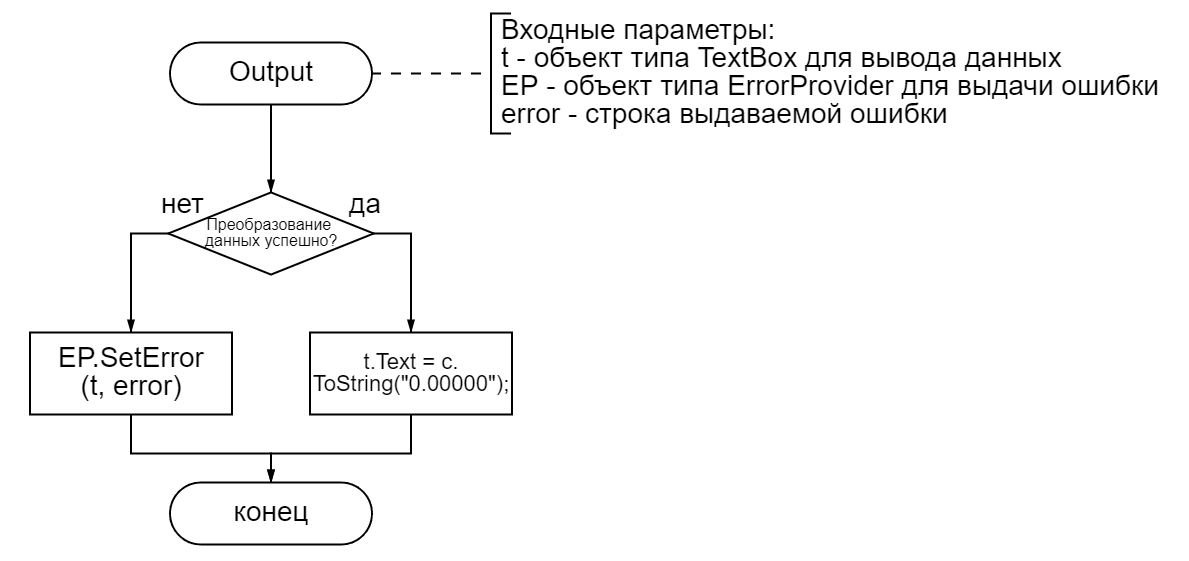


Рисунок 2 – Метод вывода Output

Алгоритм метода, решающего поставленную задачу, представлен на рисунке 3. Математические функции были взяты из математического пакета System.Math. Для файлового вывода использовалась функция System.IO.File.WriteAllText

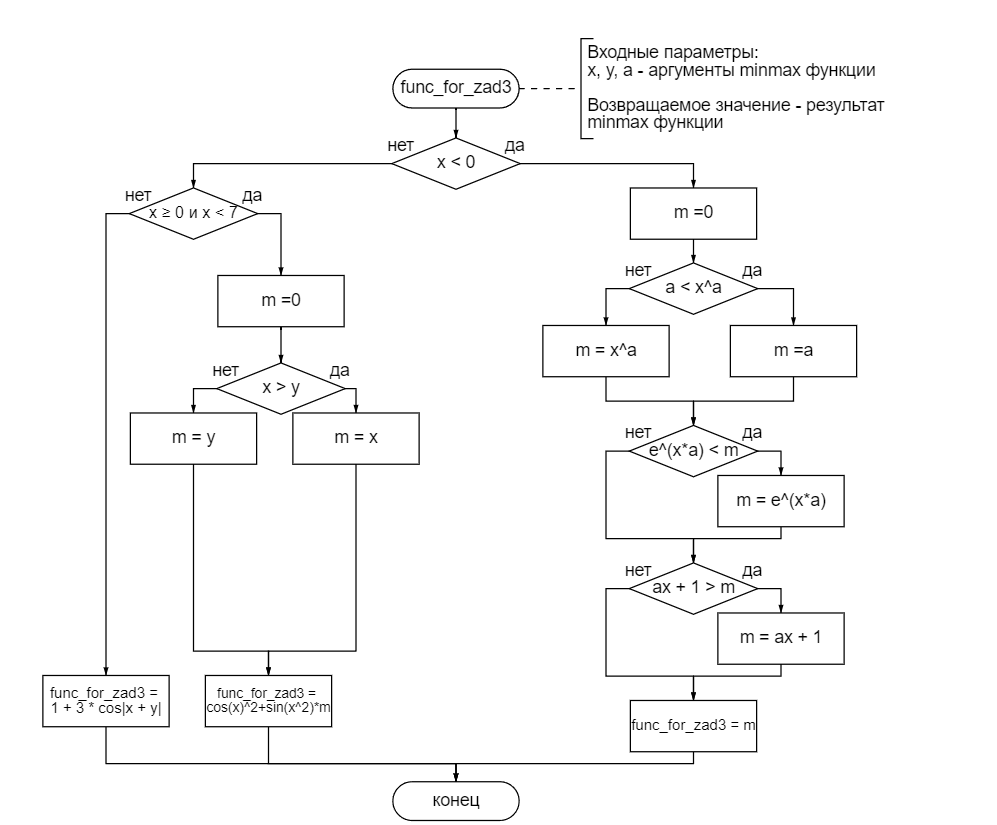


Рисунок 3 – Метод решения задачи func\_for\_zad3

**4) Программный код DLL-библиотеки**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace dll\_for\_prac

{

public class Class1

{

// Ввод данных с обработкой ошибки (входные данные могут быть не преобразованны при конвертации). Возвращает корректность процесса конвертации

public static bool Input(TextBox t, out double value, ErrorProvider EP, String error)

{

try

{

value = Convert.ToDouble(t.Text);

return true;

}

catch (Exception Ex)

{

EP.SetError(t, error);

value = 0;

return false;

}

}

// Вывод данных с обработкой ошибки (double переменная может иметь тип NaN)

public static void Output(TextBox t, double c, ErrorProvider EP, String error)

{

if (Double.IsNaN(c))

{

EP.SetError(t, error);

}

else

{

t.Text = c.ToString("0.00000");

}

}

public static double func\_for\_zad3(double x, double y, double a)

{

if (x < 0)

{

double m = 0;

if (a < Math.Pow(x, a)) m = a; else m = Math.Pow(x, a);

if (Math.Exp(x \* a) < m) m = Math.Exp(x \* a);

if (a \* x + 1 > m) m = a \* x + 1;

return m;

}

else

{

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

{

double m = 0;

if (x > y) m = x; else m = y;

return Math.Pow(Math.Cos(x), 2) + Math.Sin(Math.Pow(x, 2)) \* m;

}

else

{

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

}

}

}

}

}

**5) Программный код событийной процедуры**

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

{

double value\_return;

double x, y, a;

String final\_content = "Задание №3\n\n";

if (!Input(textBoxX, out x, errorProvider1, "В поля для ввода данных допускается лишь ввод числовых значений с плавающей точкой")) return;

if (!Input(textBoxY, out y, errorProvider1, "В поля для ввода данных допускается лишь ввод числовых значений с плавающей точкой")) return;

if (!Input(textBoxA, out a, errorProvider1, "В поля для ввода данных допускается лишь ввод числовых значений с плавающей точкой")) return;

value\_return = func\_for\_zad3(x,y,a);

Output(textBox\_return, value\_return, errorProvider1, "Результатом вычисления является не действительное число");

final\_content += "Входные данные(x, y, a): " + x + " " + y + " " + a + "\n\n";

final\_content += "Выходные данные: " + value\_return + "\n";

System.IO.File.WriteAllText("task2.txt", final\_content);

}

**6) Результаты работы программы**

На рисунке 4 представлен результат работы программы при случайных тестовых данных для проверки значения, представленного программой. Программа работает корректно на наборах тестовых данных

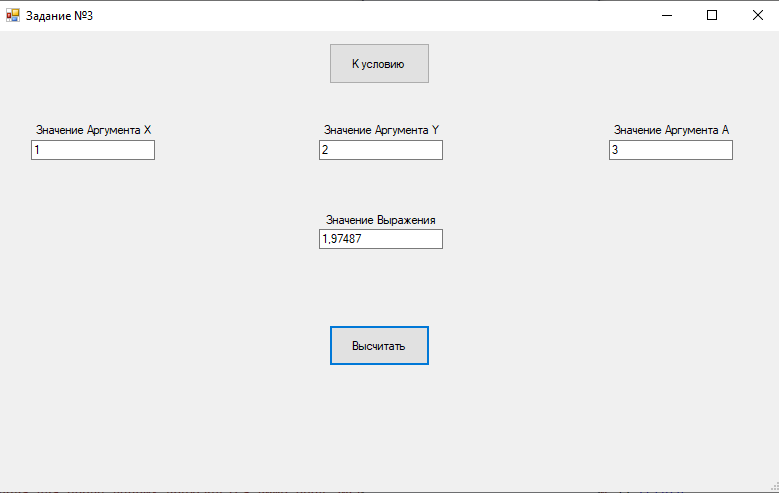


Рисунок 4 - Результаты работы при тестовых данных

Попробуем ввести недопустимые данные, например, «МТУСИ». Рисунок 5.

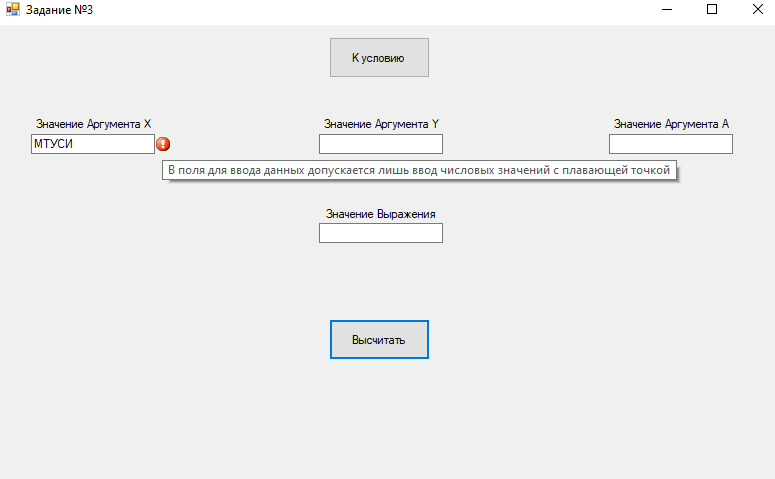


Рисунок 5 - Результат при некорректно введенных данных

**7) Список основных источников литературы**

1. Гуриков С. Р. Введение в программирование на языке Visual C# : учебное пособие / С.Р. Гуриков. — Москва : ФОРУМ : ИНФРА-М, 2020. — 447 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-00091-458-8. - Текст : электронный. - URL: https://znanium.com/catalog/product/1092167 (дата обращения: 23.06.2023). – Режим доступа: по подписке + библиотека МТУСИ