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»

**Отчеты по учебной практике**

Выполнил: студент гр. БЭИ2202

Кулешов А. С.

Вариант 15.

Проверил: доц. каф. «Информатика» Гуриков С. Р.

Москва, 2023 г.

**Отчет по заданию №1**

**1) Индивидуальное задание на разработку проекта**

Разработать программный код функции:

Осуществить вычисление тремя разными способами (при помощи return, out и ref).

**2) Техническое задание**

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

2. Следует учесть, что исходные данные в таблице вариантов даны исключительно для тестирования программы. В программе предусмотреть ввод любых данных от пользователя с последующей защитой от ошибок.

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

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

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. В ходе решения задачи своего варианта предусмотреть расчет арифметического выражения тремя способами: с модификаторами ref и out, оператором return.

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

8. 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.

9. Результаты работы программы записать в текстовый файл.

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

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

12. 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”.

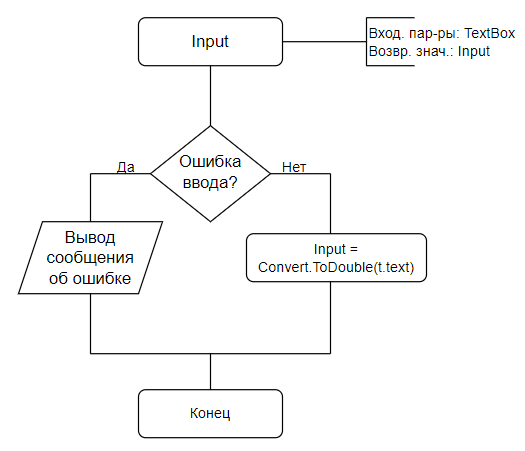


Рисунок 1 – Метод ввода данных из TextBox

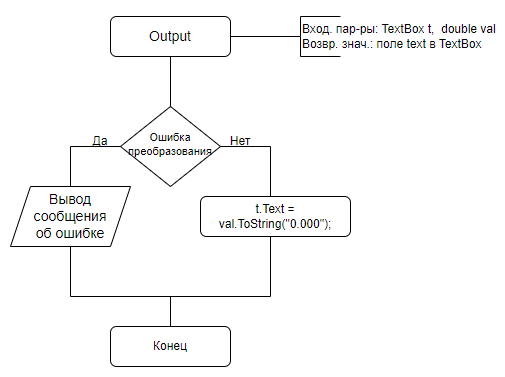


Рисунок 2 – Метод вывода данных в TextBox

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

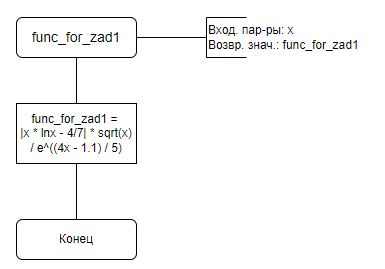


Рисунок 3 – Метод решения задачи

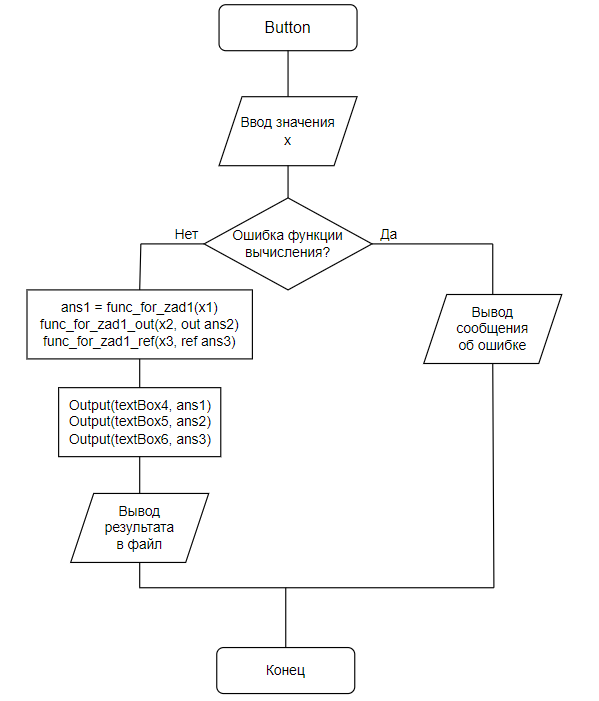


Рисунок 4 – Событийная процедура вычисления задания №1

**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");

}

}

// Функция для решения первого задания (через return)

public static double func\_for\_zad1(double x)

{

return Math.Abs(x \* Math.Log(x) - 4d / 7d) \* Math.Sqrt(x) / Math.Exp((4 \* x - 1.1d) / 5);

}

// Функция для решения первого задания (через out)

public static void func\_for\_zad1\_out(double x, out double res)

{

res = Math.Abs(x \* Math.Log(x) - 4d / 7d) \* Math.Sqrt(x) / Math.Exp((4 \* x - 1.1d) / 5);

}

// Функция для решения первого задания (через ref)

public static void func\_for\_zad1\_ref(double x, ref double res)

{

res = Math.Abs(x \* Math.Log(x) - 4d / 7d) \* Math.Sqrt(x) / Math.Exp((4 \* x - 1.1d) / 5);

}

public static void NumberLimiter(KeyPressEventArgs e, TextBox t)

{

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

{

if (!(e.KeyChar.ToString() == "," && t.Text.IndexOf(",") == -1) && !(e.KeyChar.ToString() == "-"))

e.Handled = true;

}

}

}

**5) Программный код событийной процедуры**

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

{

double x1, x2, x3;

double ans1, ans2, ans3 = 0;

String final\_content = "Задание №1\n\n";

if (Input(textBox1, out x1, errorProvider1, "В поля для ввода данных допускается лишь ввод числовых значений с плавающей точкой"))

{

try

{

ans1 = func\_for\_zad1(x1);

final\_content += "Входные данные(return): " + x1 + '\n';

final\_content += "Выходные данные(return): " + ans1 + "\n\n";

Output(textBox4, ans1, errorProvider1, "Результатом вычисления является не действительное число");

}

catch (Exception Ex)

{

errorProvider1.SetError(textBox4, "Во время вычислений произошла ошибка");

}

}

if (Input(textBox2, out x2, errorProvider1, "В поля для ввода данных допускается лишь ввод числовых значений с плавающей точкой"))

{

try

{

func\_for\_zad1\_out(x2, out ans2);

final\_content += "Входные данные(out): " + x2 + '\n';

final\_content += "Выходные данные(out): " + ans2 + "\n\n";

Output(textBox5, ans2, errorProvider1, "Результатом вычисления является не действительное число");

}

catch (Exception Ex)

{

errorProvider1.SetError(textBox5, "Во время вычислений произошла ошибка");

}

}

if (Input(textBox3, out x3, errorProvider1, "В поля для ввода данных допускается лишь ввод числовых значений с плавающей точкой"))

{

try

{

func\_for\_zad1\_ref(x3, ref ans3);

final\_content += "Входные данные(ref): " + x3 + '\n';

final\_content += "Выходные данные(ref): " + ans3 + "\n\n";

Output(textBox6, ans3, errorProvider1, "Результатом вычисления является не действительное число");

}

catch (Exception Ex)

{

errorProvider1.SetError(textBox6, "Во время вычислений произошла ошибка");

}

}

System.IO.File.WriteAllText("task1.txt", final\_content);

}

**6) Результаты работы программы**

На рисунке 5 представлен результат работы программы при данном в задании для проверки значении входного параметра и значении функции. Как видно, программа работает корректно.

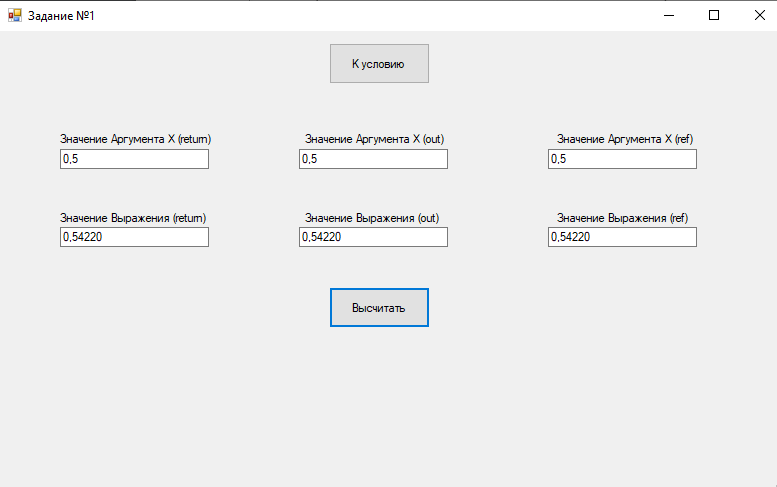


Рисунок 5 – Результат работы программы при заданном параметре

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

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.

**Отчет по заданию №2**

**1) Индивидуальное задание на разработку проекта**

Разработать программный код решения следующей задачи:

Вычислить значение площади каждого из трёх кругов, ограниченных тремя окружностями, длины которых L1, L2, L3 известны. Используя формулы для окружностей:

**Назначение функции**: Вычислить площадь круга , предварительно вычислив R по формуле

Осуществить вычисление двумя разными способами (при помощи out, ref).

**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). Кроме того, должна быть выполнена схема иерархии подпрограмм.

4. Добавить в существующую DLL-библиотеку методы, решающие предписанные задачи. Обмен данными между подпрограммами должен осуществляться через параметры, без использования глобальных переменных. Событийная процедура должна содержать только операторы вызова пользовательских подпрограмм.

5. В ходе решения задачи своего варианта предусмотреть написание двух методов с модификаторами ref и out. Данные методы должны вызывать несколько раз метод, назначение которого указано в задании.

В результатах выполнения программы необходимо показать, что результаты получились одинаковыми.

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”.

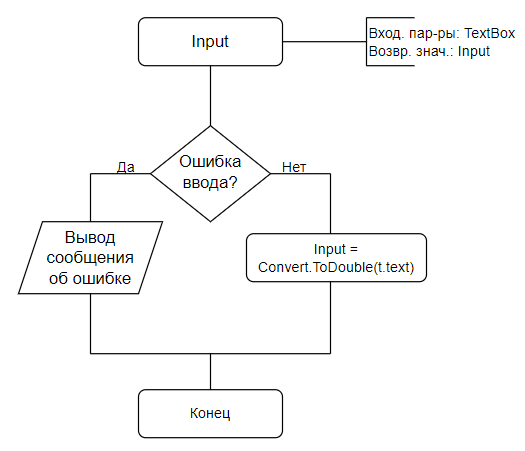


Рисунок 1 – Метод ввода данных из TextBox

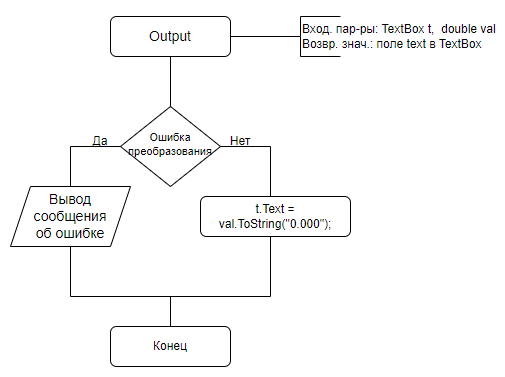


Рисунок 2 – Метод вывода данных в TextBox

Был разработан вспомогательный метод для решения задачи: алгоритм нахождения радиуса окружности от её длины, представлен на рисунке 3. Алгоритм метода, решающего поставленную задачу, представлен на рисунке 4. Математические функции были взяты из математического пакета System.Math. Для файлового вывода использовалась функция System.IO.File.WriteAllText

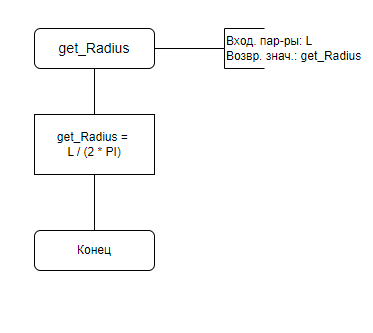


Рисунок 3 – Вспомогательный метод, находящий радиус окружности

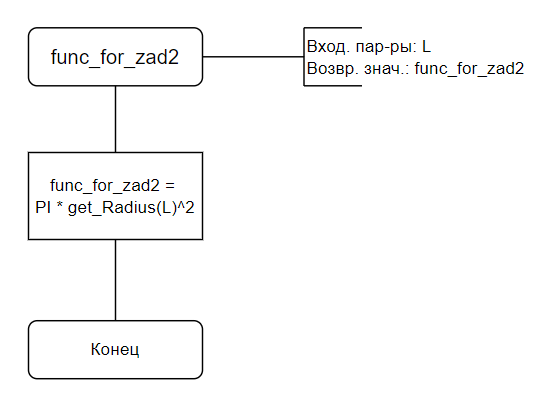


Рисунок 4 – Метод, решающий задачу

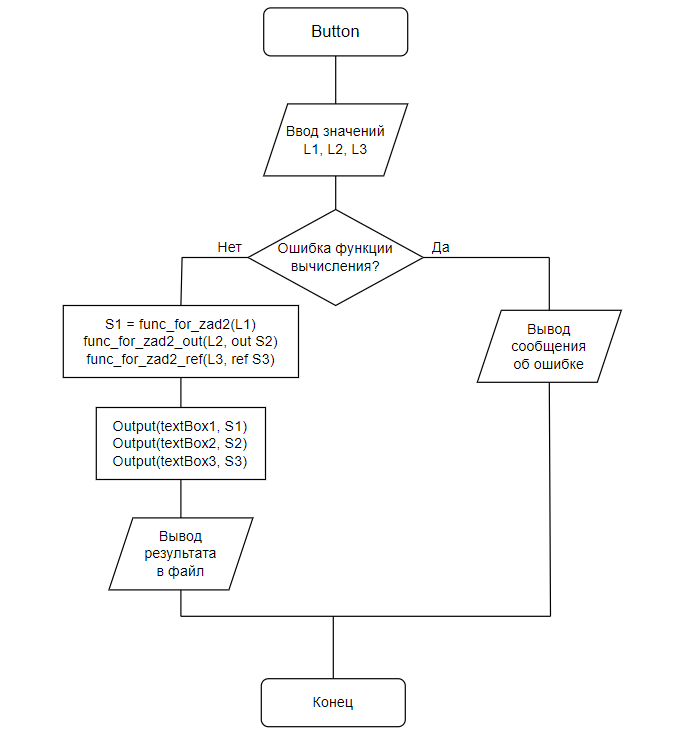


Рисунок 5 – Алгоритм событийной процедуры

Схема иерархии методов приведена на рисунке 6.

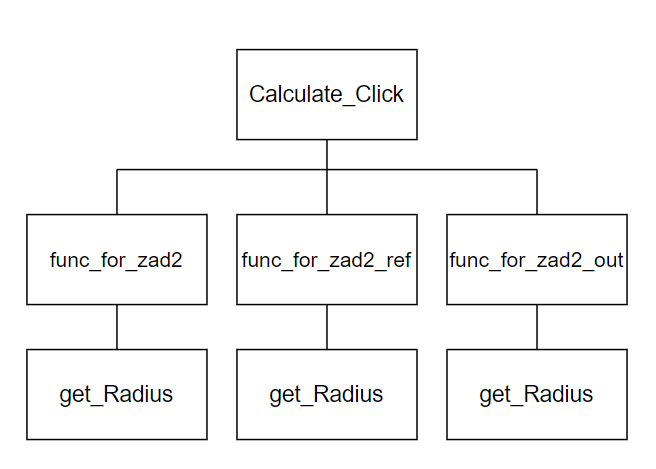


Рисунок 6 — Схема иерархии методов

**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");

}

}

// Функция для решения второго задания (через return)

public static double get\_Radius(double L)

{

// L = 2 PI R => R = L/(2 \* PI)

return L / (2 \* Math.PI);

}

// Функция для решения второго задания (через return)

public static double func\_for\_zad2(double L)

{

double S = Math.PI \* Math.Pow(get\_Radius(L),2);

return S;

}

// Функция для решения второго задания (через out)

public static void func\_for\_zad2\_out(double L, out double res)

{

double S = Math.PI \* Math.Pow(get\_Radius(L), 2);

res = S;

}

// Функция для решения второго задания (через ref)

public static void func\_for\_zad2\_ref(double L, ref double res)

{

double S = Math.PI \* Math.Pow(get\_Radius(L), 2);

res = S;

}

public static void NumberLimiter(KeyPressEventArgs e, TextBox t)

{

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

{

if (!(e.KeyChar.ToString() == "," && t.Text.IndexOf(",") == -1) && !(e.KeyChar.ToString() == "-"))

e.Handled = true;

}

}

}

}

**5) Программный код событийной процедуры**

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

{

double L1, L2, L3;

double S1, S2 = 0, S3;

String final\_content = "Задание №2\n\n";

if (Input(textBoxL1, out L1, errorProvider1, "В поля для ввода данных допускается лишь ввод числовых значений с плавающей точкой"))

{

try

{

S1 = func\_for\_zad2(L1);

final\_content += "Входные данные(return): " + L1 + '\n';

final\_content += "Выходные данные(return): " + S1 + "\n\n";

Output(textBoxS1, S1, errorProvider1, "Результатом вычисления является не действительное число");

}

catch (Exception Ex)

{

errorProvider1.SetError(textBoxS1, "Во время вычислений произошла ошибка");

}

}

if (Input(textBoxL2, out L2, errorProvider1, "В поля для ввода данных допускается лишь ввод числовых значений с плавающей точкой"))

{

try

{

func\_for\_zad2\_ref(L2, ref S2);

final\_content += "Входные данные(return): " + L2 + '\n';

final\_content += "Выходные данные(return): " + S2 + "\n\n";

Output(textBoxS2, S2, errorProvider1, "Результатом вычисления является не действительное число");

}

catch (Exception Ex)

{

errorProvider1.SetError(textBoxS2, "Во время вычислений произошла ошибка");

}

}

if (Input(textBoxL3, out L3, errorProvider1, "В поля для ввода данных допускается лишь ввод числовых значений с плавающей точкой"))

{

try

{

func\_for\_zad2\_out(L3, out S3);

final\_content += "Входные данные(return): " + L3 + '\n';

final\_content += "Выходные данные(return): " + S3 + "\n\n";

Output(textBoxS3, S3, errorProvider1, "Результатом вычисления является не действительное число");

}

catch (Exception Ex)

{

errorProvider1.SetError(textBoxS3, "Во время вычислений произошла ошибка");

}

}

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

}

}

**6) Результаты работы программы**

На рисунке 7 представлен результат работы программы при случайном наборе данных для проверки значения, представленного программой. Программа работает корректно на наборах тестовых данных

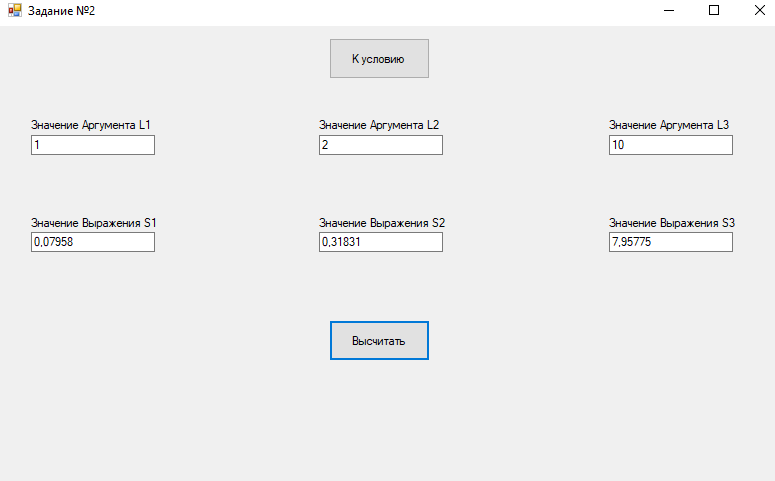


Рисунок 7 – Результаты работы при тестовых данных

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

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.

**Отчет по заданию №3**

**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”.

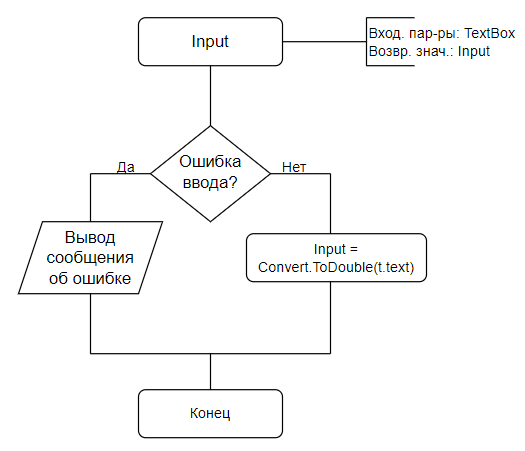


Рисунок 1 – Метод ввода данных из TextBox

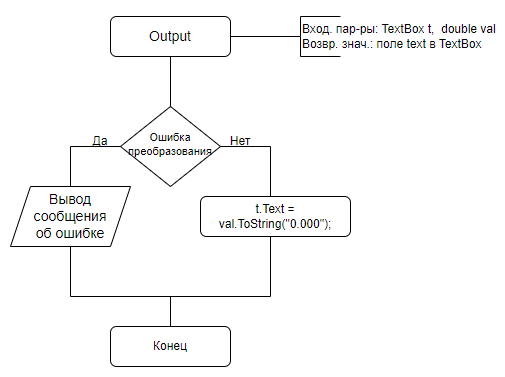


Рисунок 2 – Метод вывода данных в TextBox

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

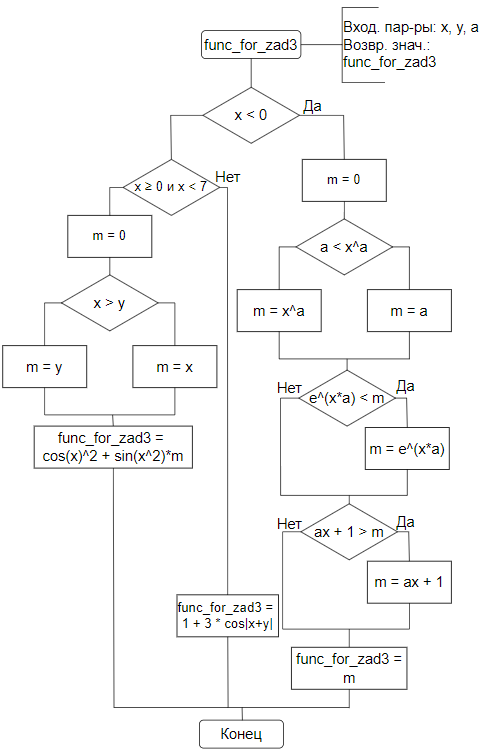


Рисунок 3 – Метод решения задачи

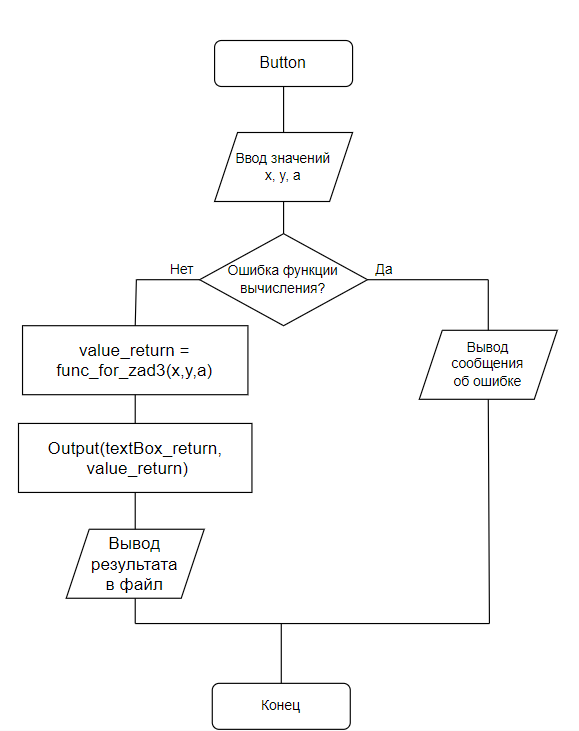


Рисунок 4 – Алгоритм событийной процедуры

**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));

}

}

}

public static void NumberLimiter(KeyPressEventArgs e, TextBox t)

{

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

{

if (!(e.KeyChar.ToString() == "," && t.Text.IndexOf(",") == -1) && !(e.KeyChar.ToString() == "-"))

e.Handled = true;

}

}

}

}

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

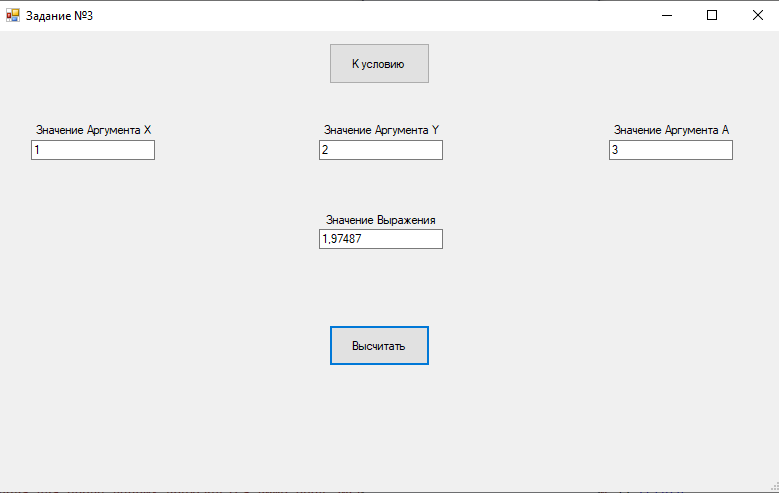


Рисунок 5 - Результаты работы при тестовых данных

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

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.

**Отчет по заданию №4**

**1) Индивидуальное задание на разработку проекта**

Получить расчет в таблице Excel на основе использования математических функций и на основе языка программирования VBA выражения.

**2) Техническое задание**

1. Выбрать вариант задания из таблицы (см. задание практики №1) согласно порядковому номеру в журнале группы.

2. Используя основной источник литературы №2 (Глава 2), получить расчет своего выражения(-ний) в таблице Excel на основе использования математических функций.

3. Построенная таблица должна содержать заголовок, выполненный на основе объединения ячеек и быть выровнен по центру. Выполнить границы.

4. Таблица должна содержать исходные данные.

5. Полученные результаты должны совпасть с результатами первого задания по практике.

6. Используя основной источник литературы №2 (Глава 8, стр. 117- 119), получить расчет своего выражения(-ний) на основе языка программирования VBA. Согласно изученному материалу, создать кнопку на листе Excel и организовать взаимосвязь с рабочим листом при вводе и выводе данных.

7. Полученные результаты должны совпасть с результатами первого задания по практике, а также с ранее полученными результатами расчета в Excel.

8. Выполнить отчет согласно ГОСТ оформления отчетов по НИР.

**3) Программный код модуля и событийной процедуры на языке VBA.**

Программный код модуля с функцией, решающей поставленную задачу представлен ниже.

Расчет задания при помощи функции

**Public** **Function** funcByVal(**ByVal** x **As** Double) **As** Double

funcByVal = Abs(x \* Log(x) - 4# / 7#) \* Sqr(x) / Exp((4 \* x - 1.1) / 5)

**End** **Function**

Расчет задания при помощи процедуры

**Public** **Sub** funcByRef(**ByVal** x **As** Double, **ByRef** res **As** Double)

res = Abs(x \* Log(x) - 4# / 7#) \* Sqr(x) / Exp((4 \* x - 1.1) / 5)

**End** **Sub**

Программный код событийной процедуры представлен ниже. Ввод и вывод вставляются автоматически в заданные ячейки

**Sub** Кнопка2\_Щелчок()

**Dim** res\_return **As** Double

**Dim** res\_ref **As** Double

Range("b2").Value = Val(InputBox("Введите значение х", "Ввод числа"))

*' Получение значений*

res\_return = funcByVal(Range("b2"))

funcByRef Range("b2"), res\_ref

MsgBox "Значение выражения (функция) = " & Format(res\_return, "0.000"), vbInformation, "Получаем ответ"

*' Вывод в Excel*

Range("b4").Value = res\_return

Range("b5").Value = res\_ref

**End** **Sub**

**4) Результаты расчета выражений**

Результаты расчета представлены на рисунке 1. Ввод значения происходит из поля b2, затем записывается поля b4 и b5 соответственно.

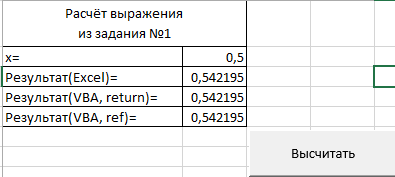


Рисунок 1 – Результаты расчета выражения

Сравним результаты с полученными в задании №1. Рисунок 2.

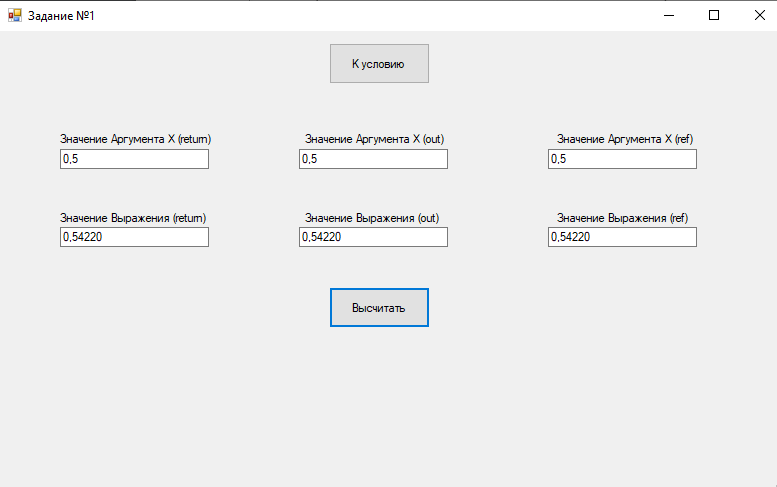


Рисунок 2 – Результат расчета в задании №1

Как видно на рисунках, результаты почти совпадают, различие возникает из-за нехватки точности в c# приложении. Программа работает корректно

**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: 15.06.2023). – Режим доступа: по подписке.

**Отчет по заданию №5**

**1) Индивидуальное задание на разработку проекта**

Построить таблицу функции y = f(x) в Excel при изменении x на отрезке [a;b] с шагом h. Найти количество и произведение значений функции

**2) Техническое задание**

1. Выбрать вариант задания из таблицы согласно порядковому номеру

в журнале группы

2. Используя основной источник литературы №2 (Глава 2, стр. 125),

получить результат табулирования функции в Excel. Действия по

форматированию полученных таблиц (установка границ,

выравнивание, объединение и проч.) не выполнять.

3. Дополнительные действия по варианту своей задачи (например,

нахождение максимальных или минимальных значений, расчет суммы,

произведения и проч.), выполнить с использованием встроенных

функций Excel.

4. Выполнить запись макроса, который осуществляет форматирование

полученных таблиц. Код макроса назначить кнопке на листе Excel.

5. Выполнить запись макроса, который строит диаграмму на листе Excel

по полученным значениям. При этом можно использовать основной

источник литературы №2 (Глава 3). Выполнить подписи по осям и

другие действия по улучшению визуализации диаграммы. Код макроса

назначить кнопке на листе Excel.

6. На зачете по учебной практике студент должен быть подготовлен к

выполнению задания по табулированию функции на основе создания

функций и процедур VBA. Теоретический и практический материал

изложен в Главе 11 основного источника литературы №2.

Организация взаимодействия с рабочим листом Excel отработана при

реализации задания №4 учебной практики.

7. На зачете по учебной практике студент должен быть подготовлен к

записи макросов в Word и созданию соответствующих кнопок на Ленте

или на Панели быстрого доступа, которые форматируют текст

согласно ГОСТ по оформлению отчетов по НИР.

8. Выполнить отчет согласно ГОСТ оформления отчетов по НИР

**3) Программный код модуля и событийной процедуры на языке VBA.**

Программный код модуля с функцией, решающей поставленную задачу представлен ниже.

**Option** **Explicit**

**Public** **Function** func(**ByVal** x **As** **Double**) **As** **Double**

func = Sin(4# \* x) - 2#

**End** **Function**

Программный код событийной процедуры представлен ниже. Ввод и вывод вставляются автоматически в заданные ячейки

**Sub** Кнопка1\_Щелчок()

Range("b3").Value = func(Range("a3"))

Range("b4").Value = func(Range("a4"))

Range("b5").Value = func(Range("a5"))

Range("b6").Value = func(Range("a6"))

Range("b7").Value = func(Range("a7"))

Range("b8").Value = func(Range("a8"))

Range("b9").Value = func(Range("a9"))

Range("b10").Value = func(Range("a10"))

Range("b11").Value = func(Range("a11"))

Range("b12").Value = func(Range("a12"))

Range("b13").Value = func(Range("a13"))

Range("b14").Value = func(Range("a14"))

Range("b15").Value = func(Range("a15"))

Range("b16").Value = func(Range("a16"))

Range("b17").Value = func(Range("a17"))

Range("b18").Value = func(Range("a18"))

Range("b19").Value = func(Range("a19"))

**End** **Sub**

**4) Результаты расчета выражений**

Результаты расчета представлены на рисунке 1. Ввод значения происходит из полей «Границы», затем формируется таблица, а при нажатии на кнопку происходит расчёт

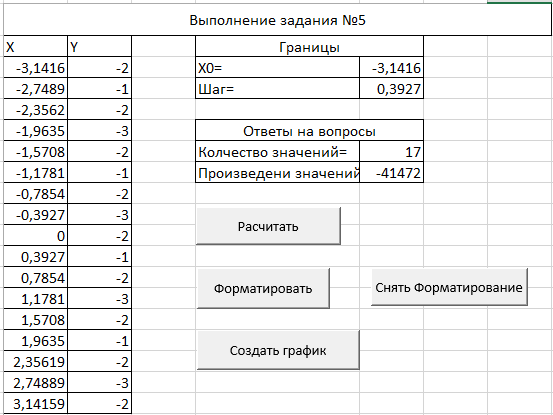


Рисунок 1 – Результаты расчета выражения

Также было добавлена кнопка для снятия и добавления форматирования с таблицы. Результат нажатия на эти кнопки можно увидеть на рисунках 2-3. Реализовано добавление и удаление форматирование макросами.

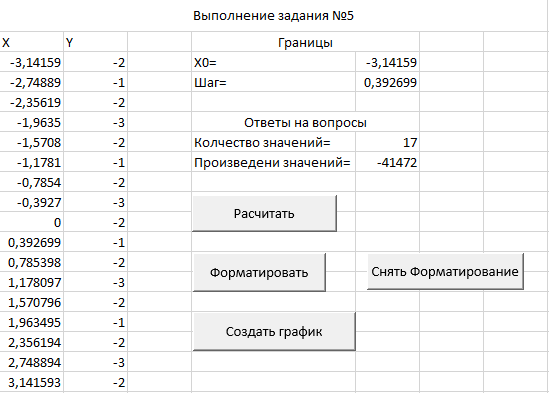


Рисунок 2 – снятие форматирования

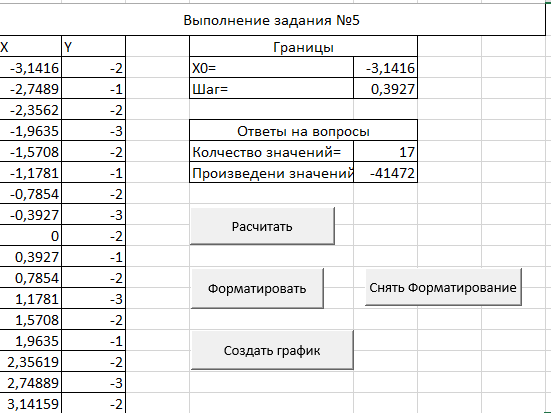


Рисунок 3 – добавление форматирования

Наконец, построение графика по нажатию на кнопку было реализовано при помощи вышеупомянутых макросов



Рисунок 4 – Созданный график

**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: 15.06.2023). – Режим доступа: по подписке.

**Отчет по заданию №6**

**1) Индивидуальное задание на разработку проекта**

Получить расчет в таблице Excel на основе использования математических функций и на основе языка программирования VBA выражения.

**2) Техническое задание**

1. Выбрать вариант задания из таблицы (см. задание учебной практики

№3) согласно порядковому номеру в журнале группы

2. Используя основной источник литературы №2 (Глава 2, пример 2.4),

получить расчет своей системы в программе Excel, используя функцию

ЕСЛИ. Написать макрос, форматирующий полученную таблицу, и

назначить его код кнопке на листе Excel.

3. Используя основной источник литературы №2 (Глава 9, задача 9.4,

листинг 9.7), получить расчет своей системы на основе использования

оптимальных алгоритмов нахождения максимальных и минимальных

значений (написать одну процедуру). При написании кода не

использовать функции ввода-вывода листинга 9.7, а воспользоваться

листингом 8.6, в котором показан пример организации взаимосвязи с

листом Excel. Подобный способ ввода-вывода данных использовался в

задании №4 учебной практики.

4. Написать макрос, форматирующий полученную таблицу, и назначить

его код кнопке на листе Excel.

5. Полученные результаты должны совпасть с результатами задания №3

учебной практики, а также с полученными результатами расчета в п.2.

6. Выполнить отчет согласно ГОСТ оформления отчетов по НИР.

**3) Программный код модуля и событийной процедуры на языке VBA.**

Программный код модуля с функцией, решающей поставленную задачу представлен ниже.

**Option** Explicit

**Public** **Sub** Sub3(**ByVal** x **As** Double, **ByVal** y **As** Double, **ByVal** a **As** Double, **ByRef** res **As** Double)

**Dim** m **As** Double

**If** x < 0 **Then**

**If** a < x ^ a **Then** m = a **Else** m = x ^ a

**If** Exp(x \* a) < m **Then** m = Exp(x \* a)

**If** a \* x + 1 > m **Then** m = a \* x + 1

res = m

**Else**

**If** x >= 0 **And** x < 7 **Then**

**If** x > y **Then** m = x **Else** m = y

res = Cos(x) ^ 2 + Sin(x ^ 2) \* m

**Else**

res = 1 + 3 \* Cos(Abs(x + y))

**End** **If**

**End** **If**

**End** **Sub**

Программный код событийной процедуры представлен ниже. Ввод и вывод вставляются автоматически в заданные ячейки

**Sub** Кнопка1\_Щелчок()

**Dim** res **As** Double

Range("b2").Value = Val(InputBox("Введите значение х", "Ввод числа"))

Range("b3").Value = Val(InputBox("Введите значение y", "Ввод числа"))

Range("b4").Value = Val(InputBox("Введите значение a", "Ввод числа"))

Sub3 Range("b2"), Range("b3"), Range("b4"), res

MsgBox "Значение выражения = " & res, vbInformation, "Получаем ответ"

Range("b7").Value = res

**End** **Sub**

**4) Результаты расчета выражений**

Результаты расчета представлены на рисунке 1. Ввод значения происходит из полей b2, b3 и b4. Одна из ячеек результатов заполняется автоматически(используя функции Excel), вторая по нажатию на кнопку

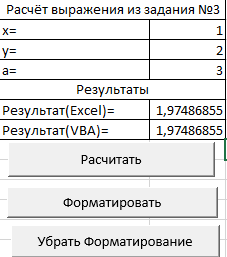


Рисунок 1 – Результаты расчета выражения

Также было добавлена кнопка для снятия и добавления форматирования с таблицы. Результат нажатия на эти кнопки можно увидеть на рисунках 2-3. Реализовано добавление и удаление форматирование макросами.

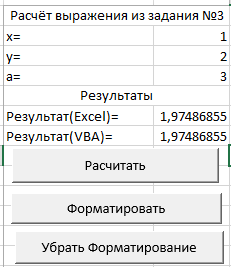


Рисунок 2 – снятие форматирования

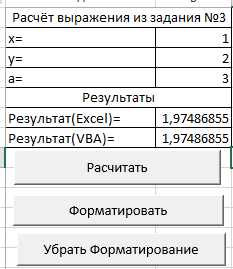


Рисунок 3 – добавление форматирования

Сравним результаты с полученными в задании №3. Рисунок 4.

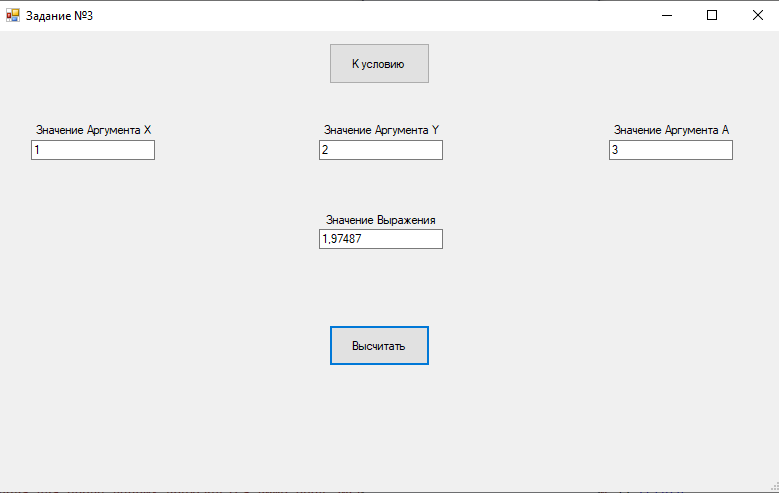


Рисунок 4 – Результат расчета в задании №3

Как видно на рисунках, результаты почти совпадают, различие возникает из-за нехватки точности в c# приложении. Программа работает корректно

**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: 15.06.2023). – Режим доступа: по подписке.