



**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ
ГОРОДА МОСКВЫ**
**Государственное бюджетное профессиональное
образовательное учреждение города Москвы**
«Колледж малого бизнеса № 4»
(ГБПОУ КМБ № 4)

МДК 05.03 Тестирование информационных систем

Специальность: 09.02.07 Информационные системы и программирование

Форма обучения: очная

Студент(ка): Козлова София Евгеньевна

Группа: ИПО-21.24

Руководитель: Рыбаков Александр Сергеевич

Отчётная работа защищена с оценкой «___» _____

Москва, 2025 г.

ПРАКТИЧЕСКАЯ РАБОТА №1

Тема. Ручная отладка программного обеспечения.

(Реализация линейного алгоритма вычисления значения w по заданной формуле в Windows Forms (.NET Framework)).

Цель. Изучить процесс отладки программного обеспечения ручным методом. разработать приложение для вычисления математического выражения с пользовательским интерфейсом, провести ручную отладку и тестирование.

Оборудование. ПК с установленной Visual Studio, ОС Windows.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Задание 1. Вариант в отчёте: реализация Windows-приложения для вычисления значения w по формуле, приведённой на изображении, с использованием Windows Forms (.NET Framework).

Задание 2. Ручная отладка:

1. Математическая модель задачи с указанием имён и назначения переменных:

ЛИНЕЙНЫЙ АЛГОРИТМ

Формула:

$$w = |\cos x - \cos y|^{1+2\sin^2 y} * \left(1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4}\right).$$

Переменные для линейного алгоритма				
Название переменной	Тип данных	Описание переменной	Значение которое хранит	Ограничения
x	double	Входные данные(ввод пользователем)	input	$-1 \leq x \leq 1$ (область определения arccos)
y	double	Входные данные(ввод пользователем)	input	Не должно приводить к делению на 0 в выражении $ x-y z+x^2$
z	double	Входные данные(ввод пользователем)	input	Нет ограничений
Cos (x)	double	Косинус угла		
Cos (y)	double	Косинус угла y		
Sin**2(y)	double			
$ \cos(x)-\cos(y) $	double			
$1 + z + z^{2/2} + z^{3/3} + z^{4/4}$	double			

- **x, y** — входные значения (углы в радианах), используются для вычисления косинусов и синусов.
- **z** — входное значение, используется в полиномиальной части формулы.
- **w** — результат вычисления (искомое значение).
- **cos x, cos y** — косинусы углов x и y.
- **Sin**2 y** — квадрат синуса угла y, используется в показателе степени.
- **|cos x – cos y|** — модуль разности косинусов, возводится в степень.
- **(1 + z + z**2/2 + z**3/3 + z**4/4)** — полином от z, множитель в формуле.

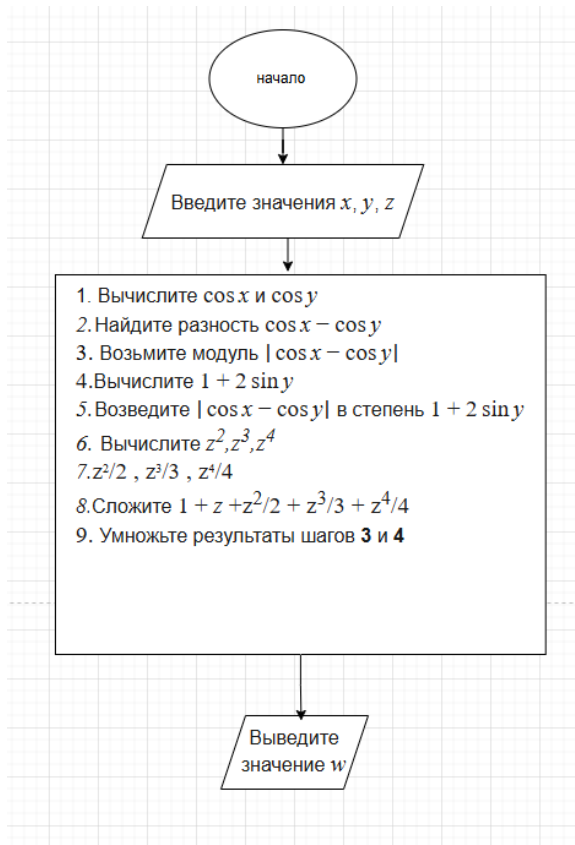
2. Спецификация программы:

- Программа должна принимать на вход три числа: x, y, z.
- Вычислять значение w по заданной формуле.
- Отображать результат в удобном для пользователя виде.
- Интерфейс должен включать:
 - поля ввода для x, y, z;

- кнопку «Вычислить»;
- поле вывода результата.
- Программа должна обрабатывать ошибки ввода (некорректные числа) и выводить соответствующие сообщения.

3. Алгоритм программы:

Линейное уравнение



4. Отладка логики программы методом «грубой силы» с помощью соседа:

- Дадаян Артур

1. `textBoxR.AppendText($"2. cos(y) = cos({y}) = {cosY:F5}\r\n");`
 В данной строке есть недочет, во всем коде указано, что после запятой будет 3 числа, а тут 5

`textBoxR.AppendText($"2. cos(y) = cos({y}) = {cosY:F3}\r\n");`

2. `textBoxR.AppendText($"Округлённый результат: {w:F3}");`
 В данной строке ошибка, задача была округлить, обычно это 2 числа после запятой, а тут три

`textBoxR.AppendText($"Округлённый результат: {w:F2}");`

5. Тестовые наборы для проверки функционала системы

$$4. w = |\cos x - \cos y|^{1+2 \sin^2 y} * \left(1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4}\right).$$

- Тестовые данные для условия $xy < 5$:

- $x=1, y=2: xy=2 < 5$.
- $x=-2, y=-3: xy=6 > 5$ (не подходит).
- $x=0, y=4: xy=0 < 5$.

- Тестовые данные для условия $xy > 7$:

- $x=3, y=3: xy=9 > 7$.
- $x=-4, y=-2: xy=8 > 7$.
- $x=2, y=4: xy=8 > 7$.

- Тестовые данные для условия «иначе»:

- $x=2, y=2: xy=4 < 5$ (не подходит).
- $x=3, y=2: xy=6$ (подходит).
- $x=-3, y=-2: xy=6$ (подходит).

- Граничные значения:

- $x=1, y=5: xy=5$ (иначе).
- $x=2, y=4: xy=8$ (иначе).

- Особые случаи:

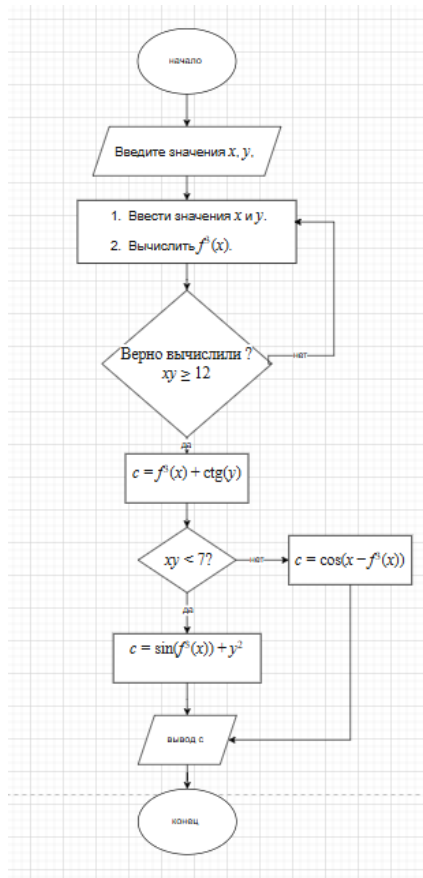
- $x=0, y=0: xy=0$ (иначе).
- $x=1, y=-1: xy=-1$ (иначе).

Разветвляющийся алгоритм

Описание переменных:

- x и y — входные параметры.
- $f(x)$ — некоторая функция от x .
- c — результат вычислений.
- Интерфейс должен включать:
 - поля ввода для x , y , z ;
 - кнопку «Вычислить»;
 - поле вывода результата.
- Программа должна обрабатывать ошибки ввода (некорректные числа) и выводить соответствующие сообщения.

3. Алгоритм программы:



4. Отладка логики программы методом «грубой силы» с помощью соседа:

- Дадаян Артур

1. richTextBox.AppendText("5) Произведение $xy = x + x + x + \dots + x + x$ ");
в данной строке ошибка

richTextBox.AppendText("5) Произведение $xy = x + xy + xy + \dots + xy + xy$ ");

2. $c = \text{Math.Sinh}(\text{Math.Pow}(fx, 2)) + y * y$;
В данной строке ошибка

$c = \text{Math.Sinh}(\text{Math.Pow}(fx, 3)) + y * y$;

5. Тестовые наборы для проверки функционала системы

- Тестовые данные для условия $xy > 12$:

- $x=4, y=4: xy=16 > 12$.
- $x=-3, y=-5: xy=15 > 12$.
- $x=6, y=2: xy=12$ (не подходит).

- Тестовые данные для условия $xy < 7$:

- $x=1, y=5: xy=5 < 7$.
- $x=-2, y=-3: xy=6 < 7$.
- $x=3, y=2: xy=6 < 7$.

- Тестовые данные для условия «иначе»:

- $x=2, y=4: xy=8$ (иначе).
- $x=-4, y=-2: xy=8$ (иначе).
- $x=3, y=3: xy=9$ (иначе).

- Граничные значения:

- $x=3, y=4: xy=12$ (иначе).
- $x=2, y=3: xy=6 < 7$.

- Особые случаи:

- $x=0, y=0: xy=0$ (иначе).
- $x=1, y=-1: xy=-1$ (иначе).

РЕЗУЛЬТАТ РАБОТЫ:

Линейное уравнение

Линейный Разветвляющийся

$$4. \quad w = |\cos x - \cos y|^{1+2\sin^2 y} * \left(1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4}\right).$$

Введите значение X:

Введите значение Y:

Введите значение Z:

1) $\cos(x) = \cos(6) = 0,960$
2) $\cos(y) = \cos(7) = 0,754$
3) $|\cos(x) - \cos(y)| = 0,206$
4) $\sin^2(y) = (\sin(7))^2 = 0,432$
5) Показат степени: $1 + 2 \times \sin^2(y) = 1,863$
6) $|\cos(x) - \cos(y)|^{(1 + 2 \times \sin^2(y))} = 0,053$
7) $1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4}$
 $= 1 + 8 + (64,000)/2 + (512,000)/3 + (4096,000)/4$
 $= 1 + 8 + 32,000 + 170,667 + 1024,000 = 1235,667$

Результат: $w = 0,053 \times 1235,667 = 65,239$
Округлённый результат: 65,24

```
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 Алгоритмы
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void label1_Click(object sender, EventArgs e)
        {
        }

        private void label2_Click(object sender, EventArgs e)
        {
        }

        private void label1_Click_1(object sender, EventArgs e)
        {
        }

        private void textBox4_TextChanged(object sender, EventArgs e)
        {
        }
    }
}
```



```

private void button1_Click(object sender, EventArgs e)
{
    // Очищаем поле результата
    textBoxR.Clear();

    try
    {
        // Получаем введённые значения, используется вещественный тип данных(десятичное
        число)//а так же преобразование строки в число
        double x = double.Parse(textBoxX.Text);
        double y = double.Parse(textBoxY.Text);
        double z = double.Parse(textBoxZ.Text);

        //1: Вычисляем cos(x) и cos(y), используется статический метод из класса Math
        double cosX = Math.Cos(x);
        double cosY = Math.Cos(y);
        textBoxR.AppendText($"1. cos(x) = cos({x}) = {cosX:F3}\r\n"); // добавление текста
        в конец //возвращение в начало строки и новая строка
        textBoxR.AppendText($"2. cos(y) = cos({y}) = {cosY:F3}\r\n"); //F3 - спецификатор,
        который покажет число с 3мя цифрами после запятой

        //2: |cos(x) - cos(y)|
        double resultSave = Math.Abs(cosX - cosY); //число джделается положительным, даже
        если оно отрицательное
        textBoxR.AppendText($"3. |cos(x) - cos(y)| = {resultSave:F3}\r\n"); //добавляет
        текст в конце

        //3: sin**2(y)
        double sin2Y = Math.Pow(Math.Sin(y), 2); //Math.Pow выводит первое число в степень
        второго
        textBoxR.AppendText($"4. sin**2(y) = (sin({y}))**2 = {sin2Y:F3}\r\n");

        //4: Показатель степени: 1 + 2*sin**2(y)
        double exponent = 1 + 2 * sin2Y;
        textBoxR.AppendText($"5. Показат степени: 1 + 2*sin**2(y) = {exponent:F3}\r\n");

        //5: Возводим |cos(x) - cos(y)| в степень
        double part1 = Math.Pow(resultSave, exponent); //число, которое будет исп для
        возведение в степень
        textBoxR.AppendText($"6. |cos(x) - cos(y)|**(1 + 2*sin**2(y)) = {part1:F3}\r\n");

        //6: Вычисляем 2 часть: 1 + z + z**2/2 + z**3/3 + z**4/4
        double z2 = Math.Pow(z, 2);
        double z3 = Math.Pow(z, 3);
        double z4 = Math.Pow(z, 4);
        double part2 = 1 + z + z2 / 2 + z3 / 3 + z4 / 4;

        textBoxR.AppendText($"7. 1 + z + z**2/2 + z**3/3 + z**4/4\r\n");
        textBoxR.AppendText($"      = 1 + {z} + ({z2:F3})/2 + ({z3:F3})/3 + ({z4:F3})/4\r\n");
        textBoxR.AppendText($"      = 1 + {z} + {z2 / 2:F3} + {z3 / 3:F3} + {z4 / 4:F3} =
        {part2:F3}\r\n");

        //7: Итог: w = part1 * part2
        double w = part1 * part2;
        textBoxR.AppendText($"      \r\n");
        textBoxR.AppendText($"Результат: w = {part1:F3} × {part2:F3} = {w:F3}\r\n");
        textBoxR.AppendText($"Округлённый результат: {w:F2}");
    }
    //обработка ошибок в программе
    catch (FormatException)
    {
        MessageBox.Show("Введите корректные числовые значения!", "Ошибка ввода",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Ошибка: {ex.Message}", "Ошибка расчёта", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}
}
}

```

Разветвляющиеся уравнение

С выбором функции Cos(x)

Линейный Разветвляющийся

$$3. c = \begin{cases} f^3(x) + \text{ctg}(y), & xy > 12 \\ \text{sh}(f^3(x)) + y^2, & xy < 7 \\ \cos(x - f^3(x)), & \text{иначе} \end{cases}$$

Введите значение X:

Введите значение Y:

Вычислить

Выберите функцию:

☒ Cos (x)

☐ Sqr (x)

☐ Ext (x)

☒ Ответ красным цветом

1) Введён X = 5
2) Введён Y = 6
3) Выбрана функция: cos(x)
4) $f(x) = 0,283662185463226$
5) Произведение $xy = 30$
6) Условие: $xy > 12$ – формула №1
7) Результат: -3,41352834329741

С выбором функции Sqr(x)

Линейный Разветвляющийся

$$3. c = \begin{cases} f^3(x) + \text{ctg}(y), & xy > 12 \\ \text{sh}(f^3(x)) + y^2, & xy < 7 \\ \cos(x - f^3(x)), & \text{иначе} \end{cases}$$

Введите значение X:

Введите значение Y:

Вычислить

Выберите функцию:

☐ Cos (x)

☒ Sqr (x)

☐ Ext (x)

☒ Ответ красным цветом

1) Введён X = 5
2) Введён Y = 6
3) Выбрана функция: sqr(x)
4) $f(x) = 25$
5) Произведение $xy = 30$
6) Условие: $xy > 12$ – формула №1
7) Результат: 15621,5636469958

С выбором функции Ext(x)

Линейный

Разветвляющийся

$$3. c = \begin{cases} f^3(x) + \text{ctg}(y), & xy > 12 \\ \text{sh}(f^3(x)) + y^2, & xy < 7 \\ \cos(x - f^3(x)), & \text{иначе} \end{cases}$$

Введите значение X:

Введите значение Y:

Вычислить

Выберите функцию:

☐ Cos (x)
 ☐ Sqr (x)
 ☒ Ext (x)

☒ Ответ красным цветом

1) Введён X = 5
 2) Введён Y = 6
 3) Выбрана функция: exp(x)
 4) f(x) = 148,413159102577
 5) Произведение xy = 30
 6) Условие: xy > 12 – формула №1
 7) Результат: 3269013,93611911

```
private void button2_Click(object sender, EventArgs e)
{
    richTextBox.Clear(); //чистит старый текст
    richTextBox.ForeColor = Color.Black; // обычный цвет для всех этапов

    // 1. Чтение X
    double x; //присвоили значение
    if (!double.TryParse(tbX.Text, out x)) // если НЕ получилось преобразовать текст в
    число ,выполнить блок if
    {
        richTextBox.Text = "1) Ошибка: неверное значение X";
        return; // выход из метода
    }
    richTextBox.AppendText("1) Введён X = " + x + "\r\n"); //добавить текст в конец
    //приклеиваем значение переменной x

    // 2. Чтение Y
    double y;
    if (!double.TryParse(tbY.Text, out y))
    {
        richTextBox.AppendText("2) Ошибка: неверное значение Y\r\n");
        return;
    }
    richTextBox.AppendText("2) Введён Y = " + y + "\r\n");

    // 3. Выбор функции f(x)
    double fx = 0; //начальное значение(на всякий)
    string func = ""; //строка запоминает, какую функцию выбрали

    if (rbCos.Checked)
    {
        fx = Math.Cos(x);
        func = "cos(x)";
    }
    else if (rbSqr.Checked)
    {

```

```

        fx = x * x;
        func = "sqr(x)";
    }
    else if (rbExt.Checked)
    {
        fx = Math.Exp(x);
        func = "exp(x)";
    }

    richTextBox.AppendText("3) Выбрана функция: " + func + "\r\n"); //func переменная с
    помощью которой запоминаем назв выбранной функции
    richTextBox.AppendText("4) f(x) = " + fx + "\r\n");

    // 4) Вычисление xy
    double xy = x * y;
    richTextBox.AppendText("5) Произведение xy = " + xy + "\r\n");

    //5) Основной алгоритм
    double c = 0; //объявляем переменную c

    if (xy > 12)
    {
        richTextBox.AppendText("6) Условие: xy > 12 – формула %1\r\n");

        double t = Math.Tan(y);
        if (Math.Abs(t) < 0.000001) //модуль
        {
            richTextBox.AppendText("7) Ошибка: ctg(y) не существует (tan(y)=0)\r\n");
            return;
        }

        c = Math.Pow(fx, 3) + 1 / t; // f^3(x) + ctg(y)
    }
    else if (xy < 7)
    {
        richTextBox.AppendText("6) Условие: xy < 7 – формула %2\r\n");

        c = Math.Sinh(Math.Pow(fx, 3)) + y * y; // sh(f^3) + y^2 //гиперболический синус
    }
    else
    {
        richTextBox.AppendText("6) Условие: 7 <= xy <= 12 – формула %3\r\n");

        c = Math.Cos(x - Math.Pow(fx, 3)); // cos(x - f^3)
    }

    // 6) Финальный вывод
    richTextBox.AppendText("7) Результат: ");

    // теперь делаем результат красным
    richTextBox.SelectionStart = richTextBox.Text.Length;
    richTextBox.SelectionColor = Color.Red;
    richTextBox.AppendText(c.ToString());
    richTextBox.SelectionColor = Color.Black;
    }

```

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чём заключается ручная отладка ПО?

Ручная отладка ПО - это процесс пошагового анализа программы для выявления и исправления ошибок. Включает:

- выполнение программы «вручную» (на бумаге или в уме) с конкретными входными данными;
- сравнение ожидаемых и фактических результатов;
- анализ логики работы программы;
- поиск и устранение ошибок (синтаксических, логических, ошибок выполнения).

2. На каком этапе проводится ручная отладка?

Ручная отладка проводится на этапе тестирования и отладки ПО, после написания кода. Она может выполняться:

- после реализации отдельных модулей для проверки их корректности;
- при обнаружении ошибок в работе программы;
- перед сдачей проекта для финального тестирования.

3. Опишите методы отладки.

Основные методы отладки:

- **ручное тестирование** - пошаговое выполнение программы с тестовыми наборами данных;
- **метод индукции** - анализ конкретных случаев для выявления общей закономерности ошибки;
- **метод дедукции** - логический анализ программы для исключения возможных причин ошибки;
- **обратное прослеживание** - анализ программы от места обнаружения ошибки к её источнику;
- **использование отладчика (debugger)** - пошаговое выполнение кода с просмотром значений переменных;
- **метод «грубой силы»** — детальный анализ каждого шага алгоритма с привлечением коллег для проверки логики.

Вывод: Делайте все задания от Александра Сергеевича, может, тогда у нас, у тупоголовых, хоть что-то отложится в голове.
И пусть... и пусть.