МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ)

ИНСТИТУТ ТРАНСПОРТНОЙ ТЕХНИКИ И СИСТЕМ УПРАВЛЕНИЯ

Лабораторная работа №3

по дисциплине «Методы программирования»

«Автоматизация компоновки ГПИ. Функции»

Выполнил: ст. гр. ТКИ-342

Никулин Д.В.

Ситало Р.В.

Проверил: доцент, к.т.н.

Сафронов А.И.

Москва – 2024 г

Оглавление

Цель работы

«Закрепить навыки разработки визуального пользовательского интерфейса, освоить работу с текстовыми файлами в среде *Microsoft Visual Studio*, научиться взаимно увязывать одни элементы управления с другими, получить представление о перерисовке и перемасштабировании».

Формулировка задачи

«В интегрированной среде разработки *Visual Studio* разработать программу в режиме *Windows Forms Application* на языке *Visual C#*, представляющую собой пользовательский интерфейс, содержащий главное меню, позволяющее:

1. Начать работу с приложением.
2. Прервать работу приложения.
3. Предоставить пользователю справочную информацию о работе с приложением.

Сама программа должна реализовывать вывод в списки значений аргумента и соответствующих им значений функций. Список функций должен обязательно содержать следующие пункты:

1. Логарифм по основанию 2,
2. Тангенс,
3. Возведение в квадрат,
4. Косинус,
5. Натуральный логарифм,
6. Арккосинус,
7. Извлечение корня,
8. Арктангенс,
9. Синус,
10. Десятичный логарифм,
11. Арксинус.

Индивидуальная задача

Варианты разделителей - &*

Варианты цветового оформления подложки – *AppWorkspace*

Знаки после запятой - *(N mod 5) + 1 = 4*

Индивидуальная функция - $((1 – x) / (1 + x))^{4/5}$

Порядок функция - 10, 9, 11, 8, 6, 5, 7, 4, 3, 2, 1.

Диаграммы классов, входящих в состав решения

Рисунок 1 – Диаграмма классов

Сеть Петри

Полная сеть Петри

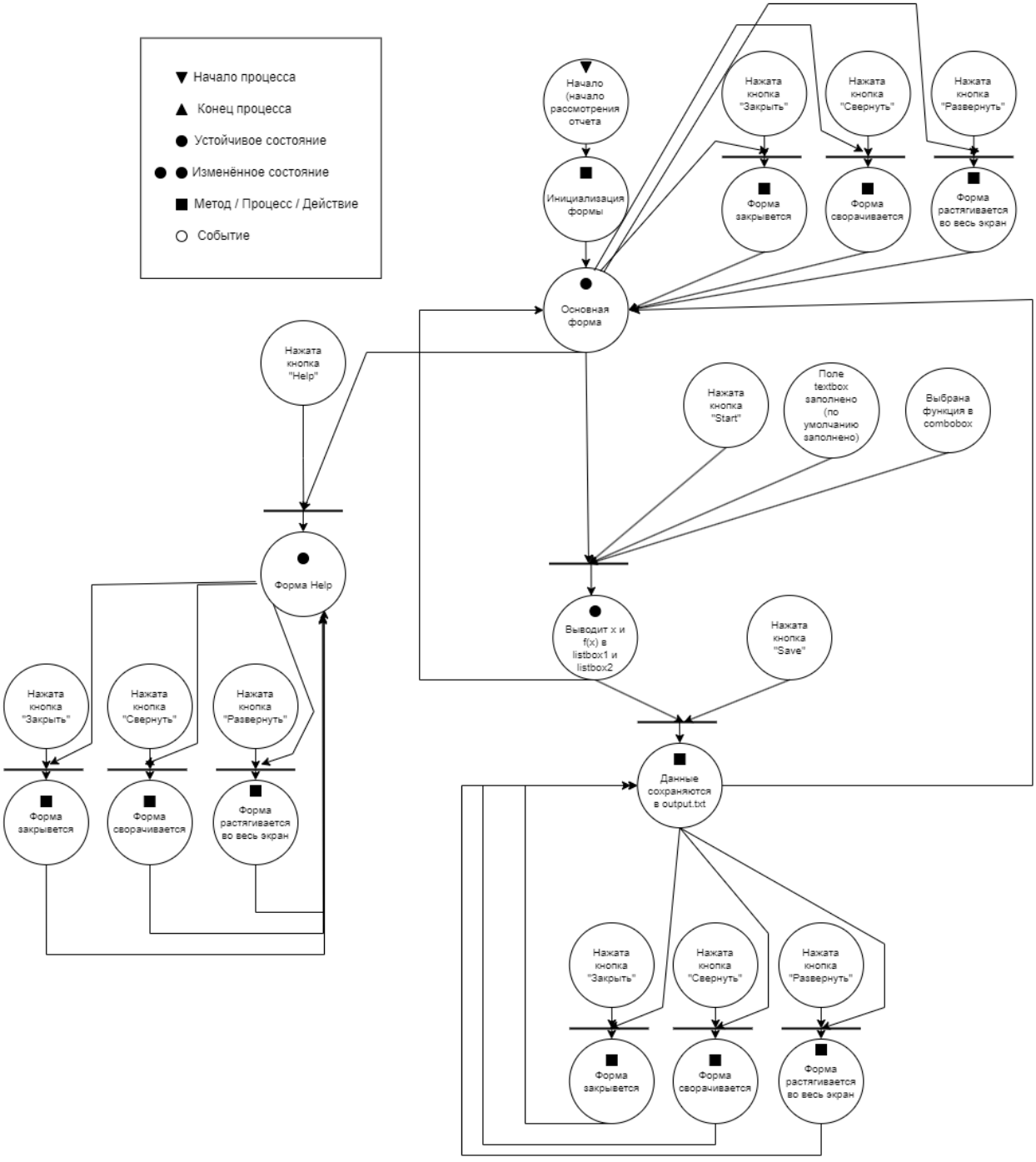

Рисунок 2 – Полная сеть Петри

Скриншоты работы программы

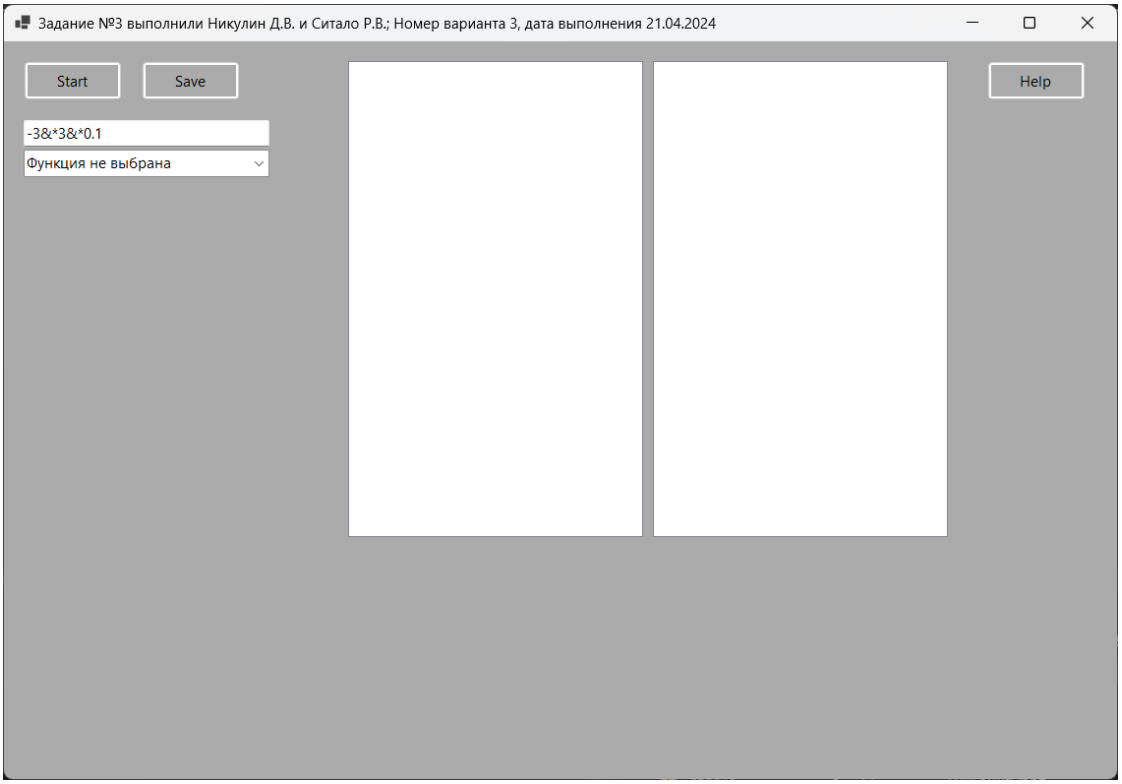

Рисунок 3 – Начальная форма

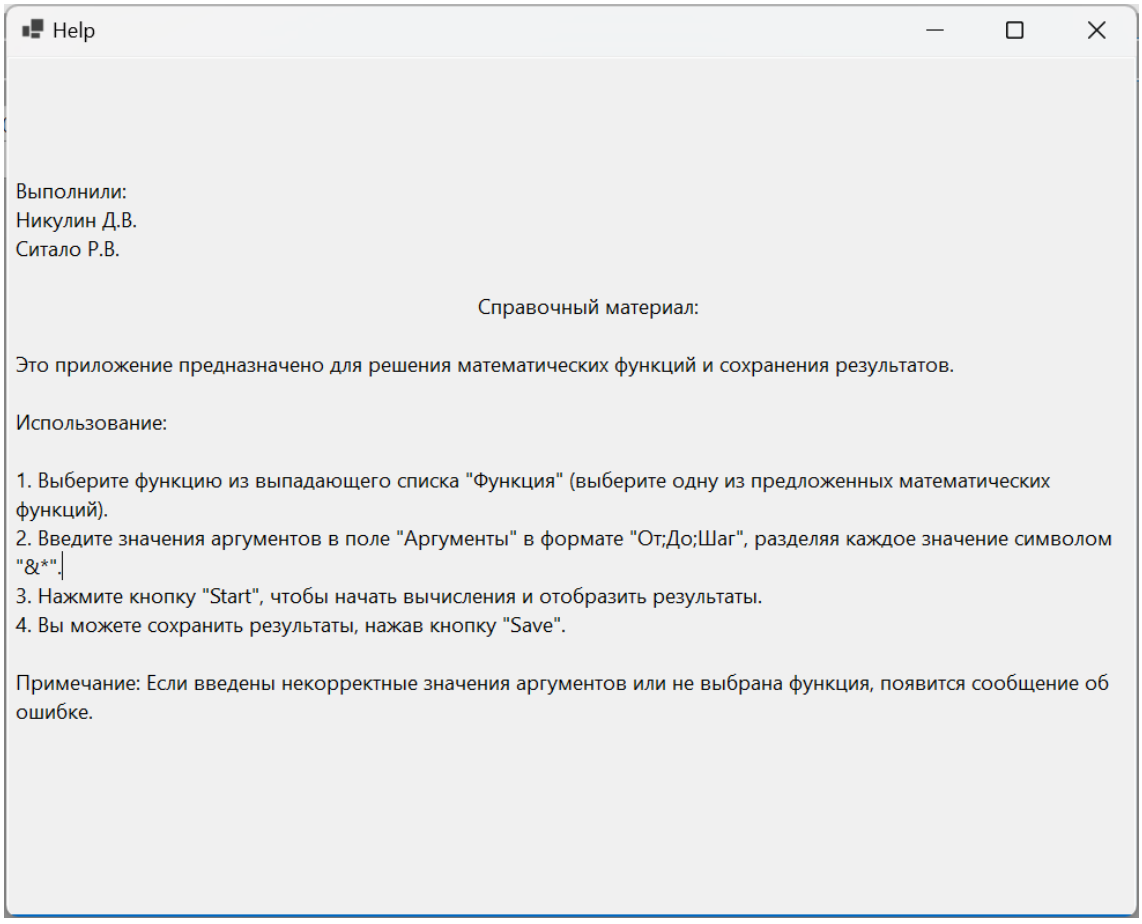

Рисунок 4 – Форма помощи

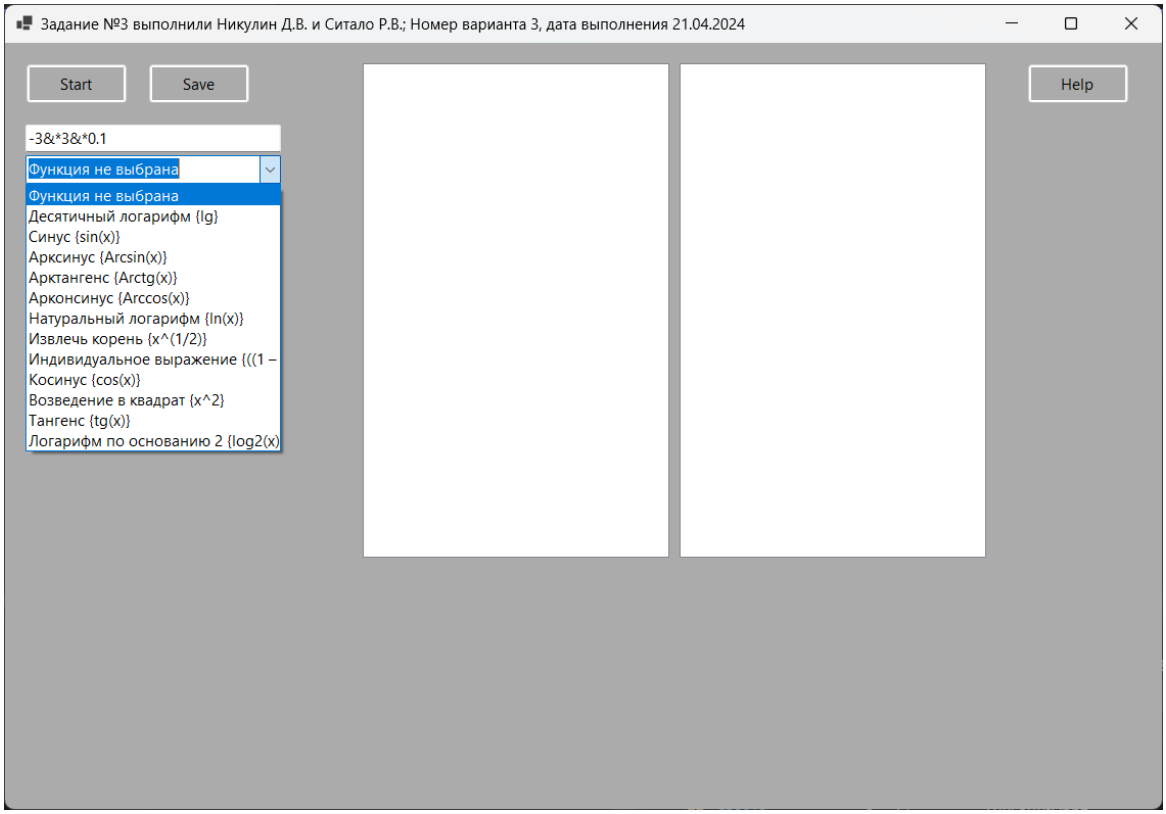

Рисунок 5 – Выбор функции

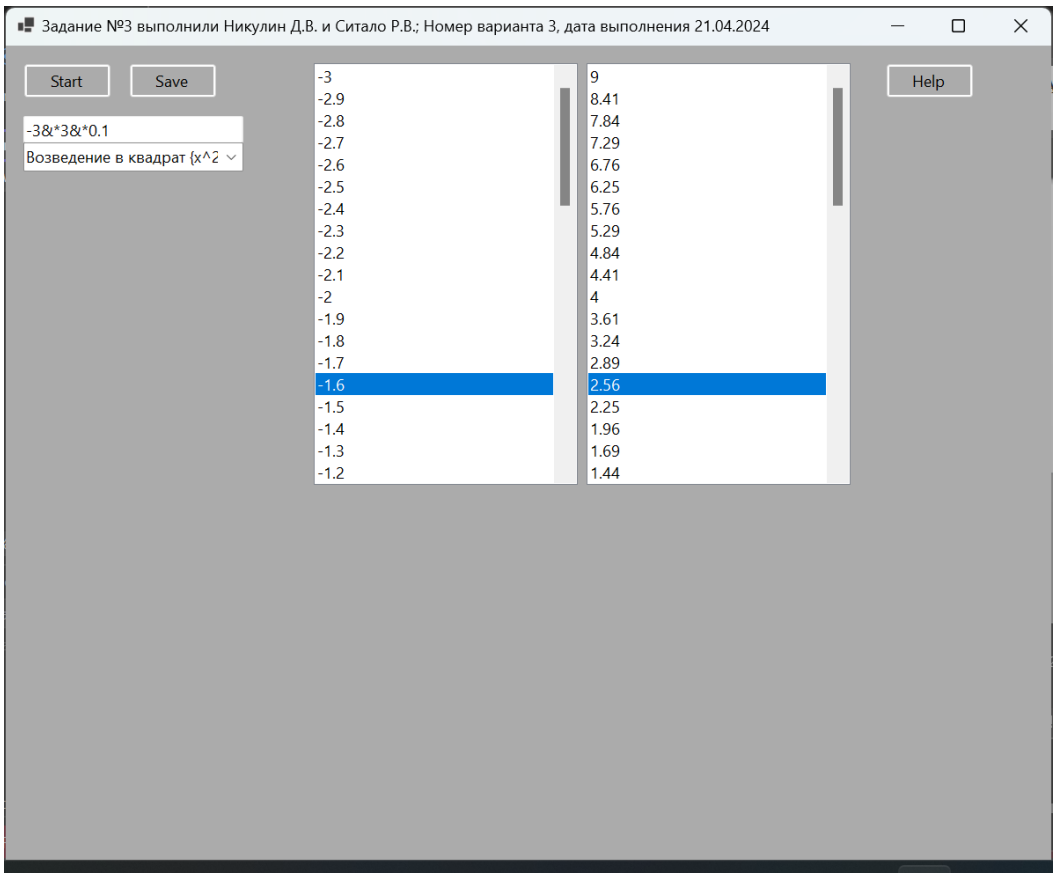

Рисунок 6 – Возведение в квадрат

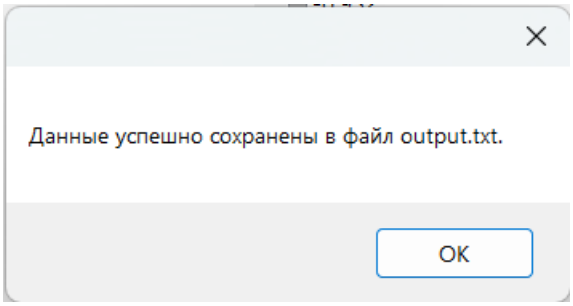

Рисунок 7 – Сохранение в файл

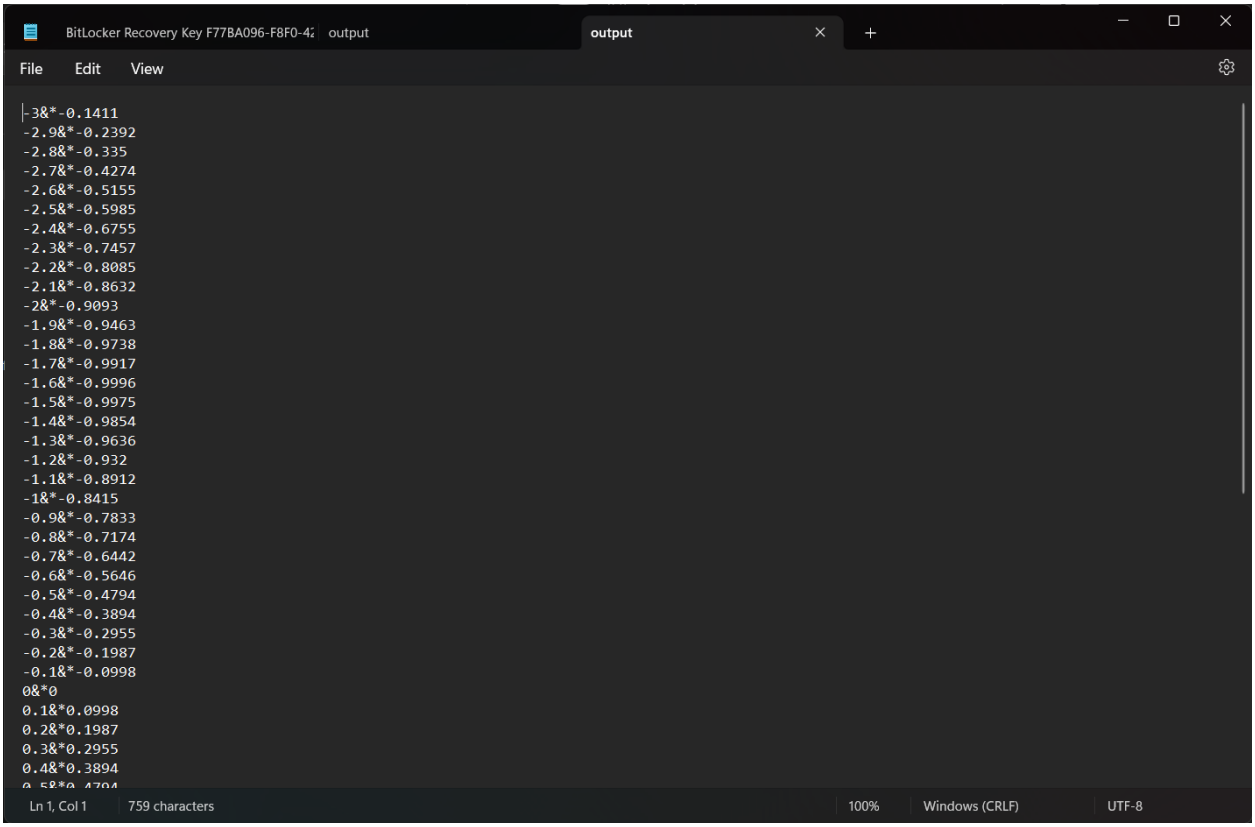

Рисунок 8 – Файл output.txt

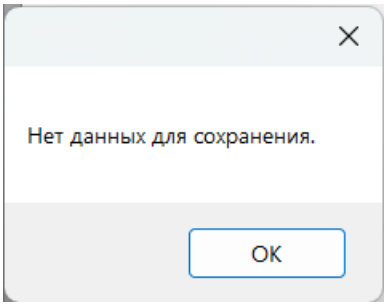

Рисунок 9 – Ошибка “Нет данных для сохранения”

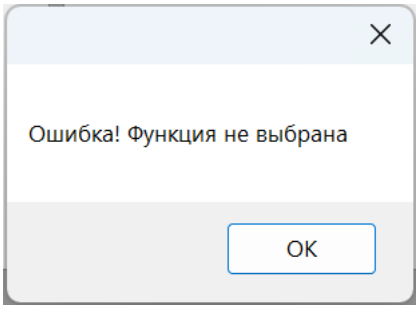

Рисунок 10 – Ошибка “Функция не выбрана”

Код программы

```
namespace WinFormsApp2
{
    public partial class MainForm : Form
    {

        // Массив доступных функций
        string[] func_arr = {
            "Функция не выбрана",
            "Десятичный логарифм {lg}",
            "Синус {sin(x)}",
            "Арксинус {Arcsin(x)}",
            "Арктангенс {Arctg(x)}",
            "Арконсинус {Arccos(x)}",
            "Натуральный логарифм {ln(x)}",
            "Извлечь корень {x^(1/2)}",
            "Индивидуальное выражение {((1 − x) / (1 + x))^(4/5)}",
            "Косинус {cos(x)}",
            "Возведение в квадрат {x^2}",
            "Тангенс {tg(x)}",
            "Логарифм по основанию 2 {log2(x)}"
        };

        // Выбранный индекс в комбобоксе
        int index_combobox = 0;

        // Флаг ошибки
        bool flag_error = false;

        // Значения аргументов по умолчанию
        double[] default_arguments = { −3, 3, 0.1 };

        // Инструменты пользовательского интерфейса
        ComboBox functionscmbbx; // Комбобокс с функциями
        Button startbtn; // Кнопка "Start"
        Button helpbtn; // Кнопка "Help"
```

```
        Button savebtn; // Кнопка "Save"
        ListBox argumentlstbx; // Список аргументов
        ListBox answerlstbx; // Список результатов
        TextBox input_argbx; // Текстовое поле для ввода аргументов



        public MainForm()
        {
            // Инициализация формы
            InitializeComponent();
            this.Text = "Задание №3 выполнили Никулин Д.В. и Ситало Р.В.; Номер
варианта 3, дата выполнения 21.04.2024";
            this.MinimumSize = new Size(950, 600);
        }


        private void ShowAllTools(object sender, EventArgs e)
        {
            // Код для добавления всех инструментов на форму
            //
            // startbtn
            //
            startbtn = new Button();
            this.Controls.Add(startbtn);
            startbtn.Click += new EventHandler(startbtnClick);
            startbtn.Text = "Start";
            startbtn.Location = new Point(20, 20);
            startbtn.Size = new Size(100, 40);
            startbtn.ForeColor = Color.Black;
            //
            // help
            //
            helpbtn = new Button();
            this.Controls.Add(helpbtn);
            helpbtn.Click += new EventHandler(helpbtnCliclk);
            helpbtn.Text = "Help";
            helpbtn.Location = new Point(1000, 20);
            helpbtn.Size = new Size(100, 40);
```

```
            helpbtn.ForeColor = Color.Black;
            //
            // savebtn
            //
            savebtn = new Button();
            this.Controls.Add(savebtn);
            savebtn.Click += new EventHandler(savebtnClick);
            savebtn.Text = "Save";
            savebtn.Location = new Point(140, 20);
            savebtn.Size = new Size(100, 40);
            savebtn.ForeColor = Color.Black;
            //
            // functionslstbx
            //
            functionscmbbx = new ComboBox();
            this.Controls.Add(functionscmbbx);
            functionscmbbx.Location = new Point(20, 110);
            functionscmbbx.Size = new Size(250, 20);
            foreach (string str in func_arr)
            {
                functionscmbbx.Items.Add(str);
            }
            functionscmbbx.SelectedIndex = 0;
            functionscmbbx.SelectedIndexChanged += new
EventHandler(FuncSelectedIndexChanged);
            //
            // argumentlstbx
            //
            argumentlstbx = new ListBox();
            this.Controls.Add(argumentlstbx);
            argumentlstbx.Location = new Point(350, 20);
            argumentlstbx.Size = new Size(300, 500);
            argumentlstbx.SelectedIndexChanged += new
EventHandler(ArgumentListSelectedIndexChanged);
            //
            // answerlstbx
            //
```

```
            answerlstbx = new ListBox();
            this.Controls.Add(answerlstbx);
            answerlstbx.Location = new Point(660, 20);
            answerlstbx.Size = new Size(300, 500);
            answerlstbx.SelectedIndexChanged += new
EventHandler(AnswerListSelectedIndexChanged);
            //
            // input_argbx
            //
            input_argbx = new TextBox();
            this.Controls.Add(input_argbx);
            input_argbx.Location = new Point(20, 80);
            input_argbx.Size = new Size(250, 20);
            input_argbx.Text = default_arguments[0].ToString() + "&*" +
default_arguments[1].ToString() + "&*" + default_arguments[2].ToString();
            input_argbx.TextChanged += new EventHandler(inputArgbxTextChanged);



            // Pin форм разным по сторонам
            startbtn.Anchor = AnchorStyles.Left | AnchorStyles.Top;
            savebtn.Anchor = AnchorStyles.Left | AnchorStyles.Top;
            helpbtn.Anchor = AnchorStyles.Right | AnchorStyles.Top;
            input_argbx.Anchor = AnchorStyles.Left | AnchorStyles.Top;
            argumentlstbx.Anchor = AnchorStyles.Right | AnchorStyles.Top;
            answerlstbx.Anchor = AnchorStyles.Right | AnchorStyles.Top;
            functionscmbbx.Anchor = AnchorStyles.Left | AnchorStyles.Top;
        }

        // Обработчик изменения выбранной функции
        private void ArgumentListSelectedIndexChanged(object sender, EventArgs e)
        {
            answerlstbx.SelectedIndex = argumentlstbx.SelectedIndex;

        }

        // Обработчик изменения текста в поле аргументов
        private void AnswerListSelectedIndexChanged(object sender, EventArgs e)
```

```
{
    argumentlstbx.SelectedIndex = answerlstbx.SelectedIndex;
}

private void FuncSelectedIndexChanged(object sender, EventArgs e)
{
    index_combobox = functionscmbbx.SelectedIndex;
}

// Обработчик текстовой формы для диапазона аргументов
private void inputArgbxTextChanged(object sender, EventArgs e)
{
    string[] words = input_argbx.Text.Split("&*");
    default_arguments = new double[3];
    int i = 0;
    flag_error = false;
    try
    {
        foreach (string word in words)
        {
            default_arguments[i] = double.Parse(word);
            i++;
        }
    }
    catch (Exception ex)
    {
        flag_error = true;
    }

}

// Обработчик клика по кнопке "Start"
private void startbtnClick(object sender, EventArgs e)
{
    if (flag_error || index_combobox == 0)
    {
        MessageBox.Show("Ошибка! Функция не выбрана");
```

```
            }
            else
            {
                argumentlstbx.Items.Clear();
                answerlstbx.Items.Clear();
                for (double i = default_arguments[0]; i <= default_arguments[1]; i
+= default_arguments[2])
                {
                    argumentlstbx.Items.Add(Math.Round(i, 4));
                    answerlstbx.Items.Add(Math.Round(SolveFunc(i), 4));
                }
            }

        }

        // Обработчик клика по кнопке "Help"
        private void helpbtnCliclk(object sender, EventArgs e)
        {

            Form help_window = new Form();
            help_window.Text = "Help";
            help_window.Width = 1000;
            help_window.Height = 800;

            TextBox helpTextBox = new TextBox();
            helpTextBox.Multiline = true;
            helpTextBox.Dock = DockStyle.Fill;
            helpTextBox.ReadOnly = true;

            // Читаем содержимое файла справки
            string helpFilePath = "help.txt";
            if (File.Exists(helpFilePath))
            {
                string helpText = File.ReadAllText(helpFilePath);
                helpTextBox.Text = helpText;
            }
            else
```

```
    {
        helpTextBox.Text = "File not found: help.txt";
    }

    help_window.Controls.Add(helpTextBox);
    help_window.ShowDialog();

}

// Обработчик клика по кнопке "Save"
private void savebtnClick(object sender, EventArgs e)
{
    if (argumentlstbx.Items.Count == 0 || answerlstbx.Items.Count == 0)
    {
        MessageBox.Show("Нет данных для сохранения.");
        return;
    }

    try
    {
        using (StreamWriter writer = new StreamWriter("output.txt"))
        {
            for (int i = 0; i < argumentlstbx.Items.Count; i++)
            {
                string argument = argumentlstbx.Items[i].ToString();
                string result = answerlstbx.Items[i].ToString();
                writer.WriteLine($"{argument}&*{result}");
            }
        }
        MessageBox.Show("Данные успешно сохранены в файл output.txt.");
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Ошибка при сохранении данных: {ex.Message}");
    }
}
```

```
// Выбор функции по индексу
private double SolveFunc(double x)
{
    switch (index_combobox)
    {
        case 1:
            return Math.Log10(x);
        case 2:
            return Math.Sin(x);
        case 3:
            return Math.Asin(x);
        case 4:
            return Math.Atan(x);
        case 5:
            return Math.Acos(x);
        case 6:
            return Math.Log(x);
        case 7:
            return Math.Sqrt(x);
        case 8:
            return Math.Pow(((1.0 - x) / (1.0 + x)), 0.8);
        case 9:
            return Math.Cos(x);
        case 10:
            return x * x;
        case 11:
            return Math.Tan(x);
        case 12:
            return Math.Log2(x);
        default:
            throw new ArgumentException("Недопустимое значение");
    }
}



// Метод загрузки формы после добавления всех инструментов
```

```
        private void LoadMainForm(object sender, EventArgs e)
        {
            InitializeComponent();
            this.Text = "Задание №3 выполнили Никулин Д.В. и Ситало Р.В.; Номер
варианта 3, дата выполнения 21.04.2024";
            this.MinimumSize = new Size(1150, 600);
            this.Shown += ShowAllTools;
            this.Cursor = Cursors.Hand;
            this.BackColor = SystemColors.AppWorkspace;
        }



    }
}
```

Вывод

В ходе выполнения лабораторной работы были закреплены навыки разработки визуального пользовательского интерфейса с использованием среды разработки Microsoft Visual Studio. Была освоена работа с текстовыми файлами, что позволило усовершенствовать навыки работы с файловой системой. Также было изучено взаимное увязывание элементов управления для создания более удобного и интуитивно понятного интерфейса.

Используя полученные знания, была реализована программа на языке Visual C#, представляющая собой пользовательский интерфейс, позволяющий выполнять ряд математических операций. В ходе работы были реализованы основные функции, а также индивидуальная функция, обеспечивающая широкие возможности для пользовательского ввода и анализа данных.