



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.....	3
Формулировка задачи	3
Индивидуальная задача	4
Диаграммы классов, входящих в состав решения	5
Сеть Петри	6
Полная сеть Петри	6
Краткая сеть Петри.....	7
Скриншоты работы программы.....	9
Код программы.....	12
Вывод	17

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 - $(N \bmod 5) + 1 = 4$

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

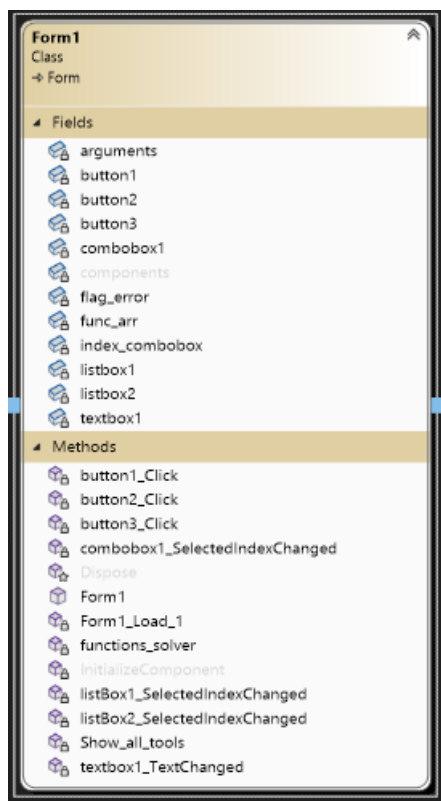


Рисунок 1 – Диаграмма классов

Сеть Петри

Полная сеть Петри

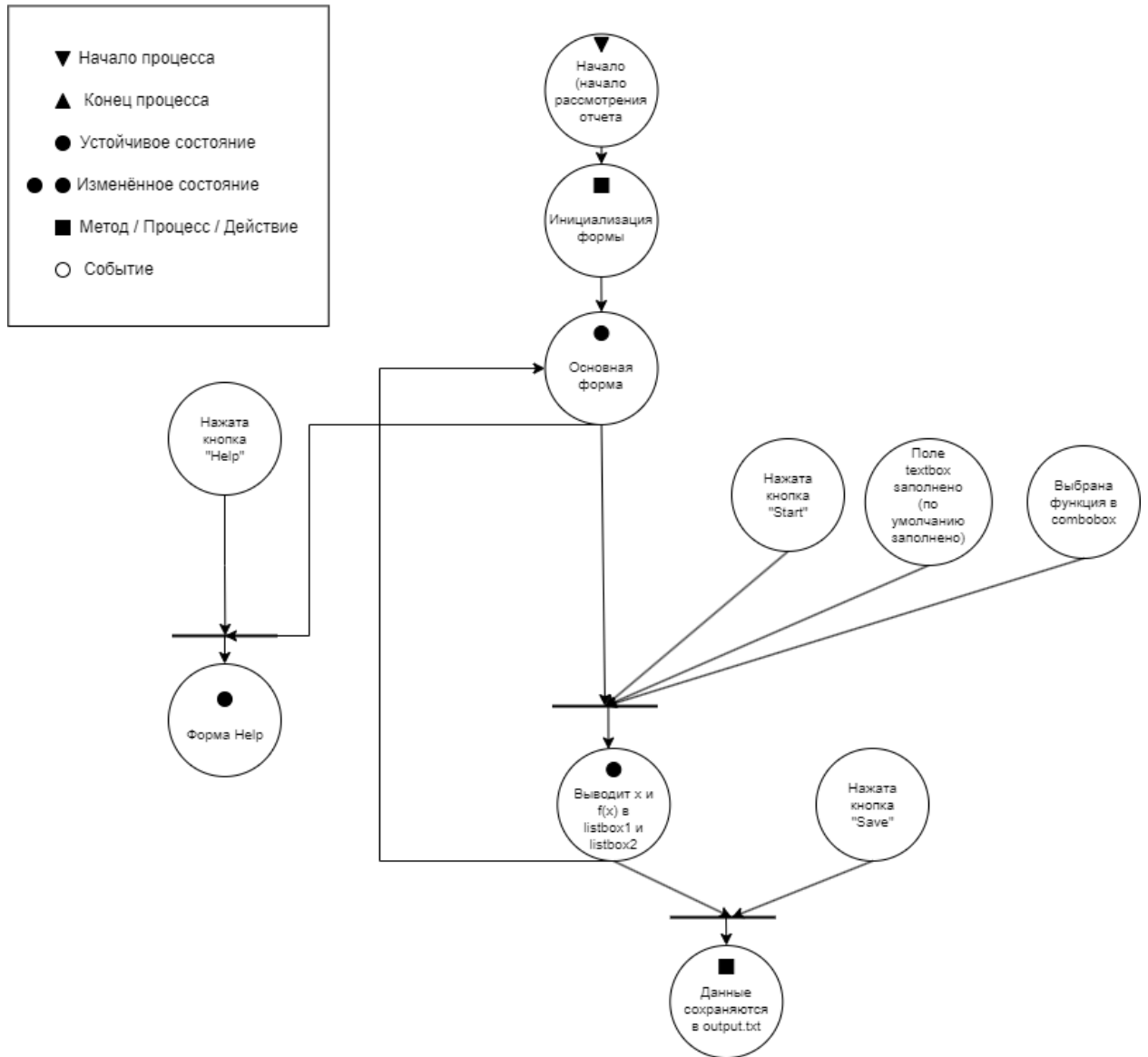


Рисунок 2 – Полная сеть Петри

Краткая сеть Петри

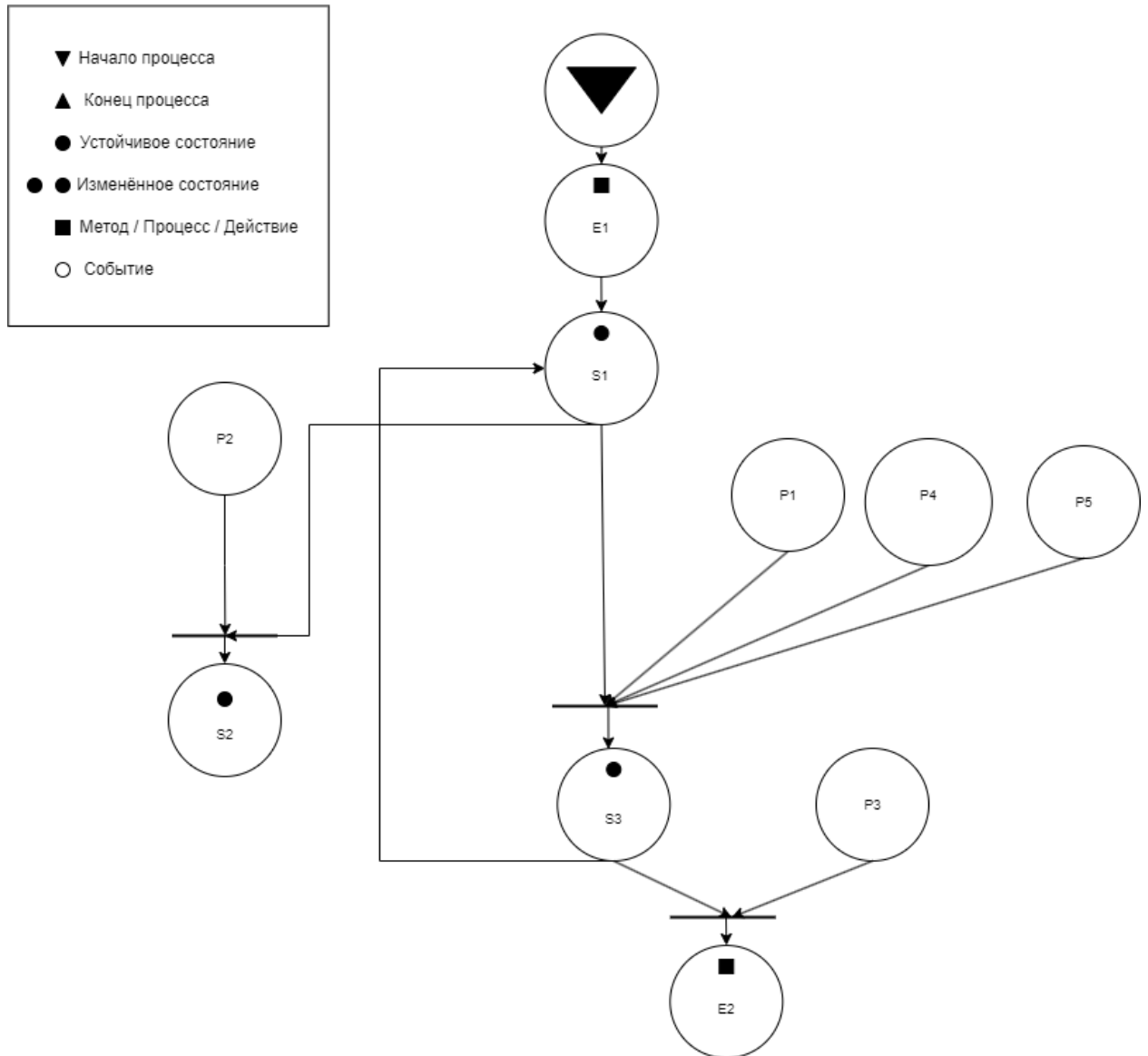


Рисунок 3 – Краткая сеть Петри

Описание:

– состояния (states):

s1 – Основная форма

s2 – Форма Help

s3 – Выводит x и $f(x)$ в listbox1 и listbox2

– действия (effects):

e1 – Инициализация формы

e2 – Данные сохраняются в output.txt

– события (prompts):

p1 – кнопка “Start”

p2 – кнопка “Help”

p3 – Кнопка “Save”

p4 – Поле textbox заполнено (по умолчанию заполнено)

p5 – Выбрана функция в combobox

Скриншоты работы программы

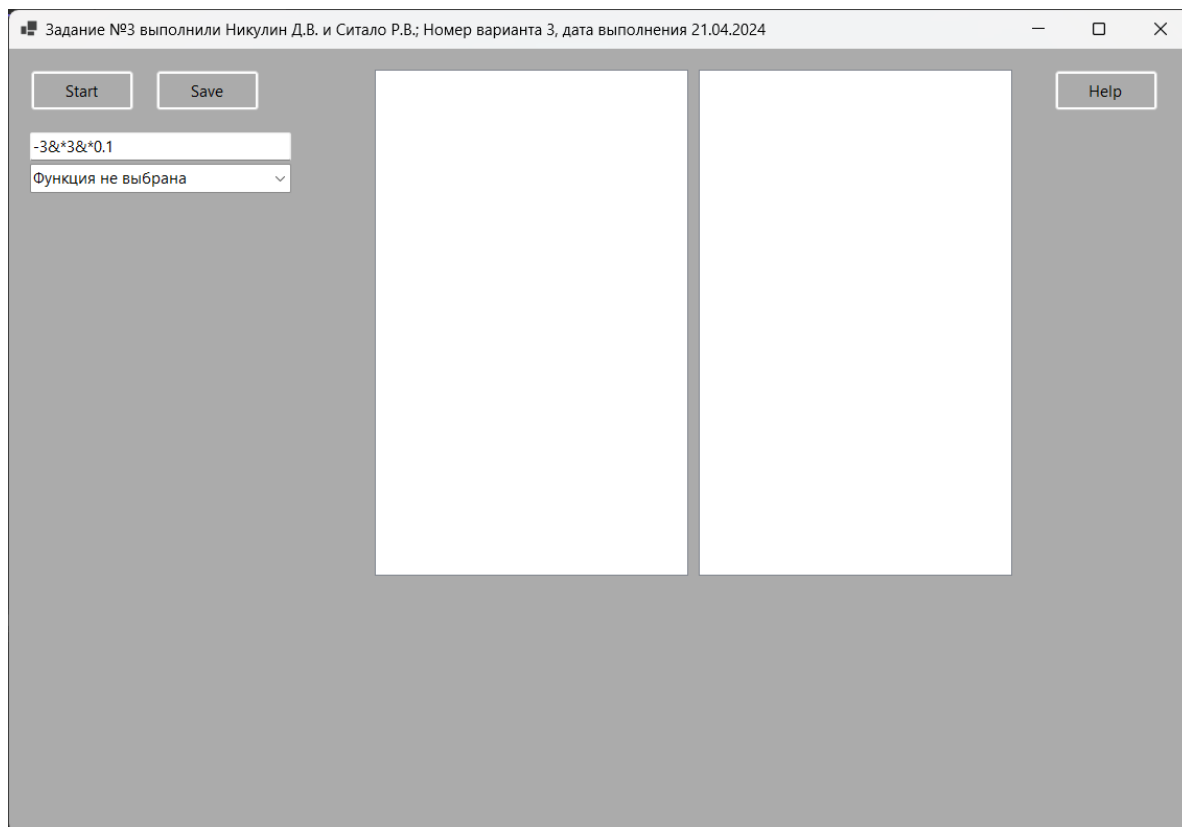


Рисунок 4 – Начальная форма

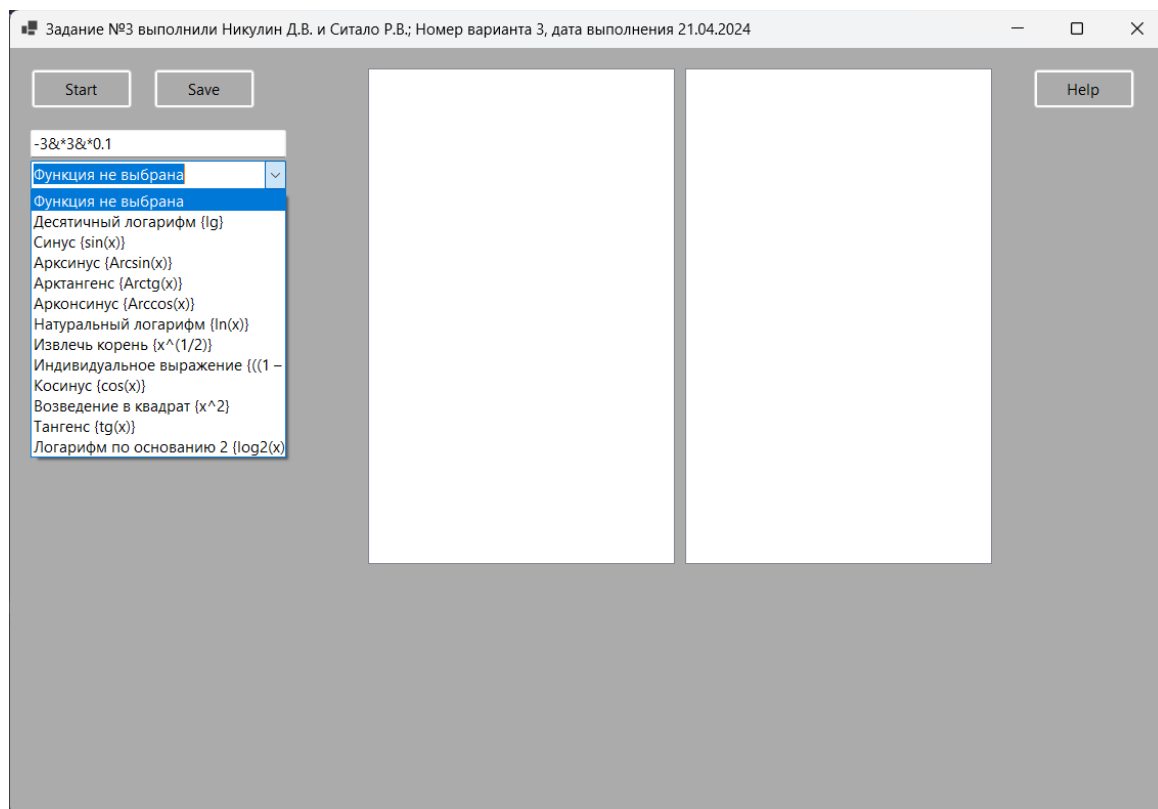


Рисунок 5 – Выбор функции

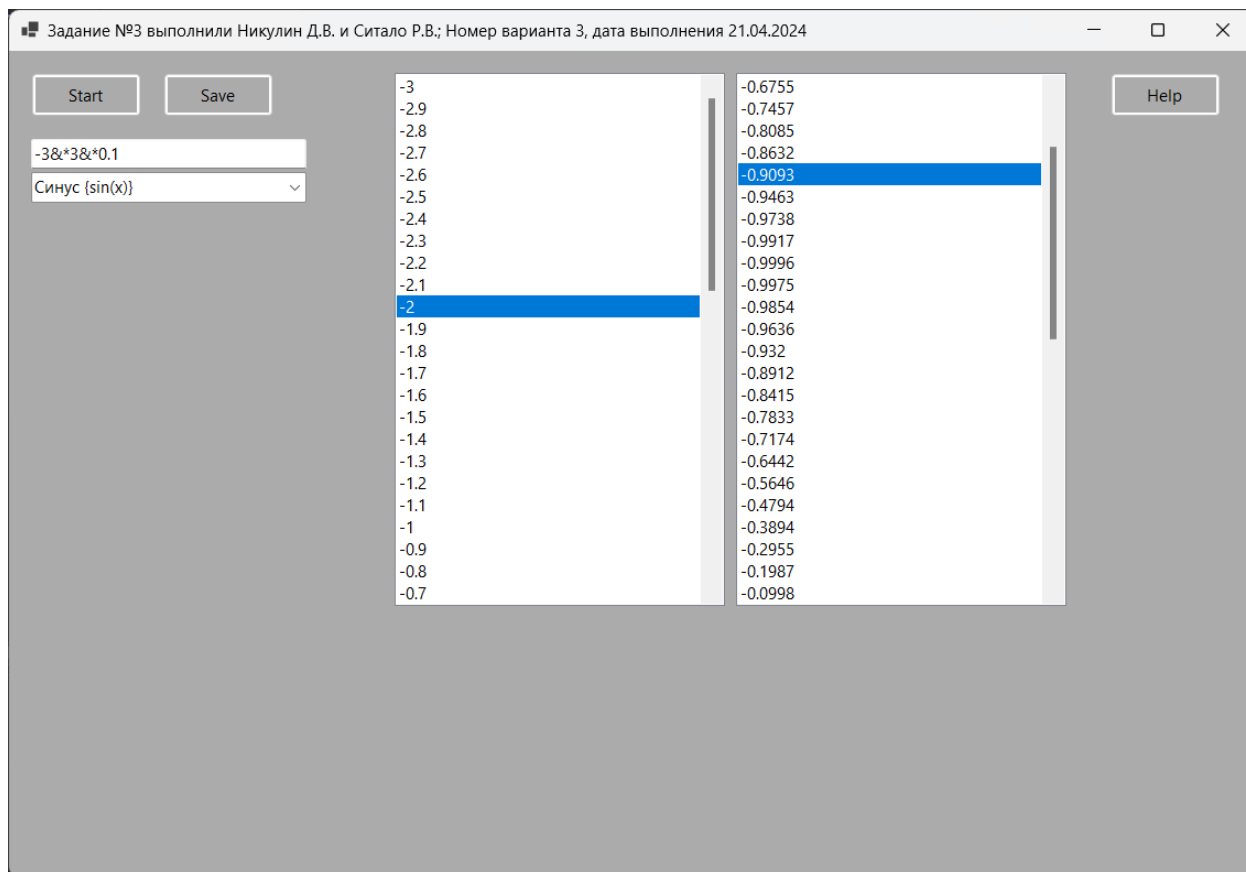


Рисунок 6 – Синус

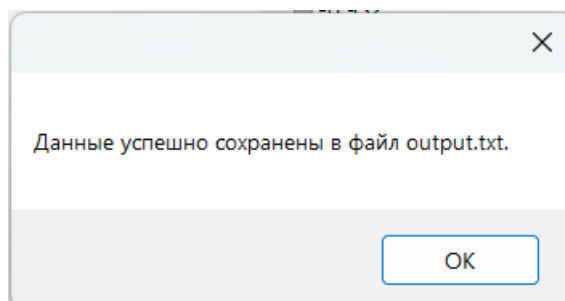


Рисунок 7 – Сохранение в файл

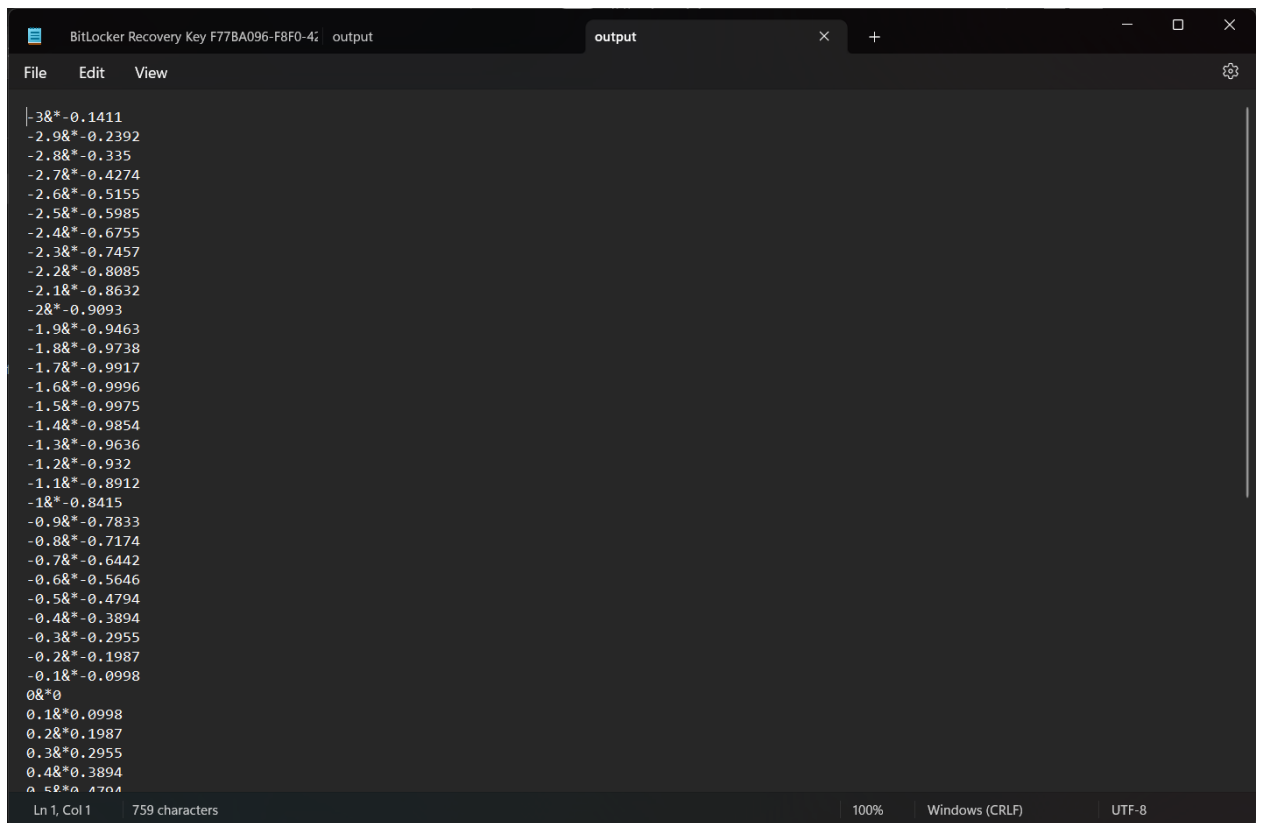


Рисунок 8 – Файл output.txt

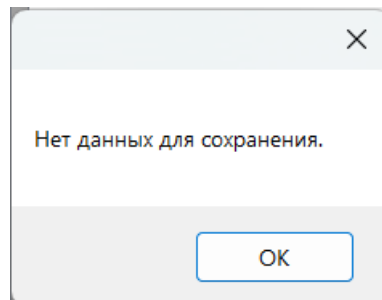


Рисунок 9 – Ошибка “Нет данных для сохранения”

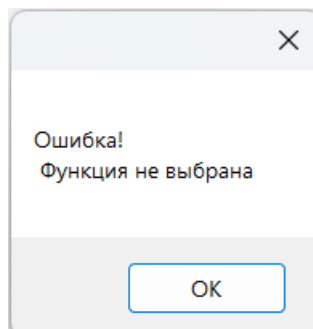


Рисунок 10 – Ошибка “Функция не выбрана”

Код программы

```
using System.Windows.Forms;
using static System.Runtime.InteropServices.JavaScript.JSType;

namespace WinFormsApp2
{
    public partial class Form1 : Form
    {
        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)}"
        };

        int index_combobox = 0;
        bool flag_error = false;
        double[] arguments = { -3, 3, 0.1 };

        // Tools
        ComboBox combobox1;
        Button button1;
        Button button2;
        Button button3;
        ListBox listbox1;
        ListBox listbox2;
        TextBox textbox1;

        public Form1()
        {
            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 Show_all_tools(object sender, EventArgs e)
        {
            //
            // button1
            //
            button1 = new Button();
            this.Controls.Add(button1);
            button1.Click += new EventHandler(button1_Click);
            button1.Text = "Start";
            button1.Location = new Point(20, 20);
            button1.Size = new Size(100, 40);
            button1.ForeColor = Color.Black;
            //
            // button2
            //
            button2 = new Button();
```

```

this.Controls.Add(button2);
button2.Click += new EventHandler(button2_Click);
button2.Text = "Help";
button2.Location = new Point(1000, 20);
button2.Size = new Size(100, 40);
button2.ForeColor = Color.Black;
//
// button3
//
button3 = new Button();
this.Controls.Add(button3);
button3.Click += new EventHandler(button3_Click);
button3.Text = "Save";
button3.Location = new Point(140, 20);
button3.Size = new Size(100, 40);
button3.ForeColor = Color.Black;
//
// combobox1
//
combobox1 = new ComboBox();
this.Controls.Add(combobox1);
combobox1.Location = new Point(20, 110);
combobox1.Size = new Size(250, 20);
foreach (string str in func_arr)
{
    combobox1.Items.Add(str);
}
combobox1.SelectedIndex = 0;
combobox1.SelectedIndexChanged += new
EventHndler(combobox1_SelectedIndexChanged);
//
// listbox1
//
listbox1 = new ListBox();
this.Controls.Add(listbox1);
listbox1.Location = new Point(350, 20);
listbox1.Size = new Size(300, 500);
listbox1.SelectedIndexChanged += new
EventHndler(listBox1_SelectedIndexChanged);
//
// listbox2
//
listbox2 = new ListBox();
this.Controls.Add(listbox2);
listbox2.Location = new Point(660, 20);
listbox2.Size = new Size(300, 500);
listbox2.SelectedIndexChanged += new
EventHndler(listBox2_SelectedIndexChanged);
//
// textbox1
//
textbox1 = new TextBox();
this.Controls.Add(textbox1);
textbox1.Location = new Point(20, 80);
textbox1.Size = new Size(250, 20);
textbox1.Text = arguments[0].ToString() + "&*" + arguments[1].ToString()
+ "&*" + arguments[2].ToString();
textbox1.TextChanged += new EventHandler(textbox1_TextChanged);

// Pin tools to sides
button1.Anchor = AnchorStyles.Left | AnchorStyles.Top;
button3.Anchor = AnchorStyles.Left | AnchorStyles.Top;
button2.Anchor = AnchorStyles.Right | AnchorStyles.Top;
textbox1.Anchor = AnchorStyles.Left | AnchorStyles.Top;

```

```

        listbox1.Anchor = AnchorStyles.Right | AnchorStyles.Top;
        listbox2.Anchor = AnchorStyles.Right | AnchorStyles.Top;
        combobox1.Anchor = AnchorStyles.Left | AnchorStyles.Top;
    }

    private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
    {
        listbox2.SelectedIndex = listbox1.SelectedIndex;
    }
    private void listBox2_SelectedIndexChanged(object sender, EventArgs e)
    {
        listbox1.SelectedIndex = listbox2.SelectedIndex;
    }

    private void combobox1_SelectedIndexChanged(object sender, EventArgs e)
    {
        index_combobox = combobox1.SelectedIndex;
    }

    // From;To;Step
    private void textbox1_TextChanged(object sender, EventArgs e)
    {
        string[] words = textbox1.Text.Split("&*");
        arguments = new double[3];
        int i = 0;
        flag_error = false;
        try
        {
            foreach (string word in words)
            {
                arguments[i] = double.Parse(word);

                i++;
            }
        }
        catch (Exception ex)
        {
            flag_error = true;
        }
    }

    // Start button
    private void button1_Click(object sender, EventArgs e)
    {
        if (flag_error || index_combobox == 0)
        {
            MessageBox.Show("Ошибка!\n Функция не выбрана");
        }
        else
        {
            listbox1.Items.Clear();
            listbox2.Items.Clear();
            for (double i = arguments[0]; i <= arguments[1]; i += arguments[2])
            {
                listbox1.Items.Add(Math.Round(i, 4));
                listbox2.Items.Add(Math.Round(functions_solver(i), 4));
            }
        }
    }

```

```

}

// Help button
private void button2_Click(object sender, EventArgs e)
{
    Form help_window = new Form();
    help_window.Text = "Help";
    help_window.Width = 500;
    help_window.Height = 500;

    Label label1 = new Label();
    label1.Location = new System.Drawing.Point(300, 100);
    help_window.Controls.Add(label1);
    label1.Width = 270;
    label1.Height = 40;
    label1.Text = "Help Window";
    label1.ForeColor = Color.Black;
    label1.Location = new Point(10, 10);
    help_window.ShowDialog();
}

// Save button
private void button3_Click(object sender, EventArgs e)
{
    if (listbox1.Items.Count == 0 || listbox2.Items.Count == 0)
    {
        MessageBox.Show("Нет данных для сохранения.");
        return;
    }

    try
    {
        using (StreamWriter writer = new StreamWriter("output.txt"))
        {
            for (int i = 0; i < listbox1.Items.Count; i++)
            {
                string argument = listbox1.Items[i].ToString();
                string result = listbox2.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}");
    }
}

// List of functions to solve
private double functions_solver(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);
    }
    return 0;
}

```

```

// Load after all tools
private void Form1_Load_1(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 += Show_all_tools;
    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.