

**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](#_Toc165376936)

[Формулировка задачи 3](#_Toc165376937)

[Индивидуальная задача 4](#_Toc165376938)

[Диаграммы классов, входящих в состав решения 5](#_Toc165376939)

[Сеть Петри 6](#_Toc165376940)

[Полная сеть Петри 6](#_Toc165376941)

[Скриншоты работы программы 7](#_Toc165376942)

[Код программы 11](#_Toc165376943)

[Вывод 16](#_Toc165376944)

# 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

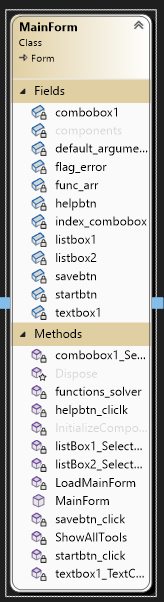


Рисунок 1 – Диаграмма классов

# Сеть Петри

## Полная сеть Петри

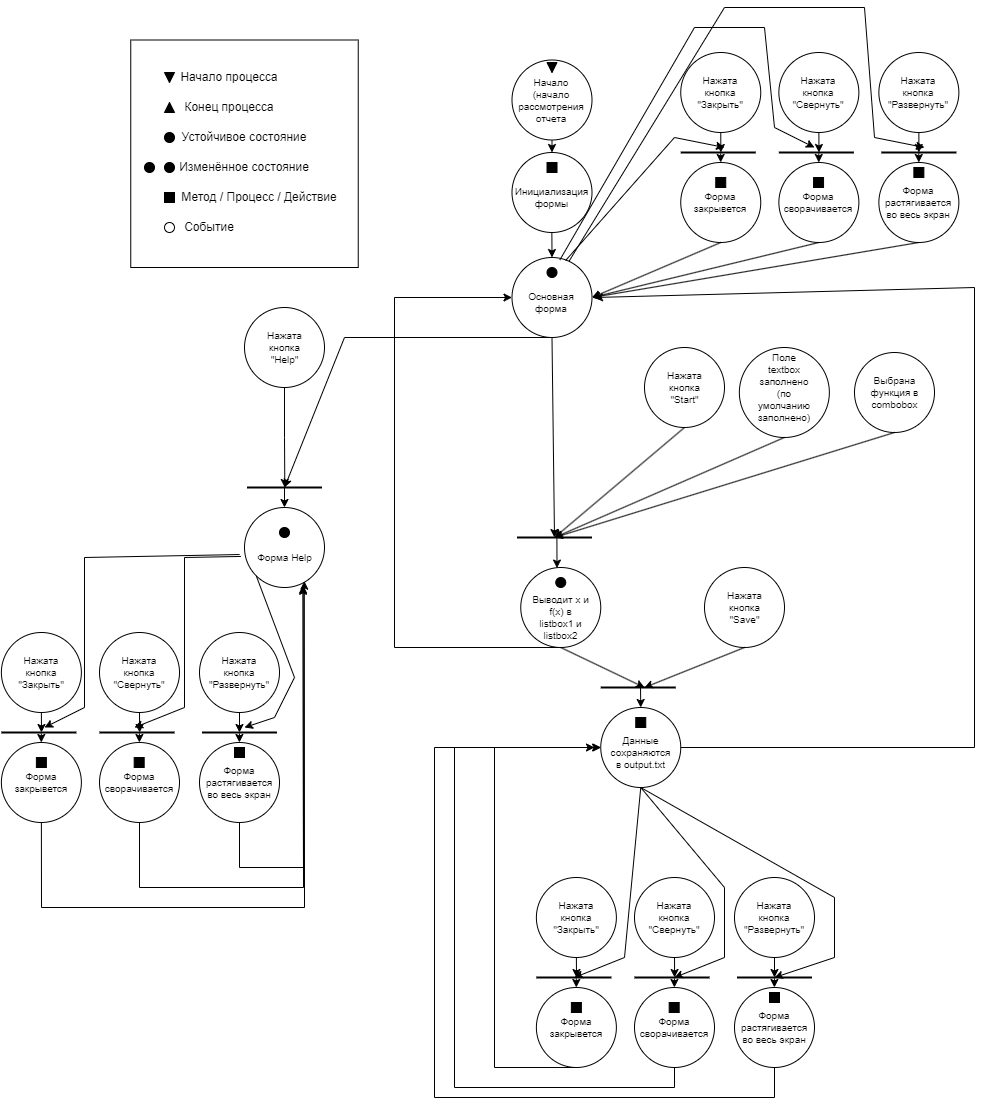


Рисунок 2 – Полная сеть Петри

# Скриншоты работы программы

****

Рисунок 3 – Начальная форма

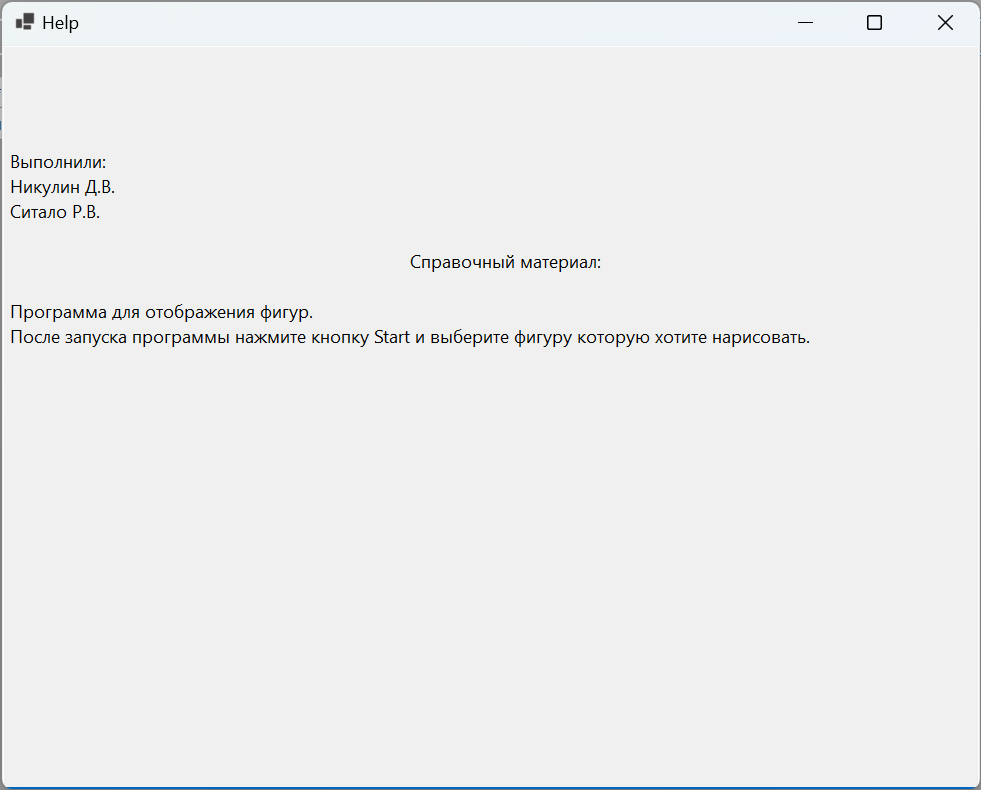


Рисунок 4 – Форма помощи

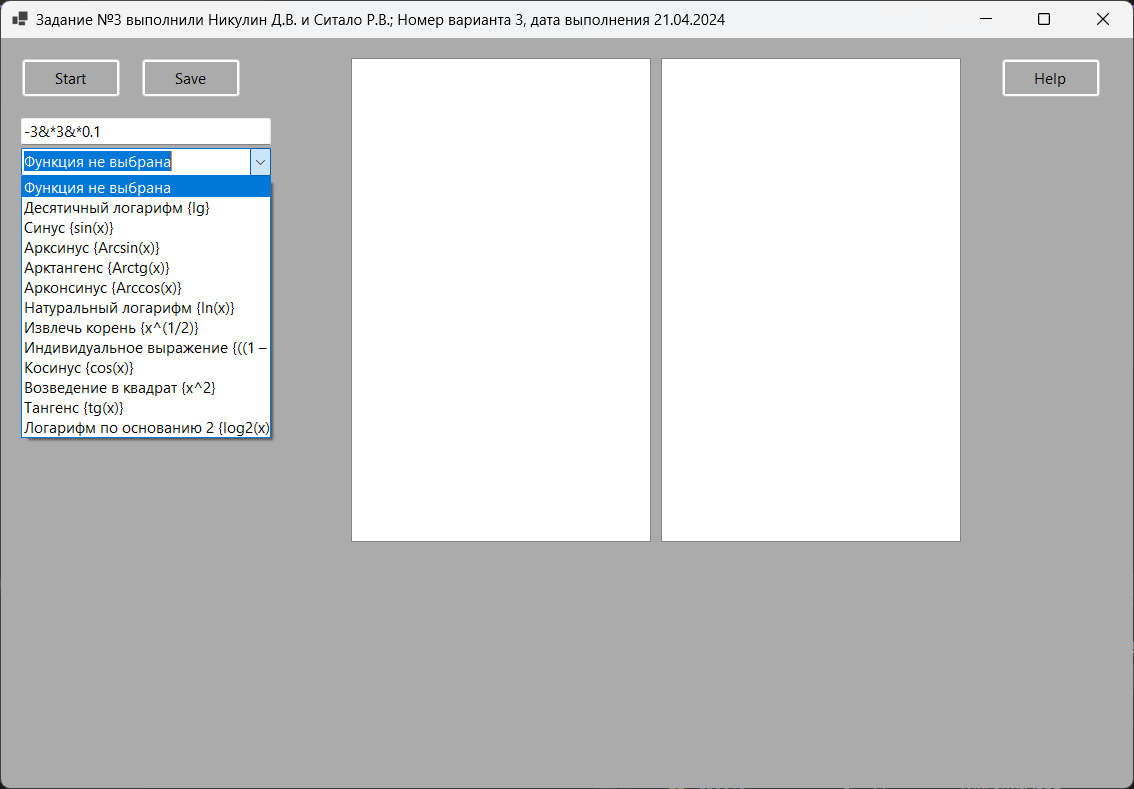


Рисунок 5 – Выбор функции

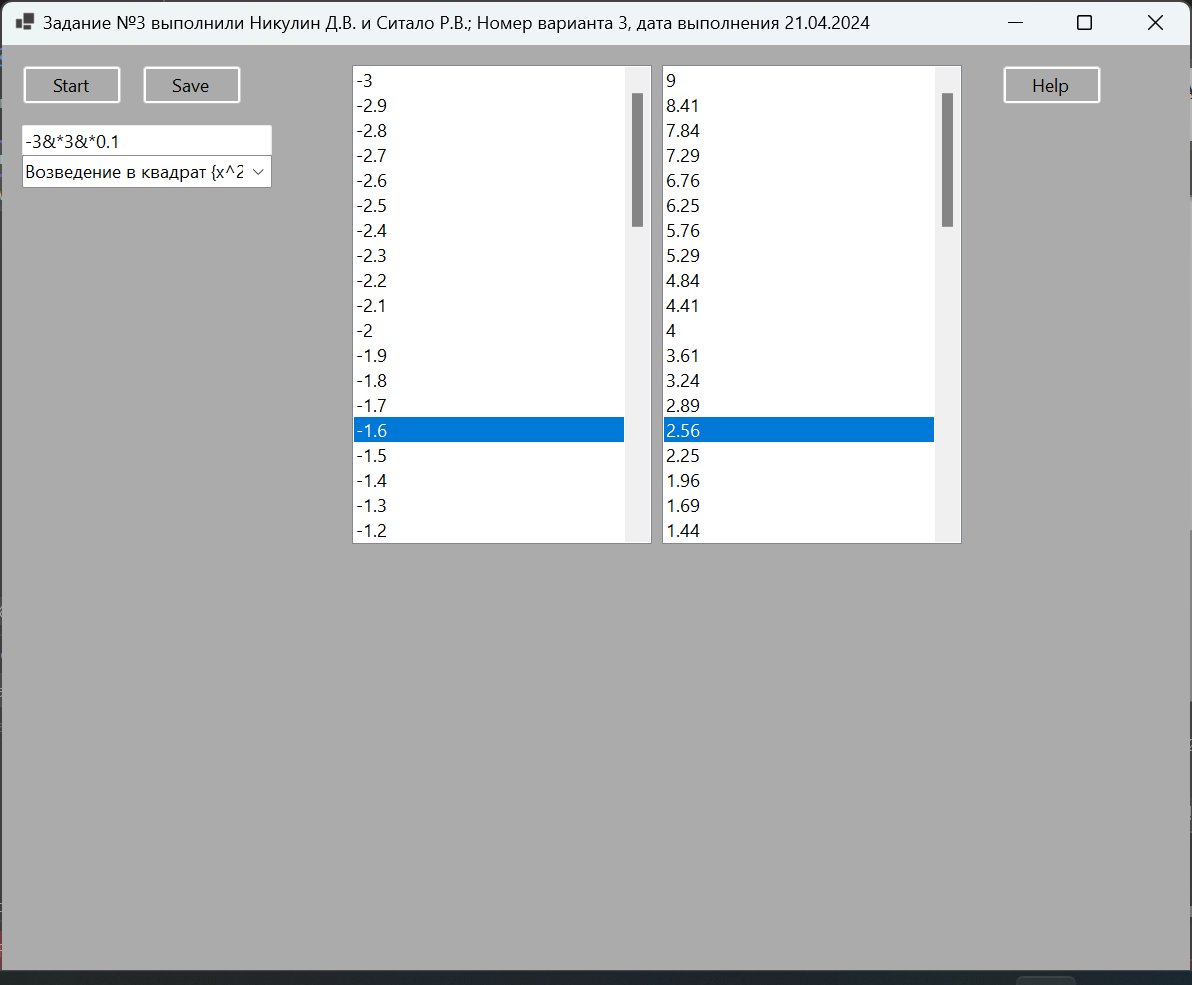


Рисунок 6 – Возведение в квадрат

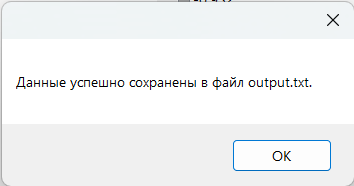


Рисунок 7 – Сохранение в файл

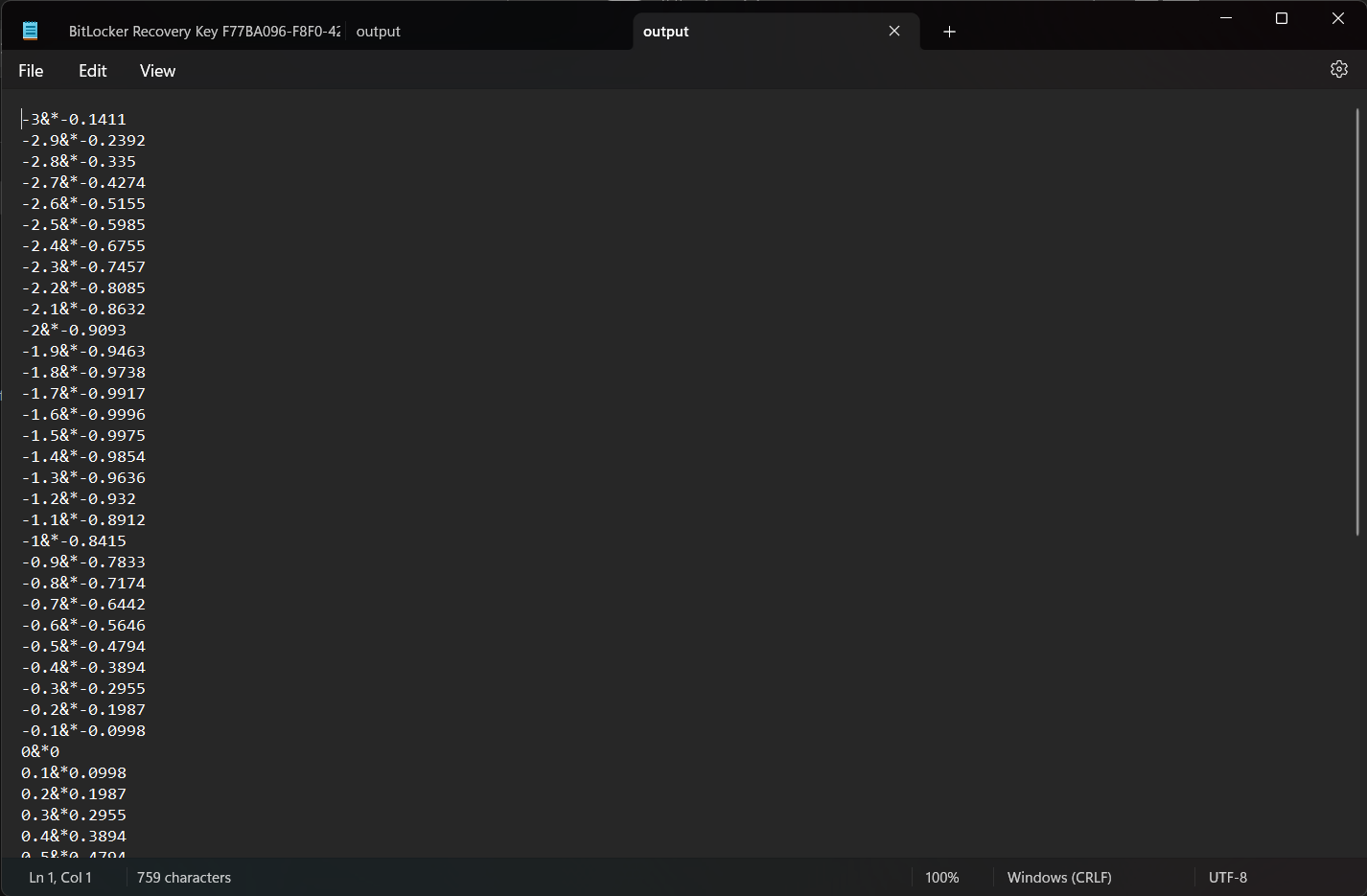


Рисунок 8 – Файл output.txt

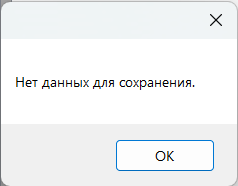


Рисунок 9 – Ошибка “Нет данных для сохранения”

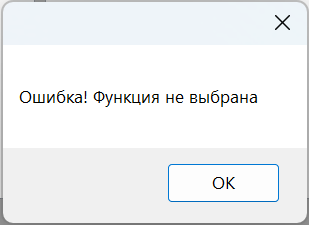
****

Рисунок 10 – Ошибка “Функция не выбрана”

# Код программы

using System.Windows.Forms;

using static System.Windows.Forms.DataFormats;

namespace WinFormsApp2

{

public partial class MainForm : 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[] default\_arguments = { -3, 3, 0.1 };

// Tools

ComboBox combobox1;

Button startbtn;

Button helpbtn;

Button savebtn;

ListBox listbox1;

ListBox listbox2;

TextBox textbox1;

public MainForm()

{

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)

{

//

// startbtn

//

startbtn = new Button();

this.Controls.Add(startbtn);

startbtn.Click += new EventHandler(startbtn\_click);

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(helpbtn\_cliclk);

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(savebtn\_click);

savebtn.Text = "Save";

savebtn.Location = new Point(140, 20);

savebtn.Size = new Size(100, 40);

savebtn.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 EventHandler(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 EventHandler(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 EventHandler(listBox2\_SelectedIndexChanged);

//

// textbox1

//

textbox1 = new TextBox();

this.Controls.Add(textbox1);

textbox1.Location = new Point(20, 80);

textbox1.Size = new Size(250, 20);

textbox1.Text = default\_arguments[0].ToString() + "&\*" + default\_arguments[1].ToString() + "&\*" + default\_arguments[2].ToString();

textbox1.TextChanged += new EventHandler(textbox1\_TextChanged);

// Pin tools to sides

startbtn.Anchor = AnchorStyles.Left | AnchorStyles.Top;

savebtn.Anchor = AnchorStyles.Left | AnchorStyles.Top;

helpbtn.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("&\*");

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;

}

}

// Start button

private void startbtn\_click(object sender, EventArgs e)

{

if (flag\_error || index\_combobox == 0)

{

MessageBox.Show("Ошибка! Функция не выбрана");

}

else

{

listbox1.Items.Clear();

listbox2.Items.Clear();

for (double i = default\_arguments[0]; i <= default\_arguments[1]; i += default\_arguments[2])

{

listbox1.Items.Add(Math.Round(i, 4));

listbox2.Items.Add(Math.Round(functions\_solver(i), 4));

}

}

}

// Help button

private void helpbtn\_cliclk(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();

}

// Save button

private void savebtn\_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 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.