Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Российский университет транспорта» (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Лабораторная работа №3

по дисциплине

**«**Методы программирования**»**

Выполнили: ст. гр. ТКИ-341

Поваляева А.В.  
Топталова С.Д.

Проверил: к.т.н. Сафронов А.И.

Москва ­– 2024 г.

1. **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.

1. **Формулировка задания**

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.

Таблица 1 – Вариант расположения расчётных функций в комбинированном или обычном списке

|  |  |
| --- | --- |
| **№вар** | **Порядок индексов функций из списка** |
| 2 | 4, 3, 8, 7, 11, 2, 10, 9, 6, 1, 5. |

**Требования к оформлению**:

1. Заголовок экранной формы должен содержать надпись вида: «Задание №3 выполнил: [Фамилия И.О. автора]; Номер варианта: [Номер]; Дата выполнения: [дд/мм/гггг];

2. Дата выполнения проставляется в момент, когда программа считается законченной и по ней можно готовить итоговый отчёт о выполнении работы;

3. Чётные варианты отключают стандартный блок управления экранной формой и создают авторские кнопки «Свернуть», «Развернуть», «Закрыть» внизу экранной формы;

4. Справочная информация должна быть вызвана в дочерней экранной форме и считана в статический по размеру ярлык из текстового файла (чётные варианты);

5. В режиме конструктора экранная форма не должна содержать ни одного элемента управления;

6. Решение (*solution*) должно содержать только одну единственную экранную форму. Экранная форма со справкой настраивается программным кодом;

7. Компоненты экранной формы со справкой должны быть чувствительными к изменению размера экранной формы по ширине и высоте (обязательно подключение события *Resize*);

8. Названия функций должны быть элементами комбинированного списка (чётные варианты);

9. Выбор конкретной функции из списка инициирует заполнение некоторого файла значениями аргумента и значениями функции. Разделитель данных подбирается согласно таблице вариантов;

Таблица 2 – Вариант разделителей, записываемых в файл

|  |  |
| --- | --- |
| **№вар** | **Разделитель** |
| 2 | % |

10. Предусмотреть кнопку, нажатие на которую инициирует заполнение двух списков (один список под значения аргумента, другой список – под значения функции, округлённых по правилу *(10 mod 5) + 1* для знаков после плавающей запятой данными о функциональной зависимости из файла. Обучающиеся с чётными номерами вариантов обновляют существующие основные списки;

11. В качестве исходного состояния функциональных компонентов главной экранной формы принимается недоступное состояние (*.Enabled = false*) – чётные варианты;

12. Кнопка «Начало работы с приложением» должна реализовывать активацию доступа пользователя к функциональным компонентам или отображение функциональных компонентов на экранной форме для пользователя;

13. Все элементы программы должны носить значащие имена переменных, в которых отражено существо этих элементов, например, экранная форма – *frmMain*, ярлык – *lblHelp*, комбинированный список – *cmbFigures* и т.д.;

14. Выбор значения абсциссы в списке должен инициировать выбор соответствующего ему значения ординаты и наоборот – выбор значения ординаты должен инициировать выбор соответствующего ему значения абсциссы (работа с параметрами *SelectedIndex* или *SelectedItem* списков).

**Вариант индивидуального задания:**

Таблица 3 – Вариант дополнительных функциональных зависимостей, рассчитываемых программой

|  |  |  |
| --- | --- | --- |
| **№вар** | **Функция** | **Позиция** |
| 2 | (x2 - x + 7) / (x + 1) | 1 |

1. **Диаграмма классов, входящих в состав решения**

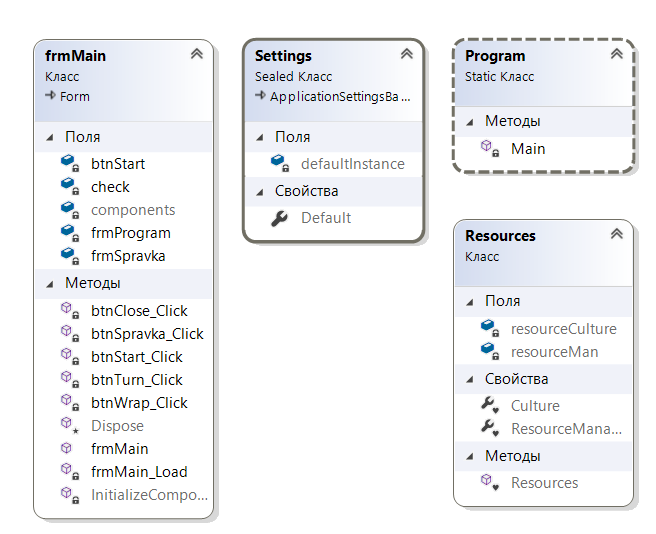
****

Рисунок 1 –Диаграмма классов, входящих в состав решения

1. **Сеть Петри запрограммированного процесса**

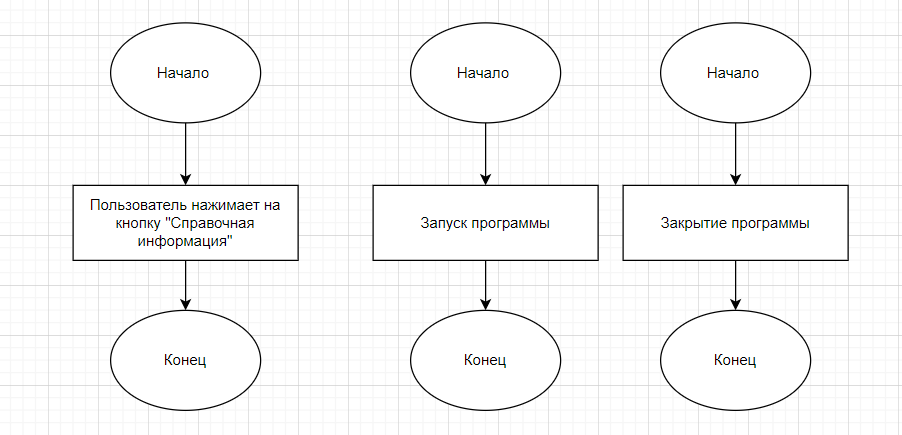
**Изображение выглядит как текст, снимок экрана, круг, черно-белый

Автоматически созданное описание**

Рисунок 2 – Сеть Петри запрограммированного процесса

1. **Схемы алгоритмов методов в составе решения, отмеченных на сети Петри в качестве «эффектов» (метка )**

* **e1**

****

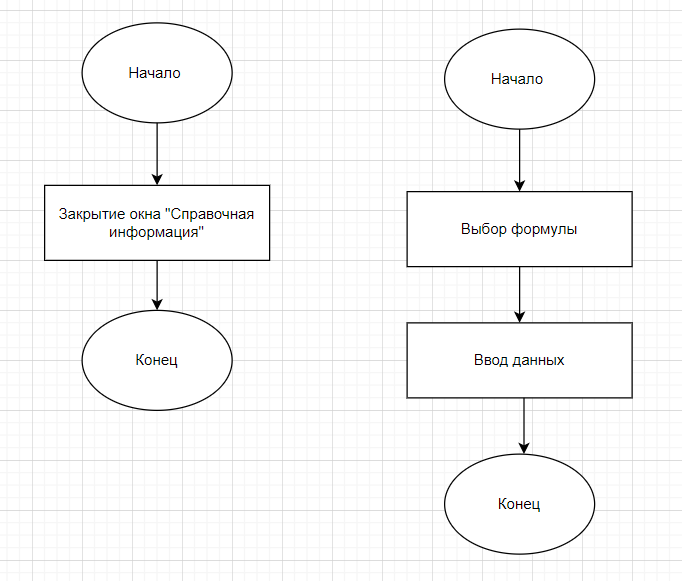
****

Рисунок 3 – Схемы алгоритмов методов в составе решения (метка e1 )

1. **Тестовые примеры**

* При нажатии кнопки «Справочная информация» происходит открытие окна со справочной информацией;
* Расчет некоторой функции;
* Округление значений по правилу *(10 mod 5) + 1*.

1. **Листинг (код) составленного программного обеспечения**

* **frmMain**

using System;

using System.Drawing;

using System.Windows.Forms;

using System.IO;

namespace Laba\_\_\_3

{

public partial class frmMain : Form

{

private System.Windows.Forms.Button btnStart;

public frmMain()

{

InitializeComponent();

}

bool check = false;

private void frmMain\_Load(object sender, EventArgs e)

{

this.Text = "Задание №3 выполнили: Поваляева А.В., Топталова С.Д.; Номер варианта: 2; Дата выполнения: 20/05/2024";

this.ClientSize = new Size(800, 500);

this.FormBorderStyle = FormBorderStyle.FixedSingle;

this.ControlBox = false;

this.BackColor = Color.PeachPuff;

this.btnStart = new System.Windows.Forms.Button();

btnStart.Size = new System.Drawing.Size(250, 100);

btnStart.Location = new Point(280, 180);

btnStart.Font = new Font("Times New Roman", 14, FontStyle.Bold);

btnStart.Text = "Старт";

btnStart.BackColor = Color.AliceBlue;

btnStart.Click += new EventHandler(btnStart\_Click);

btnStart.AutoSize = false;

this.Controls.Add(btnStart);

System.Windows.Forms.Button btnClose = new System.Windows.Forms.Button();

btnClose.Size = new System.Drawing.Size(30, 30);

btnClose.Location = new Point(755, 455);

btnClose.Font = new Font("Times New Roman", 14, FontStyle.Bold);

btnClose.Text = "X";

btnClose.BackColor = Color.AliceBlue;

btnClose.Click += new EventHandler(btnClose\_Click);

btnClose.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;

this.Controls.Add(btnClose);

System.Windows.Forms.Button btnWrap = new System.Windows.Forms.Button();

btnWrap.Size = new System.Drawing.Size(30, 30);

btnWrap.Location = new Point(725, 455);

btnWrap.Font = new Font("Times New Roman", 14, FontStyle.Bold);

btnWrap.Text = "-";

btnWrap.BackColor = Color.AliceBlue;

btnWrap.Click += new EventHandler(btnWrap\_Click);

btnWrap.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;

this.Controls.Add(btnWrap);

System.Windows.Forms.Button btnTurn = new System.Windows.Forms.Button();

btnTurn.Size = new System.Drawing.Size(30, 30);

btnTurn.Location = new Point(695, 455);

btnTurn.Font = new Font("Times New Roman", 14, FontStyle.Bold);

btnTurn.Text = "";

btnTurn.BackColor = Color.AliceBlue;

btnTurn.Click += new EventHandler(btnTurn\_Click);

btnTurn.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;

this.Controls.Add(btnTurn);

System.Windows.Forms.Button btnSpravka = new System.Windows.Forms.Button();

btnSpravka.Size = new System.Drawing.Size(300, 30);

btnSpravka.Location = new Point(15, 455);

btnSpravka.Font = new Font("Times New Roman", 14, FontStyle.Bold);

btnSpravka.Text = "Справочная информация";

btnSpravka.BackColor = Color.AliceBlue;

btnSpravka.Click += new EventHandler(btnSpravka\_Click);

btnSpravka.Anchor = AnchorStyles.Bottom | AnchorStyles.Left;

this.Controls.Add(btnSpravka);

}

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

{

if (check == false)

{

this.WindowState = FormWindowState.Maximized;

check = true;

}

else

{

this.WindowState = FormWindowState.Normal;

check = false;

}

}

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

{

this.WindowState = FormWindowState.Minimized;

}

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

{

Application.Exit();

}

Form frmSpravka = new Form();

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

{

frmSpravka.Text = "Справочная информация";

frmSpravka.Size = new Size(400, 400);

frmSpravka.BackColor = Color.PeachPuff;

frmSpravka.ControlBox = false;

System.Windows.Forms.Button btnSpravkaClose = new System.Windows.Forms.Button();

btnSpravkaClose.Text = "X";

btnSpravkaClose.Size = new System.Drawing.Size(30, 30);

btnSpravkaClose.Font = new Font("Times New Roman", 14, FontStyle.Bold);

btnSpravkaClose.BackColor = Color.AliceBlue;

btnSpravkaClose.Location = new Point(340, 325);

btnSpravkaClose.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;

btnSpravkaClose.Click += (s, ev) =>

{

frmSpravka.Close();

};

System.Windows.Forms.Button btnSpravkaWrap = new System.Windows.Forms.Button();

btnSpravkaWrap.Text = "-";

btnSpravkaWrap.Size = new System.Drawing.Size(30, 30);

btnSpravkaWrap.Font = new Font("Times New Roman", 14, FontStyle.Bold);

btnSpravkaWrap.BackColor = Color.AliceBlue;

btnSpravkaWrap.Location = new Point(310, 325);

btnSpravkaWrap.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;

btnSpravkaWrap.Click += (s, ev) =>

{

frmSpravka.WindowState = FormWindowState.Minimized;

};

System.Windows.Forms.Button btnSpravkaTurn = new System.Windows.Forms.Button();

btnSpravkaTurn.Text = "";

btnSpravkaTurn.Size = new System.Drawing.Size(30, 30);

btnSpravkaTurn.Font = new Font("Times New Roman", 14, FontStyle.Bold);

btnSpravkaTurn.BackColor = Color.AliceBlue;

btnSpravkaTurn.Location = new Point(280, 325);

btnSpravkaTurn.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;

btnSpravkaTurn.Click += (s, ev) =>

{

if (frmSpravka.WindowState == FormWindowState.Maximized)

{

frmSpravka.WindowState = FormWindowState.Normal;

btnSpravkaTurn.Text = "";

}

else

{

frmSpravka.WindowState = FormWindowState.Maximized;

btnSpravkaTurn.Text = "";

}

};

frmSpravka.Controls.Add(btnSpravkaTurn);

frmSpravka.Controls.Add(btnSpravkaClose);

frmSpravka.Controls.Add(btnSpravkaWrap);

System.Windows.Forms.Label lblSpravka = new System.Windows.Forms.Label();

lblSpravka.AutoSize = false;

lblSpravka.Size = new Size(frmSpravka.ClientRectangle.Width, frmSpravka.ClientRectangle.Height - 40);

lblSpravka.Location = new Point(0, 10);

lblSpravka.Anchor = AnchorStyles.Top | AnchorStyles.Bottom | AnchorStyles.Left | AnchorStyles.Right;

lblSpravka.Text = File.ReadAllText("spravka.txt");

frmSpravka.SizeChanged += (s, ev) =>

{

lblSpravka.Size = new Size(frmSpravka.ClientRectangle.Width, frmSpravka.ClientRectangle.Height - 40);

};

frmSpravka.Controls.Add(lblSpravka);

frmSpravka.ShowDialog();

}

Form frmProgram = new Form();

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

{

frmProgram.Text = "Задание №3 выполнили: Поваляева А.В., Топталова С.Д.; Номер варианта: 2; Дата выполнения: 20/05/2024";

frmProgram.Size = new Size(800, 500);

frmProgram.BackColor = Color.PeachPuff;

frmProgram.ControlBox = false;

System.Windows.Forms.Button btnProgramClose = new System.Windows.Forms.Button();

btnProgramClose.Text = "X";

btnProgramClose.Size = new System.Drawing.Size(30, 30);

btnProgramClose.Font = new Font("Times New Roman", 14, FontStyle.Bold);

btnProgramClose.BackColor = Color.AliceBlue;

btnProgramClose.Location = new Point(750, 420);

btnProgramClose.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;

btnProgramClose.Click += (s, ev) =>

{

frmProgram.Close();

};

System.Windows.Forms.Button btnProgramWrap = new System.Windows.Forms.Button();

btnProgramWrap.Text = "-";

btnProgramWrap.Size = new System.Drawing.Size(30, 30);

btnProgramWrap.Font = new Font("Times New Roman", 14, FontStyle.Bold);

btnProgramWrap.BackColor = Color.AliceBlue;

btnProgramWrap.Location = new Point(720, 420);

btnProgramWrap.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;

btnProgramWrap.Click += (s, ev) =>

{

frmProgram.WindowState = FormWindowState.Minimized;

};

System.Windows.Forms.Button btnProgramTurn = new System.Windows.Forms.Button();

btnProgramTurn.Text = "";

btnProgramTurn.Size = new System.Drawing.Size(30, 30);

btnProgramTurn.Font = new Font("Times New Roman", 14, FontStyle.Bold);

btnProgramTurn.BackColor = Color.AliceBlue;

btnProgramTurn.Location = new Point(690, 420);

btnProgramTurn.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;

btnProgramTurn.Click += (s, ev) =>

{

if (frmProgram.WindowState == FormWindowState.Maximized)

{

frmProgram.WindowState = FormWindowState.Normal;

btnProgramTurn.Text = "";

}

else

{

frmProgram.WindowState = FormWindowState.Maximized;

btnProgramTurn.Text = "";

}

};

System.Windows.Forms.ComboBox cbFunc = new System.Windows.Forms.ComboBox();

cbFunc.Location = new Point(40, 40);

cbFunc.Size = new System.Drawing.Size(200, 150);

cbFunc.Items.Add("Выражение (x\*x-x+7)/(x+7)");

cbFunc.Items.Add("Косинус");

cbFunc.Items.Add("Возведение в квадрат");

cbFunc.Items.Add("Арктангенс");

cbFunc.Items.Add("Извлечение корня");

cbFunc.Items.Add("Арксинус");

cbFunc.Items.Add("Тангенс");

cbFunc.Items.Add("Десятичный логарифм");

cbFunc.Items.Add("Синус");

cbFunc.Items.Add("Арккосинус");

cbFunc.Items.Add("Логарифм по основанию 2");

cbFunc.Items.Add("Натуральный логарифм");

cbFunc.SelectedIndex = 0;

System.Windows.Forms.ListBox lBZnach = new System.Windows.Forms.ListBox();

lBZnach.Location = new Point(40, 100);

lBZnach.Size = new Size(250, 300);

frmProgram.Controls.Add(lBZnach);

System.Windows.Forms.ListBox lBRezult = new System.Windows.Forms.ListBox();

lBRezult.Location = new Point(400, 100);

lBRezult.Size = new Size(250, 300);

frmProgram.Controls.Add(lBRezult);

System.Windows.Forms.TextBox tBStart = new System.Windows.Forms.TextBox();

System.Windows.Forms.TextBox tBEnd = new System.Windows.Forms.TextBox();

System.Windows.Forms.TextBox tBStep = new System.Windows.Forms.TextBox();

tBStart.Location = new Point(280, 40);

tBStart.Size = new System.Drawing.Size(50, 100);

tBEnd.Location = new Point(350, 40);

tBEnd.Size = new System.Drawing.Size(50, 100);

tBStep.Location = new Point(420, 40);

tBStep.Size = new System.Drawing.Size(50, 100);

System.Windows.Forms.Button btnSchet = new System.Windows.Forms.Button();

btnSchet.Size = new System.Drawing.Size(150, 40);

btnSchet.Font = new Font("Times New Roman", 12, FontStyle.Bold);

btnSchet.BackColor = Color.AliceBlue;

btnSchet.Text = "Рассчитать";

btnSchet.Location = new Point(40, 400);

btnSchet.AutoSize = false;

frmProgram.Controls.Add(btnSchet);

btnSchet.Click += (s, ev) =>

{

lBZnach.Items.Clear();

lBRezult.Items.Clear();

if (string.IsNullOrEmpty(tBStart.Text) || string.IsNullOrEmpty(tBEnd.Text) || string.IsNullOrEmpty(tBStep.Text))

{

MessageBox.Show("Введите все параметры");

return;

}

if (!int.TryParse(tBStart.Text, out int start) || !int.TryParse(tBEnd.Text, out int end) || !int.TryParse(tBStep.Text, out int step))

{

MessageBox.Show("Введены некорректные значения параметров.");

return;

}

string selectedFormula = cbFunc.SelectedItem.ToString();

for (int x = start; x <= end; x += step)

{

lBZnach.Items.Add(x);

double result = 0;

switch (selectedFormula)

{

case "Выражение (x\*x-x+7)/(x+7)":

result = (x \* x - x + 7) / (x + 7);

break;

case "Косинус":

result = Math.Cos(x);

break;

case "Возведение в квадрат":

result = Math.Pow(x, 2);

break;

case "Арктангенс":

result = Math.Atan(x);

break;

case "Извлечение корня":

result = Math.Sqrt(x);

break;

case "Арксинус":

result = Math.Asin(x);

break;

case "Тангенс":

result = Math.Tan(x);

break;

case "Десятичный логарифм":

result = Math.Log10(x);

break;

case "Синус":

result = Math.Sin(x);

break;

case "Арккосинус":

result = Math.Acos(x);

break;

case "Логарифм по основанию 2":

result = Math.Log(x, 2);

break;

case "Натуральный логарифм":

result = Math.Log(x);

break;

default:

result = 0; // Если формула не определена, результат будет 0

break;

}

lBRezult.Items.Add(result);

}

string filePath = "data.txt";

using (StreamWriter writer = new StreamWriter(filePath))

{

for (int i = 0; i < lBZnach.Items.Count; i++)

{

string argument = lBZnach.Items[i].ToString();

string result = lBRezult.Items[i].ToString();

writer.WriteLine($"{argument} % {result}");

}

}

};

lBZnach.SelectedIndexChanged += (s, ev) =>

{

lBRezult.SelectedIndex = lBZnach.SelectedIndex;

};

lBRezult.SelectedIndexChanged += (s, ev) =>

{

lBZnach.SelectedIndex = lBRezult.SelectedIndex;

};

System.Windows.Forms.Button btnLoadData = new System.Windows.Forms.Button();

btnLoadData.Text = "Округлить";

btnLoadData.Size = new System.Drawing.Size(150, 40);

btnLoadData.Font = new Font("Times New Roman", 12, FontStyle.Bold);

btnLoadData.BackColor = Color.AliceBlue; ;

btnLoadData.Location = new Point(200, 400);

btnLoadData.AutoSize = false;

frmProgram.Controls.Add(btnLoadData);

btnLoadData.Click += (s, ev) =>

{

lBZnach.Items.Clear();

lBRezult.Items.Clear();

string filePath = "data.txt";

string[] lines = File.ReadAllLines(filePath);

int Rounding = (2 % 5) + 1;

foreach (string line in lines)

{

string[] parts = line.Split(new[] { "%" }, StringSplitOptions.None);

if (parts.Length == 2)

{

string argument = parts[0];

double result = double.Parse(parts[1]);

result = Math.Round(result, Rounding);

lBZnach.Items.Add(argument);

lBRezult.Items.Add(result);

}

}

};

frmProgram.Controls.Add(btnProgramTurn);

frmProgram.Controls.Add(btnProgramClose);

frmProgram.Controls.Add(btnProgramWrap);

frmProgram.Controls.Add(cbFunc);

frmProgram.Controls.Add(lBZnach);

frmProgram.Controls.Add(lBRezult);

frmProgram.Controls.Add(tBStart);

frmProgram.Controls.Add(tBEnd);

frmProgram.Controls.Add(tBStep);

frmProgram.ShowDialog();

}

}

}

1. **Графический пользовательский интерфейс программного обеспечения и его описание**

* **ГПИ стартовой страницы**

На рисунке 5 показано, что на стартовой странице находится пять кнопок. Кнопка «Старт» запускает работу приложения (перебрасывает пользователя на страницу с функциями). При нажатии кнопки «Справочная информация» пользователь получает информацию о программе. Также в правом нижнем углу страницы находятся кнопки «Закрыть», «Развернуть в полноэкранный режим» и «Свернуть» для взаимодействия с формой.

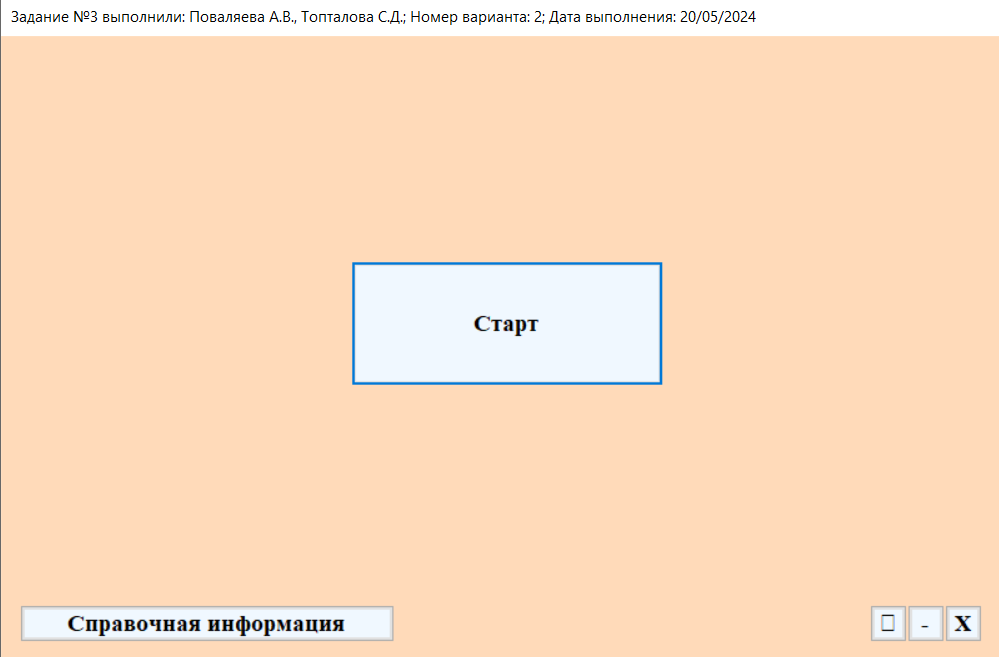
****

Рисунок 5 – ГПИ стартовой страницы

* ГПИ справочной информации

На рисунке 6 показана краткая справочная информация о программе. Справочная информация вызвана в дочерней экранной форме и считана в статический по размеру ярлык из текстового файла (spravka.txt)

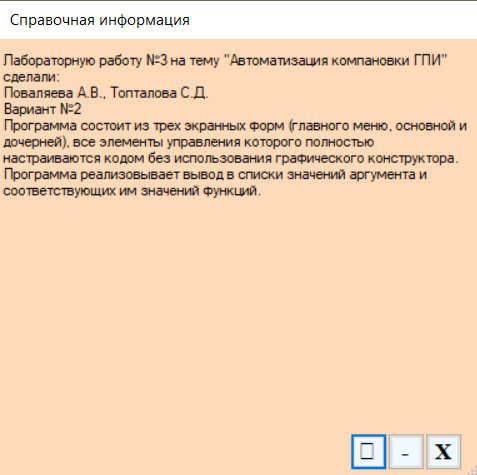
****

Рисунок 6 – ГПИ справочной информации

* ГПИ программы

На рисунке 7 показано, что в ГПИ приложения находится список ComboBox для выбора пользователем функции, по которой будет происходить расчет. Также расположены три поля TextBox для ввода пользователем начального значения, конечного значения и шага для расчетов по функциям. Находятся два ListBox. Находятся две кнопки «Рассчитать» и «Округлить».

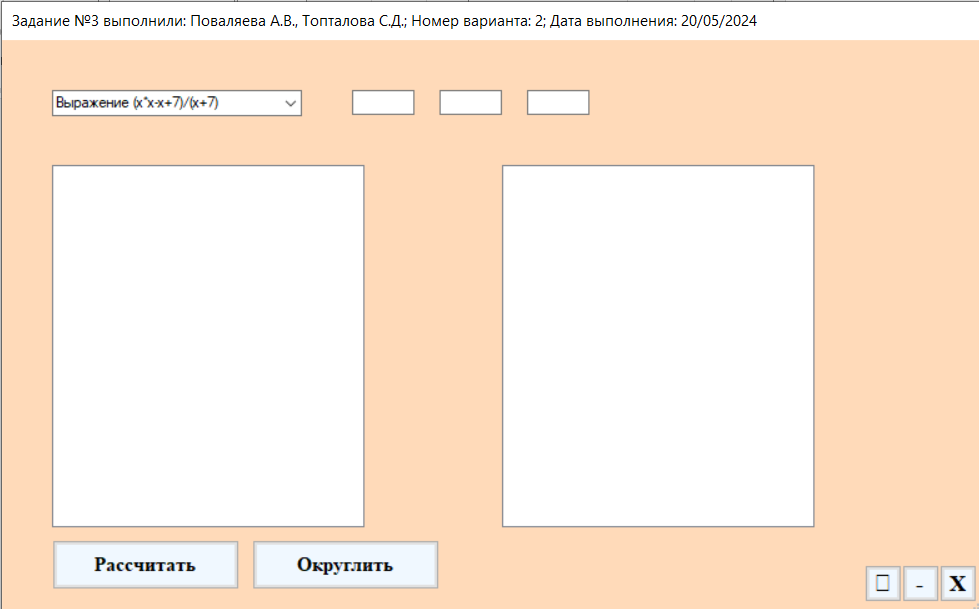
****

Рисунок 7 – ГПИ приложения

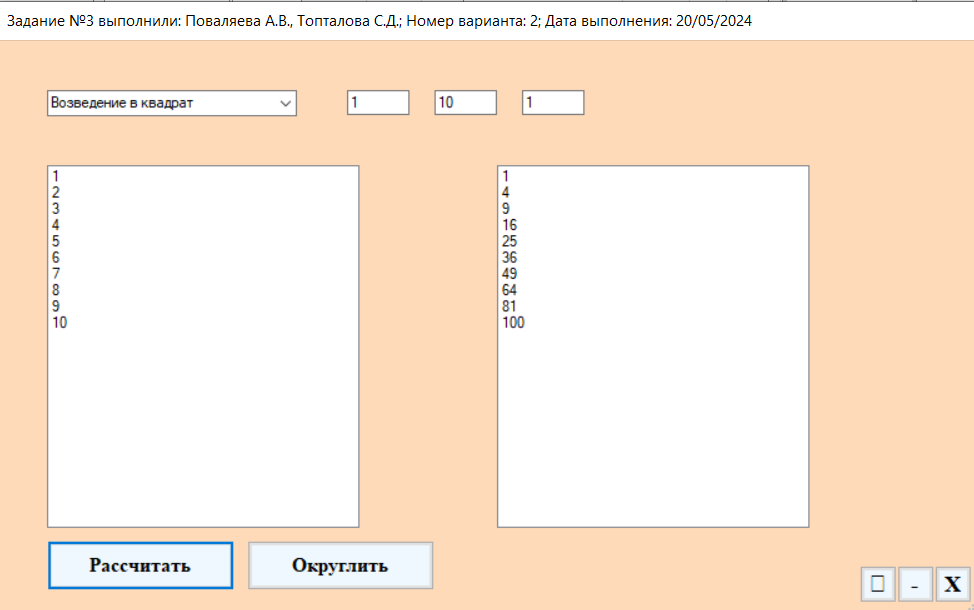


Рисунок 8 – Результат работы программы (Пользователь выбрал функцию “Возведение в квадрат”, начальное число 1, конечное число 10, шаг 1)

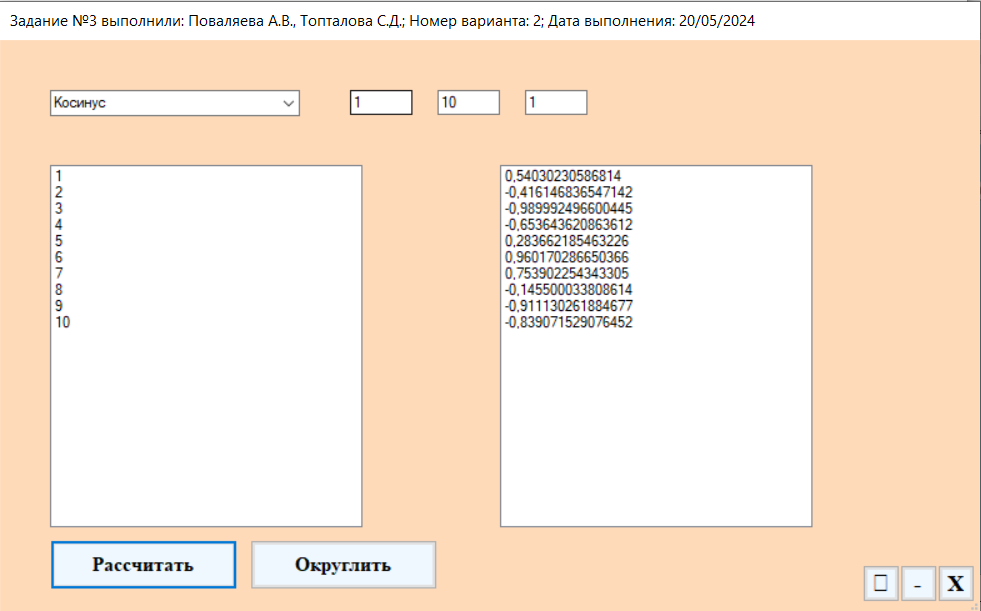


Рисунок 9 – Результат работы программы (Пользователь выбрал функцию “Косинус”, начальное число 1, конечное число 10, шаг 1)

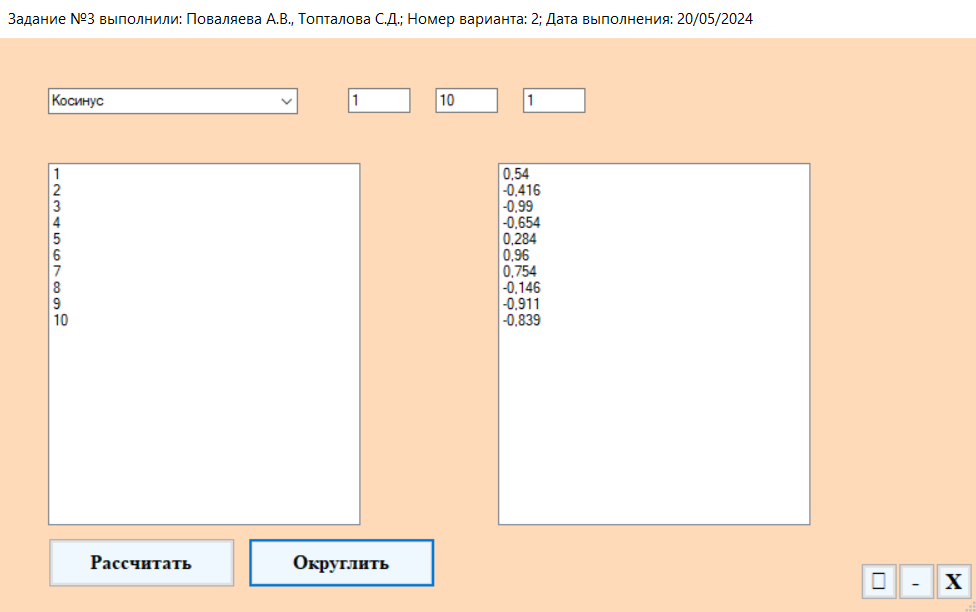


Рисунок 10 – Результат работы программы с округлением (Пользователь выбрал функцию “Косинус”, начальное число 1, конечное число 10, шаг 1)

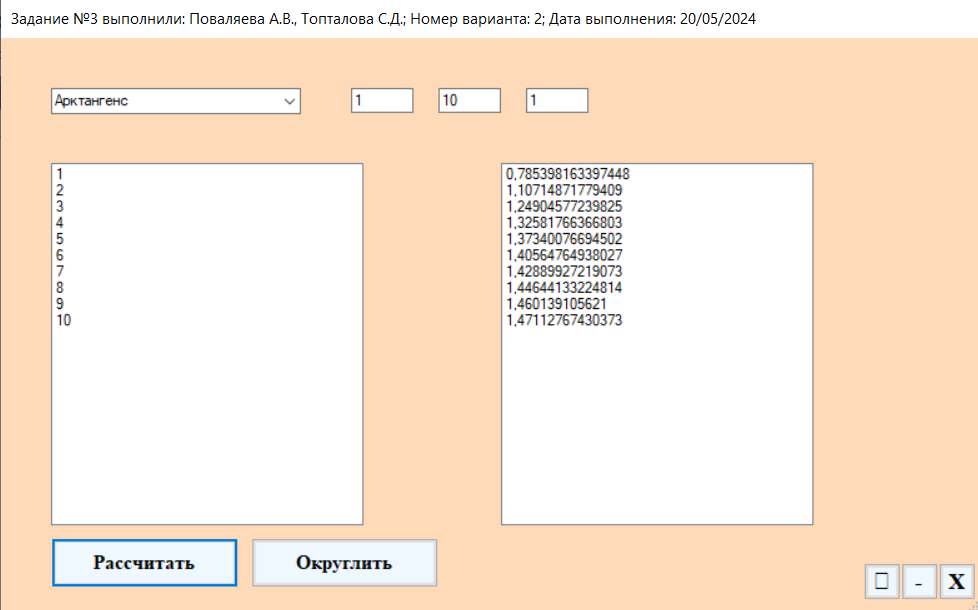


Рисунок 11 – Результат работы программы (Пользователь выбрал функцию “Арктангенс”, начальное число 1, конечное число 10, шаг 1)

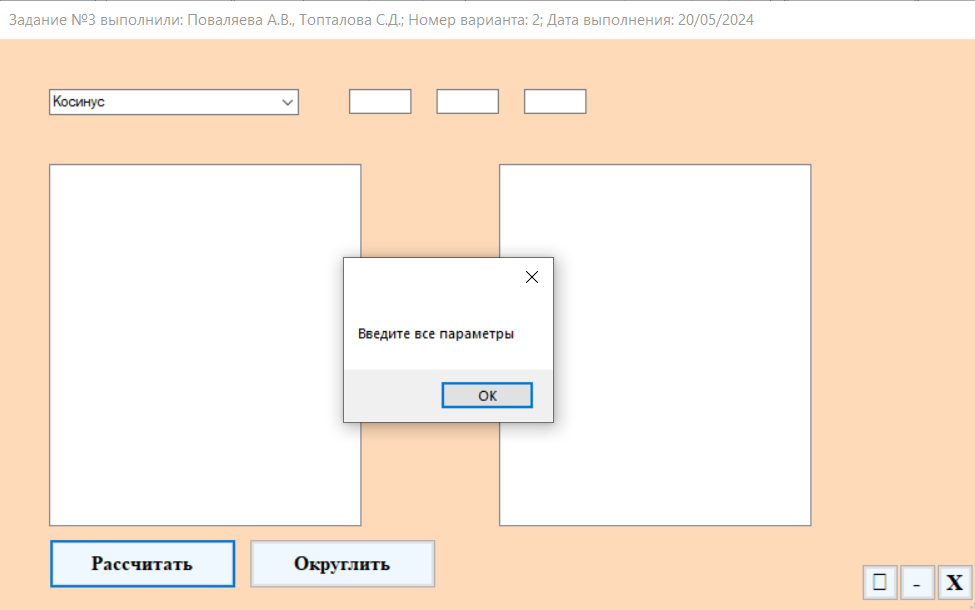


Рисунок 12 – Результат работы программы (пользователь не ввел в поля TextBox входные параметры и нажал кнопку «Рассчитать»)

1. **Подтверждение соответствия графического пользовательского интерфейса требованиям к оформлению**

* Заголовок экранной формы должен содержать надпись вида: «Задание №3 выполнил: [Фамилия И.О. автора]; Номер варианта: [Номер]; Дата выполнения: [дд/мм/гггг];

****

Рисунок 13 – Заголовок экранной формы

* Чётные варианты отключают стандартный блок управления экранной формой и создают авторские кнопки «Свернуть», «Развернуть», «Закрыть» внизу экранной формы;



Рисунок 14 – Авторские кнопки «Свернуть», «Развернуть», «Закрыть» внизу экранной формы.

* Справочная информация должна быть вызвана в дочерней экранной форме и считана в статический по размеру ярлык из текстового файла (чётные варианты);





Рисунок 15, 16 – Справочная информация вызвана в дочерней экранной форме и считана в статический по размеру ярлык из текстового файла

* В режиме конструктора экранная форма не должна содержать ни одного элемента управления;

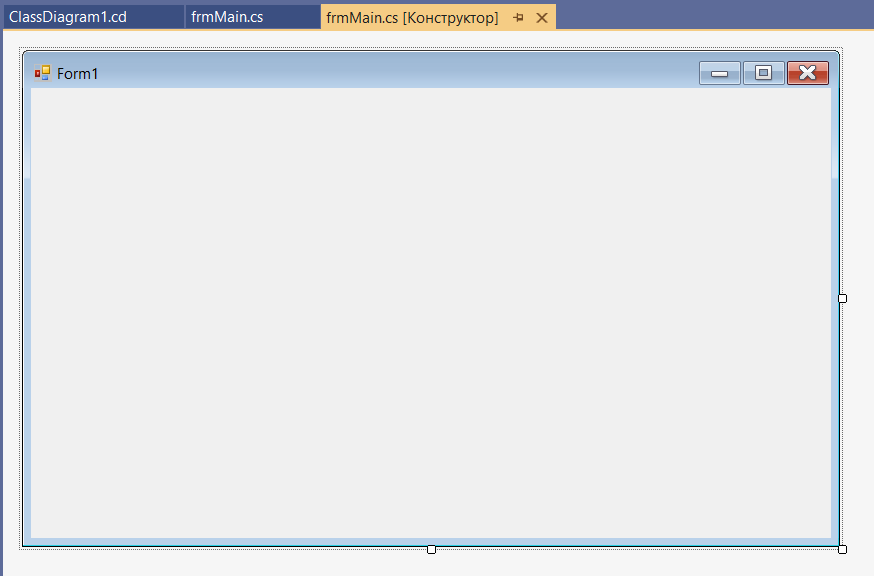


Рисунок 17 – Режим конструктора

* Решение (*solution*) должно содержать только одну единственную экранную форму. Экранная форма со справкой настраивается программным кодом;

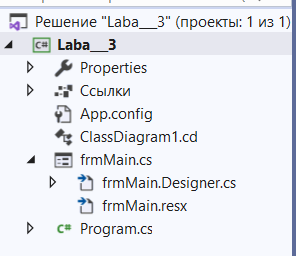


Рисунок 18 – Решение (*solution*) содержит только одну единственную экранную форму

* Компоненты экранной формы со справкой должны быть чувствительными к изменению размера экранной формы по ширине и высоте (обязательно подключение события *Resize*);

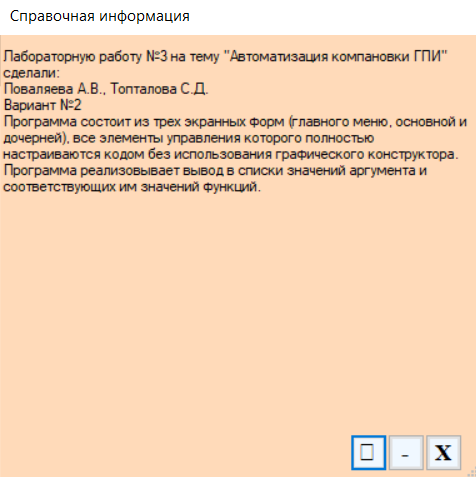


Рисунок 19 – Окно со справочной информацией до изменения размера

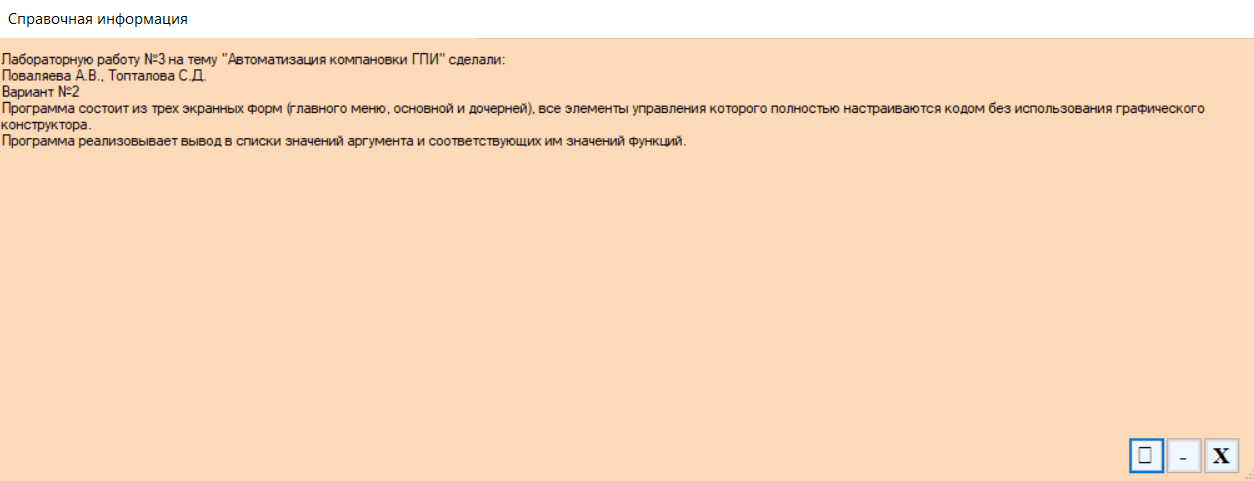


Рисунок 20 – Окно со справочной информацией после изменения размера

* Названия функций должны быть элементами комбинированного списка (чётные варианты);

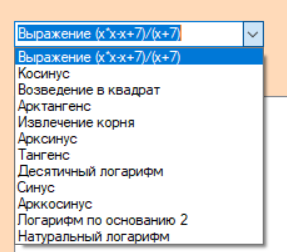


Рисунок 21 – Названия функций – это элементы комбинированного списка

* Выбор конкретной функции из списка инициирует заполнение некоторого файла значениями аргумента и значениями функции. Разделитель данных подбирается согласно таблице вариантов;

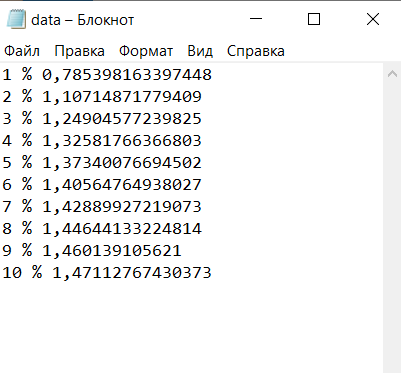


Рисунок 22– Выбор конкретной функции из списка инициирует заполнение файла data.txt значениями аргумента и значениями функции

* Предусмотреть кнопку, нажатие на которую инициирует заполнение двух списков (один список под значения аргумента, другой список – под значения функции, округлённых по правилу *(10 mod 5) + 1* для знаков после плавающей запятой данными о функциональной зависимости из файла. Обучающиеся с чётными номерами вариантов обновляют существующие основные списки;



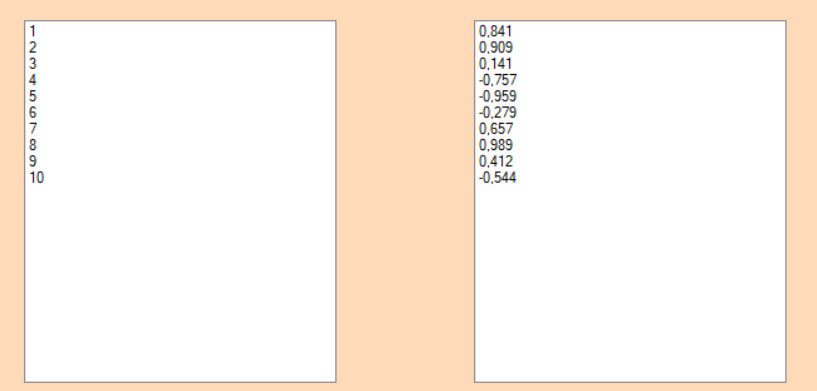


Рисунок 23, 24 – Кнопка, нажатие на которую инициирует заполнение двух списков и пример обновленных существующих основных списков

* Кнопка «Начало работы с приложением» должна реализовывать активацию доступа пользователя к функциональным компонентам или отображение функциональных компонентов на экранной форме для пользователя;

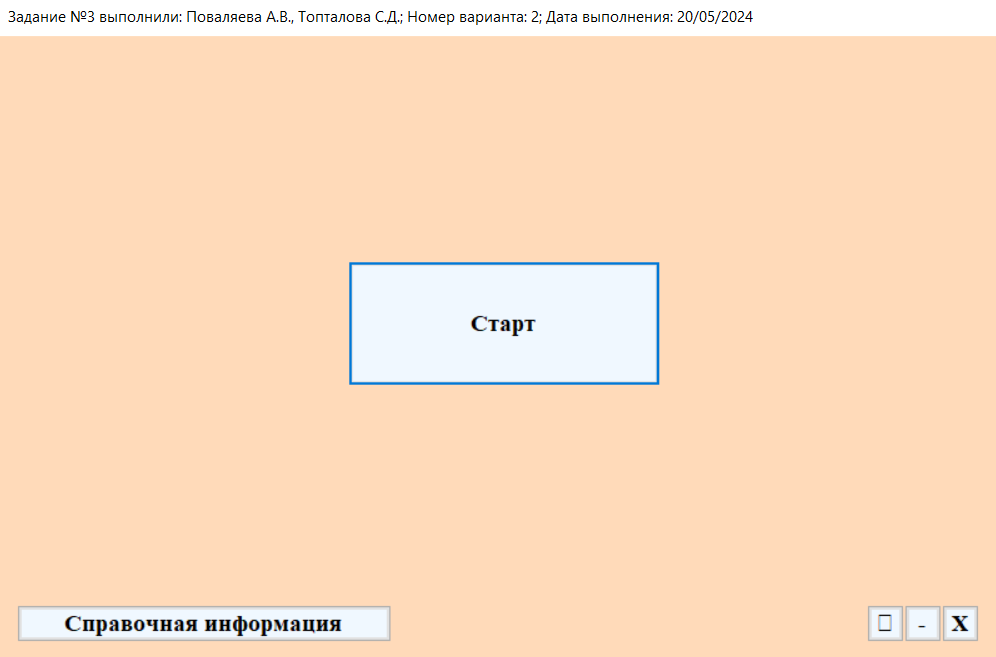


Рисунок 25 – Форма до нажатия на кнопку «Старт»

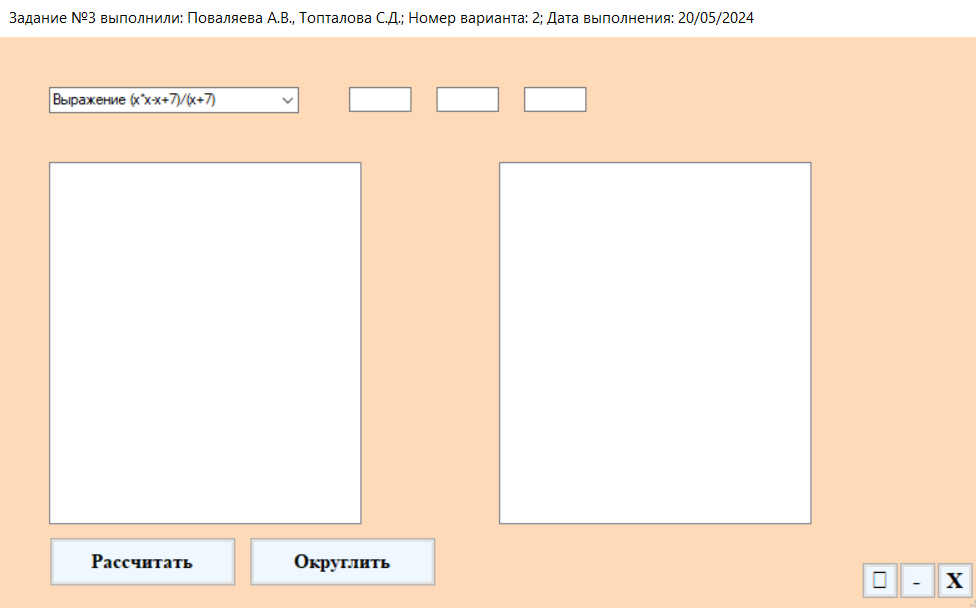
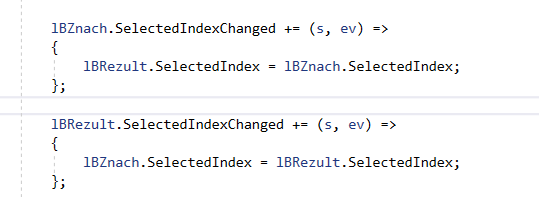


Рисунок 26 – Форма после нажатия на кнопку «Старт»

* Выбор значения абсциссы в списке должен инициировать выбор соответствующего ему значения ординаты и наоборот – выбор значения ординаты должен инициировать выбор соответствующего ему значения абсциссы (работа с параметрами *SelectedIndex* или *SelectedItem* списков).



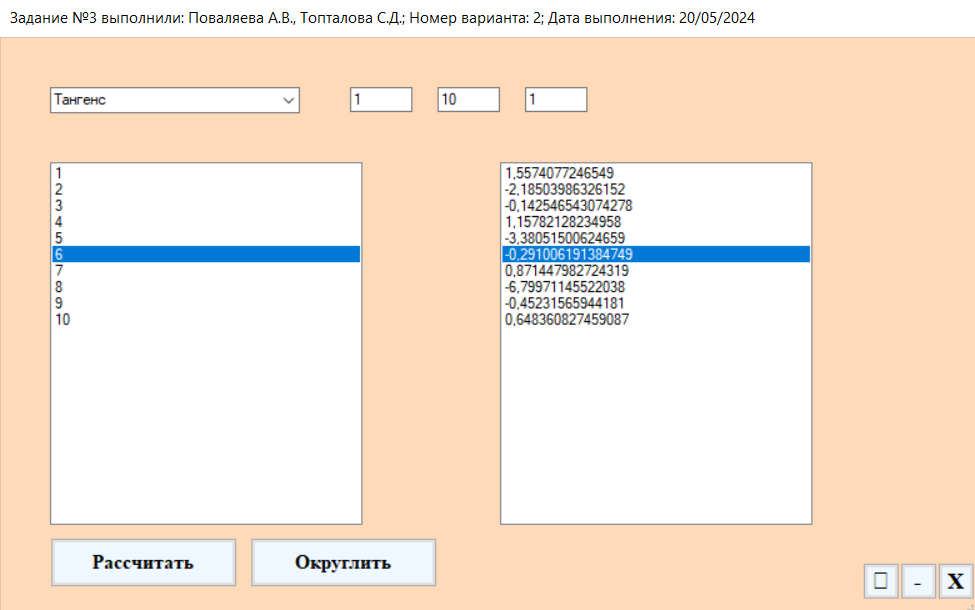


Рисунок 27, 28 – Работа с параметрами *SelectedIndex* или *SelectedItem* списков

1. **Расчёт тестовых примеров с использованием составленного программного обеспечения**

* При нажатии кнопки «Справочная информация» происходит открытие окна со справочной информацией;

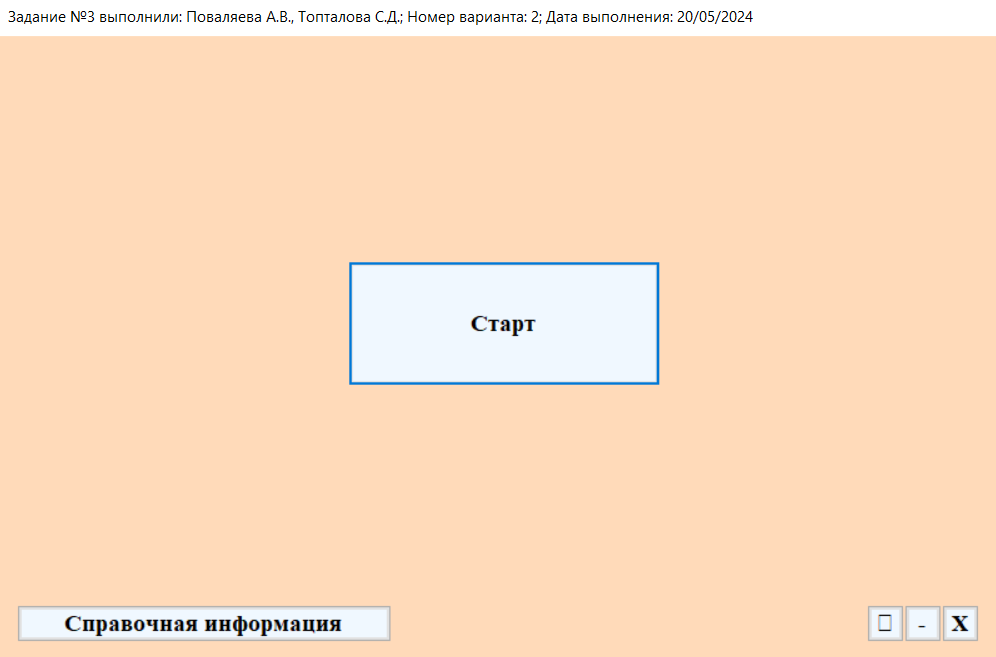


Рисунок 29 – ГПИ главного меню

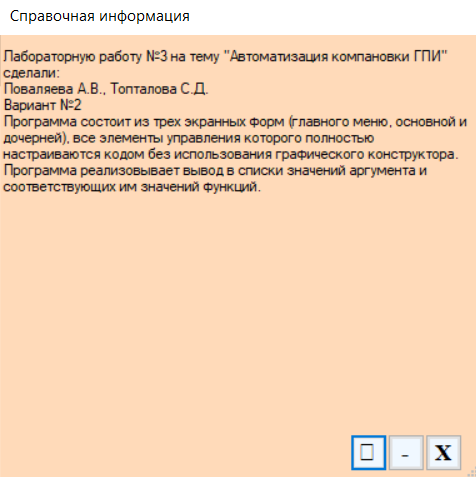


Рисунок 30 – Результат нажатия кнопки «Справочная информация» (открылось окно со справочной информацией)

* Расчет некоторой функции;

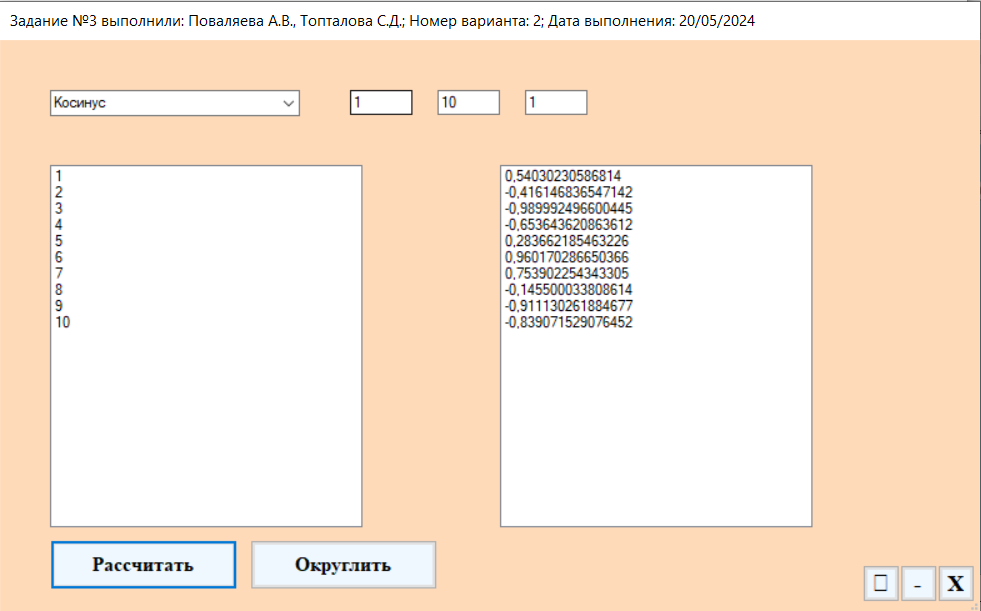


Рисунок 31 – Результат расчета некоторой функции

* Округление значений по правилу *(10 mod 5) + 1*.

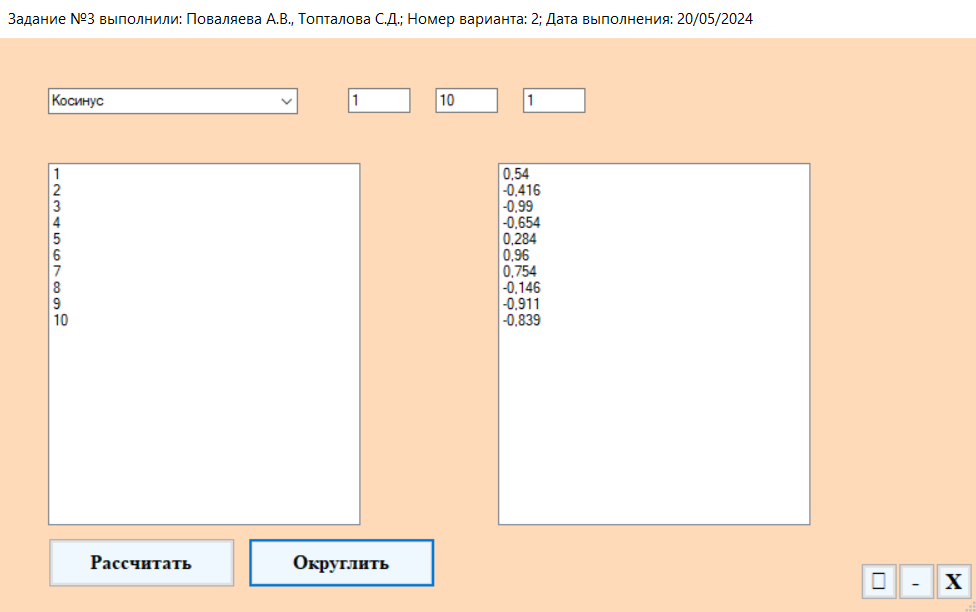


Рисунок 32 – Результат округления значений по правилу *(10 mod 5) + 1*

1. **Вывод**

В ходе выполнения лабораторной работы нами была разработана программа на языке Visual C# с использованием интегрированной среды разработки Microsoft Visual Studio. Программа представляет собой экранную форму с главным меню, позволяющим пользователю начать или прервать работу с приложением, получить справочную информацию о его использовании, а также закрыть, свернуть или развернуть в полноэкранный режим форму.

Основной функционал программы состоит в выводе в списки значений аргумента и соответствующих им значений функций. Разработанное приложение предоставляет простой и эффективный способ вычисления и анализа математических функций.

Таким образом, мы закрепили навыки разработки визуального пользовательского интерфейса, освоили работу с текстовыми файлами в среде *Microsoft Visual Studio*, научились взаимно увязывать одни элементы управления с другими, получили представление о перерисовке и перемасштабировании.