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.

Вариант №2

Проверил: доцент кафедры УиЗИ, 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 *(2 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**

**Изображение выглядит как текст, снимок экрана

Автоматически созданное описание**

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. **Тестовые примеры**

* При нажатии кнопки «Справочная информация» происходит открытие окна со справочной информацией;
* Расчет некоторой функции;
* Округление значений по правилу *(2 mod 5) + 1*.

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

using System;

using System.Drawing;

using System.Windows.Forms;

using System.IO;

namespace Lab3

{

public partial class Lab3 : Form

{

public Lab3()

{

InitializeComponent();

}

Form frm2 = new Form();

Form frm3 = new Form();

ComboBox CB3Frm2 = new ComboBox();

ComboBox CB2Frm2 = new ComboBox();

ComboBox CB1Frm2 = new ComboBox();

bool check = false;

bool checkStartFrm2 = false;

bool checkStartFrm3 = false;

string formula;

double x;

double[] result = new double[10];

string text = "";

string[] ElementsY = new string[10] { "0", "0", "0", "0", "0", "0", "0", "0", "0", "0" };

double[] Numbers = new double[] { 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5 };

double Perem;

string form;

private void Lab3\_Load(object sender, EventArgs e)

{

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

this.BackColor = Color.AliceBlue;

this.ControlBox = false;

this.Text = "MainMenu (Задание №3 выполнили: Поваляева А.В., Топталова С.Д.; Номер варианта: 2; Дата выполнения: 20/04/2024)";

Button StartB = new Button();

StartB.Size = new System.Drawing.Size(600, 70);

StartB.Location = new Point(100, 100);

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

StartB.Text = "СТАРТ";

StartB.BackColor = Color.LightGreen;

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

StartB.AutoSize = false;

StartB.Anchor = AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top;

this.Controls.Add(StartB);

Button ExitB = new Button();

ExitB.Size = new System.Drawing.Size(600, 70);

ExitB.Location = new Point(100, 250);

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

ExitB.Text = "Завершить работу";

ExitB.BackColor = Color.Crimson;

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

ExitB.AutoSize = false;

ExitB.Anchor = AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top;

this.Controls.Add(ExitB);

Label L1 = new Label();

L1.Size = new System.Drawing.Size(1000, 50);

L1.Location = new Point(-100, 15);

L1.Font = new Font("Times New Roman", 26, FontStyle.Bold);

L1.Text = "Главное меню";

L1.TextAlign = ContentAlignment.TopCenter;

L1.AutoSize = false;

L1.Anchor = AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top;

this.Controls.Add(L1);

Button CloseB = new Button();

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

CloseB.Location = new Point(730, 430);

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

CloseB.Text = "X";

CloseB.BackColor = Color.White;

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

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

this.Controls.Add(CloseB);

Button HideB = new Button();

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

HideB.Location = new Point(670, 430);

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

HideB.Text = "-";

HideB.BackColor = Color.White;

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

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

this.Controls.Add(HideB);

Button ExpandB = new Button();

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

ExpandB.Location = new Point(700, 430);

ExpandB.Font = new Font("Times New Roman",18, FontStyle.Bold);

ExpandB.Text = "▭";

ExpandB.BackColor = Color.White;

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

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

this.Controls.Add(ExpandB);

Button HelpB = new Button();

HelpB.Size = new System.Drawing.Size(300, 40);

HelpB.Location = new Point(0, 420);

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

HelpB.Text = "Справочная информация";

HelpB.BackColor = Color.White;

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

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

this.Controls.Add(HelpB);

}

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

{

Application.Exit();

}

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

{

Application.Exit();

}

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

{

this.WindowState = FormWindowState.Minimized;

}

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

{

if (check == false)

{

this.WindowState = FormWindowState.Maximized;

check = true;

}

else

{

this.WindowState = FormWindowState.Normal;

check = false;

}

}

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

{

if (checkStartFrm3 == false)

{

frm3.WindowState = FormWindowState.Normal;

frm3.FormClosed += new FormClosedEventHandler(frm3\_FormClosed);

frm3.ControlBox = false;

frm3.BackColor = Color.AliceBlue;

frm3.Size = new Size(800, 500);

frm3.FormBorderStyle = FormBorderStyle.FixedSingle;

Panel P1Frm3 = new Panel();

P1Frm3.BackColor = Color.AliceBlue;

P1Frm3.Dock = DockStyle.Bottom;

P1Frm3.Size = new Size(800, 30);

P1Frm3.Location = new Point(0, 0);

frm3.Controls.Add(P1Frm3);

TextBox TBFrm3 = new TextBox();

TBFrm3.Size = new Size(800, 420);

TBFrm3.Location = new Point(0, 60);

TBFrm3.Multiline = true;

string[] longtext = new string[30];

longtext[0] = "Лабораторную работу №3 на тему «Автоматизация компановки ГПИ» сделали:";

longtext[1] = "Поваляева А.В., Топталова С.Д.";

longtext[2] = "Программа состоит из двух экранных форм (основной и дочерней), все элементы управления которой полностью настраиваются кодом без использования графического конструктора. ";

TBFrm3.Font = new Font("Times New Roman", 12, FontStyle.Bold);

TBFrm3.Anchor = AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top | AnchorStyles.Bottom;

for (int i = 0; i < 8; i++)

{

TBFrm3.AppendText(longtext[i] + Environment.NewLine);

}

TBFrm3.BackColor = Color.Gray;

TBFrm3.ReadOnly = true;

frm3.Controls.Add(TBFrm3);

Label L1Frm3 = new Label();

L1Frm3.Size = new System.Drawing.Size(800, 50);

L1Frm3.Location = new Point(0, 10);

L1Frm3.Font = new Font("Times New Roman", 26, FontStyle.Bold);

L1Frm3.Text = "Справочная информация";

L1Frm3.TextAlign = ContentAlignment.TopCenter;

L1Frm3.AutoSize = false;

L1Frm3.Anchor = AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top;

frm3.Controls.Add(L1Frm3);

Button CloseBFrm3 = new Button();

CloseBFrm3.Size = new System.Drawing.Size(30, 30);

CloseBFrm3.Location = new Point(755, 0);

CloseBFrm3.Font = new Font("Times New Roman", 14, FontStyle.Bold);

CloseBFrm3.Text = "X";

CloseBFrm3.BackColor = Color.White;

CloseBFrm3.Click += new EventHandler(CloseBFrm3\_Click);

CloseBFrm3.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;

P1Frm3.Controls.Add(CloseBFrm3);

Button HideBFrm3 = new Button();

HideBFrm3.Size = new System.Drawing.Size(30, 30);

HideBFrm3.Location = new Point(725, 0);

HideBFrm3.Font = new Font("Times New Roman", 14, FontStyle.Bold);

HideBFrm3.Text = "-";

HideBFrm3.BackColor = Color.White;

HideBFrm3.Click += new EventHandler(HideBFrm3\_Click);

HideBFrm3.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;

P1Frm3.Controls.Add(HideBFrm3);

checkStartFrm3 = true;

frm3.Show();

}

else

{

frm3.Show();

}

}

void StartB\_Click(object sender, EventArgs e)

{

this.Hide();

if (checkStartFrm2 == false)

{

frm2.WindowState = FormWindowState.Maximized;

frm2.FormClosed += new FormClosedEventHandler(frm2\_FormClosed);

frm2.BackColor = Color.Gray;

frm2.ControlBox = false;

Panel P1Frm2 = new Panel();

P1Frm2.BackColor = Color.Khaki;

P1Frm2.Dock = DockStyle.Left;

P1Frm2.Location = new Point(300, 500);

frm2.Controls.Add(P1Frm2);

Panel P2Frm2 = new Panel();

P2Frm2.BackColor = Color.Gray;

P2Frm2.Dock = DockStyle.Bottom;

P2Frm2.Size = new Size(1000, 30);

P2Frm2.Location = new Point(0, 80);

frm2.Controls.Add(P2Frm2);

Button CloseBFrm2 = new Button();

CloseBFrm2.Size = new System.Drawing.Size(30, 30);

CloseBFrm2.Location = new Point(255, 0);

CloseBFrm2.Font = new Font("Times New Roman", 14, FontStyle.Bold);

CloseBFrm2.Text = "X";

CloseBFrm2.BackColor = Color.White;

CloseBFrm2.Click += new EventHandler(CloseBFrm2\_Click);

CloseBFrm2.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;

P2Frm2.Controls.Add(CloseBFrm2);

Button ExpandBFrm2 = new Button();

ExpandBFrm2.Size = new System.Drawing.Size(30, 30);

ExpandBFrm2.Location = new Point(227, 0);

ExpandBFrm2.Font = new Font("Times New Roman", 18, FontStyle.Bold);

ExpandBFrm2.Text = "▭";

ExpandBFrm2.BackColor = Color.White;

ExpandBFrm2.Click += new EventHandler(ExpandBFrm2\_Click);

ExpandBFrm2.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;

P2Frm2.Controls.Add(ExpandBFrm2);

Button HideBFrm2 = new Button();

HideBFrm2.Size = new System.Drawing.Size(30, 30);

HideBFrm2.Location = new Point(199, 0);

HideBFrm2.Font = new Font("Times New Roman", 14, FontStyle.Bold);

HideBFrm2.Text = "-";

HideBFrm2.BackColor = Color.White;

HideBFrm2.Click += new EventHandler(HideBFrm2\_Click);

HideBFrm2.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;

P2Frm2.Controls.Add(HideBFrm2);

Button ConfirmFrm2 = new Button();

ConfirmFrm2.Size = new System.Drawing.Size(179, 70);

ConfirmFrm2.Location = new Point(10, 130);

ConfirmFrm2.Font = new Font("Times New Roman", 14, FontStyle.Bold);

ConfirmFrm2.Text = "Сохранить без округления";

ConfirmFrm2.BackColor = Color.White;

ConfirmFrm2.Click += new EventHandler(ConfirmFrm2\_Click);

ConfirmFrm2.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;

P1Frm2.Controls.Add(ConfirmFrm2);

Button RoundFrm2 = new Button();

RoundFrm2.Size = new System.Drawing.Size(179, 70);

RoundFrm2.Location = new Point(10, 70);

RoundFrm2.Font = new Font("Times New Roman", 14, FontStyle.Bold);

RoundFrm2.Text = "Округлить и сохранить";

RoundFrm2.BackColor = Color.White;

RoundFrm2.Click += new EventHandler(RoundFrm2\_Click);

RoundFrm2.Anchor = AnchorStyles.Bottom | AnchorStyles.Right;

P1Frm2.Controls.Add(RoundFrm2);

CB1Frm2.Size = new System.Drawing.Size(180, 20);

CB1Frm2.Location = new Point(10, 50);

CB1Frm2.SelectedIndexChanged += CB1Frm2\_SelectedIndexChanged;

CB1Frm2.Font = new Font("Times New Roman", 11, FontStyle.Regular);

CB1Frm2.DropDownStyle = ComboBoxStyle.DropDownList;

CB1Frm2.Text = "Формулы";

CB1Frm2.Items.AddRange(new string[] { "(x^2-x+7)/(x+1)", "Косинус", "Арктангенс", "Извлечение корня", "Арксинус", "Тангенс", "Десятичный логарифм", "Синус", "Арккосинус", "Логарифм по основанию 2", "Натуральный логарифм" });

P1Frm2.Controls.Add(CB1Frm2);

CB2Frm2.Size = new System.Drawing.Size(180, 20);

CB2Frm2.Location = new Point(300, 50);

CB2Frm2.SelectedIndexChanged += CB2Frm2\_SelectedIndexChanged;

CB2Frm2.Font = new Font("Times New Roman", 11, FontStyle.Regular);

CB2Frm2.DropDownStyle = ComboBoxStyle.DropDownList;

CB2Frm2.Items.AddRange(new string[] { "0.5", "1", "1.5", "2", "2.5", "3", "3.5", "4", "4.5", "5" });

frm2.Controls.Add(CB2Frm2);

CB3Frm2.Size = new System.Drawing.Size(180, 20);

CB3Frm2.Location = new Point(1100, 50);

CB3Frm2.SelectedIndexChanged += CB3Frm2\_SelectedIndexChanged;

CB3Frm2.Font = new Font("Times New Roman", 11, FontStyle.Regular);

CB3Frm2.DropDownStyle = ComboBoxStyle.DropDownList;

CB3Frm2.Items.AddRange(ElementsY);

frm2.Controls.Add(CB3Frm2);

Label L1Frm2 = new Label();

L1Frm2.Size = new System.Drawing.Size(150, 50);

L1Frm2.Location = new Point(0, 10);

L1Frm2.Font = new Font("Times New Roman", 20, FontStyle.Bold);

L1Frm2.Text = "Формулы:";

L1Frm2.TextAlign = ContentAlignment.TopCenter;

L1Frm2.AutoSize = false;

L1Frm2.Anchor = AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top;

P1Frm2.Controls.Add(L1Frm2);

Label L2Frm2 = new Label();

L2Frm2.Size = new System.Drawing.Size(180, 50);

L2Frm2.Location = new Point(300, 0);

L2Frm2.Font = new Font("Times New Roman", 20, FontStyle.Bold);

L2Frm2.Text = "Значения:";

L2Frm2.TextAlign = ContentAlignment.TopCenter;

L2Frm2.AutoSize = false;

frm2.Controls.Add(L2Frm2);

Label L3Frm2 = new Label();

L3Frm2.Size = new System.Drawing.Size(180, 50);

L3Frm2.Location = new Point(1100, 0);

L3Frm2.Font = new Font("Times New Roman", 20, FontStyle.Bold);

L3Frm2.Text = "Результат:";

L3Frm2.TextAlign = ContentAlignment.TopCenter;

L3Frm2.AutoSize = false;

frm2.Controls.Add(L3Frm2); // Добавление кнопки на форму

frm2.ShowDialog();

}

else

{

frm2.Show();

}

}

private void frm2\_FormClosed(object sender, FormClosedEventArgs e)

{

this.Close();

}

private void CloseBFrm2\_Click(object sender, EventArgs e)

{

checkStartFrm2 = true;

frm2.Hide();

this.Show();

}

bool checkFrm2 = false;

private void ExpandBFrm2\_Click(object sender, EventArgs e)

{

if (checkFrm2 == false)

{

frm2.WindowState = FormWindowState.Maximized;

checkFrm2 = true;

}

else

{

frm2.WindowState = FormWindowState.Normal;

checkFrm2 = false;

}

}

private void HideBFrm2\_Click(object sender, EventArgs e)

{

frm2.WindowState = FormWindowState.Minimized;

}

void CB1Frm2\_SelectedIndexChanged(object sender, EventArgs e)

{

ComboBox CB1Frm2 = (ComboBox)sender;

string selectedFormula = (string)CB1Frm2.SelectedItem;

switch (selectedFormula)

{

case "(x^2-x+7)/(x+1)":

formula = "1";

for (int i = 0; i < 10; i++)

{

Convert.ToDouble(Perem);

Perem = (((Numbers[i] \* Numbers[i]) - Numbers[i]) + 7) / (Numbers[i] + 1);

ElementsY[i] = Convert.ToString(Perem);

}

CB3Frm2.Items.Clear();

foreach (string element in ElementsY)

{

CB3Frm2.Items.Add(element);

}

form = "(x^2-x+7)/(x+1)";

break;

case "Косинус":

formula = "2";

for (int i = 0; i < 10; i++)

{

Convert.ToDouble(Perem);

Perem = Math.Cos(Numbers[i]);

ElementsY[i] = Convert.ToString(Perem);

}

CB3Frm2.Items.Clear();

foreach (string element in ElementsY)

{

CB3Frm2.Items.Add(element);

}

form = "Косинус";

break;

case "Возведение в квадрат":

formula = "3";

for (int i = 0; i < 10; i++)

{

Convert.ToDouble(Perem);

Perem = Math.Pow(Numbers[i], 2);

ElementsY[i] = Convert.ToString(Perem);

}

CB3Frm2.Items.Clear();

foreach (string element in ElementsY)

{

CB3Frm2.Items.Add(element);

}

form = "Возведение в квадрат";

break;

case "Арктангенс":

formula = "4";

for (int i = 0; i < 10; i++)

{

Convert.ToDouble(Perem);

Perem = Math.Atan(Numbers[i]);

ElementsY[i] = Convert.ToString(Perem);

}

CB3Frm2.Items.Clear();

foreach (string element in ElementsY)

{

CB3Frm2.Items.Add(element);

}

form = "Арктангенс";

break;

case "Извлечение корня":

formula = "5";

for (int i = 0; i < 10; i++)

{

Convert.ToDouble(Perem);

Perem = Math.Sqrt(Numbers[i]);

ElementsY[i] = Convert.ToString(Perem);

}

CB3Frm2.Items.Clear();

foreach (string element in ElementsY)

{

CB3Frm2.Items.Add(element);

}

form = "Извлечение корня";

break;

case "Арксинус":

formula = "6";

for (int i = 0; i < 10; i++)

{

Convert.ToDouble(Perem);

Perem = Math.Asin(Numbers[i]);

ElementsY[i] = Convert.ToString(Perem);

}

CB3Frm2.Items.Clear();

foreach (string element in ElementsY)

{

CB3Frm2.Items.Add(element);

}

form = "Арксинус";

break;

case "Тангенс":

formula = "7";

for (int i = 0; i < 10; i++)

{

Convert.ToDouble(Perem);

Perem = Math.Tan(Numbers[i]);

ElementsY[i] = Convert.ToString(Perem);

}

CB3Frm2.Items.Clear();

foreach (string element in ElementsY)

{

CB3Frm2.Items.Add(element);

}

form = "Тангенс";

break;

case "Десятичный логарифм":

formula = "8";

for (int i = 0; i < 10; i++)

{

Convert.ToDouble(Perem);

Perem = Math.Log10(Numbers[i]);

ElementsY[i] = Convert.ToString(Perem);

}

CB3Frm2.Items.Clear();

foreach (string element in ElementsY)

{

CB3Frm2.Items.Add(element);

}

form = "Десятичный логарифм";

break;

case "Синус":

formula = "9";

for (int i = 0; i < 10; i++)

{

Convert.ToDouble(Perem);

Perem = Math.Sin(Numbers[i]);

ElementsY[i] = Convert.ToString(Perem);

}

CB3Frm2.Items.Clear();

foreach (string element in ElementsY)

{

CB3Frm2.Items.Add(element);

}

form = "Синус";

break;

case "Арккосинус":

formula = "10";

for (int i = 0; i < 10; i++)

{

Convert.ToDouble(Perem);

Perem = Math.Acos(Numbers[i]);

ElementsY[i] = Convert.ToString(Perem);

}

CB3Frm2.Items.Clear();

foreach (string element in ElementsY)

{

CB3Frm2.Items.Add(element);

}

form = "Арккосинус";

break;

case "Логарифм по основанию 2":

formula = "11";

for (int i = 0; i < 10; i++)

{

Convert.ToDouble(Perem);

Perem = Math.Log(Numbers[i]) / (Math.Log(2));

ElementsY[i] = Convert.ToString(Perem);

}

CB3Frm2.Items.Clear();

foreach (string element in ElementsY)

{

CB3Frm2.Items.Add(element);

}

form = "Логарифм по основанию 2";

break;

case "Натуральный логарифм":

formula = "12";

for (int i = 0; i < 10; i++)

{

Convert.ToDouble(Perem);

Perem = Math.Log(Numbers[i]);

ElementsY[i] = Convert.ToString(Perem);

}

CB3Frm2.Items.Clear();

foreach (string element in ElementsY)

{

CB3Frm2.Items.Add(element);

}

form = "Натуральный логарифм";

break;

}

}

void CB2Frm2\_SelectedIndexChanged(object sender, EventArgs e)

{

ComboBox CB2Frm2 = (ComboBox)sender;

string selectedNumber = (string)CB2Frm2.SelectedItem;

switch (selectedNumber)

{

case "0.5":

x = 0.5;

MessageBox.Show("x = 0.5");

break;

case "1":

x = 1;

MessageBox.Show("x = 1");

break;

case "1.5":

x = 1.5;

MessageBox.Show("x = 1.5");

break;

case "2":

x = 2;

MessageBox.Show("x = 2");

break;

case "2.5":

x = 2.5;

MessageBox.Show("x = 2.5");

break;

case "3":

x = 3;

MessageBox.Show("x = 3");

break;

case "3.5":

x = 3.5;

MessageBox.Show("x = 3.5");

break;

case "4":

x = 4;

MessageBox.Show("x = 4");

break;

case "4.5":

x = 4.5;

MessageBox.Show("x = 4.5");

break;

case "5":

x = 5;

MessageBox.Show("x = 5");

break;

default:

break;

}

CB3Frm2.SelectedIndex = CB2Frm2.SelectedIndex;

}

void RoundFrm2\_Click(object sender, EventArgs e)

{

switch (formula)

{

case "1":

for (int i = 0; i < 10; i++)

{

double math\_epression = (((Numbers[i] \* Numbers[i]) - Numbers[i]) + 7) / (Numbers[i] + 1);

result[i] = Math.Round(math\_epression, 2);

}

break;

case "2":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Round(Math.Cos(Numbers[i]), 2);

}

break;

case "3":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Round(Math.Pow(Numbers[i], 2), 2);

}

break;

case "4":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Round(Math.Atan(Numbers[i]), 2);

}

break;

case "5":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Round(Math.Sqrt(Numbers[i]), 2);

}

break;

case "6":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Round(Math.Asin(Numbers[i]), 2);

}

break;

case "7":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Round(Math.Tan(Numbers[i]), 2);

}

break;

case "8":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Round(Math.Log10(Numbers[i]), 2);

}

break;

case "9":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Round(Math.Sin(Numbers[i]), 2);

}

break;

case "10":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Round(Math.Acos(Numbers[i]), 2);

}

break;

case "11":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Round(Math.Log(Numbers[i]) / (Math.Log(2)), 2);

}

break;

case "12":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Round(Math.Log(Numbers[i]), 2);

}

break;

default:

break;

}

//MessageBox.Show("Результат = " + result);

text = form + ":\n";

for (int i = 0; i < 10; i++)

{

text += Numbers[i].ToString() + "% " + result[i].ToString() + "\n";

}

File.WriteAllText(@"..\\..\\" + "test\_rounded.txt", text);

}

void CB3Frm2\_SelectedIndexChanged(object sender, EventArgs e)

{

CB2Frm2.SelectedIndex = CB3Frm2.SelectedIndex;

}

void ConfirmFrm2\_Click(object sender, EventArgs e)

{

switch (formula)

{

case "1":

for (int i = 0; i < 10; i++)

{

result[i] = (((Numbers[i] \* Numbers[i]) - Numbers[i]) + 7) / (Numbers[i] + 1);

}

break;

case "2":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Cos(Numbers[i]);

}

break;

case "3":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Pow(Numbers[i], 2);

}

break;

case "4":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Atan(Numbers[i]);

}

break;

case "5":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Sqrt(Numbers[i]);

}

break;

case "6":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Asin(Numbers[i]);

}

break;

case "7":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Tan(Numbers[i]);

}

break;

case "8":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Log10(Numbers[i]);

}

break;

case "9":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Sin(Numbers[i]);

}

break;

case "10":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Acos(Numbers[i]);

}

break;

case "11":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Log(Numbers[i]) / (Math.Log(2));

}

break;

case "12":

for (int i = 0; i < 10; i++)

{

result[i] = Math.Log(Numbers[i]);

}

break;

default:

break;

}

text = form + ":\n";

for (int i = 0; i < 10; i++)

{

text += Numbers[i].ToString() + "% " + result[i].ToString() + "\n";

}

File.WriteAllText(@"..\\..\\" + "test\_nonrounded.txt", text);

}

private void frm3\_FormClosed(object sender, FormClosedEventArgs e)

{

this.Close();

}

private void CloseBFrm3\_Click(object sender, EventArgs e)

{

checkStartFrm3 = true;

frm3.Hide();

}

private void HideBFrm3\_Click(object sender, EventArgs e)

{

frm3.WindowState = FormWindowState.Minimized;

}

}

}

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

* **ГПИ стартовой страницы**

На рисунке 5 показано, что на стартовой странице находится 6 кнопок. Кнопка «Старт» запускает работу приложения (перебрасывает пользователя на страницу с функциями), кнопка «Завершить работу» закрывает приложение. При нажатии кнопки «Справочная информация» пользователь получает информацию о программе. В правом нижнем углу страницы находятся кнопки «Закрыть», «Развернуть в полноэкранный режим» и «Свернуть» для взаимодействия с формой.

**Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание**

Рисунок 5 – ГПИ стартовой страницы

* ГПИ справочной информации

На рисунке 6 показана краткая справочная информация о программе. Справочная информация вызвана в дочерней экранной форме.

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание**

Рисунок 6 – ГПИ справочной информации

* ГПИ программы

**Изображение выглядит как текст, снимок экрана, диаграмма

Автоматически созданное описание**

Рисунок 7 – ГПИ приложения

Изображение выглядит как текст, снимок экрана, диаграмма

Автоматически созданное описание

Рисунок 8 – Результат работы программы (пользователь выбрал извлечение корня и выбрал число 4)

Изображение выглядит как текст, снимок экрана, диаграмма

Автоматически созданное описание

Рисунок 9 – Результат работы программы (пользователь выбрал в синус, число 2.5)

Изображение выглядит как текст, снимок экрана, диаграмма

Автоматически созданное описание

Рисунок 10 – Результат работы программы (пользователь выбрал тангенс, число 1)

Изображение выглядит как текст, снимок экрана, диаграмма

Автоматически созданное описание

Рисунок 11 – Результат работы программы (пользователь выбрал натуральный логарифм, число 5)

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 12 – Заголовок экранной формы

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



Рисунок 13– Авторские кнопки «Свернуть», «Развернуть», «Закрыть» внизу экранной формы.

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

Изображение выглядит как текст, снимок экрана, программное обеспечение, компьютер

Автоматически созданное описание

Рисунок 14 – Режим конструктора

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

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рисунок 15 – 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*);

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 16 – Окно со справочной информацией до изменения размера

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 17 – Окно со справочной информацией после изменения размера

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

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 18 – 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;

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 19 – 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 *(2 mod 5) + 1* для знаков после плавающей запятой данными о функциональной зависимости из файла.

Изображение выглядит как текст, Шрифт, снимок экрана, типография

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 20,21 – 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;

**Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание**

Рисунок 22 – Форма до нажатия на кнопку «Старт»

Изображение выглядит как текст, снимок экрана, диаграмма

Автоматически созданное описание

Рисунок 23 – Форма после нажатия на кнопку «Старт»

* 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.;

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Рисунок 24 – Значащие имена переменных

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

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

**Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание**

Рисунок 25 – ГПИ главного меню

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 26 – Результат нажатия кнопки «Справочная информация» (открылось окно со справочной информацией)

* Расчет некоторой функции;

Изображение выглядит как текст, снимок экрана, диаграмма

Автоматически созданное описание

Рисунок 32 – Результат расчета некоторой функции

* Округление значений по правилу *(2 mod 5) + 1*.

Изображение выглядит как текст, Шрифт, снимок экрана, типография

Автоматически созданное описание

Рисунок 33 – Результат округления значений по правилу *(2 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.