**МИНИСТЕРСТВО ТРАНСПОРТА**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ**

**АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО**

**ОБРАЗОВАНИЯ**

**“РОССИЙСКИЙ УНИВЕРСИТЕТ**

**ТРАНСПОРТА” (МИИТ)**

**Институт транспортной техники и систем**

**управления**

**(ИТТСУ)**

**Кафедра “Управление и защита информации”**

**Тема: “Лабораторная работа №3”**

**Выполнили студенты группы ТКИ-342**

**Савин Лев Андреевич   
Потапов Артём Константинович**

**Специальность 10.05.01 “Компьютерная безопасность”**

**Преподаватель: к.т.н. Сафронов А.И.**

**Москва**

**2024**

1. Формулировка цели работы:

Разработать программу в среде Visual Studio с использованием Windows Forms Application на языке Visual C#, которая представляет собой пользовательский интерфейс для работы с математическими функциями. Программа должна предоставлять возможность начать и прервать работу, а также получить справочную информацию о работе с приложением.

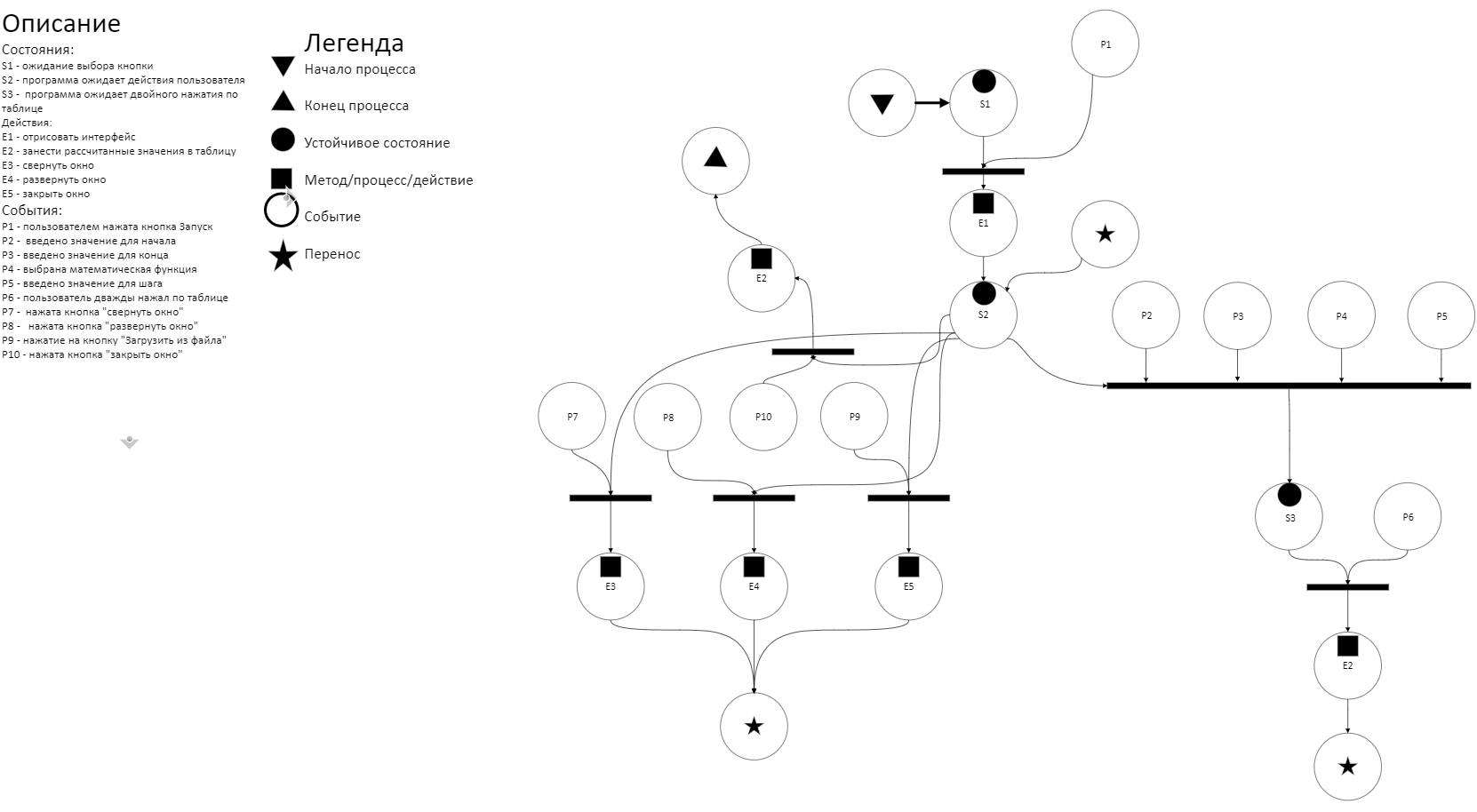
2. Описание задачи:

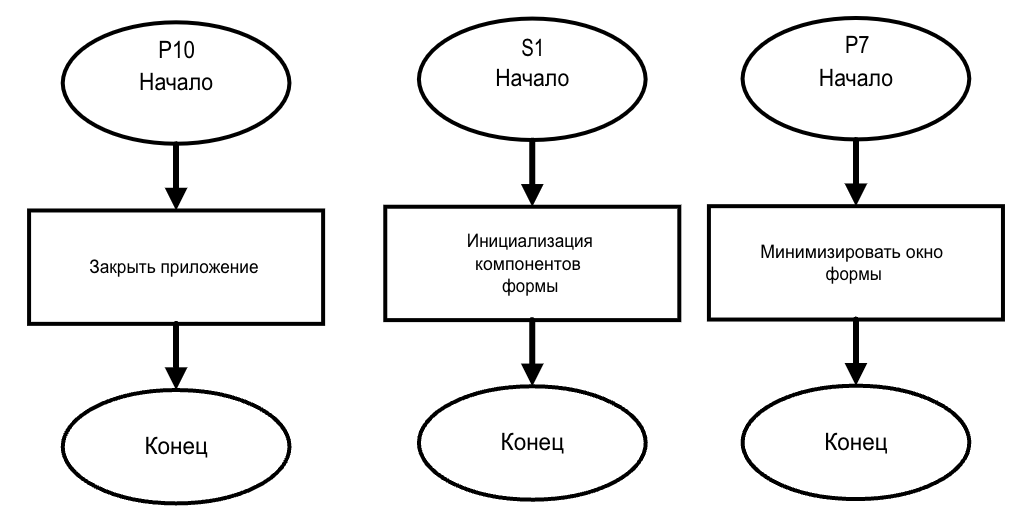
Разработать программу, которая выводит списки значений аргумента и соответствующих им значений функций. Список функций должен включать в себя 11 пунктов, таких как логарифм по основанию 2, тангенс, косинус и др. Кроме того, предусмотрена возможность заполнения списков данными из файла и обновление основных списков для разных вариантов.

3. Составление диаграммы классов:

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание



4. Составление сети Петри запрограммированного технологического процесса:   
  
5. Составление схем алгоритмов методов в составе решения, отмеченных на сети Петри в качестве «эффектов» (метка).  
  
Изображение выглядит как текст, круг, диаграмма, снимок экрана

Автоматически созданное описание

Изображение выглядит как диаграмма, зарисовка, рисунок, Технический чертеж

Автоматически созданное описание

6. Подбор тестовых примеров:

Для тестирования программы выберем следующие цели: расчёт значений некоторой функции; выгрузка значений из файла в дополнительные поля; автоматический выбор абсциссы(ординаты), соответствующей вручную выбранной ординате(абсциссе); открытие окна со справкой; попытка заставить программу построить миллион значений.

7. Листинг (код) программного обеспечения:

using System;

using System.Drawing;

using System.IO;

using System.Windows.Forms;

namespace ProgMethods\_Work3\_

{

public partial class FrmMain : Form

{

public FrmMain()

{

InitializeComponent();

}

private Button btnStart;

private TextBox tbFuncStart;

private TextBox tbFuncStep;

private TextBox tbFuncStop;

private Button btnInfo;

private ListBox lbX;

private ListBox lbY;

private ListBox lbFunc;

private ListBox lbFileX;

private ListBox lbFileY;

private Button btnLoad;

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

{

InitComponents();

this.MinimumSize = new System.Drawing.Size(500, 320);

foreach (Control cntrl in this.Controls)

{

cntrl.Visible = false;

}

btnStart.Visible = true;

}

private void InitComponents()

{

BackColor = SystemColors.ActiveCaption;

Cursor = Cursors.Hand;

//btnStart

btnStart = new Button

{

Text = "Запуск!",

Location = new Point(20, 20),

Size = new Size(75, 25),

BackColor = Color.White

};

btnStart.Click += BtnStart\_Click;

//btnInfo

btnInfo = new Button

{

Text = "Info",

Location = new Point(this.Width - 75, 20),

Size = new Size(50, 25),

BackColor = Color.White,

Anchor = AnchorStyles.Right | AnchorStyles.Top

};

btnInfo.Click += BtnInfo\_Click;

//tbFuncStart

tbFuncStart = new TextBox

{

Text = "Start",

ForeColor = Color.Silver,

Location = new Point(btnStart.Right + 10, 22),

Size = new Size(50, 20)

};

tbFuncStart.Enter += TbFuncStart\_Enter;

tbFuncStart.Leave += TbFuncStart\_Leave;

tbFuncStart.KeyPress += TbFuncStart\_KeyPress;

//tbFuncStep

tbFuncStep = new TextBox

{

Text = "Step",

ForeColor = Color.Silver,

Location = new Point(tbFuncStart.Right + 10, 22),

Size = new Size(50, 20)

};

tbFuncStep.Enter += TbFuncStep\_Enter;

tbFuncStep.Leave += TbFuncStep\_Leave;

tbFuncStep.KeyPress += TbFuncStep\_KeyPress;

//tbFuncStop

tbFuncStop = new TextBox

{

Text = "Stop",

ForeColor = Color.Silver,

Location = new Point(tbFuncStep.Right + 10, 22),

Size = new Size(50, 20)

};

tbFuncStop.Enter += TbFuncStop\_Enter;

tbFuncStop.Leave += TbFuncStop\_Leave;

tbFuncStop.KeyPress += TbFuncStop\_KeyPress;

//lbX

lbX = new ListBox

{

Location = new Point(tbFuncStart.Left, 62),

Size = new Size(tbFuncStart.Width + tbFuncStep.Left - tbFuncStart.Right + tbFuncStep.Width / 2, 200)

};

lbX.DoubleClick += LbX\_DoubleClick;

lbX.SelectedIndexChanged += LbX\_SelectedIndexChanged;

//lbY

lbY = new ListBox

{

Location = new Point(tbFuncStep.Right - tbFuncStep.Width / 2, 62),

Size = new Size(tbFuncStart.Width + tbFuncStep.Left - tbFuncStart.Right + tbFuncStep.Width / 2, 200)

};

lbY.DoubleClick += LbX\_DoubleClick;

lbY.SelectedIndexChanged += LbY\_SelectedIndexChanged;

//lbFunc

lbFunc = new ListBox

{

Location = new Point(btnStart.Left, 62),

Size = new Size(btnStart.Width, 170),

};

lbFunc.Items.Add("Log\_e X");

lbFunc.Items.Add("X^2");

lbFunc.Items.Add("1 / (Log\_2 X)");

lbFunc.Items.Add("ArcSin X");

lbFunc.Items.Add("Cos X");

lbFunc.Items.Add("Log\_2 X");

lbFunc.Items.Add("Sin X");

lbFunc.Items.Add("Log\_10 X");

lbFunc.Items.Add("ArcCos X");

lbFunc.Items.Add("Sqrt X");

lbFunc.Items.Add("ArcTg X");

lbFunc.Items.Add("Tan X");

//lbFileX

lbFileX = new ListBox

{

Location = new Point(lbY.Right + 20, lbY.Top),

Size = new Size(lbX.Width, lbX.Height)

};

lbFileX.SelectedIndexChanged += LbFileX\_SelectedIndexChanged;

//lbFiley

lbFileY = new ListBox

{

Location = new Point(lbFileX.Right, lbFileX.Top),

Size = new Size(lbY.Width, lbY.Height)

};

lbFileY.SelectedIndexChanged += LbFileY\_SelectedIndexChanged;

//btnLoad

btnLoad = new Button

{

Location = new Point(tbFuncStop.Right + 20, tbFuncStop.Top),

Size = new Size(btnStart.Width, btnStart.Height),

Text = "Загрузить",

BackColor = Color.White

};

btnLoad.Click += BtnLoad\_Click;

Controls.Add(btnStart);

Controls.Add(btnInfo);

Controls.Add(tbFuncStart);

Controls.Add(tbFuncStep);

Controls.Add(tbFuncStop);

Controls.Add(lbX);

Controls.Add(lbY);

Controls.Add(lbFunc);

Controls.Add(lbFileX);

Controls.Add(lbFileY);

Controls.Add(btnLoad);

}

private void TbFuncStop\_KeyPress(object sender, KeyPressEventArgs e)

{

char num = e.KeyChar;

if (!Char.IsDigit(num) && num != 8 && num != 44)

{

e.Handled = true;

}

}

private void TbFuncStep\_KeyPress(object sender, KeyPressEventArgs e)

{

char num = e.KeyChar;

if (!Char.IsDigit(num) && num != 8 && num != 44)

{

e.Handled = true;

}

}

private void TbFuncStart\_KeyPress(object sender, KeyPressEventArgs e)

{

char num = e.KeyChar;

if (!Char.IsDigit(num) && num != 8 && num != 44)

{

e.Handled = true;

}

}

private void LbFileY\_SelectedIndexChanged(object sender, EventArgs e)

{

lbFileX.SelectedIndex = lbFileY.SelectedIndex;

}

private void LbFileX\_SelectedIndexChanged(object sender, EventArgs e)

{

lbFileY.TopIndex = lbFileX.TopIndex;

lbFileY.SelectedIndex = lbFileX.SelectedIndex;

}

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

{

lbFileX.Visible = true;

lbFileY.Visible = true;

LoadDataFromFile();

}

private void LoadDataFromFile()

{

lbFileX.Items.Clear();

lbFileY.Items.Clear();

StreamReader streamReader = new StreamReader("data.txt", encoding:System.Text.Encoding.Unicode);

string strX = "";

int x;

while (!streamReader.EndOfStream)

{

strX = "";

while ((x = streamReader.Read()) != 36)

{

strX += (char)x;

}

lbFileX.Items.Add(string.Format("{0:2f}", strX));

lbFileY.Items.Add(Math.Round(Convert.ToDouble(streamReader.ReadLine()), 2));

}

streamReader.Close();

}

private void LbX\_SelectedIndexChanged(object sender, EventArgs e)

{

lbY.SelectedIndex = lbX.SelectedIndex;

lbY.TopIndex = lbX.TopIndex;

}

private void LbY\_SelectedIndexChanged(object sender, EventArgs e)

{

lbX.SelectedIndex = lbY.SelectedIndex;

}

private void LbX\_DoubleClick(object sender, EventArgs e)

{

if (lbFunc.SelectedIndex != -1 && tbFuncStart.Text != "Start" && tbFuncStep.Text != "Step" && tbFuncStop.Text != "Stop")

{

lbX.Items.Clear();

lbY.Items.Clear();

double.TryParse(tbFuncStop.Text, out double stop);

double.TryParse(tbFuncStep.Text, out double step);

StreamWriter streamWriter = new StreamWriter("data.txt", false, encoding:System.Text.Encoding.Unicode);

int counter = 0;

for (double.TryParse(tbFuncStart.Text, out double i); i <= stop; i += step)

{

++counter;

lbX.Items.Add(i);

streamWriter.Write(i + "$");

switch (lbFunc.SelectedIndex)

{

case 0:

lbY.Items.Add(Math.Log(i));

break;

case 1:

lbY.Items.Add(i \* i);

break;

case 2:

lbY.Items.Add(1.0 / (Math.Log(i, 2)));

break;

case 3:

lbY.Items.Add(Math.Asin(i));

break;

case 4:

lbY.Items.Add(Math.Cos(i));

break;

case 5:

lbY.Items.Add(Math.Log(i, 2));

break;

case 6:

lbY.Items.Add(Math.Sin(i));

break;

case 7:

lbY.Items.Add(Math.Log10(i));

break;

case 8:

lbY.Items.Add(Math.Acos(i));

break;

case 9:

lbY.Items.Add(Math.Sqrt(i));

break;

case 10:

lbY.Items.Add(Math.Atan(i));

break;

case 11:

lbY.Items.Add(Math.Tan(i));

break;

default:

lbY.Items.Add("NaN!!!");

break;

}

streamWriter.WriteLine(lbY.Items[lbY.Items.Count - 1]);

if (counter > 1500)

{

MessageBox.Show($"You try to calculate more than {counter - 1} items! Break");

break;

}

}

lbX.EndUpdate();

streamWriter.Close();

}

}

private void TbFuncStop\_Enter(object sender, EventArgs e)

{

if (tbFuncStop.Text == "Stop")

{

tbFuncStop.Text = "";

tbFuncStop.ForeColor = Color.Black;

}

}

private void TbFuncStop\_Leave(object sender, EventArgs e)

{

if (tbFuncStop.Text == "")

{

tbFuncStop.Text = "Stop";

tbFuncStop.ForeColor = Color.Silver;

}

}

private void TbFuncStep\_Enter(object sender, EventArgs e)

{

if (tbFuncStep.Text == "Step")

{

tbFuncStep.Text = "";

tbFuncStep.ForeColor = Color.Black;

}

}

private void TbFuncStep\_Leave(object sender, EventArgs e)

{

if (tbFuncStep.Text == "")

{

tbFuncStep.Text = "Step";

tbFuncStep.ForeColor = Color.Silver;

}

}

private void TbFuncStart\_Enter(object sender, EventArgs e)

{

if (tbFuncStart.Text == "Start")

{

tbFuncStart.Text = "";

tbFuncStart.ForeColor = Color.Black;

}

}

private void TbFuncStart\_Leave(object sender, EventArgs e)

{

if (tbFuncStart.Text == "")

{

tbFuncStart.Text = "Start";

tbFuncStart.ForeColor = Color.Silver;

}

}

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

{

foreach (Control cntrl in this.Controls)

{

cntrl.Visible = true;

}

lbFileX.Visible = false;

lbFileY.Visible = false;

}

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

{

FrmInfo frmInfo = new FrmInfo();

frmInfo.Owner = this;

frmInfo.StartPosition = FormStartPosition.CenterScreen;

frmInfo.ShowDialog();

frmInfo.Dispose();

}

}

public partial class FrmInfo : Form

{

private TextBox tbInfo;

public FrmInfo()

{

tbInfo = new TextBox

{

Location = new Point(10, 10),

Size = new Size(Width - 40, Height - 60),

Text = "Справка о программе.\n" +

"В данной программе реализована возможность автоматического построения " +

"значений аргумента и функции при заданном начальном положении, шаге и конечном" +

"положении. Для использования программы заполните соответствующие поля, выберите функцию из списка, " +

"а затем выполните двойное нажатие ЛКМ по одному из полей, зарезервированных для абсциссы и ординаты." +

"Результатом станет заполнение этих полей данными, а также выгрузка зависимости в файл. Существует возможность" +

"выгрузить данные из файла с окргулением до двух знаков при нажатии соответствующей кнопки.",

Font = new Font("Times New Roman", 14),

Multiline = true,

ReadOnly = true

};

Controls.Add(tbInfo);

Resize += FrmInfo\_Resize;

MinimumSize = new Size(150, 150);

}

private void FrmInfo\_Resize(object sender, EventArgs e)

{

tbInfo.Size = new Size(Width - 40, Height - 60);

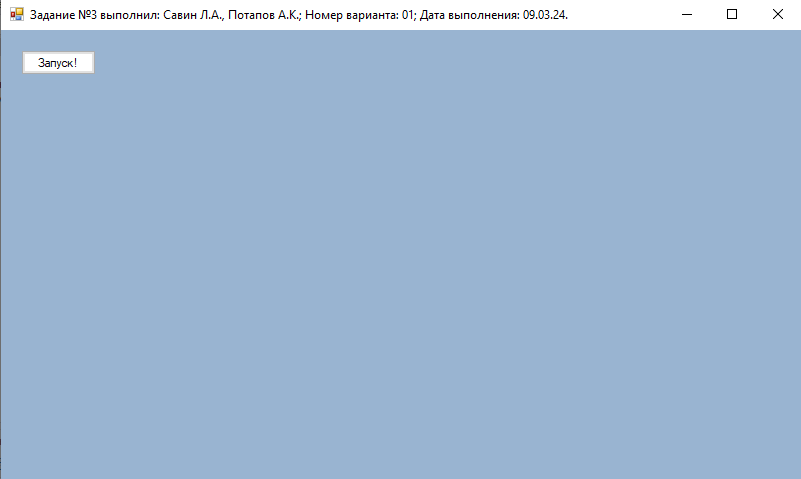
}

}

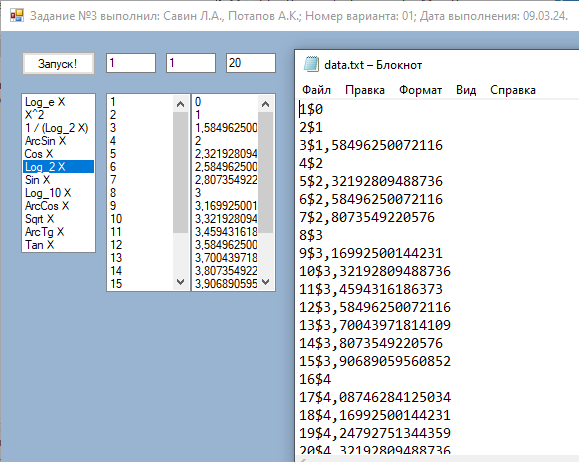
}

8. Графический пользовательский интерфейс:

При запуске программы открывается следующее окно, в котором функциональные элементы скрыты для пользователя:

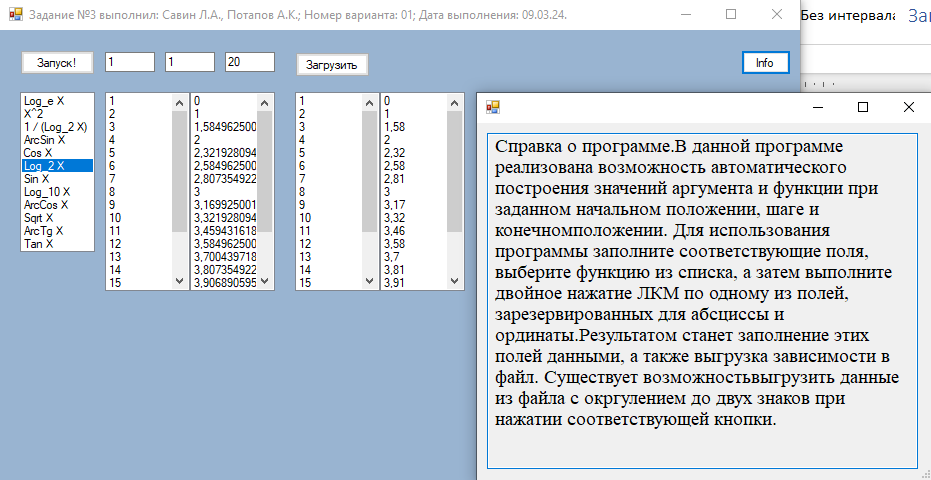


Нажмём кнопку «Запуск». Перед нами появится множество активных полей, среди которых есть поля «Start», «Step», «Stop». Они отвечают за выставление начала значений абсцисс, их шага, и конечного значения. Левее располагается окно с функциями, зависимости для которых могут быть построены с использованием данного ПО. Когда значения во всех этих окнах выставлены, можно выполнить двойной клик по области, предназначенной для абсцисс и ординат, что приведёт к построению зависимости и загрузки значений в соответствующие поля и в файл. Данное состояние окна отображено на следующем рисунке.



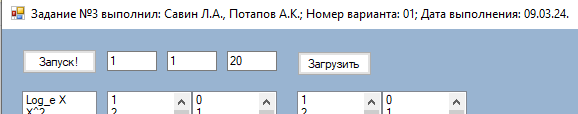
Как можно видеть, данные в файле разделены построчно и с помощью знака «$».

В программе возможно загрузить данные из файла в дополнительные поля с округлением при нажатии на кнопку «Загрузить». Также имеется возможность вывести справку при нажатии кнопки «Info». Итог показан на следующем рисунке.



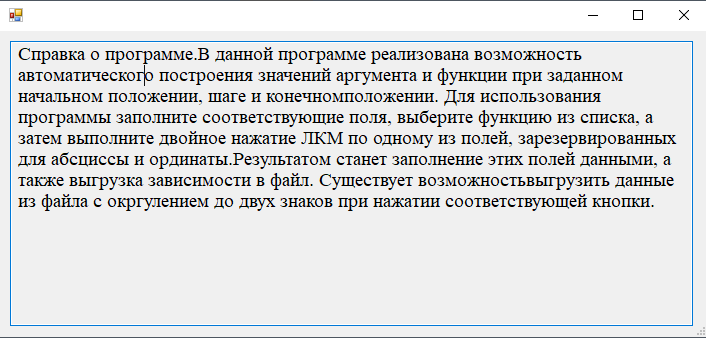
Как видим, программа вызвала дочернюю форму со справкой, и также заполнила дополнительные поля, отсутствовавшие ранее, округлёнными значениями нашей функции.

9. Подтверждение соответствия графического интерфейса заданному описанию

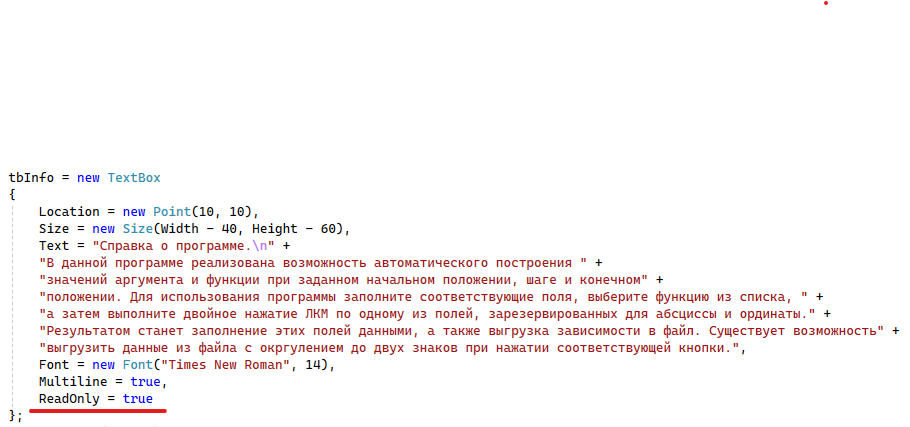


Соответствие заголовка экранной формы с проставленной датой

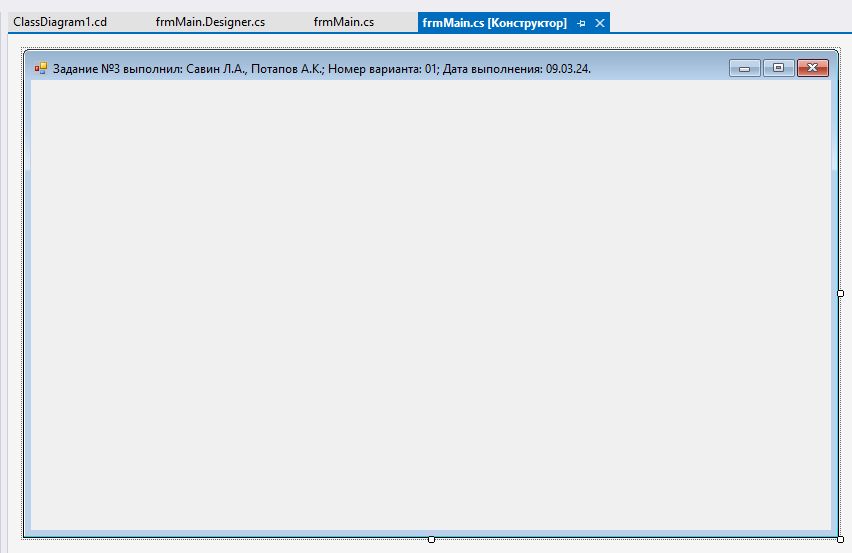
|  |
| --- |
|  |
| Курсор-рука на экранной форме |



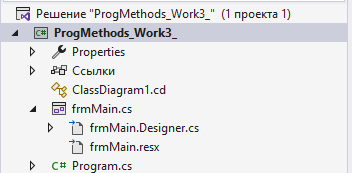
Вывод окна со справочной информацией в текстовом поле



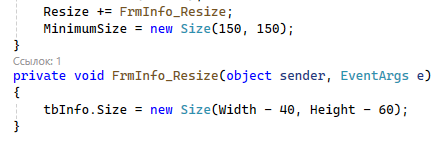
Текстовое поле имеет запрет на редактирование



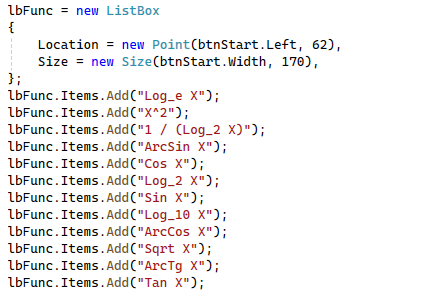
В режиме конструктора нет ни одного объекта



Имеется всего одна экранная форма

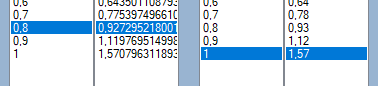


Для дочерней формы подключено событие Resize()

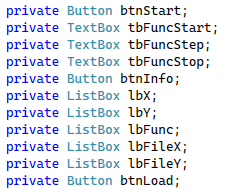


Список функций – обычный список (ListBox)

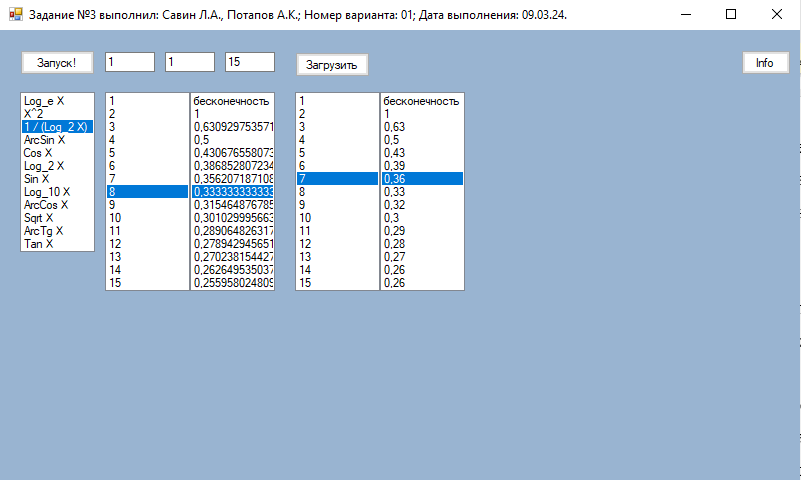
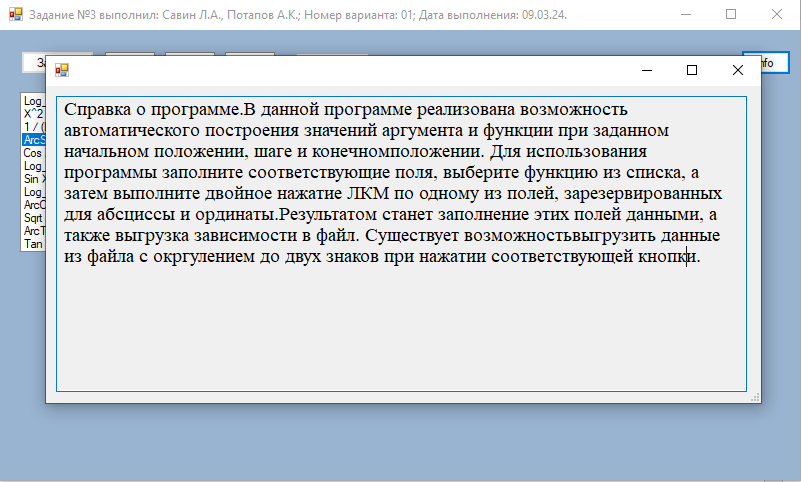
Заполнение данных в файл с разделителем, изменение цвета подложки, наличие кнопки заполнения округлёнными значениями дополнительных меню, начальное невидимое состояние и кнопка запуска программы были приведены в предыдущей главе.



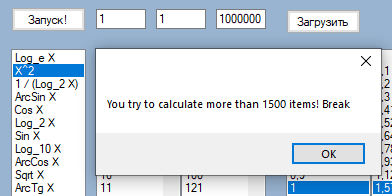
Выбор абсцисс(ординат) и соответствующих им ординат(абсцисс)

  
"Говорящие" названия

10. Расчёт тестовых примеров

  
Расчёт, запись в файл, синхронизация списков, чтение из файла с округлением  
  


Открытие окна со справкой



Попытка расчёта миллиона значений. Расчёт прерван на 1500 значениях

10. Вывод:

Разработанная программа успешно реализует функционал, представленный в постановке задачи. Графический интерфейс интуитивно понятен, программа работает стабильно и соответствует требованиям, установленным заказчиком. В ходе тестирования было выявлено, что функции программы работают корректно и в соответствии с ожиданиями.