Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Российский университет транспорта»

(ФГАОУ ВО РУТ(МИИТ), РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Лабораторная работа № 20

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Диаграммы»

Выполнил: ст. гр. ТУУ-211

Волобуев Е.Д.

Вариант №2

11.12.2024

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

\_\_\_\_\_\_\_\_\_\_\_

(дата приёмки)

Москва – 2024 г.

Оглавление

[1. Формулировка цели работы 3](#_Toc184610187)

[2. Описание задачи 3](#_Toc184610188)

[4. Схемы алгоритмов методов в составе решения, отмеченных на сети Петри в качестве «эффектов» (метка ) 5](#_Toc184610189)

[5. Подбор тестовых примеров 11](#_Toc184610190)

[6. Листинг (код) программного обеспечения 13](#_Toc184610191)

[7. Графический пользовательский интерфейс программного обеспечения 21](#_Toc184610192)

[8. Подтверждение соответствия графического пользовательского интерфейса к требованиям оформления 22](#_Toc184610193)

[9. Расчет тестовых примеров с использованием составленного программного обеспечения 24](#_Toc184610194)

[10. Формулировка вывода о проделанной работе 25](#_Toc184610195)

## **1. Формулировка цели работы**

Освоить навыки подключения внешних модулей и их использования в разрабатываемых программах; освоить навыки программируемой настройки элемента экранной формы *«Chart»*.

## **2. Описание задачи**

В интегрированной среде разработки *Microsoft Visual Studio* разработать программу в режиме *Windows Forms Application* на языке *Visual C#*, представляющую собой средство взаимодействия с электронными таблицами *Microsoft Office Excel*. Загружать данные, представленные на номограммах, в программное обеспечение из *csv*-файла. Предоставить пользователю возможность просмотра как сразу всех рядов данных, так и одного, указанного ряда данных для обеих номограмм.

**3. Сеть Петри запрограммированного технологического процесса**

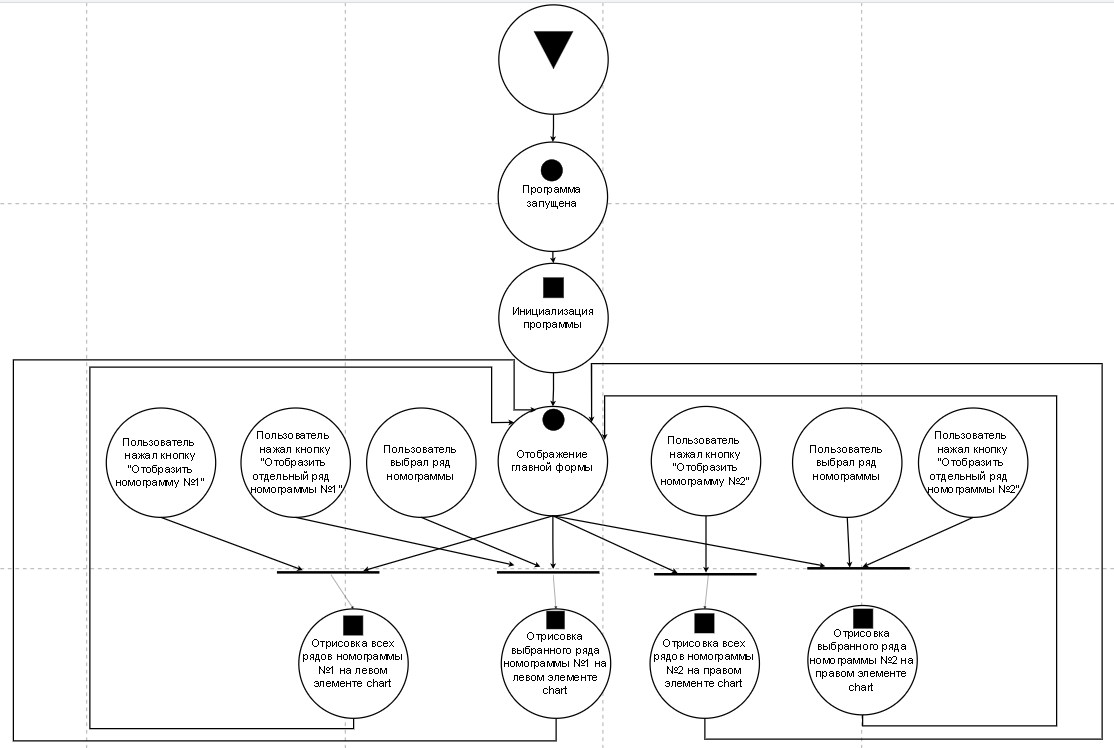


Рисунок 1 – Сеть Петри запрограммированного технологического процесса

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

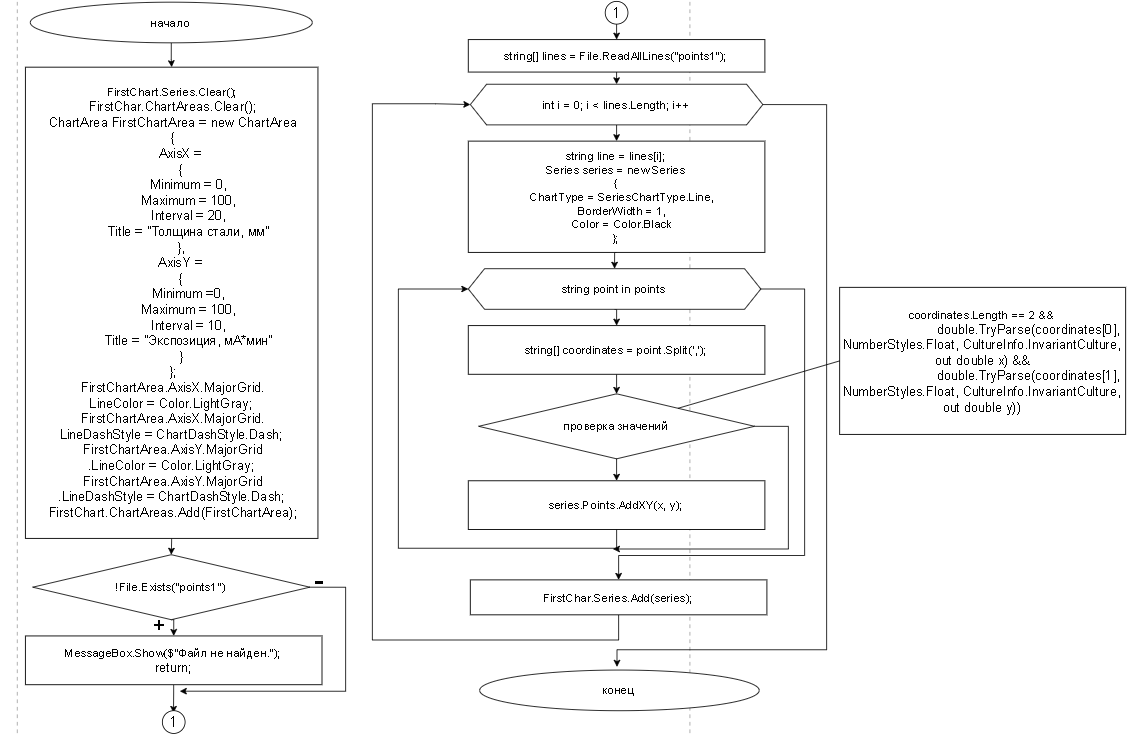


Рисунок 2 – Блок-схема метода *CreateFirstGraph*

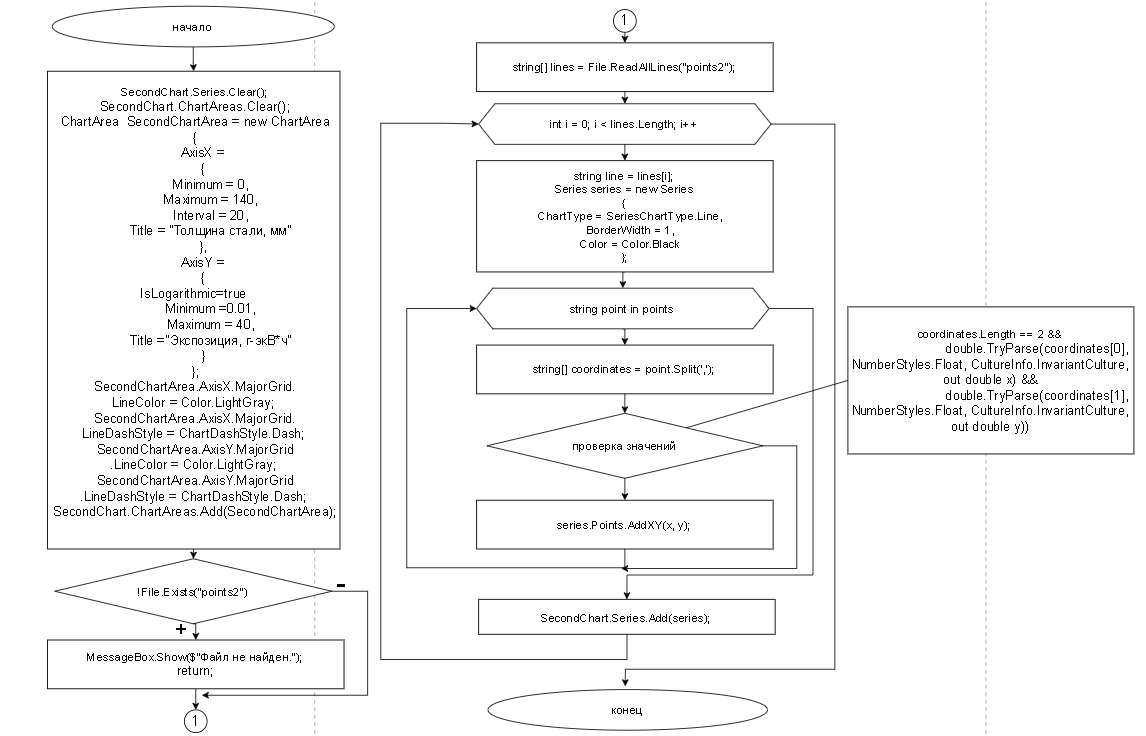
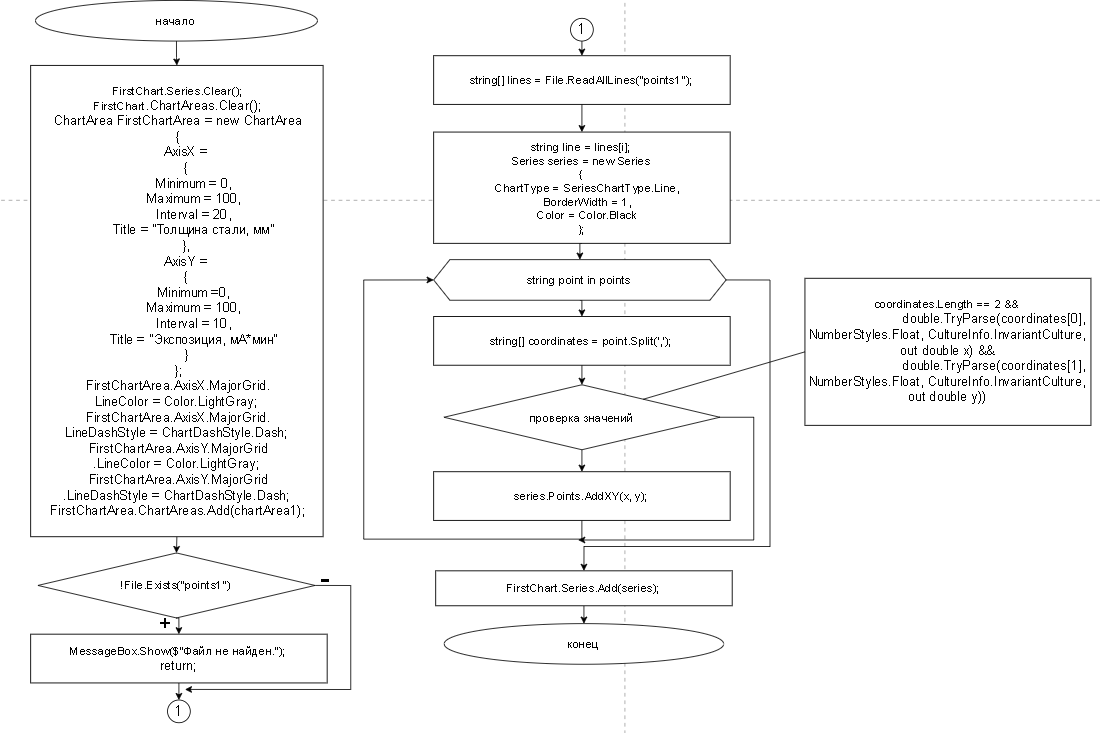
 Рисунок 3 – Блок-схема метода *CreateSecondGraph*

Рисунок 4 – Блок-схема метода *FirstGraphRowSelectedButton\_Click*

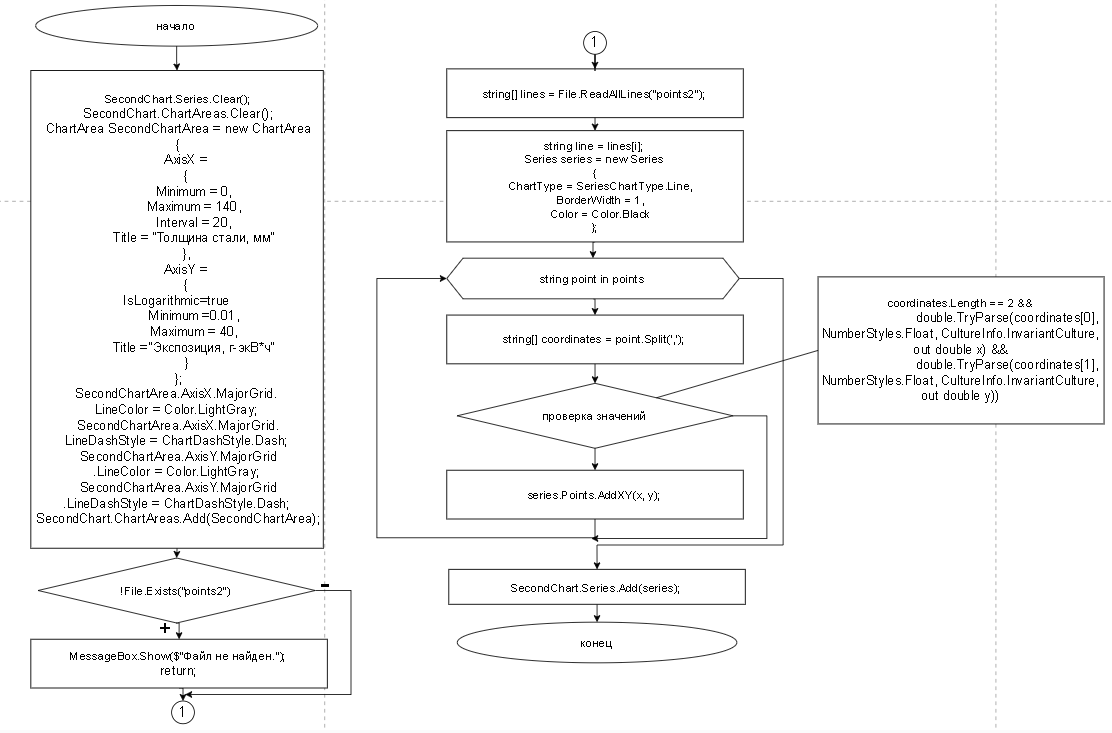


Рисунок 5 – Блок-схема метода *SecondGraphRowSelectedButton\_Click*

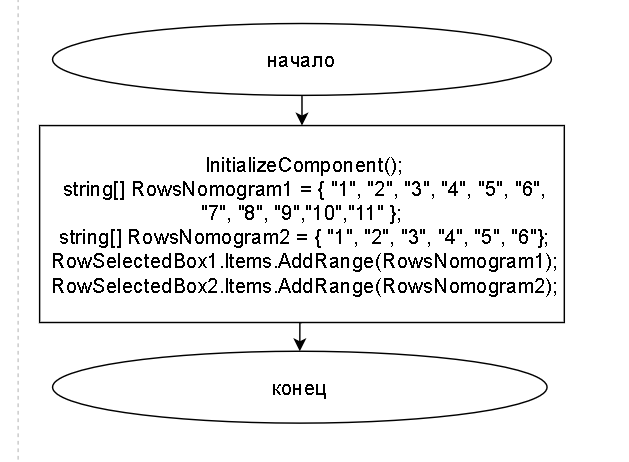


Рисунок 6– Блок-схема метода *Nomograms*

## **5. Подбор тестовых примеров**

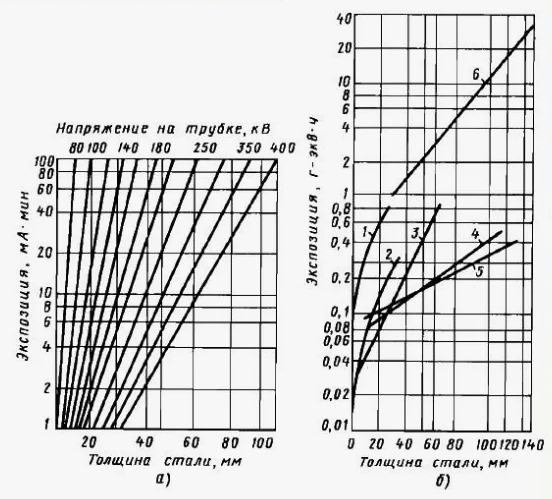


Рисунок 7 – Индивидуальный вариант задания

**Первый тестовый пример**

При нажатии на кнопку «Отобразить номограмму №1», на элементе экранной формы *«chart»* должен появиться график, схожий с рисунком 7(а). График строится по точкам, записанным в *CSV*-файле «*points1.csv*» (рисунок 8).

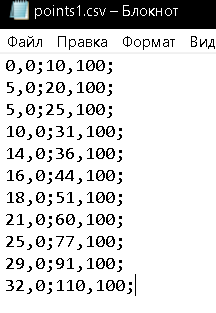


Рисунок 8 – Содержание файла «*points1.csv*»

**Второй тестовый пример**

При нажатии на кнопку «Отобразить номограмму №2», на элементе экранной формы *«chart»* должен появиться график, схожий с рисунком 7(б). График строится по точкам, записанным в *CSV*-файле «*points2.csv*» (рисунок 9).

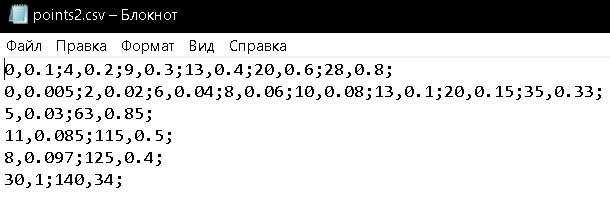


Рисунок 9 – Содержание файла «*points2.csv*»

**Третий тестовый пример**

При нажатии на кнопку «Отобразить отдельный ряд номограммы №1» и выборе номера «2» из выпадающего списка на элементе экранной формы «*chart*» должен отобразиться второй ряд первой номограммы

**Четвертый тестовый пример**

При нажатии на кнопку «Отобразить отдельный ряд номограммы №2» и выборе номера «3» из выпадающего списка на элементе экранной формы «*chart*» должен отобразиться третий ряд второй номограммы

## **6. Листинг (код) программного обеспечения**

using System;

using System.Drawing;

using System.Globalization;

using System.IO;

using System.Reflection;

using System.Windows.Forms;

using System.Windows.Forms.DataVisualization.Charting;

namespace Rabota\_20

{

public partial class Nomograms : Form

{

public Nomograms()

{

InitializeComponent();

string[] RowsFirstNomogram = { "1", "2", "3", "4", "5", "6", "7", "8", "9","10","11" };

string[] RowsSecondNomogram = { "1", "2", "3", "4", "5", "6"};

FirstNomogramRowSelectedBox.Items.AddRange(RowsFirstNomogram);

SecondNomogramRowSelectedBox.Items.AddRange(RowsSecondNomogram);

}

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

{

CreateFirstGraph();

}

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

{

CreateSecondGraph();

}

private void CreateFirstGraph()

{

FirstChar.Series.Clear();

FirstChar.ChartAreas.Clear();

ChartArea FirstChartArea = new ChartArea

{

AxisX =

{

Minimum = 0,

Maximum = 100,

Interval = 20,

Title = "Толщина стали, мм"

},

AxisY =

{

Minimum =0,

Maximum = 100,

Interval = 10,

Title = "Экспозиция, мА\*мин"

}

};

FirstChartArea.AxisX.MajorGrid.LineColor = Color.LightGray;

FirstChartArea.AxisX.MajorGrid.LineDashStyle = ChartDashStyle.Dash;

FirstChartArea.AxisY.MajorGrid.LineColor = Color.LightGray;

FirstChartArea.AxisY.MajorGrid.LineDashStyle = ChartDashStyle.Dash;

FirstChar.ChartAreas.Add(FirstChartArea);

if (!File.Exists("points1.csv"))

{

MessageBox.Show($"Файл не найден.");

return;

}

string[] lines = File.ReadAllLines("points1.csv");

for (int i = 0; i < lines.Length; i++)

{

string line = lines[i];

Series series = new Series

{

ChartType = SeriesChartType.Line,

BorderWidth = 1,

Color = Color.Black

};

string[] points = line.Split(';');

foreach (string point in points)

{

string[] coordinates = point.Split(',');

if (coordinates.Length == 2 &&

double.TryParse(coordinates[0], NumberStyles.Float, CultureInfo.InvariantCulture, out double x) &&

double.TryParse(coordinates[1], NumberStyles.Float, CultureInfo.InvariantCulture, out double y))

{

series.Points.AddXY(x, y);

}

}

FirstChar.Series.Add(series);

}

}

private void CreateSecondGraph()

{

SecondChart.Series.Clear();

SecondChart.ChartAreas.Clear();

ChartArea SecondChartArea = new ChartArea

{

AxisX =

{

Minimum = 0,

Maximum = 140,

Interval = 20,

Title = "Толщина стали, мм"

},

AxisY =

{

IsLogarithmic=true,

Minimum =0.01,

Maximum = 40,

Title = "Экспозиция, г-экВ\*ч"

}

};

SecondChartArea.AxisX.MajorGrid.LineColor = Color.LightGray;

SecondChartArea.AxisX.MajorGrid.LineDashStyle = ChartDashStyle.Dash;

SecondChartArea.AxisY.MajorGrid.LineColor = Color.LightGray;

SecondChartArea.AxisY.MajorGrid.LineDashStyle = ChartDashStyle.Dash;

SecondChart.ChartAreas.Add(SecondChartArea);

if (!File.Exists("points2.csv"))

{

MessageBox.Show($"Файл не найден.");

return;

}

string[] lines = File.ReadAllLines("points2.csv");

for (int i = 0; i < lines.Length; i++)

{

string line = lines[i];

Series series = new Series

{

ChartType = SeriesChartType.Line,

BorderWidth = 1,

Color = Color.Black

};

string[] points = line.Split(';');

foreach (string point in points)

{

string[] coordinates = point.Split(',');

if (coordinates.Length == 2 &&

double.TryParse(coordinates[0], NumberStyles.Float, CultureInfo.InvariantCulture, out double x) &&

double.TryParse(coordinates[1], NumberStyles.Float, CultureInfo.InvariantCulture, out double y))

{

series.Points.AddXY(x, y);

}

}

SecondChart.Series.Add(series);

}

}

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

{

FirstChar.Series.Clear();

FirstChar.ChartAreas.Clear();

ChartArea FirstChartArea = new ChartArea

{

AxisX =

{

Minimum = 0,

Maximum = 100,

Interval = 20,

Title = "Толщина стали, мм"

},

AxisY =

{

Minimum =0,

Maximum = 100,

Interval = 10,

Title = "Экспозиция, мА\*мин"

}

};

FirstChartArea.AxisX.MajorGrid.LineColor = Color.LightGray;

FirstChartArea.AxisX.MajorGrid.LineDashStyle = ChartDashStyle.Dash;

FirstChartArea.AxisY.MajorGrid.LineColor = Color.LightGray;

FirstChartArea.AxisY.MajorGrid.LineDashStyle = ChartDashStyle.Dash;

FirstChar.ChartAreas.Add(FirstChartArea);

if (!File.Exists("points1.csv"))

{

MessageBox.Show($"Файл не найден.");

return;

}

string[] lines = File.ReadAllLines("points1.csv");

int index = FirstNomogramRowSelectedBox.SelectedIndex;

string line = lines[index];

Series series = new Series

{

ChartType = SeriesChartType.Line,

BorderWidth = 1,

Color = Color.Black

};

string[] points = line.Split(';');

foreach (string point in points)

{

string[] coordinates = point.Split(',');

if (coordinates.Length == 2 &&

double.TryParse(coordinates[0], NumberStyles.Float, CultureInfo.InvariantCulture, out double x) &&

double.TryParse(coordinates[1], NumberStyles.Float, CultureInfo.InvariantCulture, out double y))

{

series.Points.AddXY(x, y);

}

}

FirstChar.Series.Add(series);

}

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

{

SecondChart.Series.Clear();

SecondChart.ChartAreas.Clear();

ChartArea SecondChartArea = new ChartArea

{

AxisX =

{

Minimum = 0,

Maximum = 140,

Interval = 20,

Title = "Толщина стали, мм"

},

AxisY =

{

IsLogarithmic=true,

Minimum =0.01,

Maximum = 40,

Title = "Экспозиция, г-экВ\*ч"

}

};

SecondChartArea.AxisX.MajorGrid.LineColor = Color.LightGray;

SecondChartArea.AxisX.MajorGrid.LineDashStyle = ChartDashStyle.Dash;

SecondChartArea.AxisY.MajorGrid.LineColor = Color.LightGray;

SecondChartArea.AxisY.MajorGrid.LineDashStyle = ChartDashStyle.Dash;

this.SecondChart.ChartAreas.Add(SecondChartArea);

if (!File.Exists("points2.csv"))

{

MessageBox.Show($"Файл не найден.");

return;

}

string[] lines = File.ReadAllLines("points2.csv");

int index = SecondNomogramRowSelectedBox.SelectedIndex;

string line = lines[index];

Series series = new Series

{

ChartType = SeriesChartType.Line,

BorderWidth = 1,

Color = Color.Black

};

string[] points = line.Split(';');

foreach (string point in points)

{

string[] coordinates = point.Split(',');

if (coordinates.Length == 2 &&

double.TryParse(coordinates[0], NumberStyles.Float, CultureInfo.InvariantCulture, out double x) &&

double.TryParse(coordinates[1], NumberStyles.Float, CultureInfo.InvariantCulture, out double y))

{

series.Points.AddXY(x, y);

}

}

this.SecondChart.Series.Add(series);

}

}

}

# **7. Графический пользовательский интерфейс программного обеспечения**

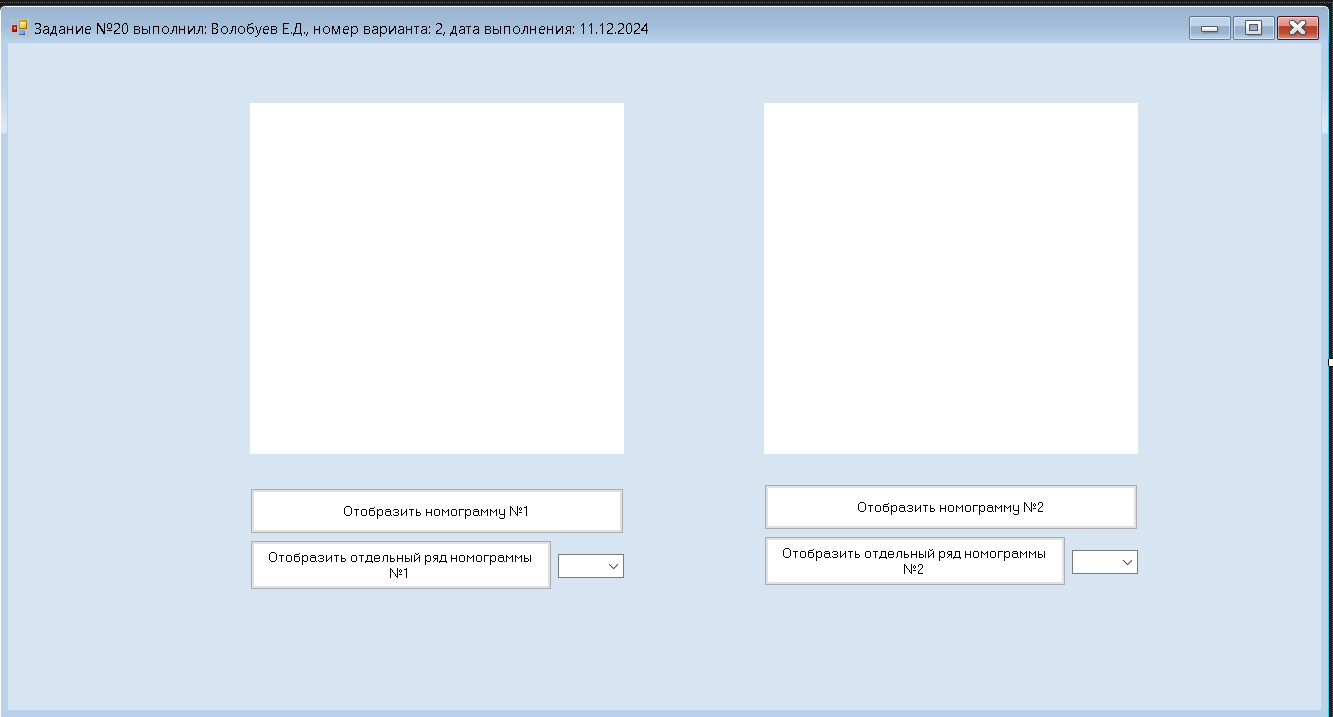


Рисунок 10 – Интерфейс главной формы при запуске

На главной форме располагается два элемента экранной формы *«chart»*, кнопки «Отобразить номограмму №1», «Отобразить номограмму №2», а также выпадающие списки для выбора ряда для двух номограмм.

# **8. Подтверждение соответствия графического пользовательского интерфейса к требованиям оформления**

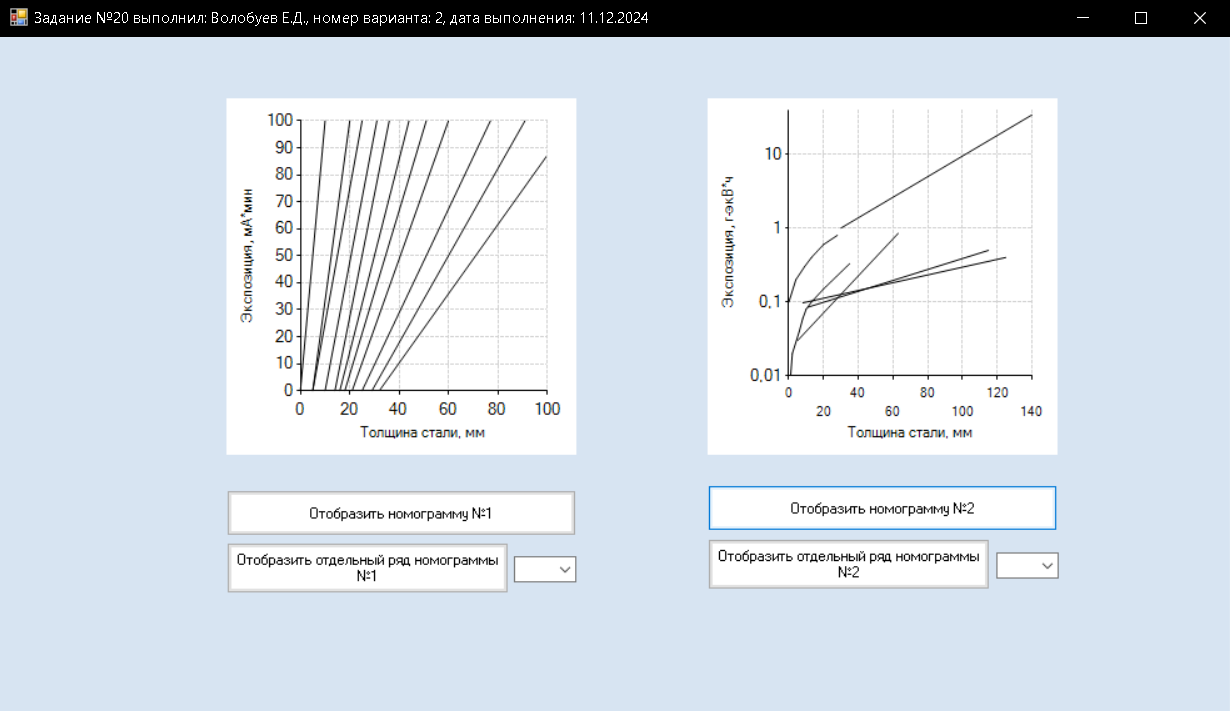


Рисунок 11 – Интерфейс главной формы при нажатии на кнопки «Отобразить номограмму №1» и «Отобразить номограмму №2»

При нажатии на кнопки «Отобразить номограмму №1» и «Отобразить номограмму №2», элементы экранной формы *«chart»* отражают заданные по индивидуальному заданию рисунки.

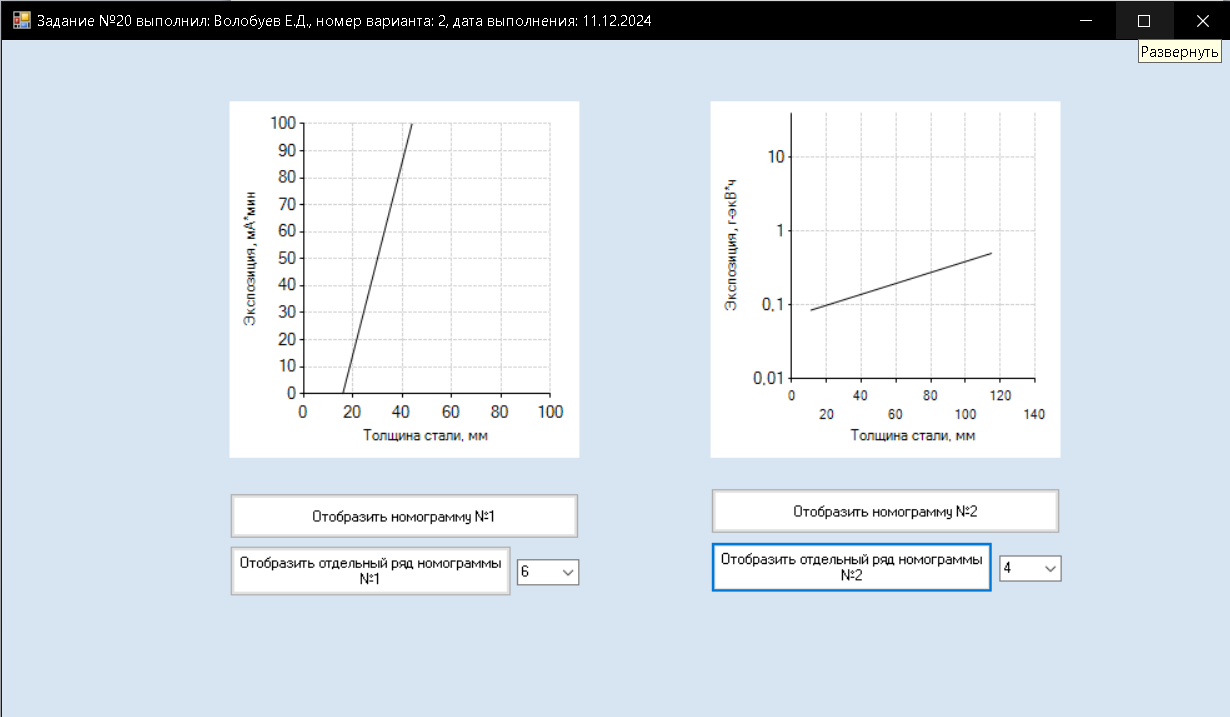


Рисунок 12 – Интерфейс главной формы при нажатии кнопки «Отобразить отдельный ряд номограммы №1» и «Отобразить отдельный ряд номограммы №2»

При нажатии на кнопки «Отобразить отдельный ряд номограммы №1» и «Отобразить отдельный ряд номограммы №2», элементы экранной формы *«chart»* отражает заданные ряды для двух номограмм.

## **9. Расчет тестовых примеров с использованием составленного программного обеспечения**

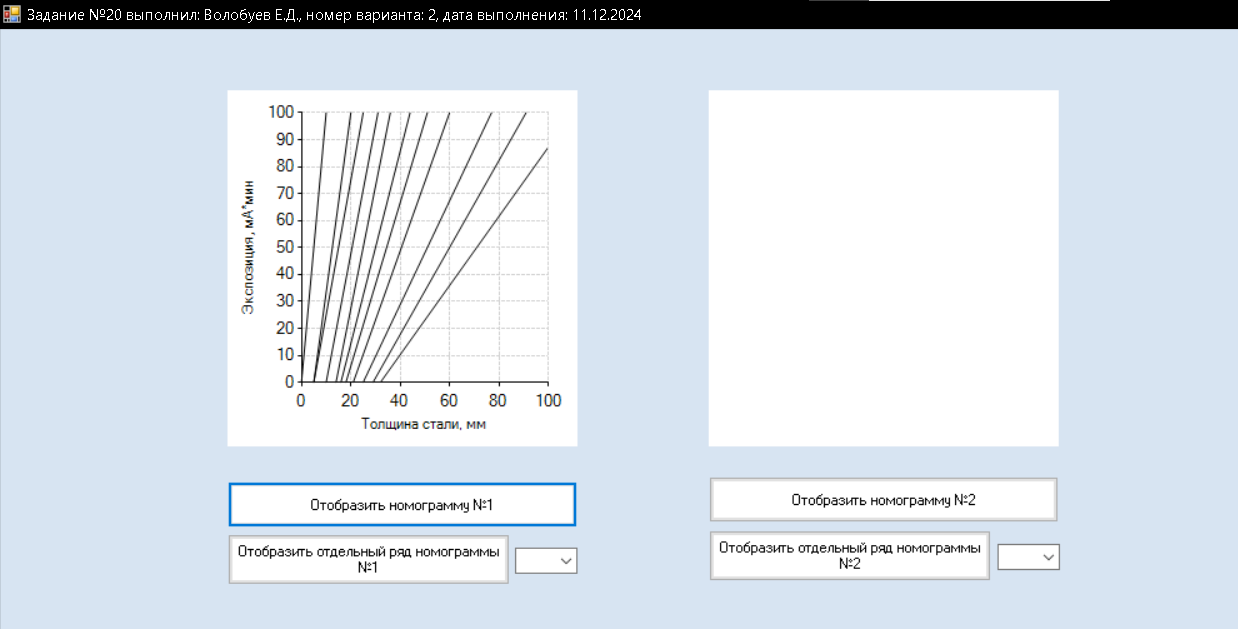


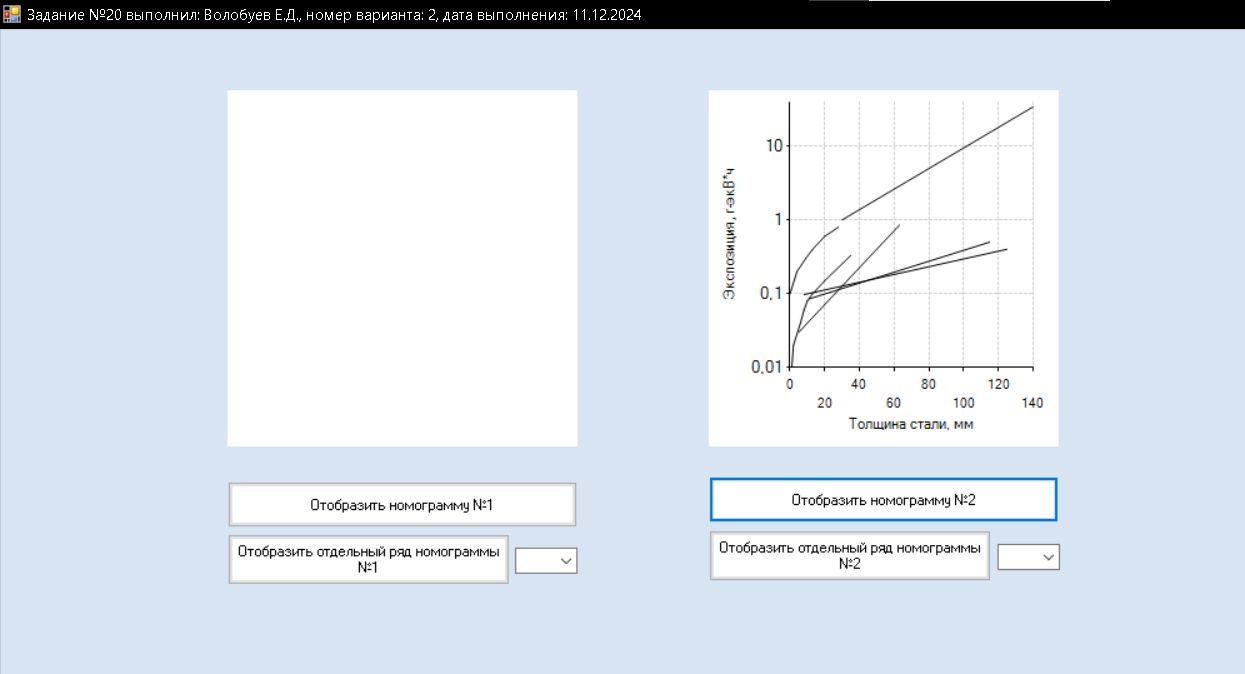
Рисунок 13 – Расчет первого тестового примера 

Рисунок 14 – Расчет второго тестового примера

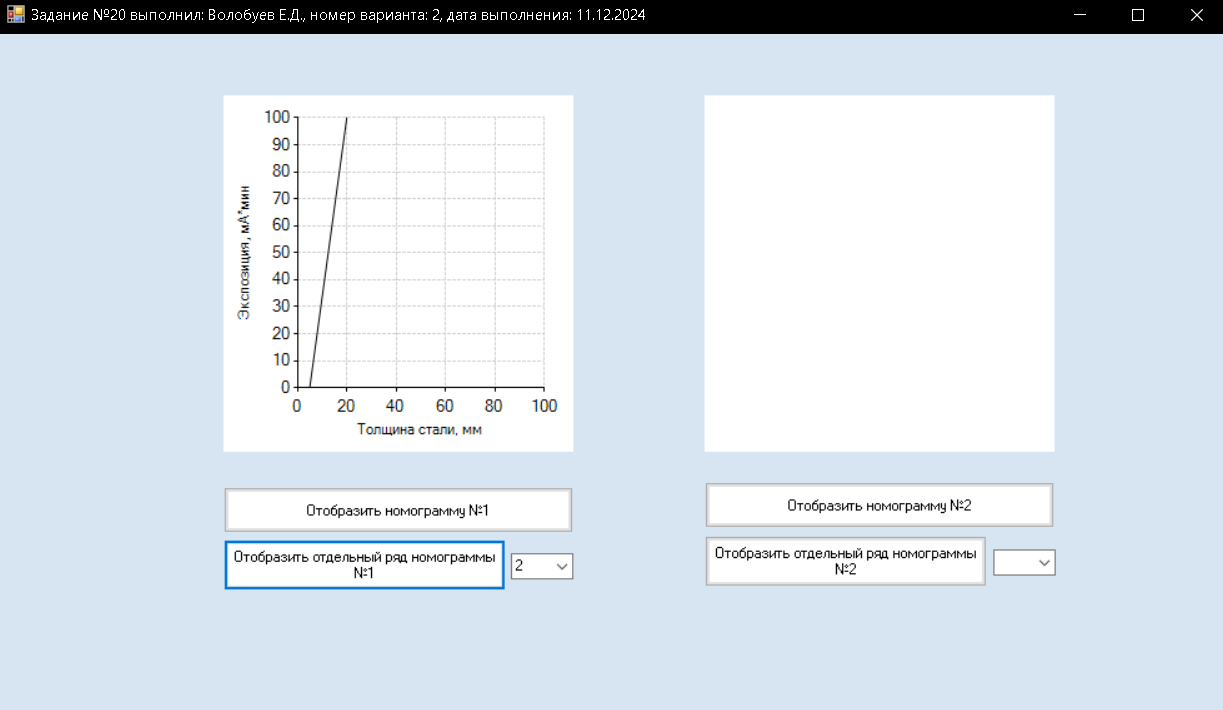


Рисунок 15 – Расчет третьего тестового примера

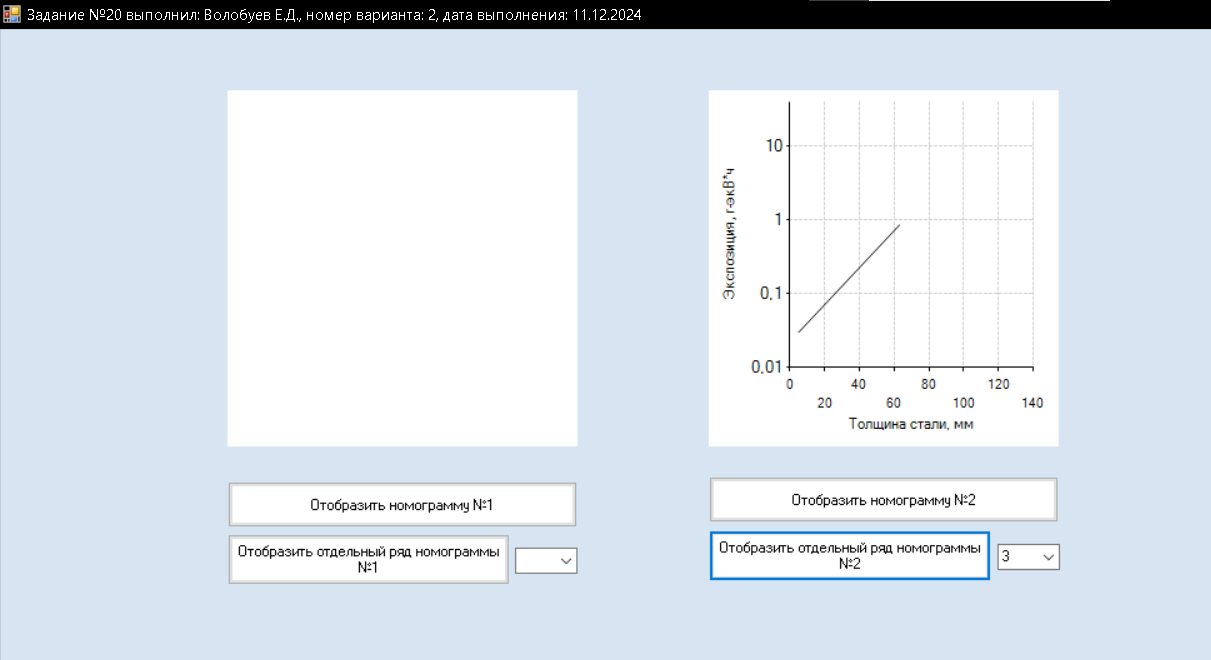


Рисунок 16 – Расчет четвертого тестового примера

## **10. Формулировка вывода о проделанной работе**

В процессе работы освоил работу с элементом экранной формы *«chart»*, закрепил для себя подключение и использование внешних модулей для работы с диаграммами, освоил редактирование и дизайн диаграмм, закрепил для себя методы работы с файлами, научился настраивать оси координат для графиков, закрепил способы подключения различных библиотек классов предназначенных для работы в режиме разработки Windows Forms.