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

Баранов А.А.

Вариант №7

09.12.2024

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

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

(дата приёмки)

Москва – 2024 г.

Оглавление

[1. Формулировка цели работы 2](#_Toc184610187)

[2. Описание задачи 2](#_Toc184610188)

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

[5. Подбор тестовых примеров 8](#_Toc184610190)

[6. Листинг (код) программного обеспечения 9](#_Toc184610191)

[7. Графический пользовательский интерфейс программного обеспечения 17](#_Toc184610192)

[8. Подтверждение соответствия графического пользовательского интерфейса к требованиям оформления 18](#_Toc184610193)

[9. Расчет тестовых примеров с использованием составленного программного обеспечения 20](#_Toc184610194)

[10. Формулировка вывода о проделанной работе 22](#_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 номограмме, в программное обеспечение из *csv*-файла. Предоставить пользователю возможность просмотра как сразу всех рядов данных, так и одного, указанного ряда данных. Предоставить пользователю возможность просмотра нескольких указанных (не всех) рядов данных. Обязательно формировать легенду для номограммы. Допускается реализация не с чёрно-белым представлением, а с цветным представлением рядов данных.

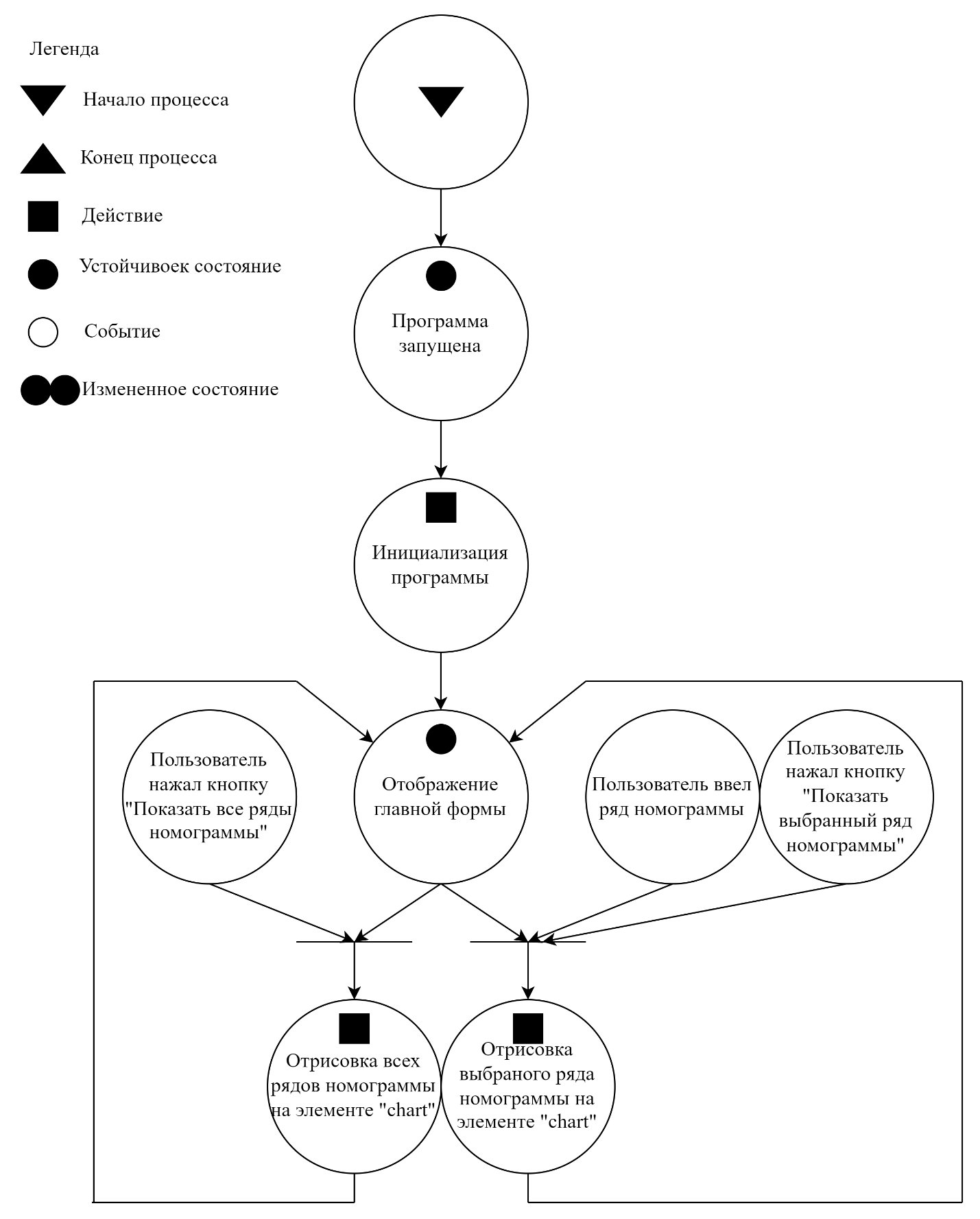
**3. Сеть Петри запрограммированного технологического процесса**

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 *showAllRows\_Click*



Рисунок 3 – Блок-схема метода *showAllRows\_Click*

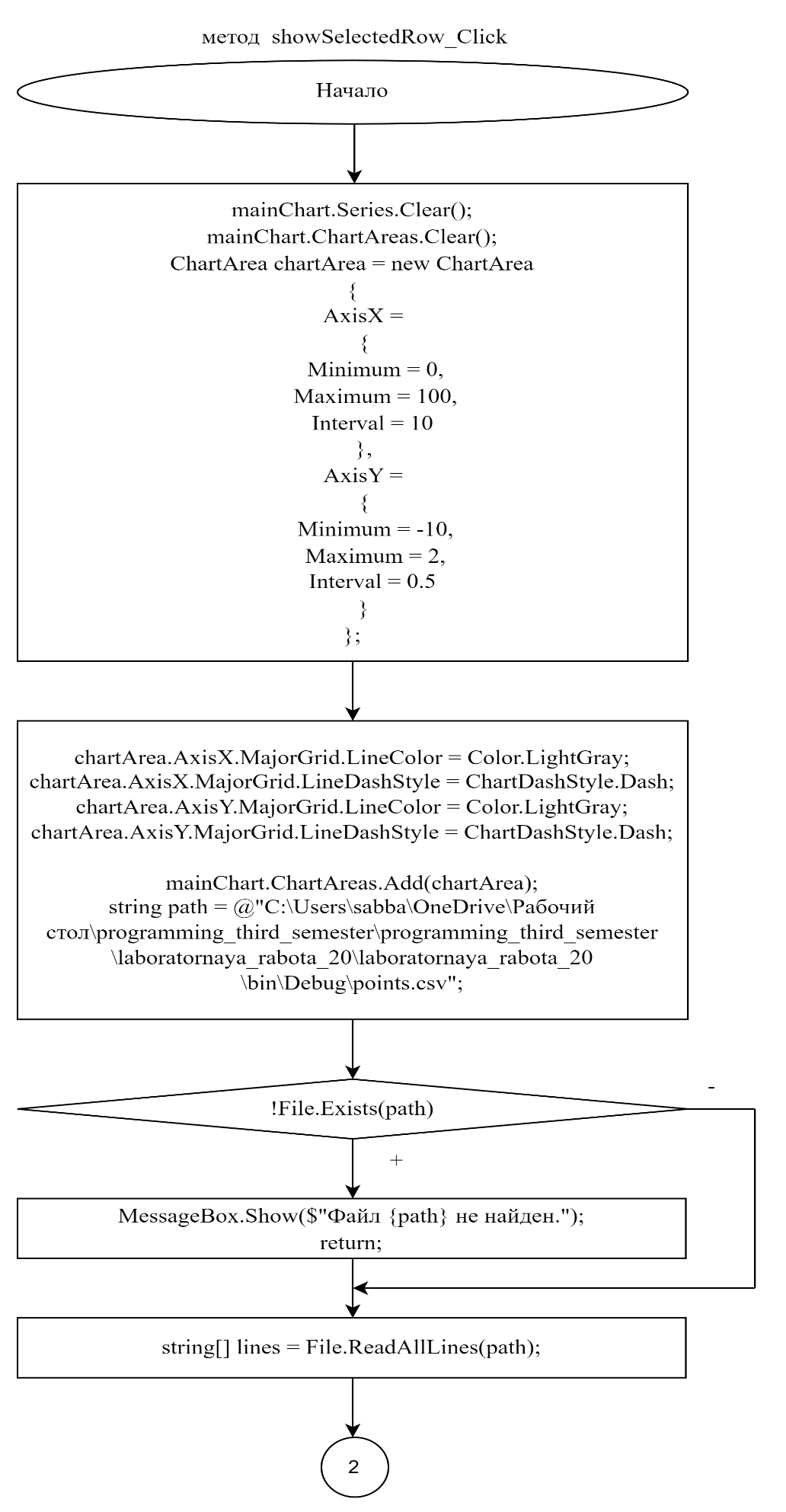


Рисунок 4 – Блок-схема метода *showSelectedRow\_Click*

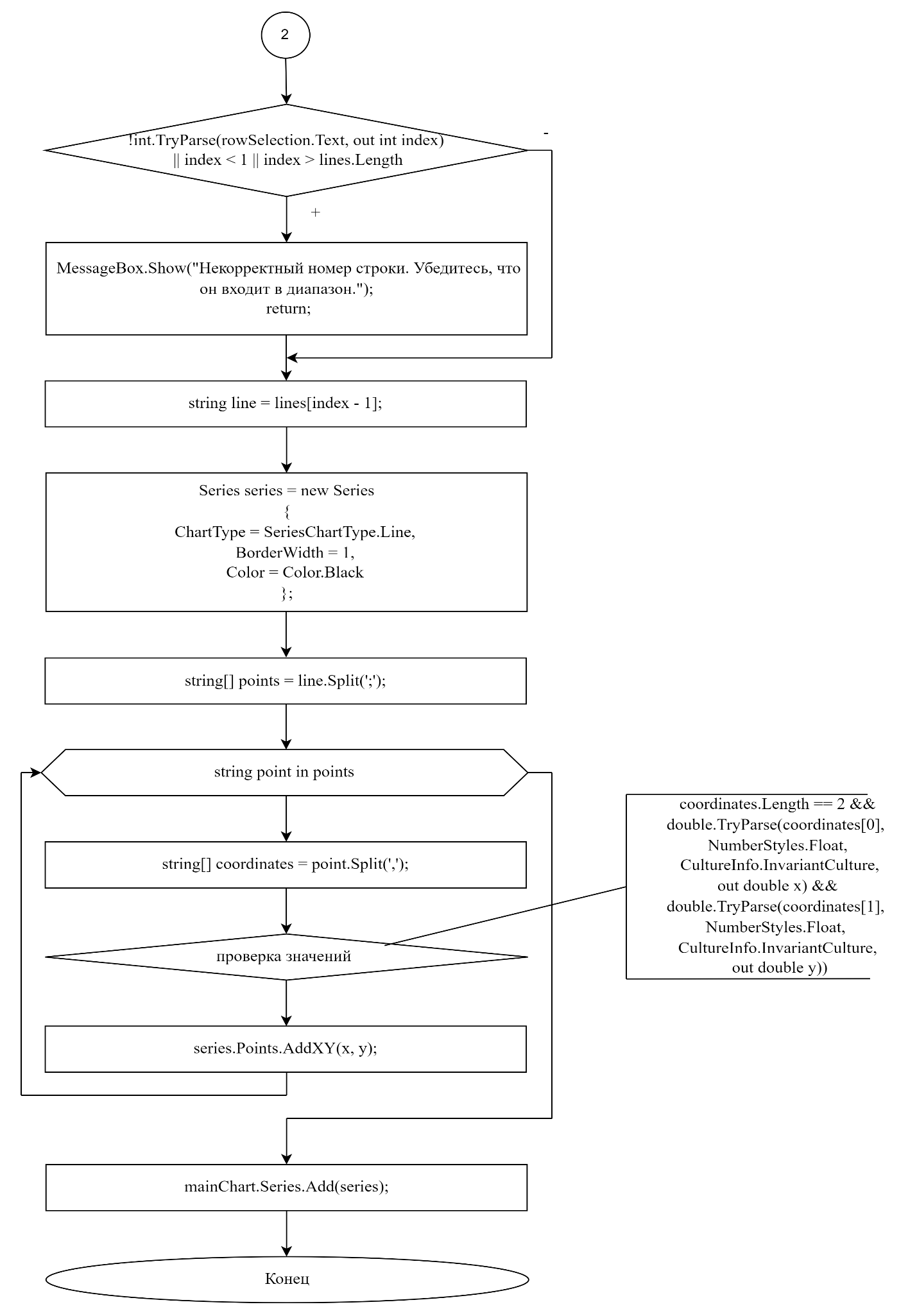


Рисунок 5 – Блок-схема метода *showSelectedRow\_Click*

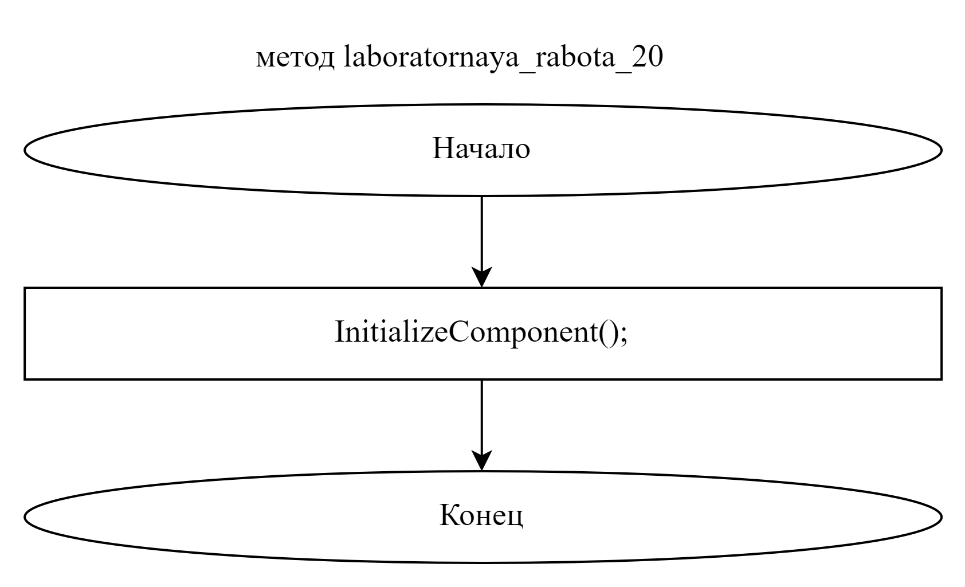


Рисунок 6 – Блок-схема метода *showSelectedRow\_Click*

## **5. Подбор тестовых примеров**

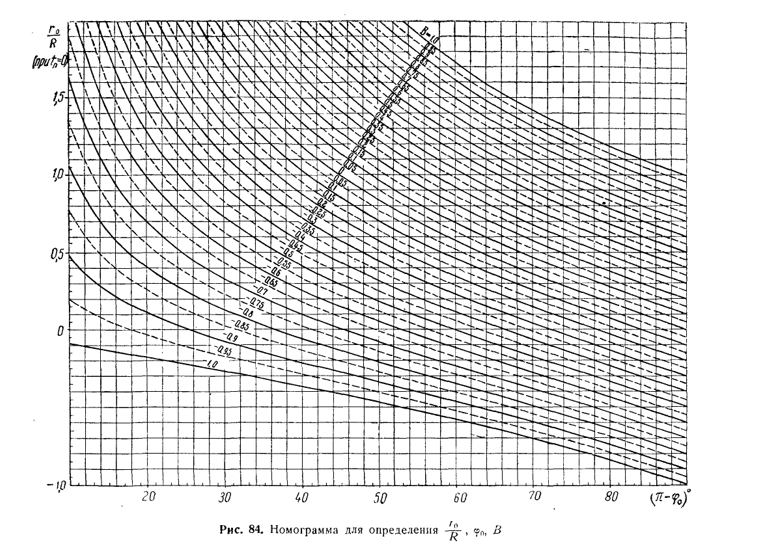


Рисунок 7 – Индивидуальный вариант задания

**Первый тестовый пример**

При нажатии на кнопку «Показать все ряды номограммы», на элементе экранной формы *«chart»* должен появиться график, схожий с Рисунком 7. График строится по то точкам, записанным в *CSV*-файле «*points*.csv» (Рисунок 8).

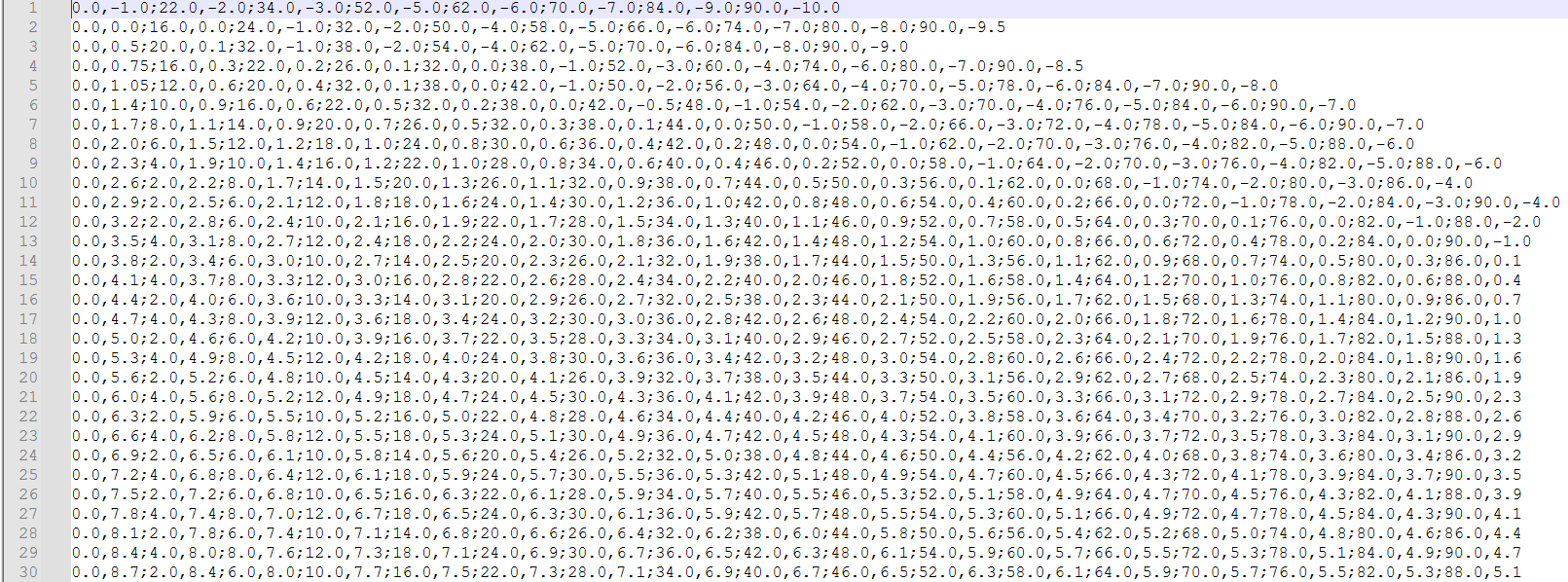


Рисунок 8 – Содержание файла «*points.csv*»

**Второй тестовый пример**

При нажатии на кнопку «Показать выбранный ряд номограммы» и вводе в строковое поле числа 2, на элементе экранной формы *«chart»* должен появиться второй снизу ряд номограммы, схожий с таким же на Рисунке 7. График строится по то точкам из второго ряда *CSV*-файла «*points.csv*» (Рисунок 8).

**Третий тестовый пример**

При вводе строки «a» в строковое поле для выбора ряда номограммы, должно вывестись сообщение об ошибке с текстом: «Некорректный номер строки. Убедитесь, что он входит в диапазон».

## 

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

using System;

using System.Drawing;

using System.Globalization;

using System.IO;

using System.Windows.Forms;

using System.Windows.Forms.DataVisualization.Charting;

namespace laboratornaya\_rabota\_20

{

public partial class laboratornaya\_rabota\_20 : Form

{

public laboratornaya\_rabota\_20()

{

InitializeComponent();

}

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

{

mainChart.Series.Clear();

mainChart.ChartAreas.Clear();

ChartArea chartArea = new ChartArea

{

AxisX =

{

Minimum = 0,

Maximum = 100,

Interval = 10

},

AxisY =

{

Minimum = -10,

Maximum = 2,

Interval = 0.5

}

};

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

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

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

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

mainChart.ChartAreas.Add(chartArea);

string path = @"C:\Users\sabba\OneDrive\Рабочий стол\programming\_third\_semester\programming\_third\_semester\laboratornaya\_rabota\_20\laboratornaya\_rabota\_20\bin\Debug\points.csv";

if (!File.Exists(path))

{

MessageBox.Show($"Файл {path} не найден.");

return;

}

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

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);

}

}

mainChart.Series.Add(series);

}

}

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

{

mainChart.Series.Clear();

mainChart.ChartAreas.Clear();

ChartArea chartArea = new ChartArea

{

AxisX =

{

Minimum = 0,

Maximum = 100,

Interval = 10

},

AxisY =

{

Minimum = -10,

Maximum = 2,

Interval = 0.5

}

};

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

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

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

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

mainChart.ChartAreas.Add(chartArea);

string path = @"C:\Users\sabba\OneDrive\Рабочий стол\programming\_third\_semester\programming\_third\_semester\laboratornaya\_rabota\_20\laboratornaya\_rabota\_20\bin\Debug\points.csv";

if (!File.Exists(path))

{

MessageBox.Show($"Файл {path} не найден.");

return;

}

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

if (!int.TryParse(rowSelection.Text, out int index) || index < 1 || index > lines.Length)

{

MessageBox.Show("Некорректный номер строки. Убедитесь, что он входит в диапазон.");

return;

}

string line = lines[index - 1];

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);

}

}

mainChart.Series.Add(series);

}

}

}

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

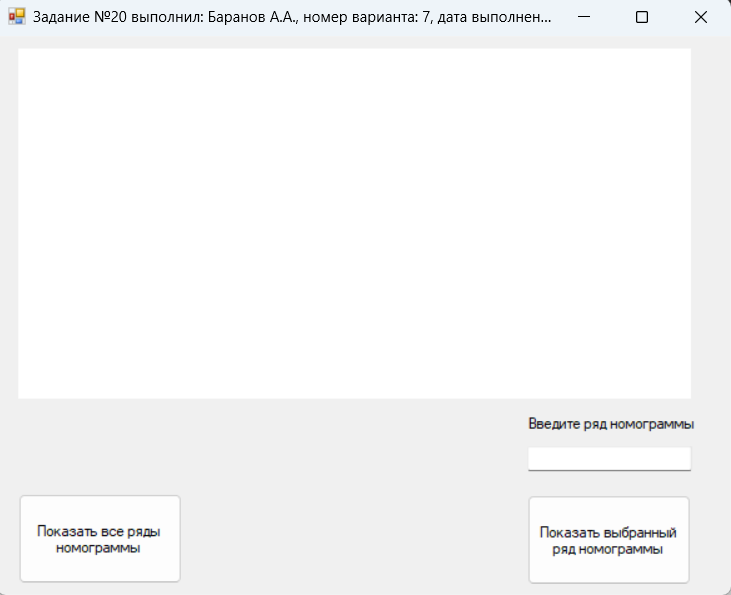


Рисунок 9 – Интерфейс главной формы при запуске

На главной форме располагается элемент экранной формы *«chart»*, кнопки «Показать все ряды номограммы» и «Показать выбранный ряд номограммы», а также тестовое поля для ввода ряда номограммы.

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

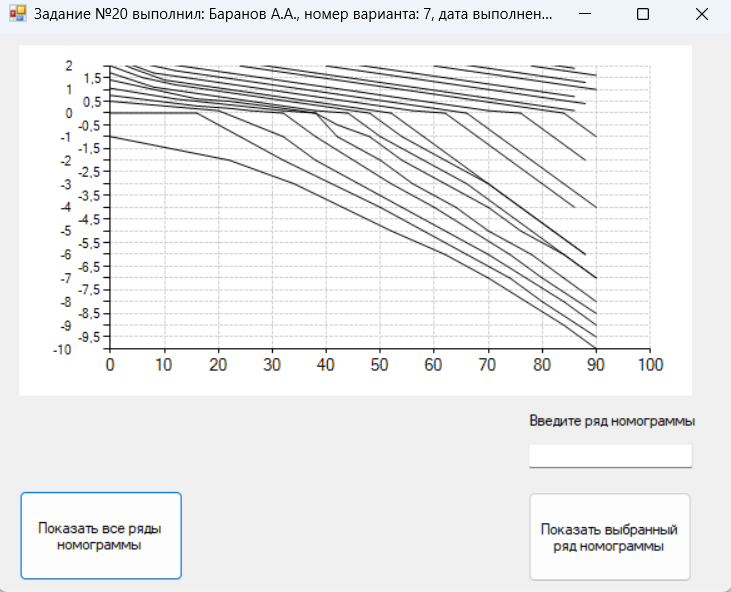


Рисунок 10 – Интерфейс главной формы при нажатии кнопки «Показать все ряды номограммы»

При нажатии на кнопку «Показать все ряды номограммы», элемент экранной формы *«chart»* отражает заданный по индивидуальному заданию рисунок.

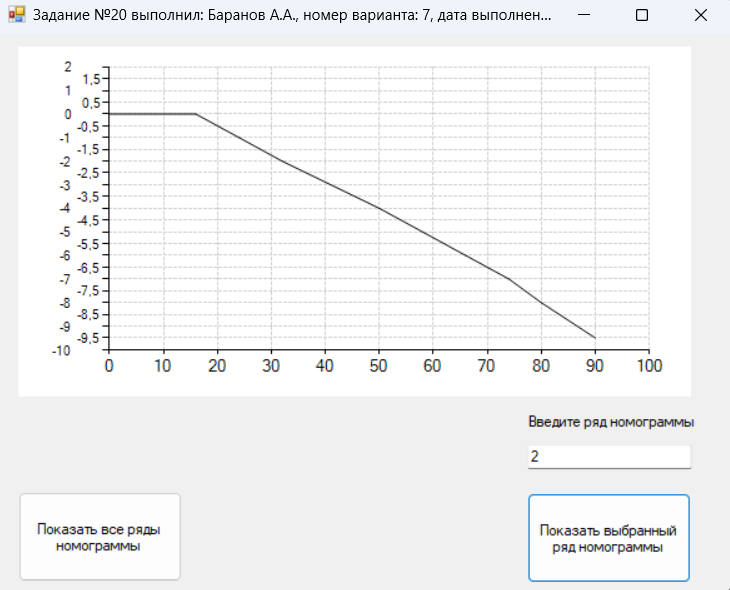


Рисунок 11 – Интерфейс главной формы при нажатии кнопки «Показать выбранный ряд номограммы»

При нажатии на кнопку «Показать все ряды номограммы», элемент экранной формы *«chart»* отражает заданный ряд номограммы.

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

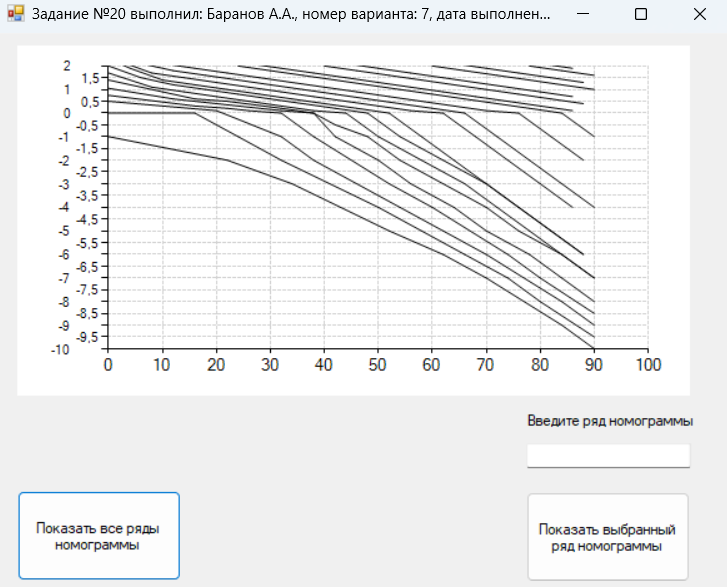


Рисунок 12 – Расчет первого тестового примера

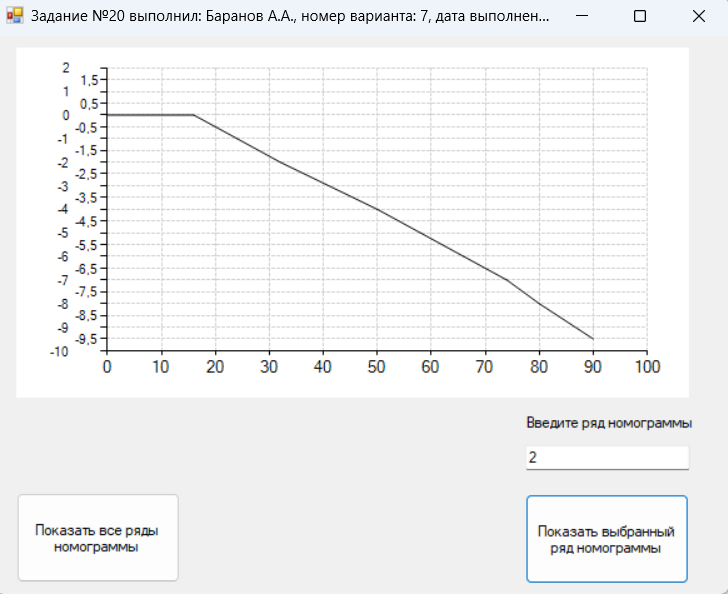


Рисунок 13 – Расчет второго тестового примера

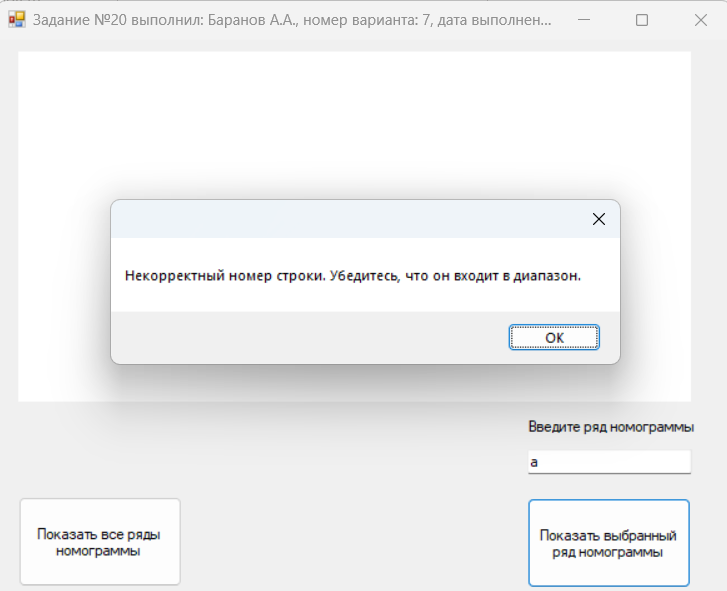


Рисунок 14 – Расчет третьего тестового примера

## 

## **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, закрепил для себя методы работы с файлами, а так же использовал внешний модуль *«Globalization»,* который помог в обработке десятичных чисел из файла «points.csv».