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

Грачева Наталья Сергеевна

Вариант №5

26.12.2024

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

\_\_.12.2024

(дата приёмки)

Москва – 2024 г.

**1. Цель работы**

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

**2. Формулировка задачи**



Загружать данные, представленные на круговых диаграммах, в программное обеспечение из *csv*-файла. Представлены шесть круговых диаграмм. Точное воспроизведение цветовой схемы диаграмм не требуется. Цветовая схема подбирается автором самостоятельно среди доступных средств фреймворка *.NET Framework*, но необходимо предоставить пользователю возможность просмотра как данных с каждой диаграммы в отдельности, так и данных со всех диаграмм сразу (согласно проиллюстрированному порядку). Предусмотреть пользовательскую возможность для последовательного подключения к уже демонстрируемым диаграммам указываемых пользователем, причём порядок размещения определяется выбором очередного названия ряда данных. В этом случае порядок вывода диаграмм на экран будет отличаться от представленного на исходной иллюстрации.

**3. Сети Петри запрограммированного технологического процесса**

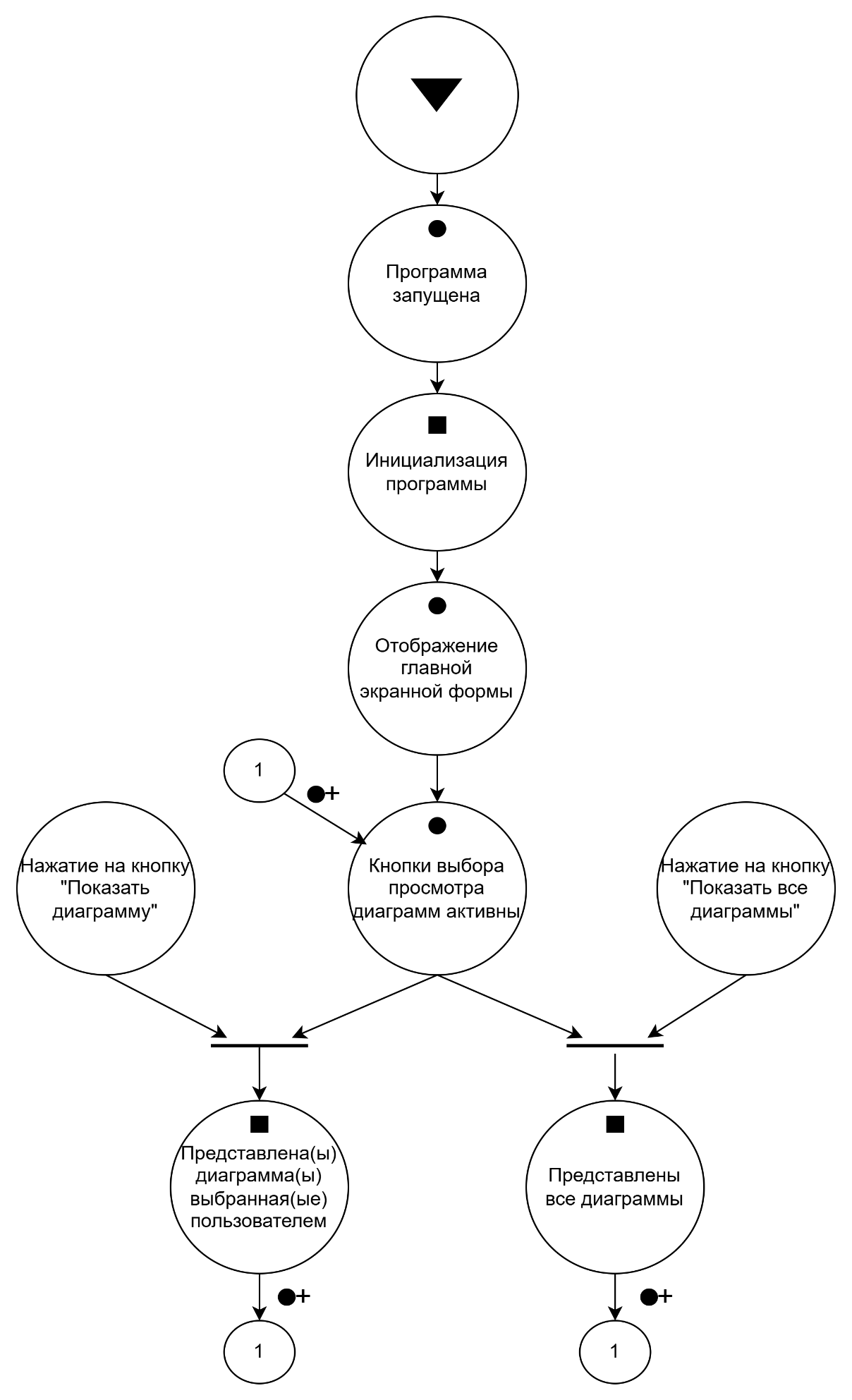


Рисунок 1 – Сеть Петри

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

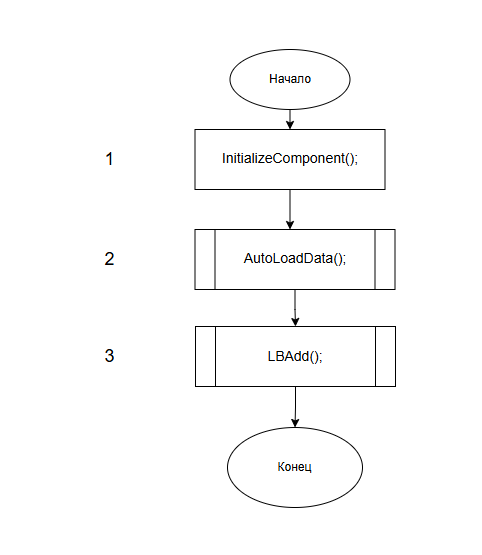


Рисунок 2 – Инициализация программы

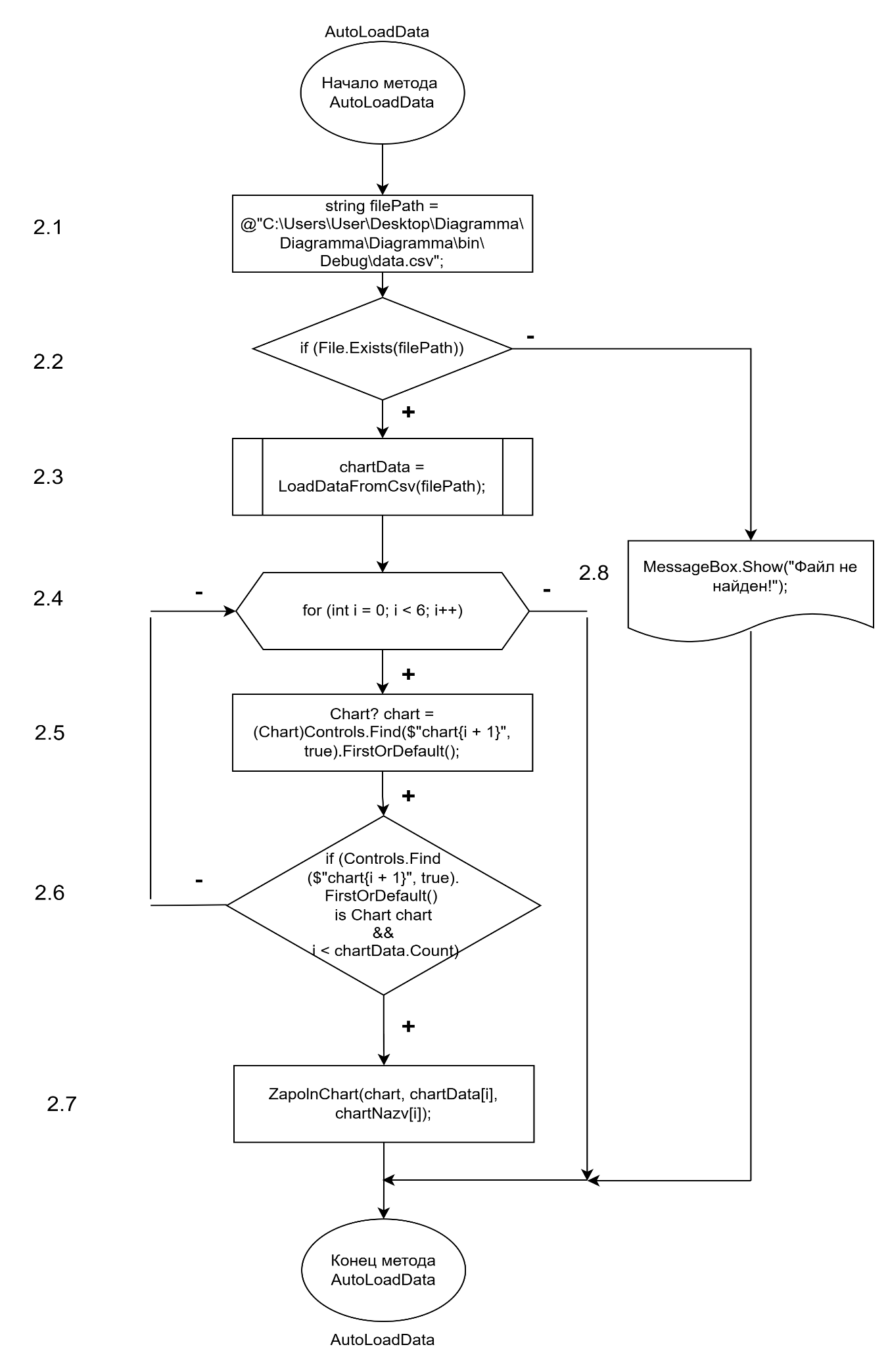


Рисунок 3 – Блок-схема методаAutoLoadData

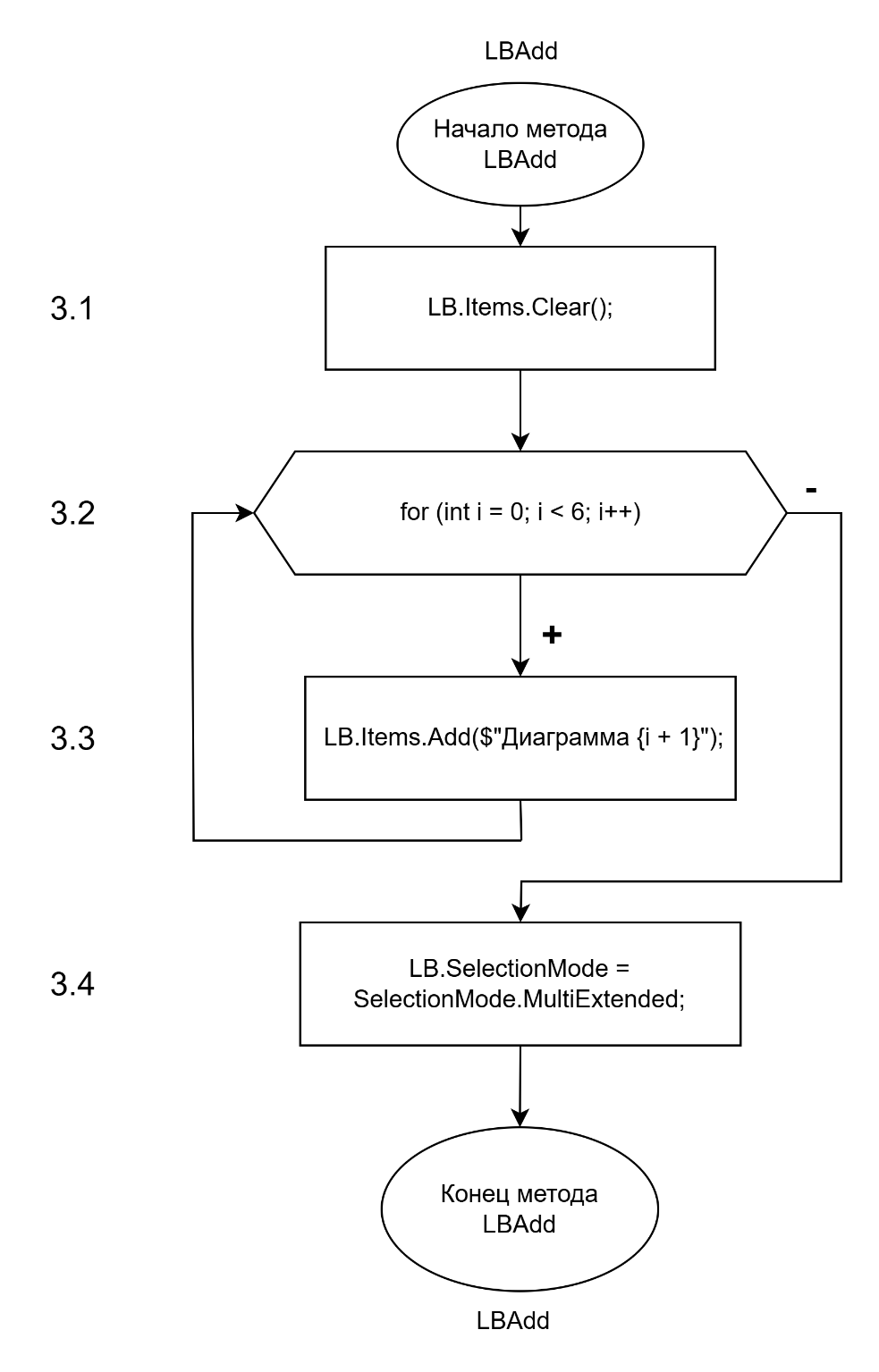


Рисунок 4 – Блок-схема методаLBAdd

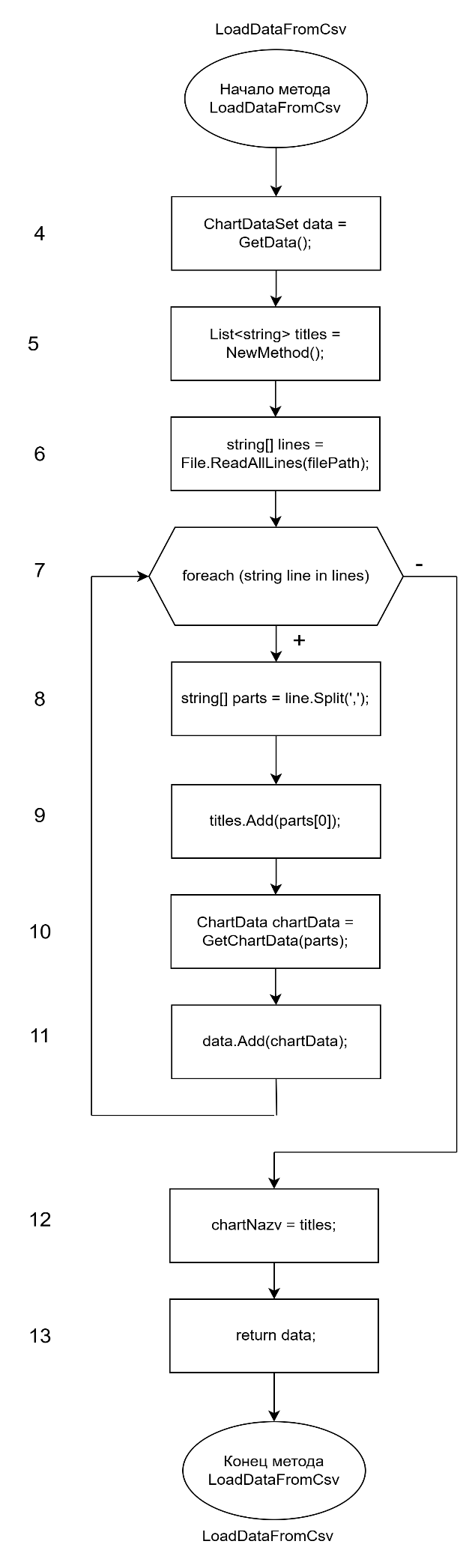


Рисунок 5 – Блок-схема методаLoadDataFromCsv

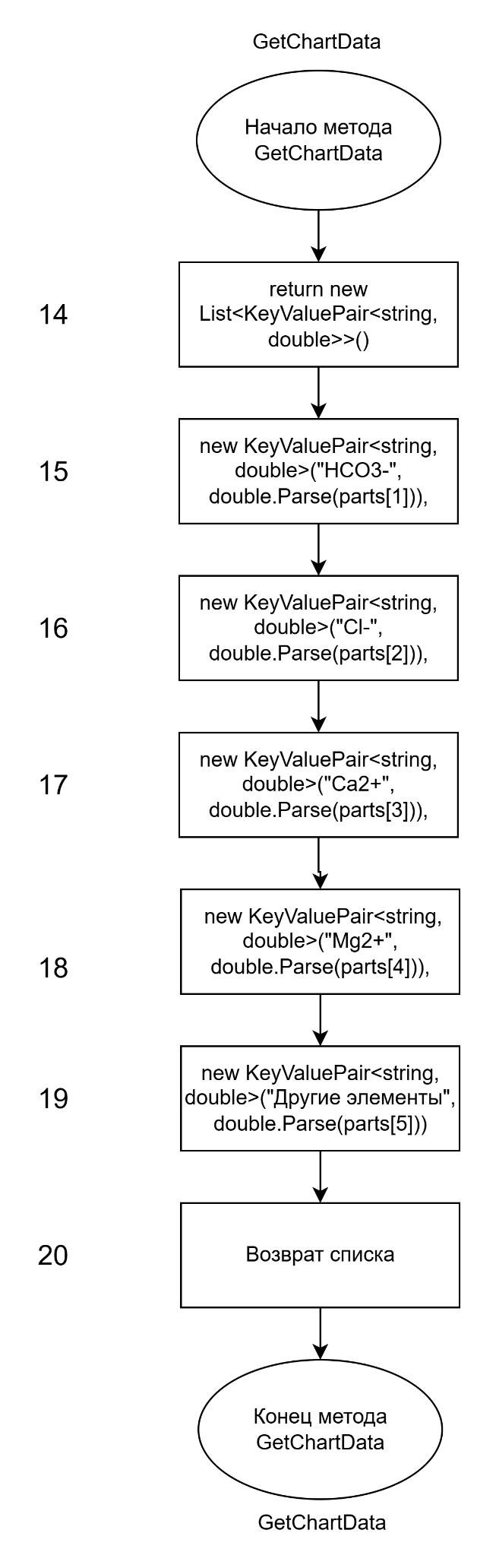


Рисунок 6 – Блок-схема метода GetChartData

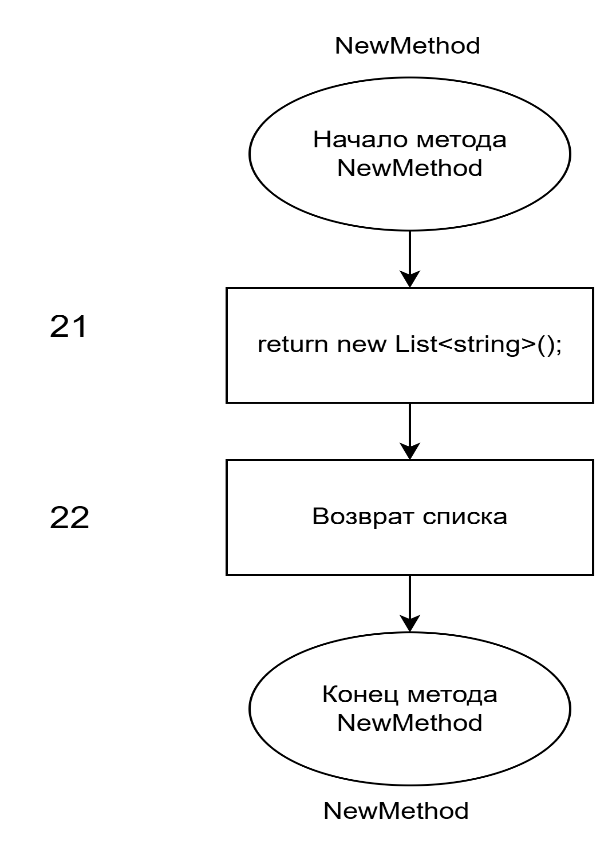


Рисунок 7 – Блок-схема метода NewMethod

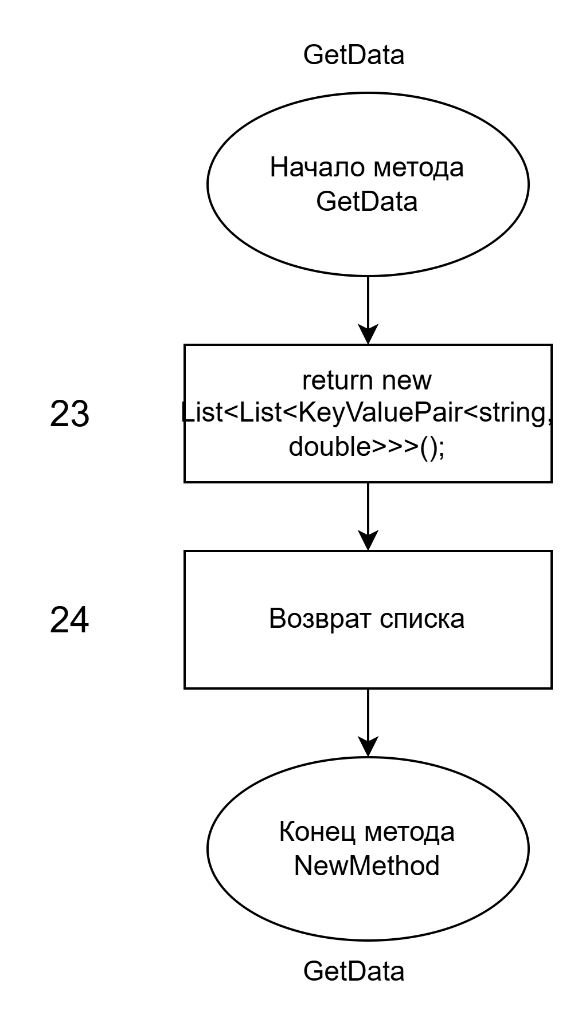
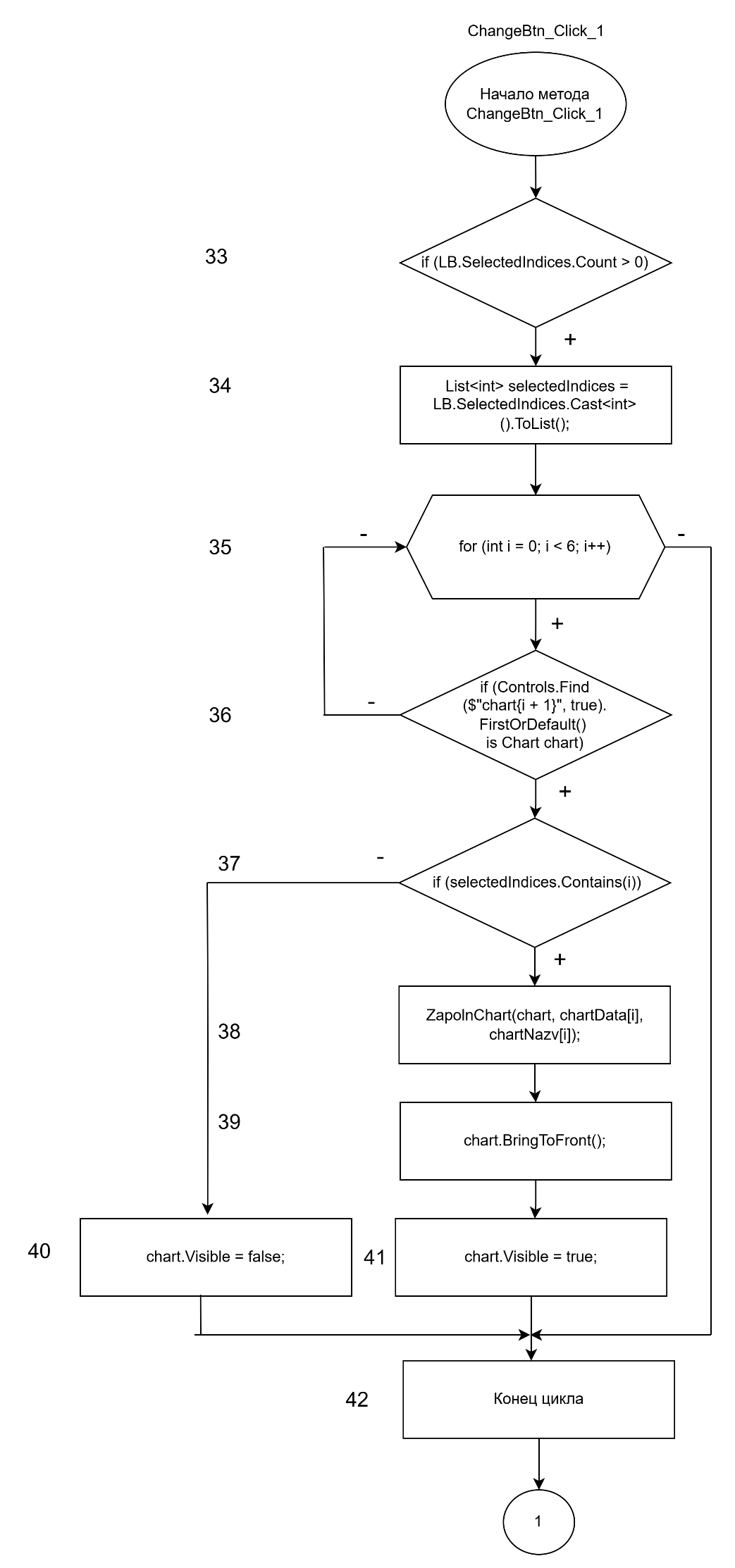


Рисунок 7 – Блок-схема метода GetData



Рисунок 7 – Блок-схема метода ZapolnChart



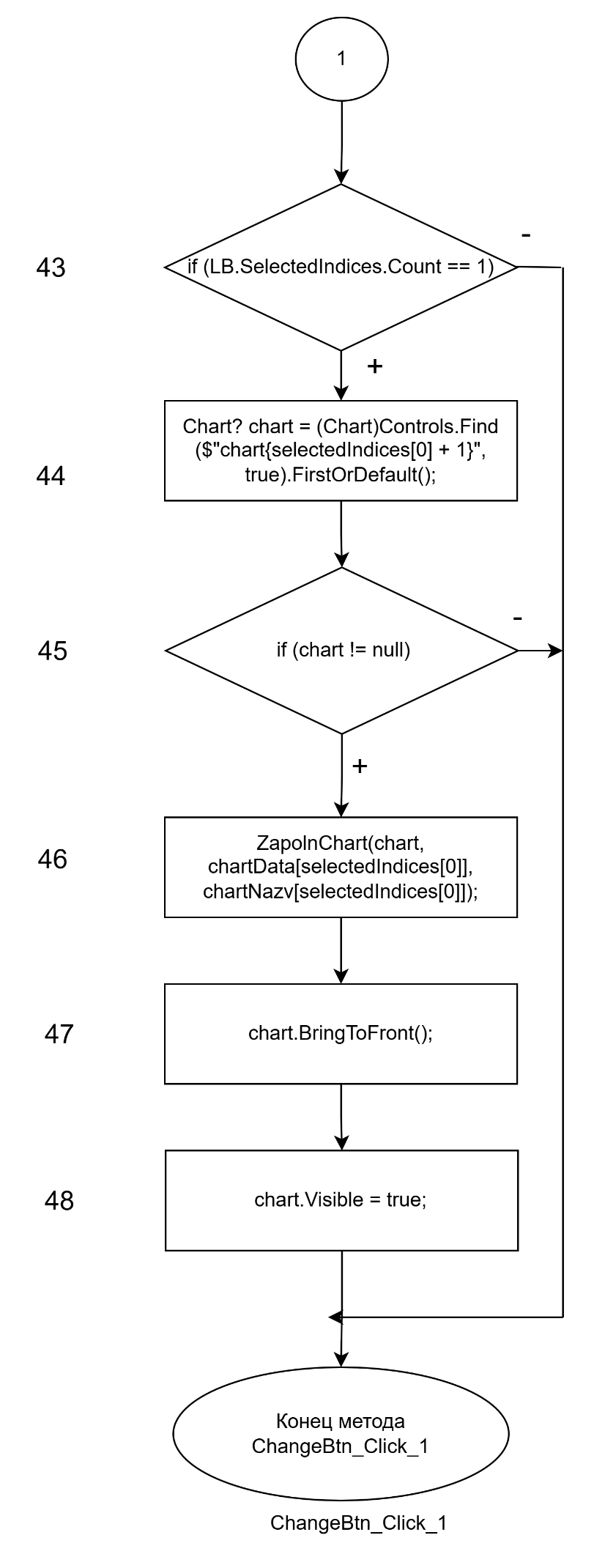


Рисунок 8 – Блок-схема метода ChangeBtn\_Click\_1

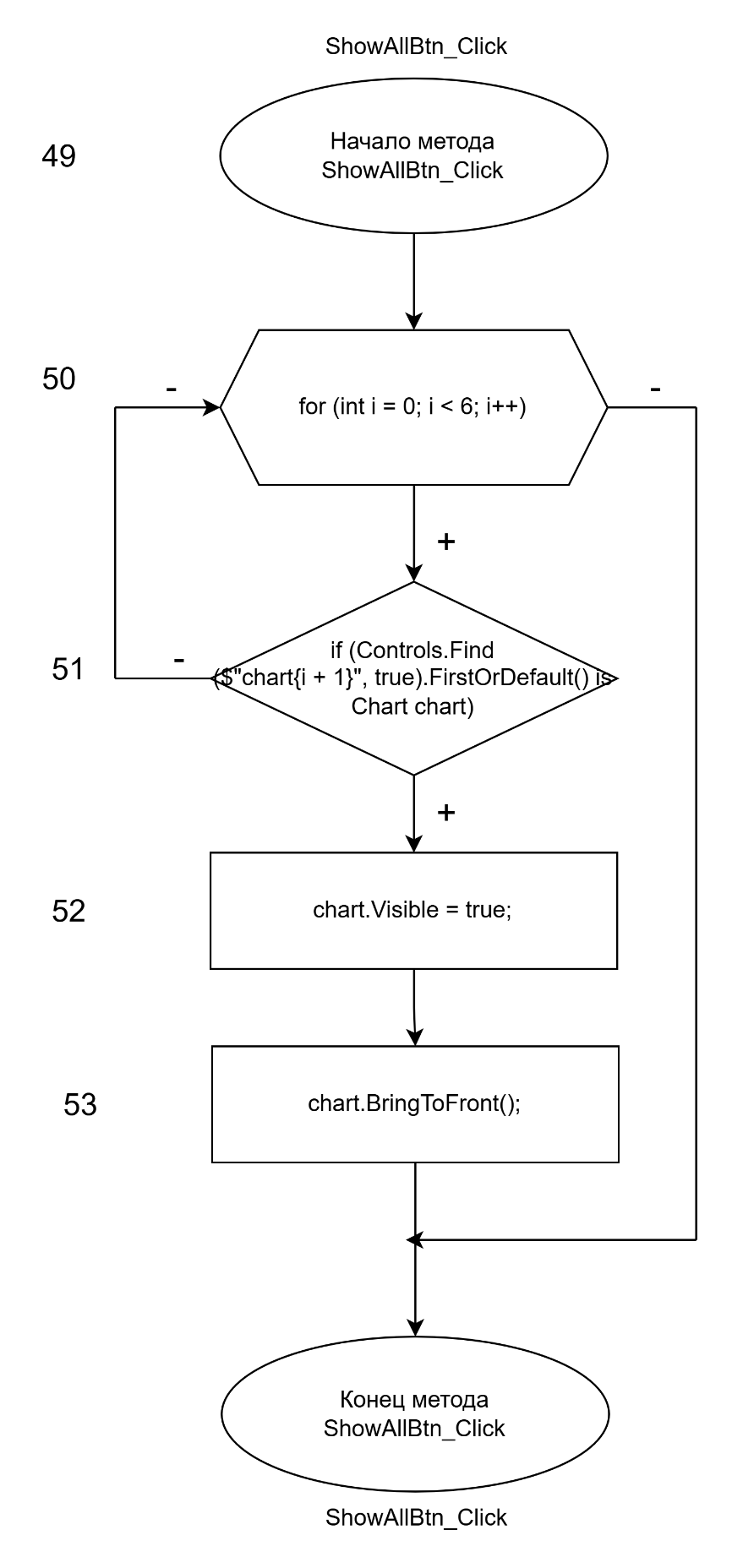


Рисунок 9 – Блок-схема метода ShowAllBtn\_Click

**5. Подбор тестовых примеров**

Пример 1:

Нажатие на кнопку «Показать диаграмму» предоставляет пользователю диаграмму, которую он выбрал.

Пример 2:

Нажатие на кнопку «Показать диаграмму» предоставляет пользователю выбора нескольких диаграмм.

Пример 3:

Нажатие на кнопку «Показать все диаграммы» показывает пользователю все диаграммы.

**6. Листинг (Код) программы**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Windows.Forms;

using System.Windows.Forms.DataVisualization.Charting;

namespace Diagramma

{

public partial class MyDiagramma : Form

{

ChartDataSet chartData = new();

List<string> chartNazv = new();

public MyDiagramma()

{

InitializeComponent();

LBAdd();

AutoLoadData();

}

private void AutoLoadData()

{

string filePath = @"C:\Users\User\Desktop\Diagramma\Diagramma\Diagramma\bin\Debug\data.csv";

if (File.Exists(filePath))

{

chartData = LoadDataFromCsv(filePath);

for (int i = 0; i < 6; i++)

{

Chart? chart = (Chart)Controls.Find($"chart{i + 1}", true).FirstOrDefault();

if (chart != null && i < chartData.Count)

{

ZapolnChart(chart, chartData[i], chartNazv[i]);

}

}

}

else

{

MessageBox.Show("Файл не найден!");

}

}

private void LBAdd()

{

LB.Items.Clear();

for (int i = 0; i < 6; i++)

{

LB.Items.Add($"Диаграмма {i + 1}");

}

LB.SelectionMode = SelectionMode.MultiExtended;

}

private ChartDataSet LoadDataFromCsv(string filePath) //загружаем данные из файла csv

{

ChartDataSet data = GetData(); //данные диаграмм

List<string> titles = NewMethod();//названия диаграмм

string[] lines = File.ReadAllLines(filePath);//считывание всех строк файла

foreach (string line in lines)//перебираем строки

{

string[] parts = line.Split(',');//разбиваем на части (то есть запятыми)

titles.Add(parts[0]);//первую часть(в файле это заголовок) добавляем в заголовок в диаграмме

ChartData chartData = GetChartData(parts);

data.Add(chartData); //добавляем этот список в общий список

}

chartNazv = titles; //присваиваем заголовки

return data;

}

private static ChartData GetChartData(string[] parts)

{

return new List<KeyValuePair<string, double>>()//список для данных диаграмм(тут это значения, то есть какой цвет на диаграмме за что отвечает)

{

new KeyValuePair<string, double>("HCO3-", double.Parse(parts[1])),

new KeyValuePair<string, double>("Cl-", double.Parse(parts[2])),

new KeyValuePair<string, double>("Ca2+", double.Parse(parts[3])),

new KeyValuePair<string, double>("Mg2+", double.Parse(parts[4])),

new KeyValuePair<string, double>("Другие элементы", double.Parse(parts[5]))

};

}

private static List<string> NewMethod()

{

return new List<string>();

}

private static ChartDataSet GetData()

{

return new List<List<KeyValuePair<string, double>>>();

}

private void ZapolnChart(Chart chart, ChartData data, string title)//заполняем диаграммы данными

{

chart.Series.Clear();//очищаем все серии диаграммы

Series series = new Series//создаем новую серию

{

ChartType = SeriesChartType.Pie,//то, что крутовая диаграмма

IsValueShownAsLabel = true//значения будут отображаться в виде меток на долях

};

for (int i = 0; i < data.Count; i++)//перебираем элементы данных, чтобы задать начальный круг диаграммы

{

series.Points.AddXY(data[i].Key, data[i].Value);//добавляем точки данных

}

series["PieStartAngle"] = "180"; //ставим, что начальный круг - 180 градусов верх

chart.Series.Add(series); //добавляем данные

chart.Titles.Clear(); //заголовки очищаем

chart.Titles.Add(title); //добавляем новые

}

private void ChangeBtn\_Click\_1(object sender, EventArgs e) //выбираем определенную диаграмму

{

if (LB.SelectedIndices.Count > 0) //если элементы в листбоксе выбрали

{

List<int> selectedIndices = LB.SelectedIndices.Cast<int>().ToList(); //получаем индексы элементов, которые выбрали

for (int i = 0; i < 6; i++) //перебираем все диаграммы

{

if (Controls.Find($"chart{i + 1}", true).FirstOrDefault() is Chart chart) //если диаграмму нашли

{

if (selectedIndices.Contains(i)) // и если индекс содержится в выбранных

{

ZapolnChart(chart, chartData[i], chartNazv[i]); //то заполняем данными

chart.BringToFront(); // и выносим ее вперед

chart.Visible = true; //и чтобы ее было видно

}

else //иначе - делаем ее невидимой

{

chart.Visible = false;

}

}

}

if (LB.SelectedIndices.Count == 1) //если выбран только 1 элемент

{

Chart? chart = (Chart)Controls.Find($"chart{selectedIndices[0] + 1}", true).FirstOrDefault(); //ищем соответствующий

if (chart != null) //если диаграмма найдена

{

ZapolnChart(chart, chartData[selectedIndices[0]], chartNazv[selectedIndices[0]]); //заполняем ее

chart.BringToFront();//выносим вперед

chart.Visible = true; //делаем видимой

}

}

}

}

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

{

for (int i = 0; i < 6; i++) //перебираем все диаграммы

{

if (Controls.Find($"chart{i + 1}", true).FirstOrDefault() is Chart chart)//если диаграммы найдены

{

chart.Visible = true;//делаем видимыми

chart.BringToFront();//и выносим вперед

}

}

}

}

}

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

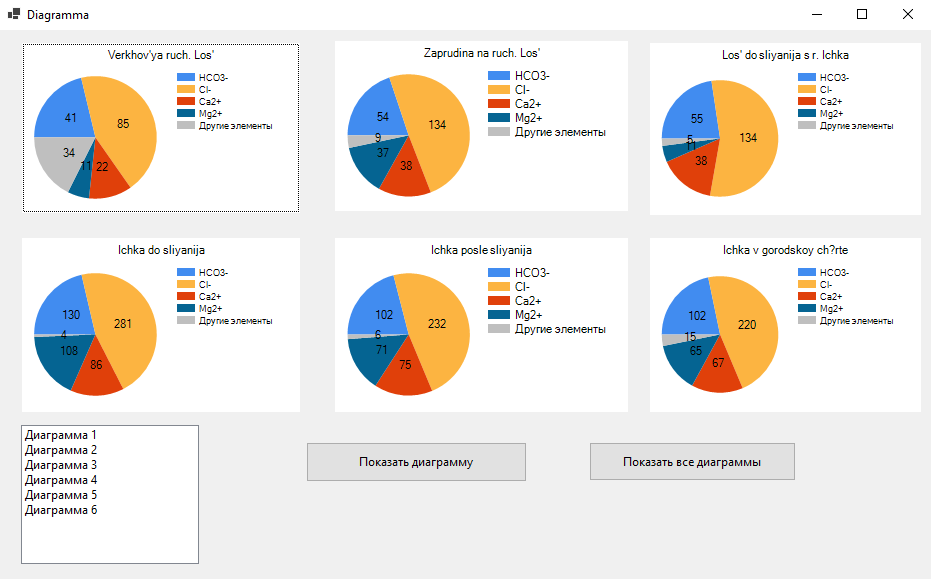
****

Рисунок 10 – Графический пользовательский интерфейс программного обеспечения

Описание:

На форме присутствуют две кнопки: «Показать диаграмму» и «Показать все диаграммы».

Нажатие на кнопку «Показать диаграмму» предоставляет пользователю выбрать одну или несколько диаграмм для просмотра.

Нажатие на кнопку «Показать все диаграммы» предоставляет пользователю все диаграммы.

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

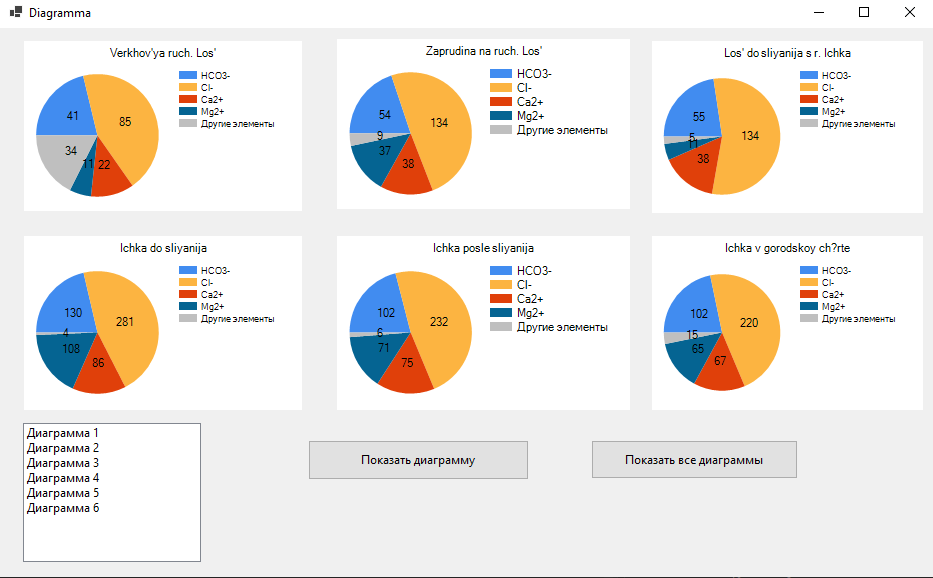


Рисунок 11 – Графический пользовательский интерфейс программного обеспечения

Представлены шесть круговых диаграмм. Точное воспроизведение цветовой схемы диаграмм не требуется. Пользователю предоставлена возможность выбора как одной диаграммы, так и нескольких, а также просмотра всех диаграмм.

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

Пример 1:

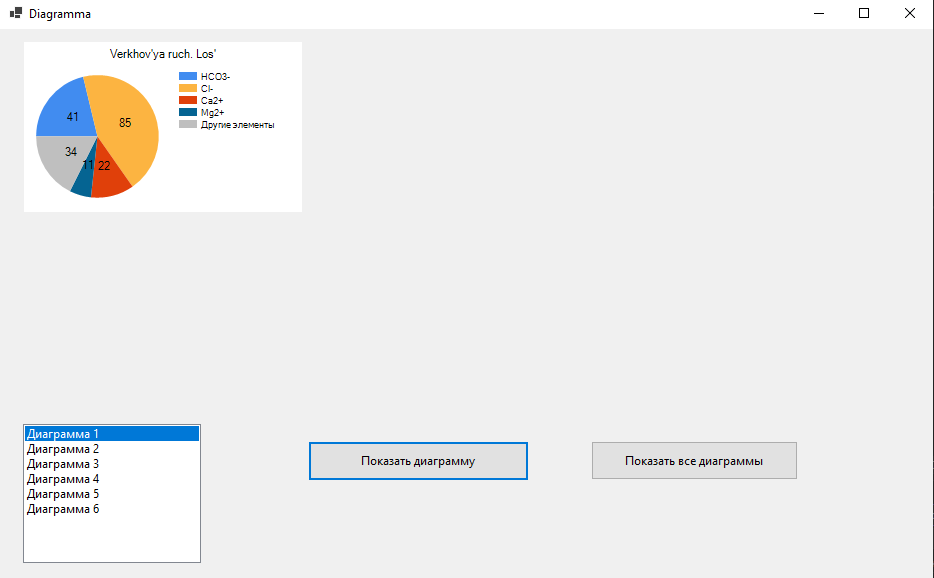
****

Рисунок 12 – Расчёт тестового примера №1 с использованием составленного программного обеспечения

Пример 2:

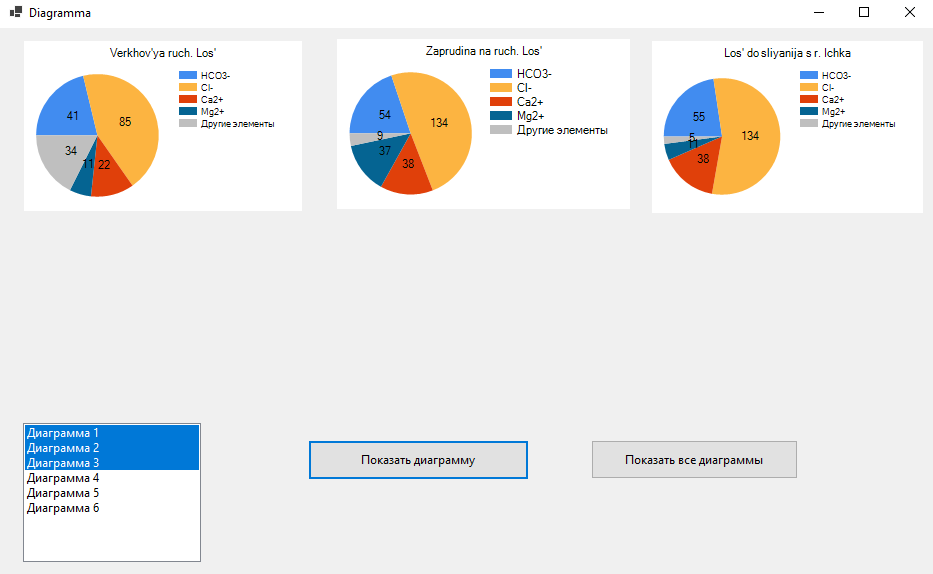
****

Рисунок 13 – Расчёт тестового примера №2 с использованием составленного программного обеспечения

Пример 3:

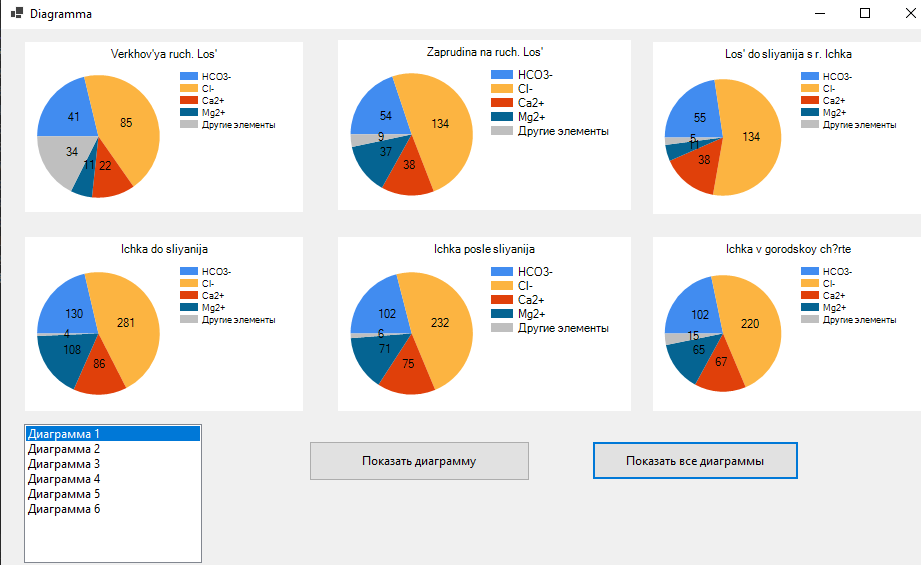


Рисунок 14 – Расчёт тестового примера №3 с использованием составленного программного обеспечения

**10. Вывод**

В процессе работы освоила работу с элементом экранной формы «*Сhart*», закрепила для себя подключение и использование внешних модулей для работы с диаграммами. В процессе работы я также изучила различные типы диаграмм и их настройки, что расширило мои навыки визуализации данных. Также научилась обрабатывать и представлять данные. Данная работа позволила закрепить теоретические знания и применить их на практике.