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»

Курсовая работа

по дисциплине:

«Методы программирования»

Выполнил: ст. гр. ТКИ-341

Черкас Н.В., Родин К.П.

Вариант №12

Проверил: к.т.н., доцент Сафронов А.И.

Москва – 2024 г.

Содержание

[1 Введение 3](#_Toc166350403)

[2 Цель работы 3](#_Toc166350404)

[3 Формулировка задачи 3](#_Toc166350405)

[4 Содержательная часть работы 4](#_Toc166350406)

[4.1 Анализ печатной формы 4](#_Toc166350407)

[4.2 Описание предназначения печатной формы 4](#_Toc166350408)

[4.3 Разнообразие печатных форм и выделение различий 4](#_Toc166350409)

[4.4 Унификация печатных форм для удобстав обработки данных 4](#_Toc166350410)

[4.5 Разработка экранной формы для импорта сведений из печатной формы 5](#_Toc166350411)

[4.6 Разработка функции экспорта сведений из печатной формы в файл текстовый формата CSV 7](#_Toc166350412)

[4.7 Разработка функции импорта сведений из оригинального и изменённого текстовых файлов формата CSV с воссозданием Word 8](#_Toc166350413)

[4.8 Разработка функции экспорта сведений из экранной формы в Excel 10](#_Toc166350414)

[4.9 Графический анализ данных, содержащихся в печатных формах 11](#_Toc166350415)

5 [Инструкция пользователя 17](#_Toc166350419)

[6 Заключение 19](#_Toc166350420)

[7 Выводы 19](#_Toc166350421)

# Введение

В рамках данной работы рассмотрена автоматизация чтения данных из документа Word и ее последующее экспортирование в CSV или Excel. Так же по документам строится статистика в виде столбчатых диаграмм. Автоматизация подобных задач является актуальной в любой сфере деятельности, так как ведение документации является обязательной частью любой деятельности.

# Цель работы

Закрепить навыки, полученные в рамках курса «Методы программирования», в том числе: подключение внешних модулей (библиотек классов) и их использования в разрабатываемых программах, программируемая настройка электронных таблиц Excel и документов Word. Освоить навыки работы с элементом Chart и построением графиков и диаграмм.

# Формулировка задачи

1. Проанализировать печатные формы.

2. Уметь воссоздавать структуру печатной формы.

3. Уметь читать печатную форму и представлять информацию из неё на экранной форме.

4. Уметь конвертировать печатную форму Word - Excel / Excel - Word (в зависимости от исходно выданного документа).

5. Уметь анализировать данные, получаемые из нескольких печатных форм за различные периоды (и представлять результаты анализа в Chart-элементе).

6. Технологический процесс работы приложения отразить на карте, выполненной в нотации сетей Петри, в сочетании с классическими схемами алгоритма для этапов, отмеченных как "эффекты".

# Содержательная часть работы

## Анализ печатных форм

Печатные формы представляют из себя списки присутствующих преподавателей, а также тем для обсуждения на заседаниях кафедры «Управление и защита информации» в 2014 – 2015 учебном году.

## Описание предназначения печатных форм

Формы предназначены для хранения информации о присутствующих сотрудниках и темах для обсуждения на заседаниях кафедры «Управление и защита информации».

## Разнообразие печатных форм и выделение различий

Главное различие печатных форм заключается в дате проведения заседаний кафедры, а также в составе присутствующих на заседаниях и списке тем для обсуждения.

## Унификация печатных форм для удобства обработки данных

Для унификации печатных форм были удалены все заголовки и символы, мешающие чтению текста (перенос каретки, новый параграф, разрыв страницы и т.д.).

## Разработка экранной формы для импорта сведений из печатной формы

Так как основную информацию в себе несет текстовый файл, в рамках данной работы было решено представить информацию в элементе Label. Для удобства данные считываются из файла Microsoft Word. Пример работы данной программы представлен на рисунках 1 и 2:

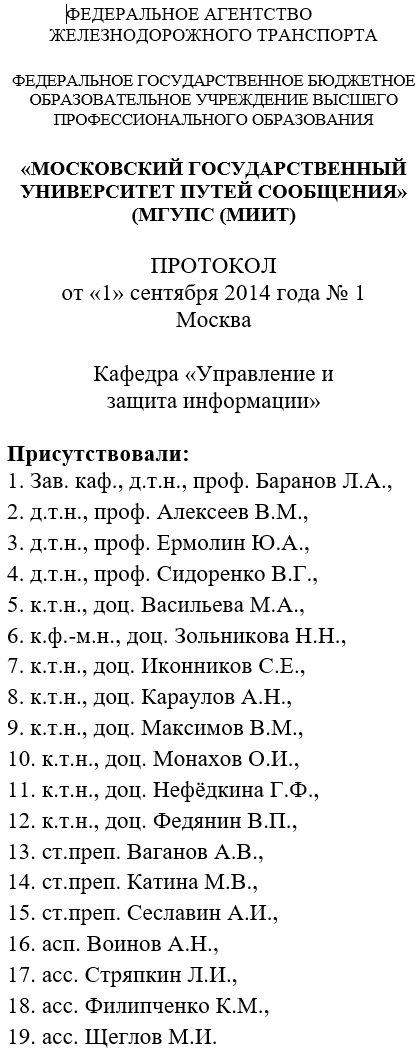


Рисунок – Исходный текстовый файл MS Word

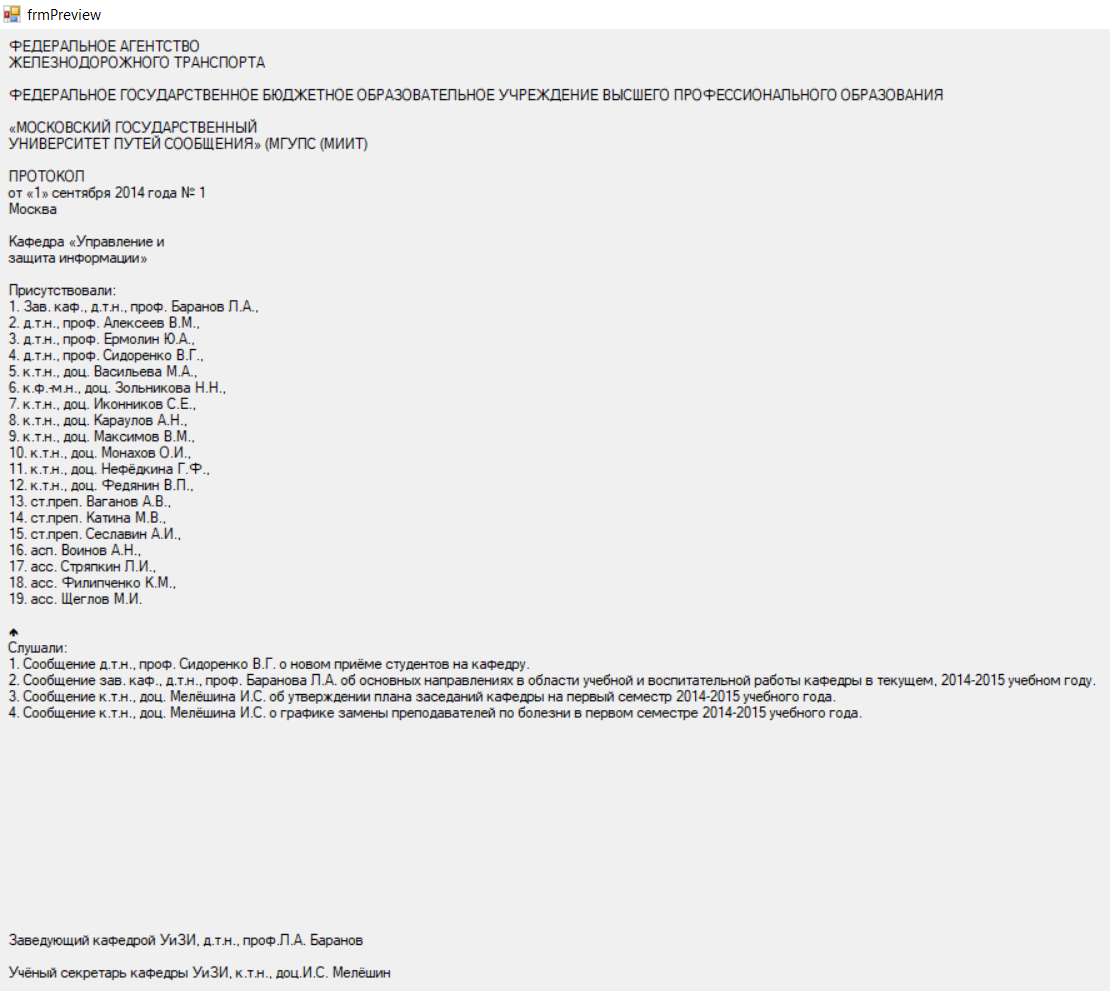


Рисунок - Текст, считанный в элемент Label

Листинг кода экранной формы:

using System;

using System.Windows.Forms;

using Word = Microsoft.Office.Interop.Word;

namespace Coursework

{

public partial class frmPreview : Form

{

private string filename;

public frmPreview(string filename)

{

this.filename = filename;

InitializeComponent();

}

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

{

this.WindowState = FormWindowState.Maximized;

Word.Application app = new Word.Application();

Word.Document doc = app.Documents.Open(filename);

this.lblpreview.Text = doc.Content.Text;

doc.Close();

app.Quit();

}

}

}

Сеть Петри формирования предпросмотра:

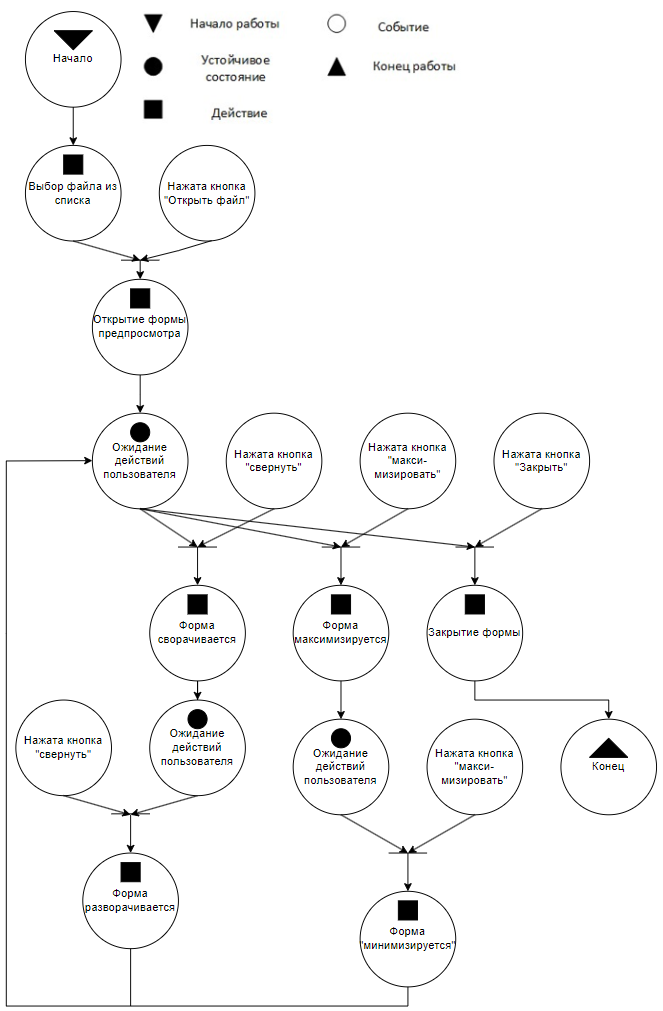


Рисунок 3 – Сеть Петри формирования предпросмотра

Составление схем алгоритмов методов в составе решения, отмеченных на сети Петри в качестве «эффектов» (метка ):

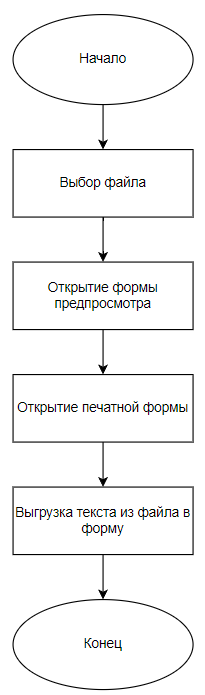


Рисунок 4 – Схема алгоритма метода формирования предпросмотра

## Разработка функции экспорта сведений из печатной формы в файл текстовый формата CSV

Данные считываются из печатной формы и записываются в CSV файл. Пример работы представлен на Рисунках 1 и 3:

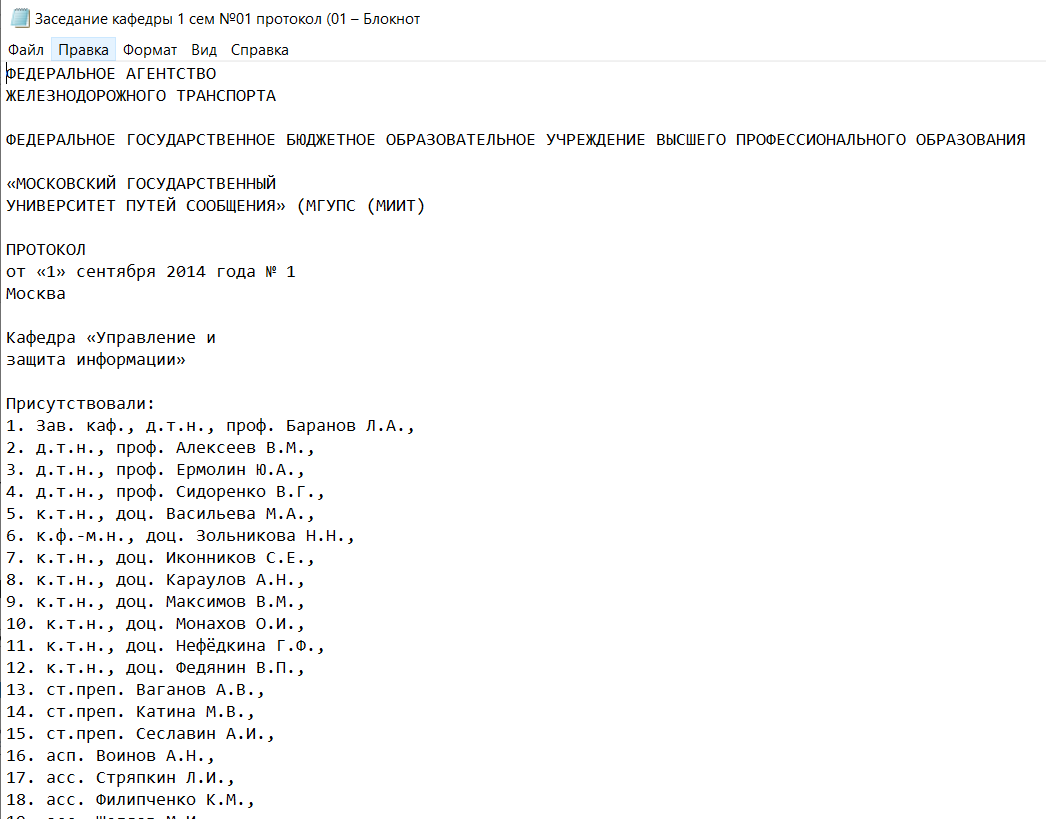


Рисунок 5 - Генерируемый программой CSV файл

Листинг кода функции экспорта в CSV файл:

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

{

wordapp = new Word.Application();

wd = wordapp.Documents.Open(this.files[this.cmbFiles.SelectedIndex], ReadOnly:true);

string csvContent = string.Empty;

foreach (Word.Paragraph paragraph in wd.Paragraphs)

{

csvContent += $"{paragraph.Range.Text}";

}

File.WriteAllText(Application.StartupPath + "\\csv\\" + ChngExtension(this.cmbFiles.SelectedItem.ToString(), ".csv"), csvContent);

wd.Close();

wordapp.Quit();

}

public string ChngExtension(string filename, string ext)

{

string[] temp = filename.Split('.');

return temp[0] + ext;

}

Сеть Петри экспорта в CSV:

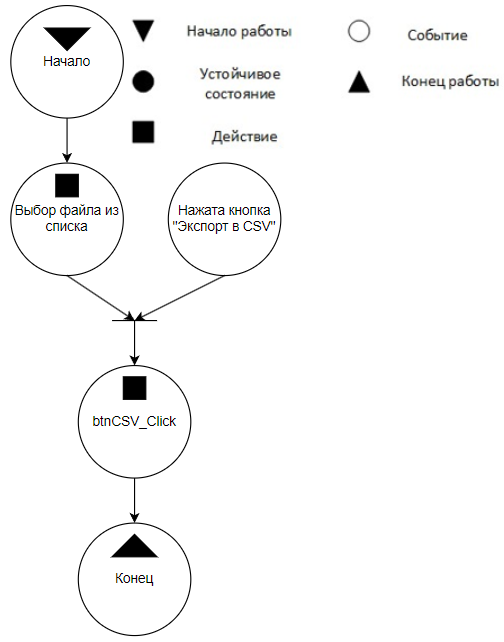


Рисунок 6 – Сеть Петри Импорта в CSV

Составление схем алгоритмов методов в составе решения, отмеченных на сети Петри в качестве «эффектов» (метка ):

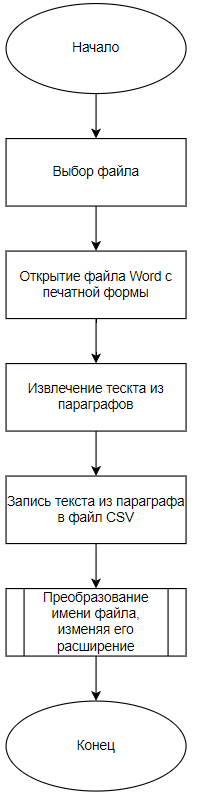


Рисунок 7 – Схема алгоритма метода btnCSV\_Click

## Разработка функции импорта сведений из оригинального и изменённого текстовых файлов формата CSV с воссозданием Word

Данные построчно считываются из CSV файла и записываются в текстовый документ MS Word. Пример работы представлен на рисунках 5 и 8:

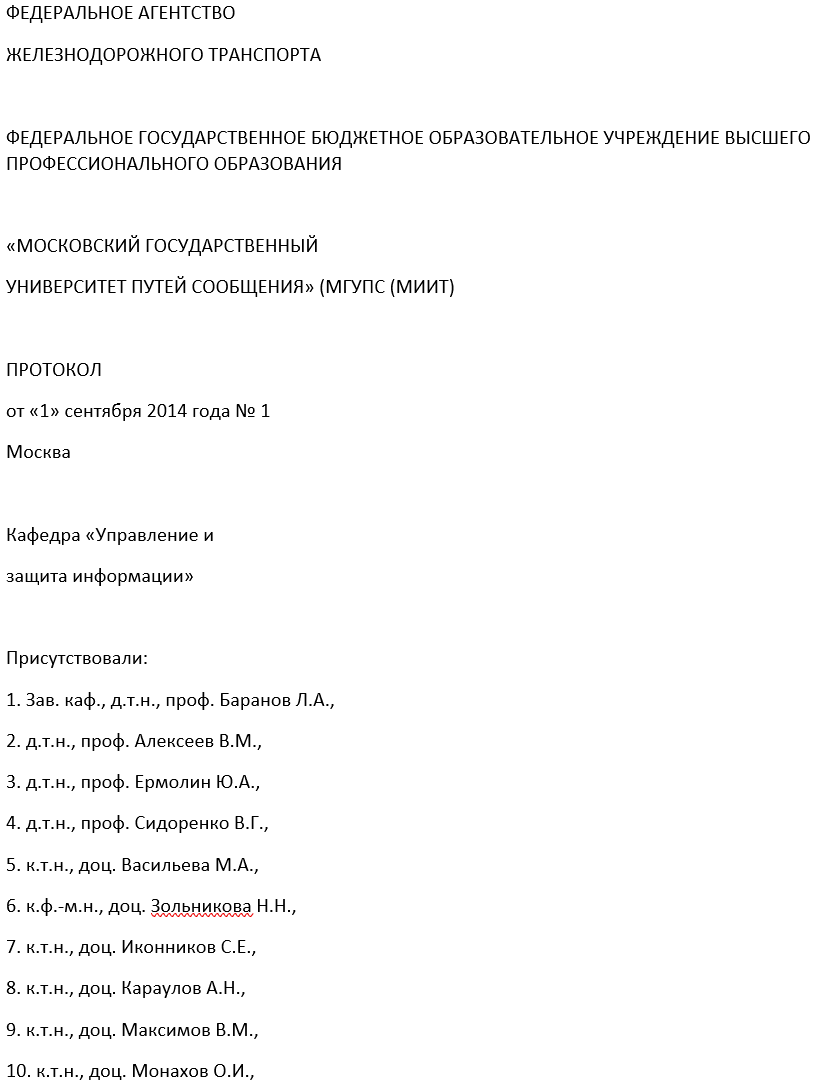


Рисунок 8 - Восстановленный из CSV файла документ

Листинг кода функции импорта из CSV:

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

{

Word.Application wordApp = new Word.Application();

Word.Document wordDoc = wordApp.Documents.Add();

string[] csvLines = File.ReadAllLines(Application.StartupPath + "\\csv\\" + ChngExtension(this.cmbFiles.SelectedItem.ToString(), ".csv"));

foreach (string line in csvLines)

{

wordDoc.Content.Text += line;

}

wordDoc.Save();

wordDoc.Close();

}

Сеть Петри импорта из CSV:

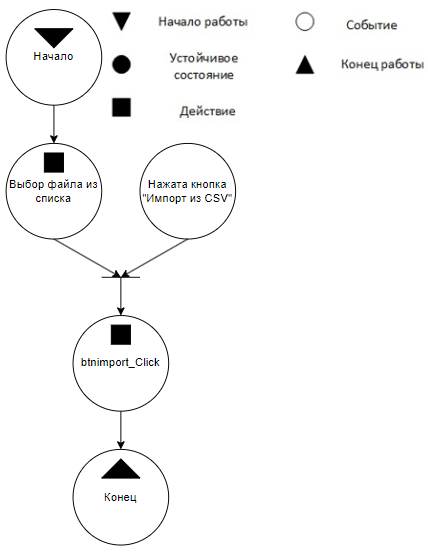


Рисунок 9 – Сеть Петри импорта из CSV

Составление схем алгоритмов методов в составе решения, отмеченных на сети Петри в качестве «эффектов» (метка ):

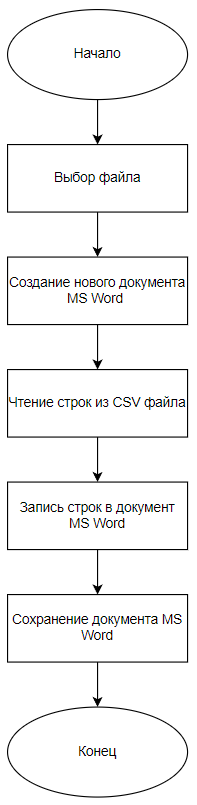


Рисунок 10 – Схема алгоритма метода btnimport\_Click

## Разработка функции экспорта сведений из экранной формы в Excel

Для удобства данные для записи в Excel файл считываются из CSV файла. Пример работы представлен на рисунках 5 и 11:



Рисунок 11 - Генерируемый программой файл Excel

Листинг кода функции экспорта в Excel:

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

{

Excel.Application excelApp = new Excel.Application();

excelApp.Workbooks.Add();

Excel.Worksheet workSheet = excelApp.Worksheets[1];

string[] csvLines = File.ReadAllLines(Application.StartupPath + "\\csv\\" + ChngExtension(this.cmbFiles.SelectedItem.ToString(), ".csv"));

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

{

string[] csvValues = csvLines[i].Split(';');

for (int j = 0; j < csvValues.Length; j++)

{

workSheet.Cells[i + 1, j + 1] = csvValues[j];

}

}

excelApp.ActiveWorkbook.SaveAs(Application.StartupPath + "\\ExcelFiles\\" + ChngExtension(this.cmbFiles.SelectedItem.ToString(), ".xlsx"));

excelApp.Quit();

}

Сеть Петри для экспорта в Excel:

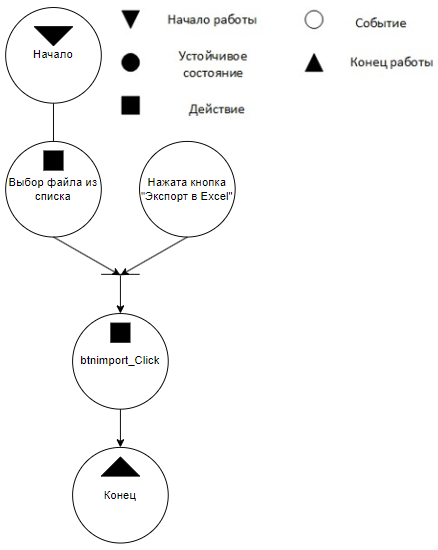


Рисунок 12 – Сеть Петри для экспорта в Excel

Составление схем алгоритмов методов в составе решения, отмеченных на сети Петри в качестве «эффектов» (метка ):

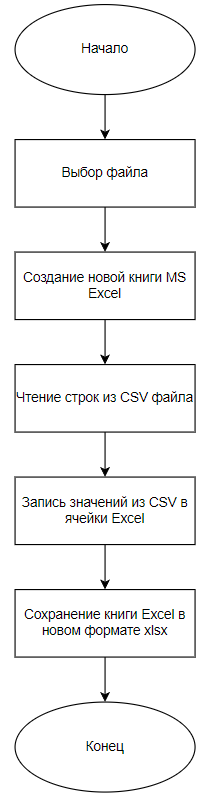


Рисунок 13 – Схема алгоритма метода btnExcel\_Click

## Графический анализ данных, содержащихся в печатных формах

Для решения данной задачи было использовано поле для отрисовки графиков Chart. Программа извлекает информацию из CSV файлов, а затем на основе полученных данных строит график, который выбрал пользователь. Ниже на рисунках 13 – 24 представлены графики и диаграммы, выводимые программой.

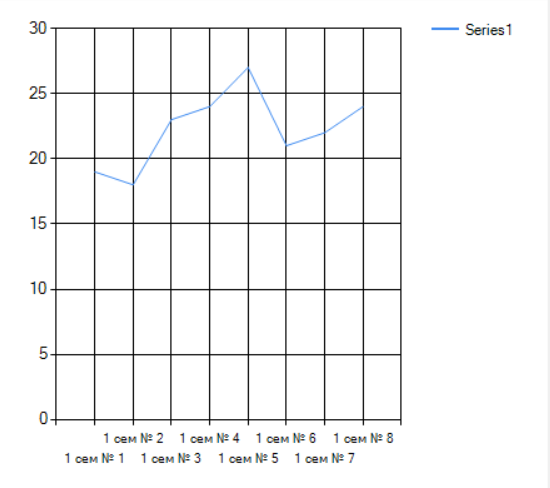


Рисунок 13 – График посещаемости всех сотрудников за 2014 год

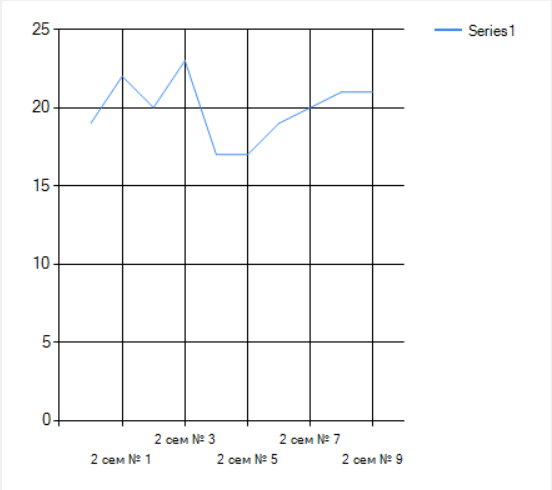


Рисунок 14 – График посещаемости всех сотрудников за 2015 год

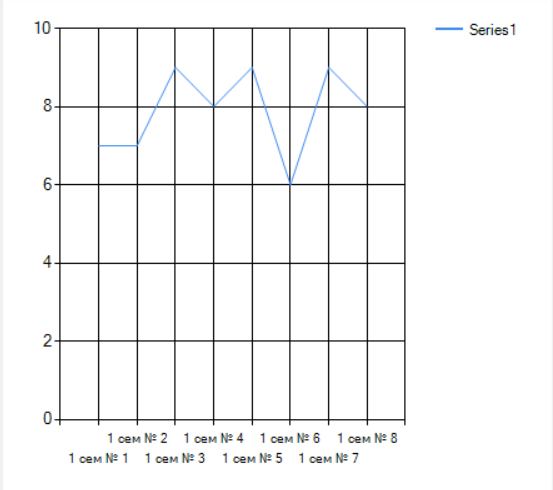


Рисунок 15 – График посещаемости профессорско-преподавательского состава за 2014 год

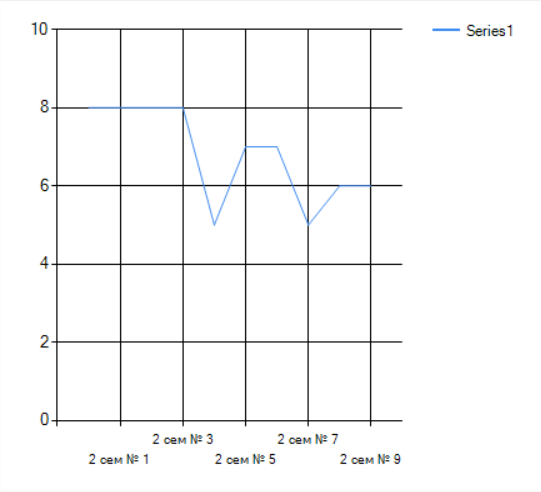


Рисунок 16 – График посещаемости профессорско-преподавательского состава за 2015 год

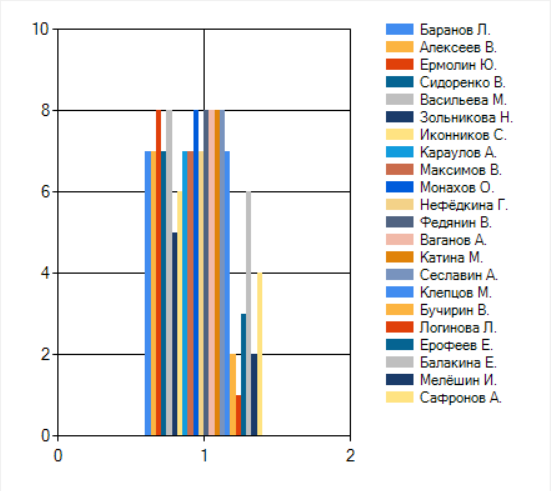


Рисунок 17 – Диаграмма индивидуальной посещаемости за 2014 год

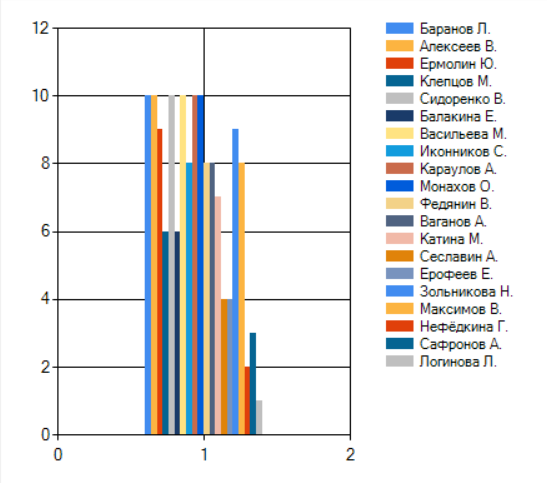


Рисунок 18 – Диаграмма индивидуальной посещаемости за 2015 год

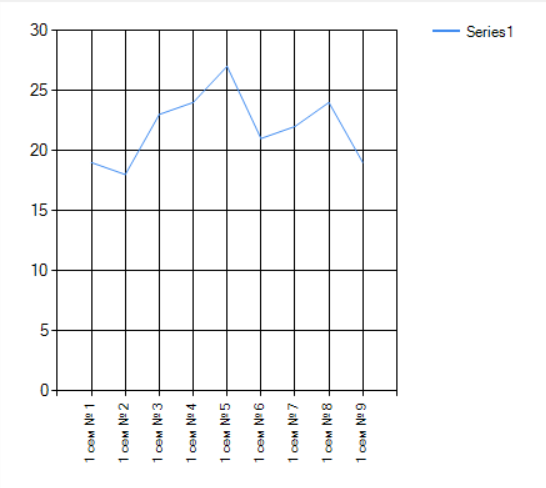


Рисунок 19 – График посещаемости всех сотрудников за первый семестр

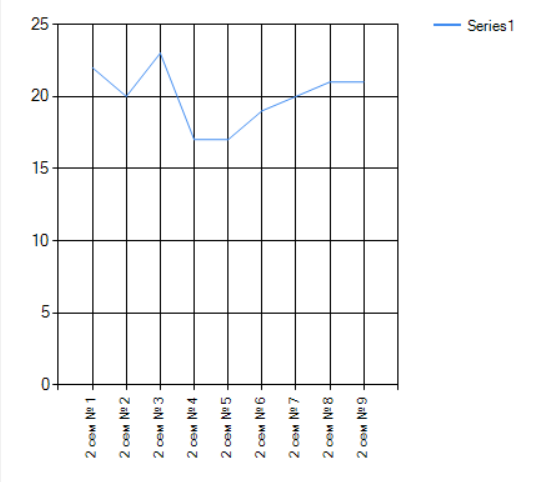


Рисунок 20 – График посещаемости всех сотрудников за второй семестр

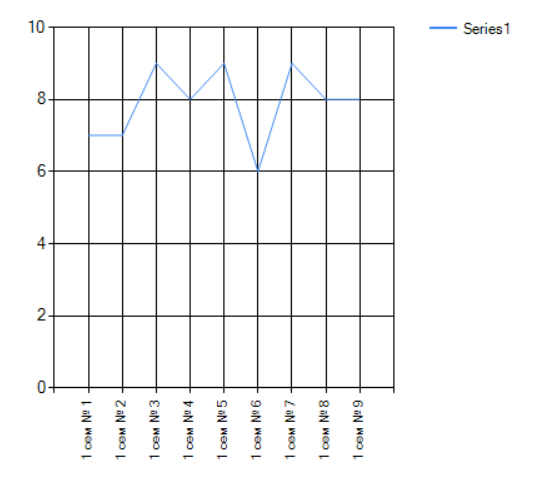


Рисунок 21 – График посещаемости профессорско-преподавательского состава за первый семестр

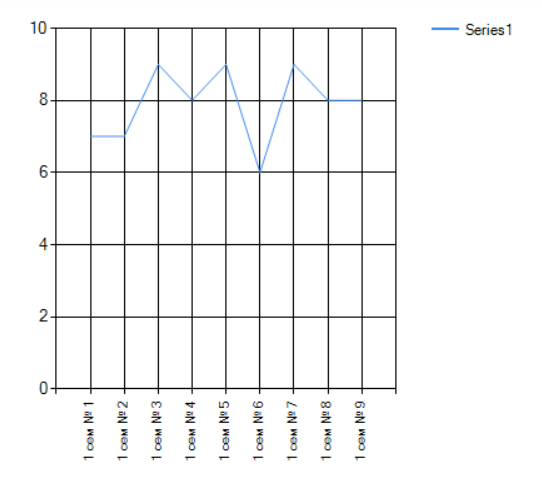


Рисунок 22 – График посещаемости профессорско-преподавательского состава за второй семестр

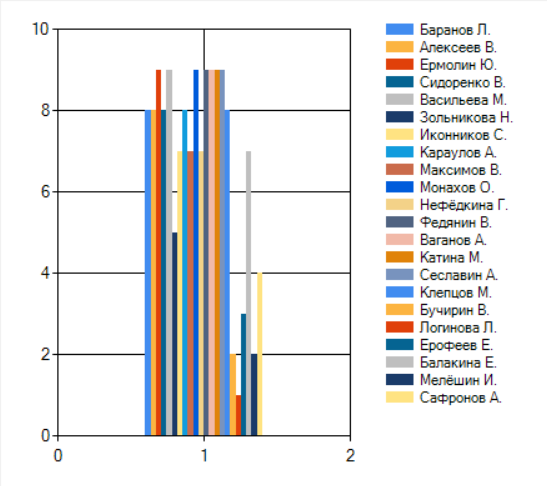


Рисунок 23 – Диаграмма индивидуальной посещаемости за первый семестр

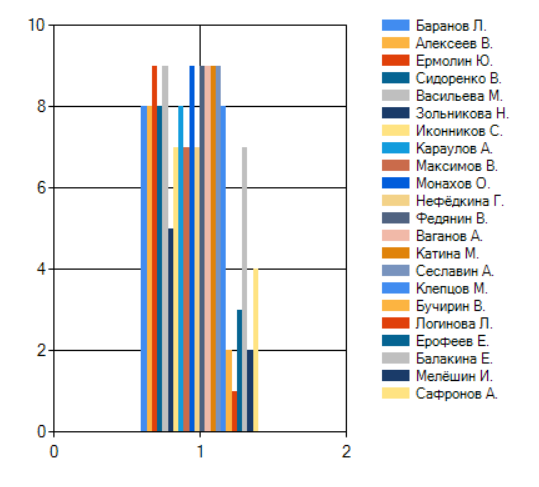


Рисунок 24 – Диаграмма индивидуальной посещаемости за второй семестр

Листинг кода формы со статистикой:

using System;

using System.Collections.Generic;

using System.Windows.Forms;

using System.IO;

using System.Text.RegularExpressions;

using System.Windows.Forms.DataVisualization.Charting;

namespace Coursework

{

public partial class frmStat : Form

{

private string afilename;

public frmStat(string filename)

{

this.afilename = filename;

InitializeComponent();

}

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

{

this.comboBox1.SelectedIndex = 0;

this.comboBox2.SelectedIndex = 0;

}

private void btnFuncPos1\_Click(object sender, EventArgs e)

{

Series s1 = new Series();

s1.ChartType = SeriesChartType.Line;

string[] files = Directory.GetFiles(Application.StartupPath + "\\csv");

Dictionary<string, int> visits = new Dictionary<string, int>();

foreach (string filename in files)

{

List<string> authors = new List<string>();

StreamReader sr = new StreamReader(filename);

string temp;

string year;

string[] prisut;

string[] separat1 = { "Слушали" };

string[] separat2 = { "Присутствовали:" };

temp = sr.ReadToEnd();

year = Regex.Match(temp, "\\d\\d\\d\\d года").Value.Split(' ')[0];

temp = temp.Split(separat1, System.StringSplitOptions.RemoveEmptyEntries)[0];

string numer = Regex.Match(temp, "№ \\d").Value.Split(' ')[1];

temp = temp.Split(separat2, System.StringSplitOptions.RemoveEmptyEntries)[1];

prisut = temp.Split(',');

temp = prisut[prisut.Length - 1];

string amount = Regex.Match(temp, "\\d+\\.").Value.Split('.')[0];

Console.WriteLine(numer);

Console.WriteLine(amount);

Console.WriteLine(year);

string semester = Regex.Match(filename, "\\d сем").Value.Split(' ')[0];

if (semester == (this.comboBox2.SelectedIndex+1).ToString())

{

visits[filename] = int.Parse(amount);

s1.Points.AddXY(Regex.Match(filename, "\\d сем").Value + " № " + numer, int.Parse(amount));

}

sr.Close();

}

ChartArea ca1 = new ChartArea();

this.chrtStat.ChartAreas.Clear();

this.chrtStat.ChartAreas.Add(ca1);

this.chrtStat.Series.Clear();

this.chrtStat.Series.Add(s1);

}

private void btnFuncPos2\_Click(object sender, EventArgs e)

{

Series s1 = new Series();

s1.ChartType = SeriesChartType.Line;

string[] files = Directory.GetFiles(Application.StartupPath + "\\csv");

Dictionary<string, int> visits = new Dictionary<string, int>();

foreach (string filename in files)

{

List<string> authors = new List<string>();

StreamReader sr = new StreamReader(filename);

string temp;

string year;

string[] prisut;

string[] separat1 = { "Слушали" };

string[] separat2 = { "Присутствовали:" };

temp = sr.ReadToEnd();

year = Regex.Match(temp, "\\d\\d\\d\\d года").Value.Split(' ')[0];

temp = temp.Split(separat1, System.StringSplitOptions.RemoveEmptyEntries)[0];

string numer = Regex.Match(temp, "№ \\d").Value.Split(' ')[1];

temp = temp.Split(separat2, System.StringSplitOptions.RemoveEmptyEntries)[1];

prisut = temp.Split(',');

temp = prisut[prisut.Length - 1];

string amount = Regex.Match(temp, "\\d+\\.").Value.Split('.')[0];

Console.WriteLine(numer);

Console.WriteLine(amount);

Console.WriteLine(year);

if (year == this.comboBox1.SelectedItem.ToString())

{

visits[filename] = int.Parse(amount);

s1.Points.AddXY(Regex.Match(filename, "\\d сем").Value + " № " + numer, int.Parse(amount));

}

sr.Close();

}

ChartArea ca1 = new ChartArea();

this.chrtStat.ChartAreas.Clear();

this.chrtStat.ChartAreas.Add(ca1);

this.chrtStat.Series.Clear();

this.chrtStat.Series.Add(s1);

}

private void btnFuncPos3\_Click(object sender, EventArgs e)

{

Series s1 = new Series();

s1.ChartType = SeriesChartType.Line;

string[] files = Directory.GetFiles(Application.StartupPath + "\\csv");

Dictionary<string, int> visits = new Dictionary<string, int>();

foreach (string filename in files)

{

List<string> authors = new List<string>();

StreamReader sr = new StreamReader(filename);

string temp;

string year;

string[] prisut;

string[] separat1 = { "Слушали" };

string[] separat2 = { "Присутствовали:" };

temp = sr.ReadToEnd();

year = Regex.Match(temp, "\\d\\d\\d\\d года").Value.Split(' ')[0];

temp = temp.Split(separat1, System.StringSplitOptions.RemoveEmptyEntries)[0];

string numer = Regex.Match(temp, "№ \\d").Value.Split(' ')[1];

temp = temp.Split(separat2, System.StringSplitOptions.RemoveEmptyEntries)[1];

prisut = Regex.Split(temp, "[А-Я].,");

int amount = 0;

foreach (string member in prisut)

{

if (member.Contains("проф") || member.Contains("преп")) amount++;

}

Console.WriteLine(numer);

Console.WriteLine(year);

if (year == this.comboBox1.SelectedItem.ToString())

{

s1.Points.AddXY(Regex.Match(filename, "\\d сем").Value + " № " + numer, amount);

}

sr.Close();

}

ChartArea ca1 = new ChartArea();

this.chrtStat.ChartAreas.Clear();

this.chrtStat.ChartAreas.Add(ca1);

this.chrtStat.Series.Clear();

this.chrtStat.Series.Add(s1);

}

private void btnFuncPos4\_Click(object sender, EventArgs e)

{

Series s1 = new Series();

s1.ChartType = SeriesChartType.Line;

string[] files = Directory.GetFiles(Application.StartupPath + "\\csv");

Dictionary<string, int> visits = new Dictionary<string, int>();

foreach (string filename in files)

{

List<string> authors = new List<string>();

StreamReader sr = new StreamReader(filename);

string temp;

string year;

string[] prisut;

string[] separat1 = { "Слушали" };

string[] separat2 = { "Присутствовали:" };

temp = sr.ReadToEnd();

year = Regex.Match(temp, "\\d\\d\\d\\d года").Value.Split(' ')[0];

temp = temp.Split(separat1, System.StringSplitOptions.RemoveEmptyEntries)[0];

string numer = Regex.Match(temp, "№ \\d").Value.Split(' ')[1];

temp = temp.Split(separat2, System.StringSplitOptions.RemoveEmptyEntries)[1];

prisut = Regex.Split(temp, "[А-Я].,");

int amount = 0;

foreach (string member in prisut)

{

if (member.Contains("проф") || member.Contains("преп")) amount++;

}

Console.WriteLine(numer);

Console.WriteLine(year);

string semester = Regex.Match(filename, "\\d сем").Value.Split(' ')[0];

if (semester == (this.comboBox2.SelectedIndex + 1).ToString())

{

s1.Points.AddXY(Regex.Match(filename, "\\d сем").Value + " № " + numer, amount);

}

sr.Close();

}

ChartArea ca1 = new ChartArea();

this.chrtStat.ChartAreas.Clear();

this.chrtStat.ChartAreas.Add(ca1);

this.chrtStat.Series.Clear();

this.chrtStat.Series.Add(s1);

}

private void btnFuncPos5\_Click(object sender, EventArgs e)

{

string[] files = Directory.GetFiles(Application.StartupPath + "\\csv");

Dictionary<string, int> visits = new Dictionary<string, int>();

foreach (string filename in files)

{

List<string> authors = new List<string>();

StreamReader sr = new StreamReader(filename);

string temp;

string year;

string[] prisut;

string[] separat1 = { "Слушали" };

string[] separat2 = { "Присутствовали:" };

temp = sr.ReadToEnd();

year = Regex.Match(temp, "\\d\\d\\d\\d года").Value.Split(' ')[0];

temp = temp.Split(separat1, System.StringSplitOptions.RemoveEmptyEntries)[0];

string numer = Regex.Match(temp, "№ \\d").Value.Split(' ')[1];

temp = temp.Split(separat2, System.StringSplitOptions.RemoveEmptyEntries)[1];

prisut = Regex.Split(temp, "[А-Я].,");

int amount = 0;

foreach (string member in prisut)

{

Console.WriteLine(member);

string tst = Regex.Match(member, "[А-Я][а-ё]+ [А-Я]\\.").Value;

if (year == this.comboBox1.SelectedItem.ToString() && (member.Contains("проф") || member.Contains("преп") || member.Contains("доц")))

{

if (visits.ContainsKey(tst)) visits[tst]++;

else visits[tst] = 1;

}

}

Console.WriteLine(numer);

Console.WriteLine(year);

sr.Close();

}

this.chrtStat.ChartAreas.Clear();

ChartArea ca1 = new ChartArea();

this.chrtStat.ChartAreas.Add(ca1);

this.chrtStat.Series.Clear();

foreach (KeyValuePair<string, int> member in visits)

{

Series series = this.chrtStat.Series.Add(member.Key);

series.Points.Add(member.Value);

}

}

private void btnFuncPos6\_Click(object sender, EventArgs e)

{

string[] files = Directory.GetFiles(Application.StartupPath + "\\csv");

Dictionary<string, int> visits = new Dictionary<string, int>();

foreach (string filename in files)

{

List<string> authors = new List<string>();

StreamReader sr = new StreamReader(filename);

string temp;

string year;

string[] prisut;

string[] separat1 = { "Слушали" };

string[] separat2 = { "Присутствовали:" };

temp = sr.ReadToEnd();

year = Regex.Match(temp, "\\d\\d\\d\\d года").Value.Split(' ')[0];

temp = temp.Split(separat1, System.StringSplitOptions.RemoveEmptyEntries)[0];

string numer = Regex.Match(temp, "№ \\d").Value.Split(' ')[1];

temp = temp.Split(separat2, System.StringSplitOptions.RemoveEmptyEntries)[1];

prisut = Regex.Split(temp, "[А-Я].,");

int amount = 0;

foreach (string member in prisut)

{

Console.WriteLine(member);

string tst = Regex.Match(member, "[А-Я][а-ё]+ [А-Я]\\.").Value;

string semester = Regex.Match(filename, "\\d сем").Value.Split(' ')[0];

if (semester == (this.comboBox2.SelectedIndex + 1).ToString() && (member.Contains("проф") || member.Contains("преп") || member.Contains("доц")))

{

if (visits.ContainsKey(tst)) visits[tst]++;

else visits[tst] = 1;

}

}

Console.WriteLine(numer);

Console.WriteLine(year);

sr.Close();

}

this.chrtStat.ChartAreas.Clear();

ChartArea ca1 = new ChartArea();

this.chrtStat.ChartAreas.Add(ca1);

this.chrtStat.Series.Clear();

foreach (KeyValuePair<string, int> member in visits)

{

Series series = this.chrtStat.Series.Add(member.Key);

series.Points.Add(member.Value);

}

}

}

}

Сеть Петри для графического анализа:

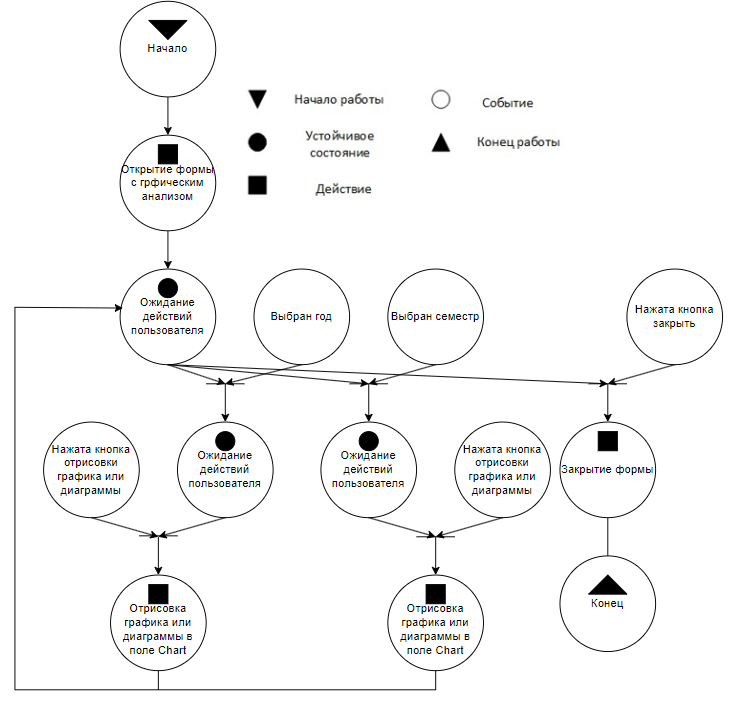


Рисунок 25 – Сеть Петри для графического анализа

Составление схем алгоритмов методов в составе решения, отмеченных на сети Петри в качестве «эффектов» (метка ):

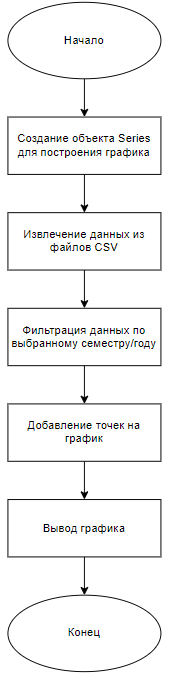
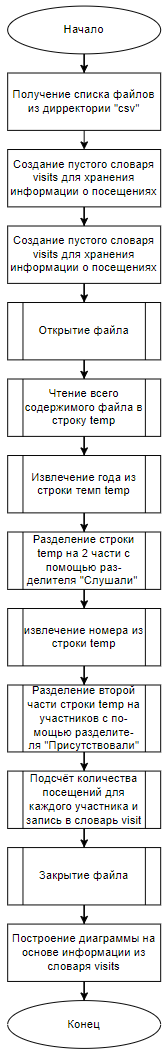
 

Рисунок 26-27 – Схема алгоритма метода построения графиков и диаграмм соответственно

# Инструкция пользователя

Перед открытием программы убедитесь, что в папке с программой имеются: папка WordFiles с печатными формами, папка temp, ExcelFiles и папка csv.

При открытии программы появится интерфейс главной формы:

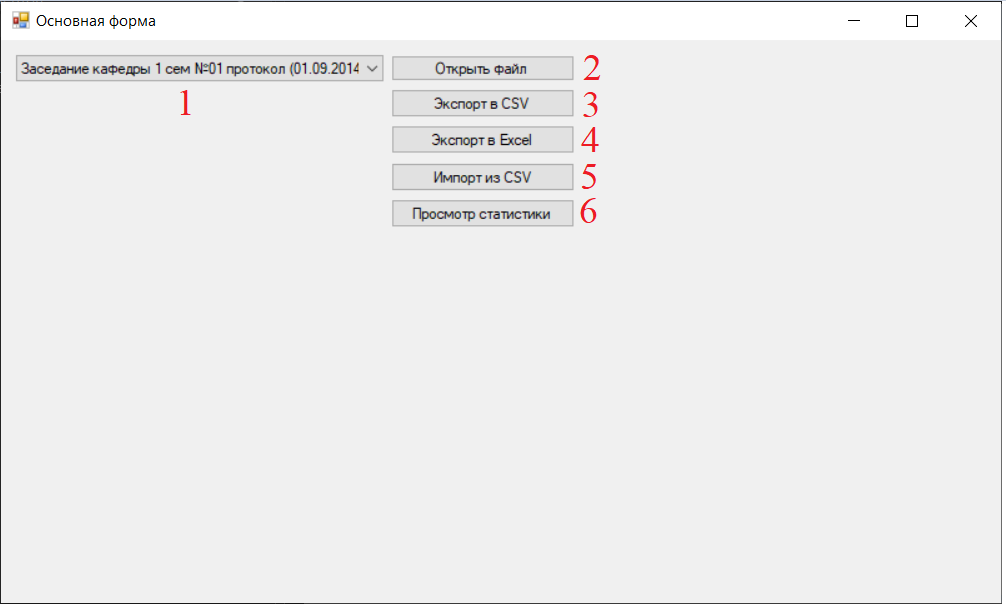


Рисунок 28 – Интерфейс главной формы

На данной форме присутствует 6 основных элементов управления в виде одного ComboBox (1) и пяти кнопок.

Для начала выберите нужный вам файл в ComboBox (1). Затем, нажмите кнопку «Экспорт в CSV» (3). Теперь программа готова к работе.

При нажатии на кнопку «Открыть файл» (2) появится форма с Label, содержащим информацию из файла. Пример представлен на рисунке 2.

При нажатии на кнопку «Экспорт в CSV» (3) информация из печатной формы экспортируется в .csv файл. Данный файл будет помещен в папку csv. Пример представлен на рисунке 5.

При нажатии на кнопку «Экспорт в Excel» (4) информация из печатной формы экспортируется в .xlsx файл. Данный файл будет помещен в папку output. Пример представлен на рисунке 11.

При нажатии на кнопку «Импорт из CSV» (5) информация из CSV файла будет считана и будет воссоздан .docx файл. Данный файл будет помещен в папку output. Пример представлен на рисунке 8.

При нажатии на кнопку «Просмотр статистики», откроется форма с возможностью просмотра различных графиков. Интерфейс данной формы представлен на рисунке 10:

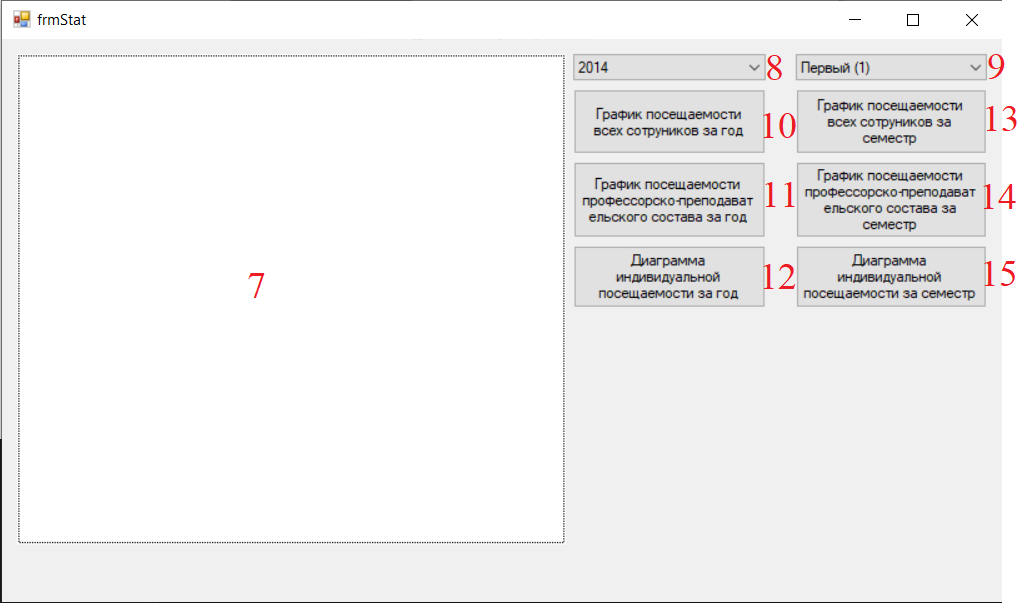


Рисунок 29 – Форма для просмотра статистики

В данной форме присутствует поле для отрисовки графиков Chart (7) и три кнопки управления.

Также на данной форме присутствует элементы управления в виде двух ComboBox (8),(9). В ComboBox (8) пользователю предлагается выбрать год (2014 или 2015), а в ComboBox (9) предлагается выбрать семестр (первый или второй).

При нажатии на кнопку «График посещаемости всех сотрудников за год» (10) в поле для отрисовки графиков Chart (7) отобразится график посещаемости всех сотрудников за выбранный год.

При нажатии на кнопку «График посещаемости профессорско-преподавательского состава за год» (11) в поле для отрисовки графиков Chart (7) отобразится график посещаемости профессорско-преподавательского состава за выбранный год.

При нажатии на кнопку «Диаграмма индивидуальной посещаемости за год» (12) в поле для отрисовки графиков Chart (7) отобразится диаграмма индивидуальной посещаемости за выбранный год.

При нажатии на кнопку «График посещаемости всех сотрудников за семестр» (13) в поле для отрисовки графиков Chart (7) отобразится График посещаемости всех сотрудников за выбранный семестр.

При нажатии на кнопку «График посещаемости профессорско-преподавательского состава за семестр» (14) в поле для отрисовки графиков Chart (7) отобразится график посещаемости профессорско-преподавательского состава за выбранный семестр.

При нажатии на кнопку «Диаграмма индивидуальной посещаемости за семестр» (15) в поле для отрисовки графиков Chart (7) отобразится диаграмма индивидуальной посещаемости за выбранный семестр.

# Заключение

В рамках данной работы были выполнены задачи:

* Анализ печатной формы
* Описание предназначения печатной формы
* Выявление различий в печатных формах
* Унификация печатных форм
* Разработка экранной формы для импорта сведений из печатной формы
* Разработка функций экспорта сведений из печатной формы в файл тексторый формата CSV
* Разработка функций импорта сведений из оригинального и изменённого текстовых файлов формата CSV с воссозданием Word
* Разработка функций экспорта сведений из экранной формы в Excel
* Графический анализ данных, содержащихся в печатной форме
* Создание инструкции пользователя

# Выводы

Язык программирования C# в сочетании с Windows Forms позволяет создать удобное Desktop приложение, для автоматизации работы с любого вида документацией в рамках приложений Microsoft Office. Написание данной программы довольно трудоемкий процесс, однако в дальнейшем написанное ПО значительно сократит время, затрачиваемое на составление документов.