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»

Курсовая работа

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Электронный документооборот»

Выполнил: ст. гр. ТУУ-211

Баранов А.А.

Вариант №16

30.12.2024

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

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

(дата приёмки)

Москва – 2024 г.

# **1. Содержание**

[1. Содержание 2](#_Toc186657505)

[2. Введение 3](#_Toc186657506)

[3. Цель работы 4](#_Toc186657507)

[4. Задачи работы 4](#_Toc186657508)

[5. Содержательная часть работы 6](#_Toc186657509)

[5.1. Анализ печатной формы 6](#_Toc186657510)

[5.2 Описание предназначение печатной формы 6](#_Toc186657511)

[5.3. Разнообразие печатных форм и выявление отличий 7](#_Toc186657512)

[5.4. Унификация печатных форм для удобства обработки данных 7](#_Toc186657513)

[5.5. Разработка экранной формы для импорта сведений из печатной формы 7](#_Toc186657514)

[5.6. Разработка функций экспорта сведений из печатной формы в файл текстовый формата CSV 8](#_Toc186657515)

[5.6.1 Листинг 8](#_Toc186657516)

[5.6.2 Сеть Петри запрограммированного технологического процесса 11](#_Toc186657517)

[5.6.3 Блок-схема алгоритма 12](#_Toc186657518)

[5.7 Разработка функций импорта сведений из оригинального и изменённого текстовых файлов формата CSV с воссозданием Word 14](#_Toc186657519)

[5.7.1 Листинг 14](#_Toc186657520)

[5.7.2 Сеть Петри запрограммированного технологического процесса 18](#_Toc186657521)

[5.7.3 Блок-схема алгоритма 19](#_Toc186657522)

[5.8. Разработка функций экспорта сведений из экранной формы в Excel 23](#_Toc186657523)

[5.8.1 Листинг 23](#_Toc186657524)

[5.8.2 Сеть Петри запрограммированного технологического процесса 27](#_Toc186657525)

[5.8.3 Блок-схема алгоритма 28](#_Toc186657526)

[5.9. Графический анализ данных, содержащихся в печатной форме 29](#_Toc186657527)

[5.9.1 Пример работы графического анализа данных, содержащихся в печатных формах 31](#_Toc186657528)

[6. Инструкция пользователя 35](#_Toc186657529)

[7. Заключение 38](#_Toc186657530)

[8. Выводы 38](#_Toc186657531)

[9. Библиографический список 39](#_Toc186657532)

2. Введение

Электронный документооборот (ЭДО) — это система, которая позволяет создавать, обрабатывать, хранить и анализировать документы в цифровом формате. Такой подход помогает значительно ускорить процессы обработки данных и автоматизировать рутинные задачи [1]. Внедрение ЭДО способствует сокращению затрат на бумагу, печать и хранение физических документов, что делает его особенно актуальным для современных организаций, стремящихся повысить свою эффективность и перейти на цифровые технологии.

Существуют различные виды ЭДО, такие как бухгалтерский, управленческий, кадровый, складской и архивный. Каждый из них ориентирован на специфические задачи, будь то обработка финансовых документов, управление кадровыми делами или хранение архивов [2]. Разработанное в рамках данной работы программное обеспечение ЭДО предоставляет возможность конвертировать данные печатных форм в различные форматы, такие как *CSV*, *DOCX* и *XLS*. Это позволяет работать с документами в привычных текстовых и табличных редакторах и обеспечивает их легкую интеграцию в рабочие процессы. Помимо конвертации, программа предлагает инструменты графического анализа данных, что позволяет визуализировать информацию, содержащуюся в документах.

Внедрение такого программного обеспечения автоматизирует обработку данных, устраняет ошибки, связанные с ручным вводом, и сокращает время, затрачиваемое на выполнение задач. Оно упрощает анализ данных и предоставляет удобные механизмы для взаимодействия с документами, повышая общую производительность труда. Благодаря этим возможностям ЭДО помогает организациям адаптироваться к современным требованиям цифровой трансформации [3].

# **3. Цель работы**

В среде *Microsoft* *Visual Studio* на языке *Visual C#* создать программное обеспечение с графическим пользовательским интерфейсом, исполняющее функцию электронного документооборота. Разработать эргономичную экранную форму программного обеспечения, разработать функцию экспорта сведений из печатной формы в файл текстового формата *CSV*, разработать функцию импорта сведений из текстового файла формата *CSV* с воссозданием в *Word*, разработать функцию экспорта сведений из экранной формы в *Excel*, разработать функцию графического анализа данных. Освоить навыки работы с *Word* и *Excel* программным способом, а также освоить графический анализ данных.

# **4. Задачи работы**

1. Проанализировать печатные формы

2. Уметь переносить содержимое печатных форм в *CSV*

3. Уметь воссоздавать структуру печатной формы

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

5. Уметь конвертировать печатную форму *Word* – *Excel*

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

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

Индивидуальное задание:

1. Показывать столбчатую диаграмму количества докладов за всё время только тех авторов, которые указали в названии слово, вводимое в текстовое поле.

2. Показывать столбчатую диаграмму количества докладов за указанный период только тех авторов, которые указали в названии слово, вводимое в текстовое поле.

3. Показывать столбчатую диаграмму распределения докладов по секциям за всё время.

4. Показывать столбчатую диаграмму распределения заявок по секциям за всё время.

5. Показывать столбчатую диаграмму распределения докладов по секциям за указанный период.

6. Показывать столбчатую диаграмму распределения заявок по секциям за указанный период.

7. Показывать столбчатую диаграмму количества уникальных авторов докладов в указанном году.

8. Показывать столбчатую диаграмму количества докладов только от не уникальных авторов за весь период.

9. Показывать столбчатую диаграмму количества докладов, составленных в соавторстве, приходящихся на каждый календарный год всего периода.

10. Показывать столбчатую диаграмму количества заявок за всё время, приходящихся на различные должности авторов-докладчиков.

11. Показывать столбчатую диаграмму количества заявок, поданных студентами за указанный календарный год, с распределением их по учебным группам.

12. Показывать столбчатую диаграмму количества заявок, приходящихся на каждый календарный год всего периода, в которых наименования доклада и статьи в точности совпадают.

13. Показывать столбчатую диаграмму количества докладов, завленных за весь рассматриваемый период от коллективов из 2-х, 3-х, 4-х, 5-ти соавторов.

14. Показывать столбчатую диаграмму количества заявок, приходящихся на конференцию за весь рассматриваемый период.

5. Содержательная часть работы

5.1. Анализ печатной формы

Каждая печатная форма представляет собой *Word*-документ, который содержит в себе заявки на выступление с докладом на научно-практической конференции. На один доклад может приходиться несколько заявок от разных выступающих с одинаковым докладом. Все печатные формы имеют в заголовке элементы, которые могут изменяться, в зависимости от документа. Эти элементы: название научно-практической конференции и дата её проведения. После заголовка в каждой печатной форме располагается таблица, имеющая фиксированную структуру – 10 строк и 2 колонки. В таблице содержатся сведения о докладчике, докладе и проведении выступления, которые меняются, в зависимости от документа.

5.2 Описание предназначение печатной формы

Каждая печатная форма – заявка на выступление с докладом в научно-практической конференции. На один доклад могут подать заявку разные люди. Информация, содержащаяся в печатной форме призвана для регистрации докладчика на научно-практической конференции.

5.3. Разнообразие печатных форм и выявление отличий

Печатные формы могут различаться между собой только наименование научно-практической конференции, датой ее проведения, а также данными о докладчике и докладе во второй колонке таблицы.

5.4. Унификация печатных форм для удобства обработки данных

Унификация печатных форм для удобства обработки данных не требуется.

5.5. Разработка экранной формы для импорта сведений из печатной формы

На Рисунке 1 показано … .

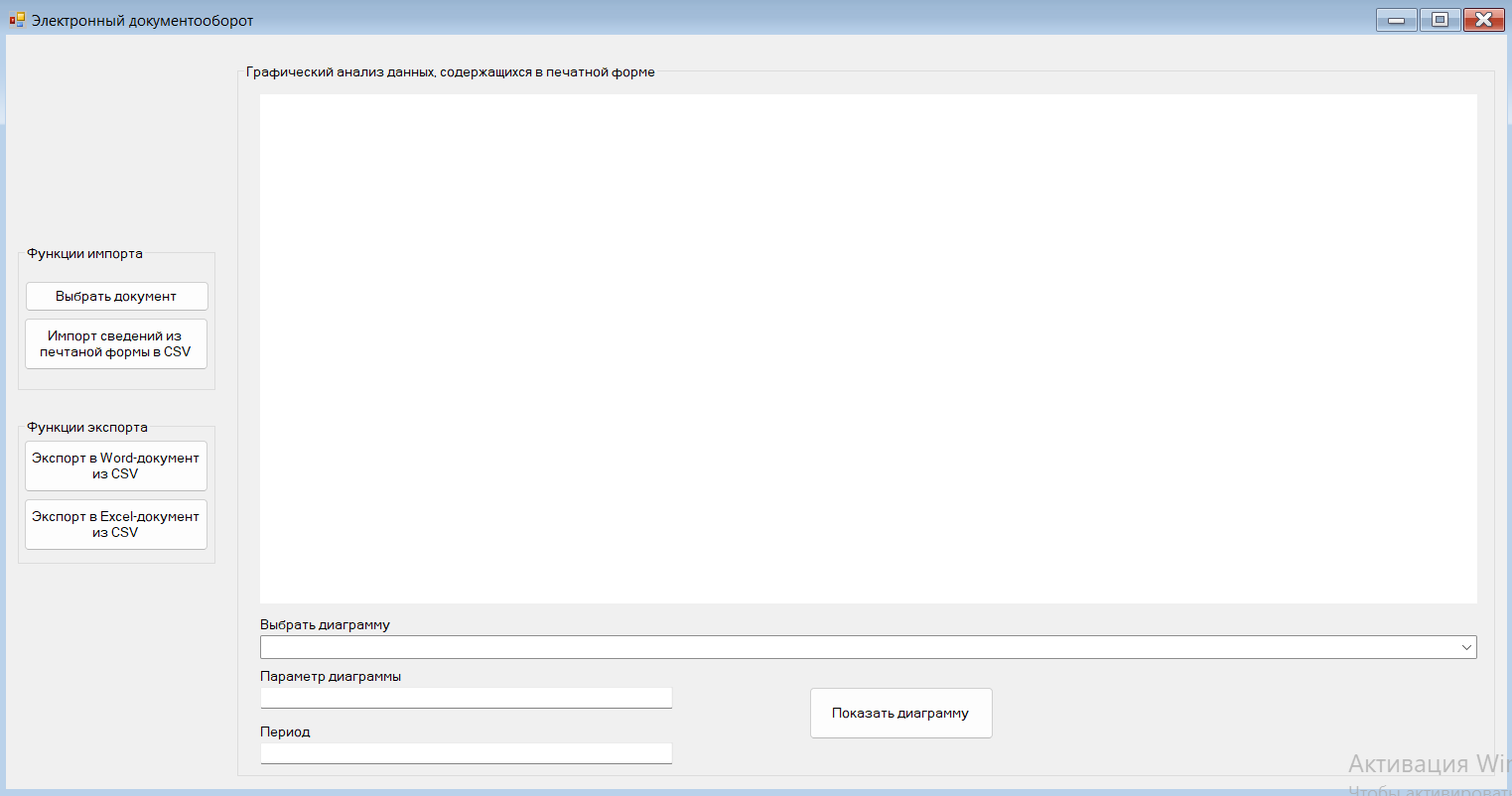


Рисунок 1 – Экранная форма для импорта сведений из печатной формы

Экранная форма для импорта сведений из печатной формы содержит в себе 2 элемента *GroupBox,*, содержащие в себе функции, предназначенные для конвертации данных в разные типы: *GroupBox* c название «Функции импорта», который группирует кнопку для выбора импортируемой в *CSV* печатной формы и кнопку, запускающую процесс импорта сведений из выбранной печатной формы в *CSV*, *GroupBox* с названием «Функции экспорта», который группирует кнопку, запускающую процесс экспорта данных в *Word*-документ из *CSV* и кнопку, запускающую процесс экспорта данных в *Excel*-документ из *CSV*. Также на экранной форме располагается *GroupBox* с названием «Графический анализ данных, содержащихся в экранной форме», который группирует элементы, предназначенные для выбора диаграммы и настройки диаграммы. Эти элементы: элемент *Chart*, предназначен для визуализации диаграмм, элемент *ComboBox*, отмеченный на экранной форме при помощи элемента *Label*, как «Выбрать диаграмму», предназначен для выбора диаграммы, элемент *TextBox*, отмеченный на экранной форме при помощи элемента *Label*, как «Параметр диаграммы», предназначен для ввода параметра диаграммы, элемент *TextBox*, отмеченный на экранной форме при помощи элемента *Label*, как «Период», предназначен для ввода периода, элемент *Button* с названием «Показать диаграмму», предназначен для инициализации отображения диаграммы на экранном элементе *Chart*. В зависимости от выбранной диаграммы, текстовые поля могут менять своё состояние, чтобы не позволить пользователю вписать данные, если они не требуются.

5.6. Разработка функций экспорта сведений из печатной формы в файл текстовый формата *CSV*

5.6.1 Листинг

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

{

if (fileName == null)

{

MessageBox.Show("Выберите файл!");

}

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

oWord.Visible = false;

Word.Document oDoc = oWord.Documents.Open(fileName, ReadOnly:true);

StreamWriter writer = new StreamWriter(pathToCSV, append: false, Encoding.UTF8);

Word.Table table = oDoc.Tables[1];

foreach (Word.Paragraph paragraph in oDoc.Paragraphs)

{

if (paragraph.Range.Tables.Count > 0)

{

for (int i = 1; i <= table.Rows.Count; i++)

{

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

for (int j = 1; j <= table.Columns.Count; j++)

{

string cellText = table.Cell(i, j).Range.Text

.Replace("\r", "")

.Replace("\n", "")

.TrimEnd('\r', '\a')

.Trim();

rowData.Add(cellText.Trim());

}

writer.WriteLine(string.Join(";", rowData));

}

break;

}

else

{

string paragraphText = paragraph.Range.Text.Trim();

writer.WriteLine(paragraphText);

}

}

writer.Close();

MessageBox.Show("Импорт в CSV выполнен");

oDoc.Close();

}

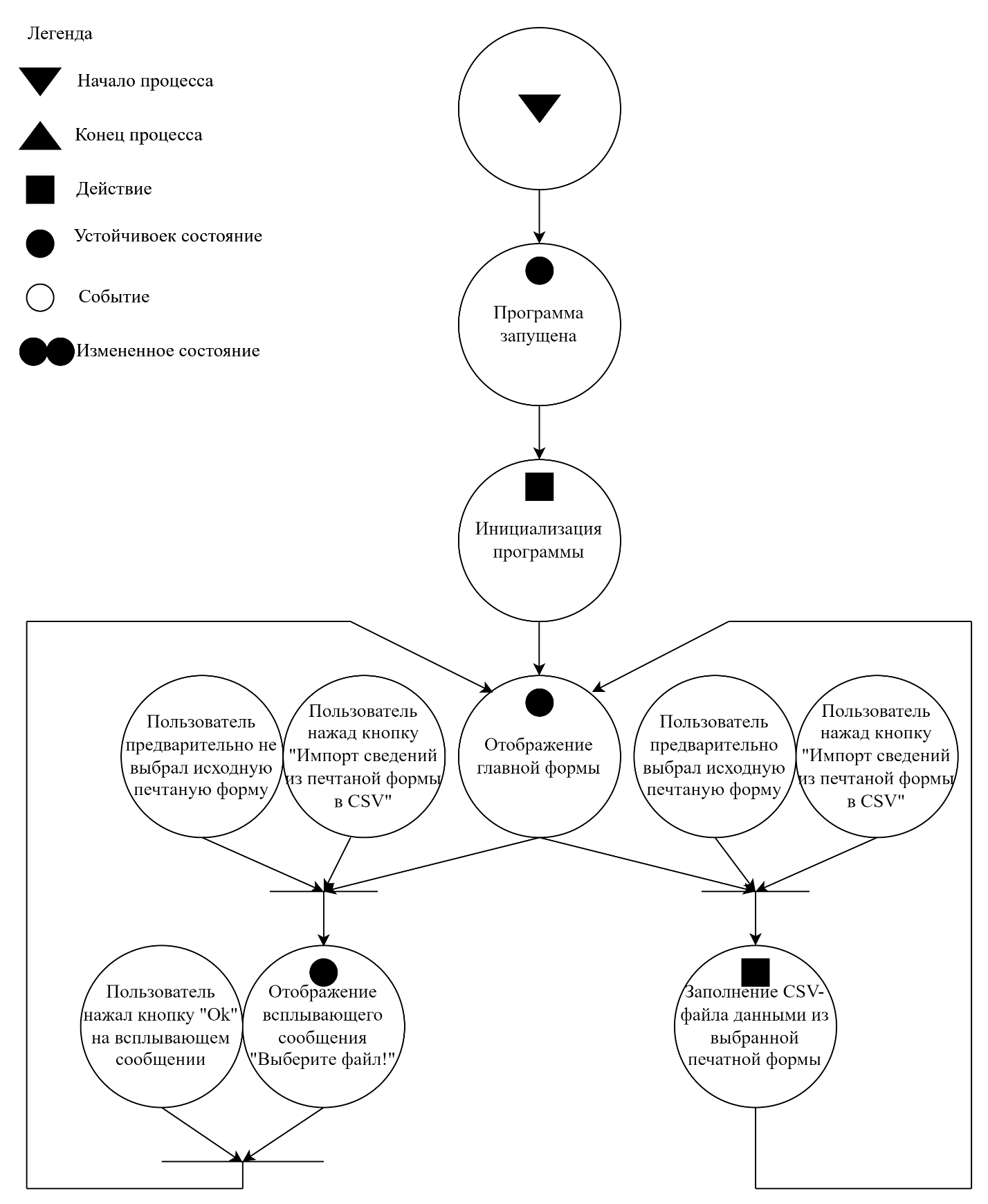
5.6.2 Сеть Петри запрограммированного технологического процесса

Рисунок 2 – Сеть Петри запрограммированного технологического процесса

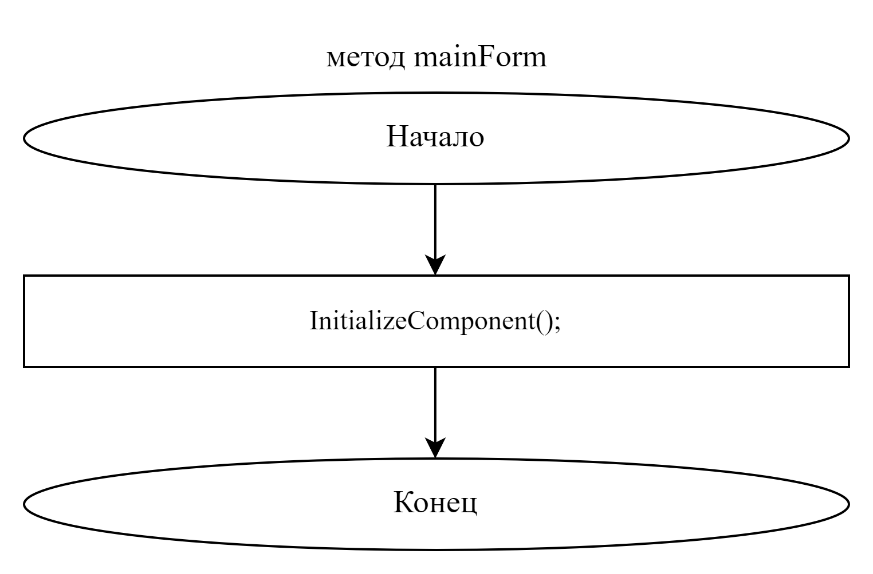
5.6.3 Блок-схема алгоритма

Рисунок 3 – Блок-схема метода *mainForm*

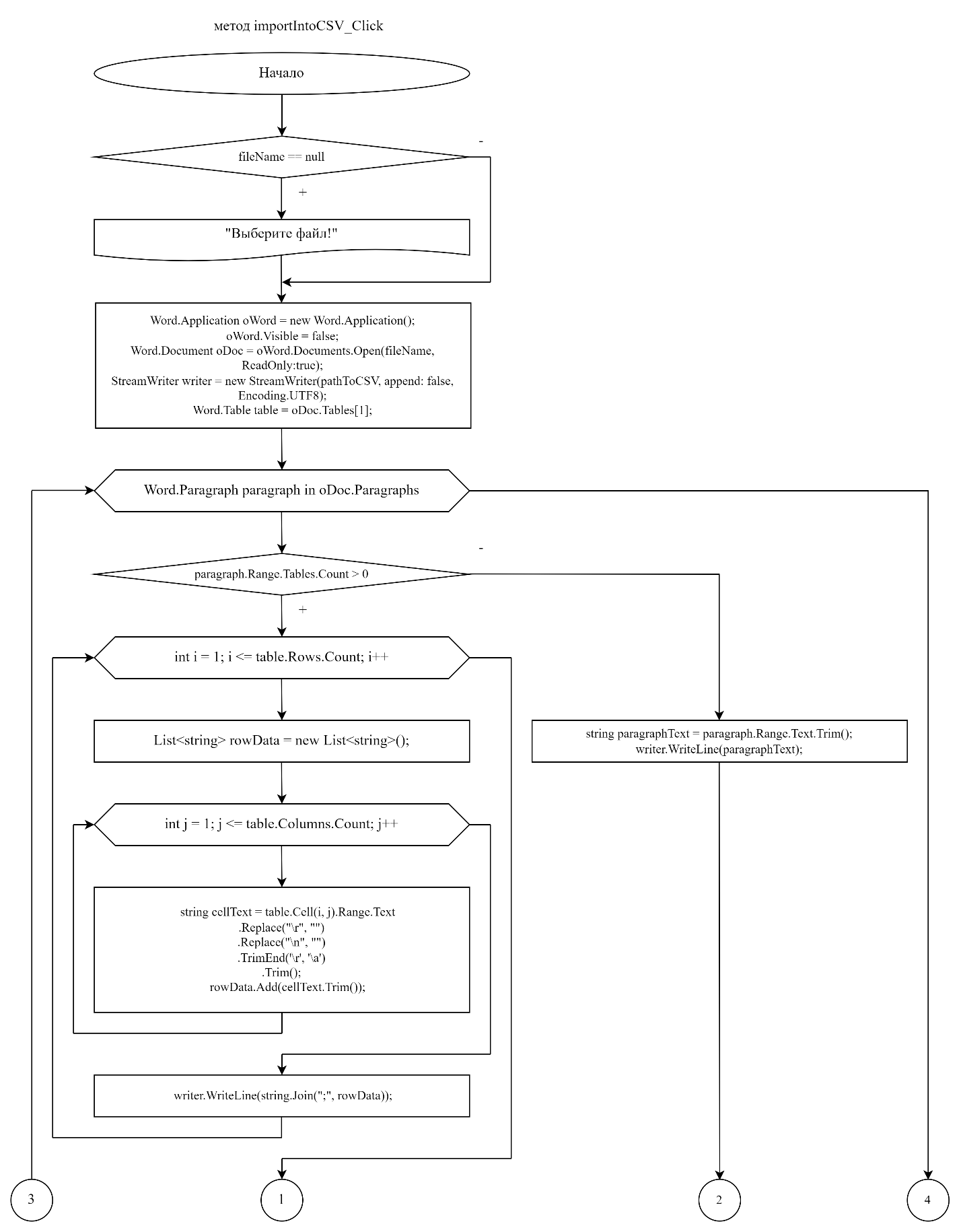


Рисунок 4 – Блок-схема метода *importIntoCSV\_Click*

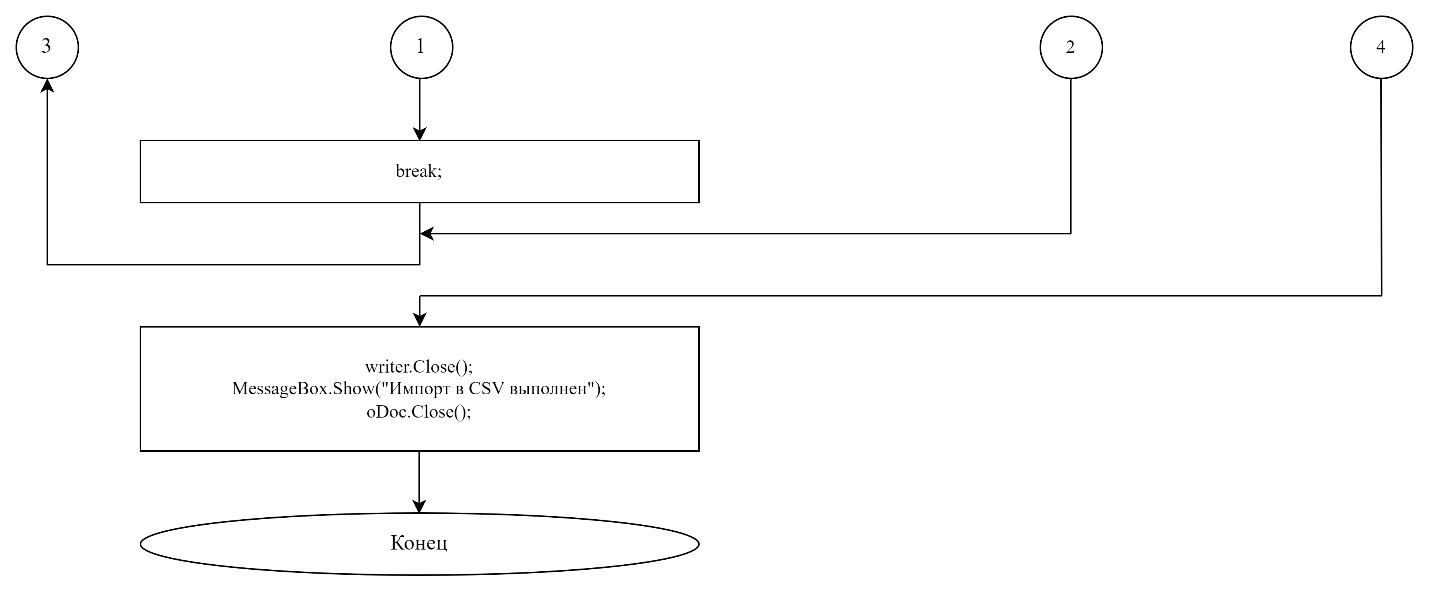


Рисунок 5 – Блок-схема метода *importIntoCSV\_Click*

5.7 Разработка функций импорта сведений из оригинального и изменённого текстовых файлов формата *CSV* с воссозданием *Word*

5.7.1 Листинг

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

{

Word.Document oDoc;

Word.Paragraph oPr;

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

oDoc = oWord.Documents.Add();

oPr = oDoc.Paragraphs.Add();

Word.Range range = oDoc.Content;

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

int index = 0;

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

{

SetParagraphText(oPr, lines[i], 14, "Times new roman", "Center", 1);

oPr.Range.ParagraphFormat.LineSpacingRule = Word.WdLineSpacing.wdLineSpace1pt5;

if (lines[i] == "")

{

oPr.Range.ParagraphFormat.LineSpacingRule = Word.WdLineSpacing.wdLineSpaceSingle;

break;

}

}

Word.Paragraph emptyParagraph = oDoc.Paragraphs.Add();

emptyParagraph.Range.Text = "";

Word.Table table = oDoc.Tables.Add(emptyParagraph.Range, 10, 2);

int idexOfEmptyString = Array.IndexOf(lines, "");

string[] tableData = lines.Skip(idexOfEmptyString + 1).ToArray();

for(int i = 0; i < 10; i++)

{

string[] row = tableData[i].Split(';');

for(int j = 1; j <= 2; j++)

{

table.Cell(i + 1, j).Range.Text = row[j - 1];

if(j == 1)

{

table.Cell(i + 1, j).Range.Font.Size = 14;

table.Cell(i + 1, j).Range.Font.Bold = 0;

table.Cell(i + 1, j).VerticalAlignment = Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter;

table.Cell(i + 1, j).Range.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;

}

else

{

table.Cell(i + 1, j).Range.Font.Size = 12;

table.Cell(i + 1, j).Range.Font.Bold = 1;

table.Cell(i + 1, j).VerticalAlignment = Word.WdCellVerticalAlignment.wdCellAlignVerticalTop;

table.Cell(i + 1, j).Range.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;

}

}

}

foreach(Word.Cell cell in table.Range.Cells)

{

cell.Borders.Enable = 1;

}

oDoc.SaveAs2(System.Windows.Forms.Application.StartupPath + "\\Импортированный Word-документ.docx");

oWord.Quit();

MessageBox.Show("Импорт в Word-документ из CSV выполнен");

}

5.7.2 Сеть Петри запрограммированного технологического процесса

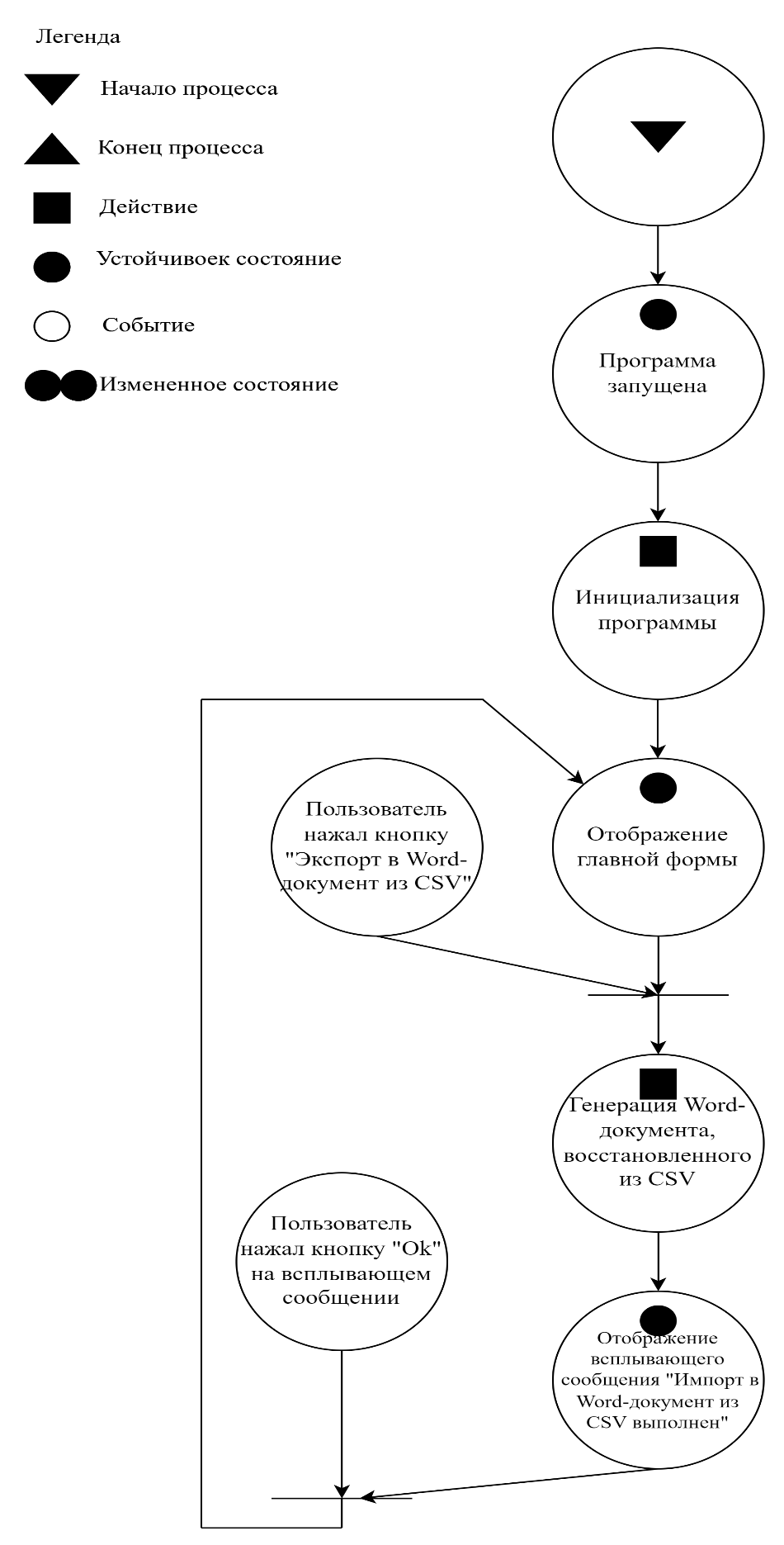


Рисунок 6 – Сеть Петри запрограммированного технологического процесса

5.7.3 Блок-схема алгоритма

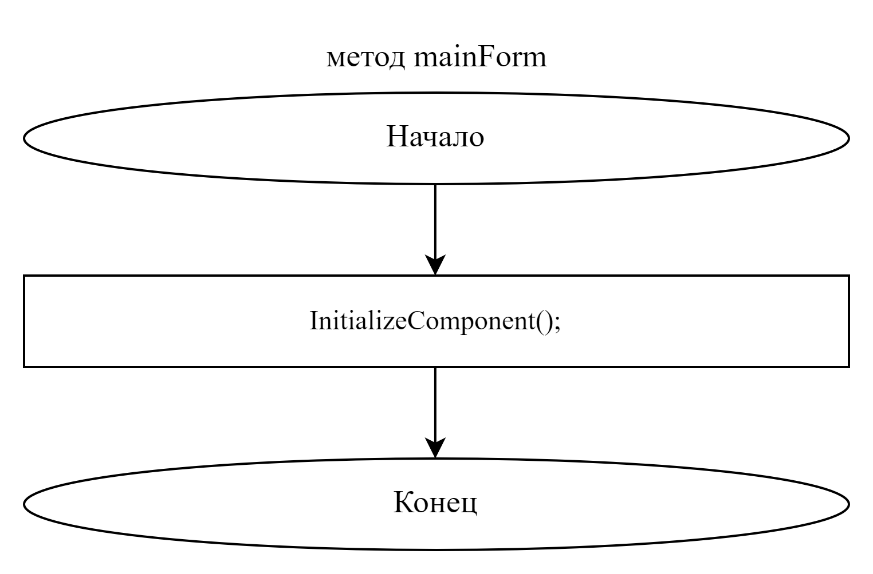


Рисунок 7 – Блок-схема метода *mainForm*



Рисунок 8 – Блок-схема метода *exportCSVToWord\_Click*

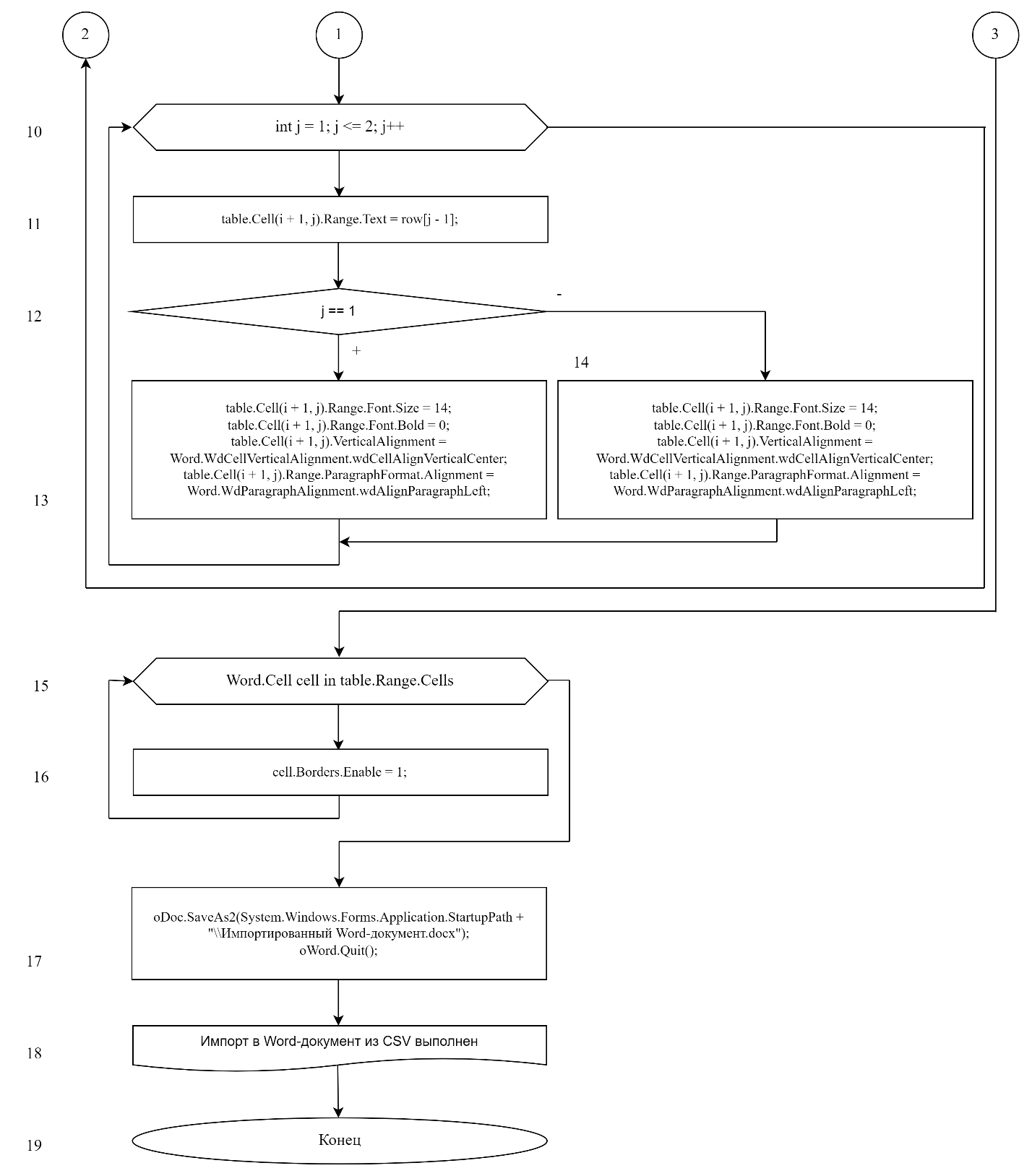


Рисунок 9 – Блок-схема метода *exportCSVToWord\_Click*

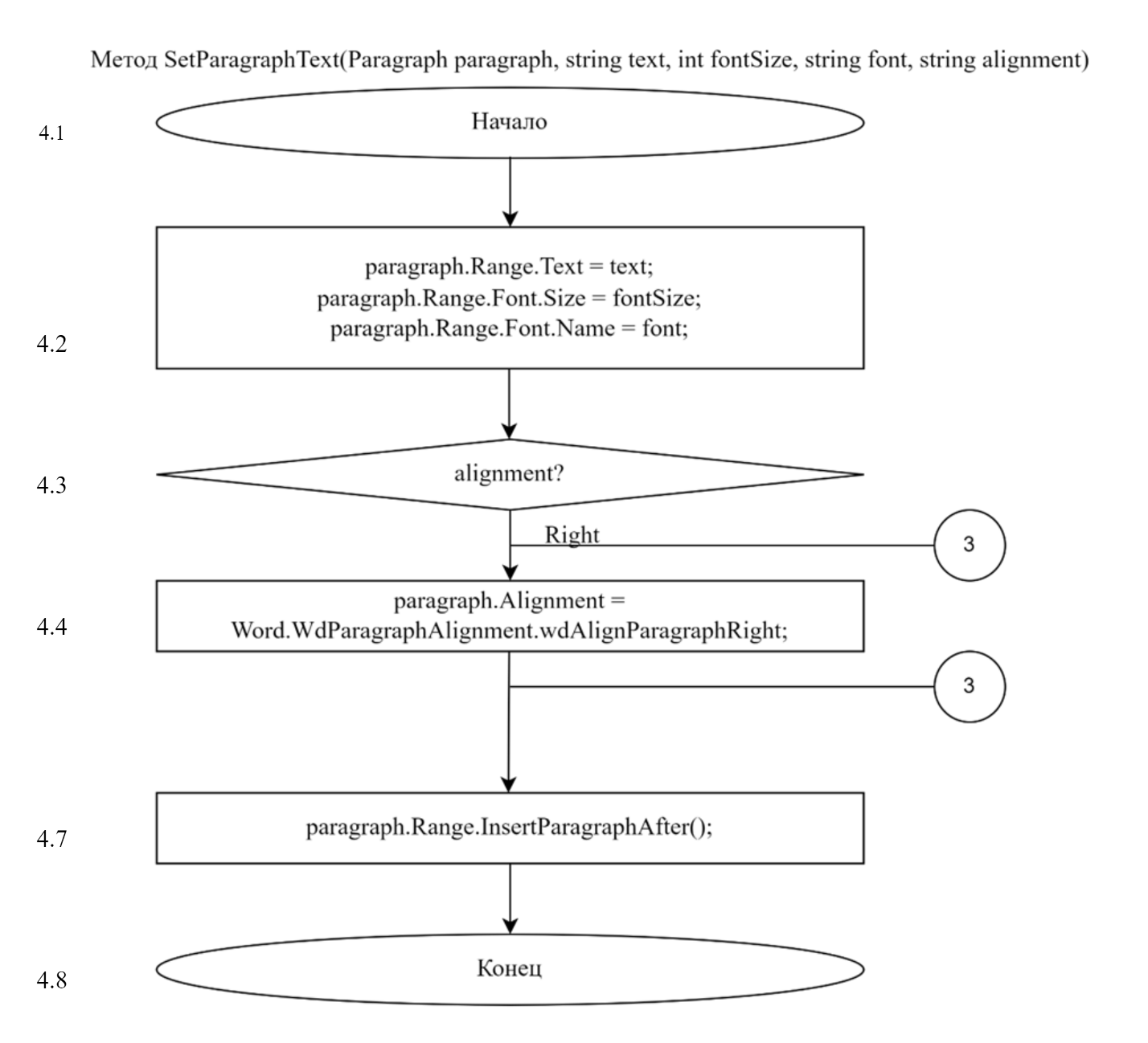


Рисунок 10 – Блок-схема метода *SetParagraphText*

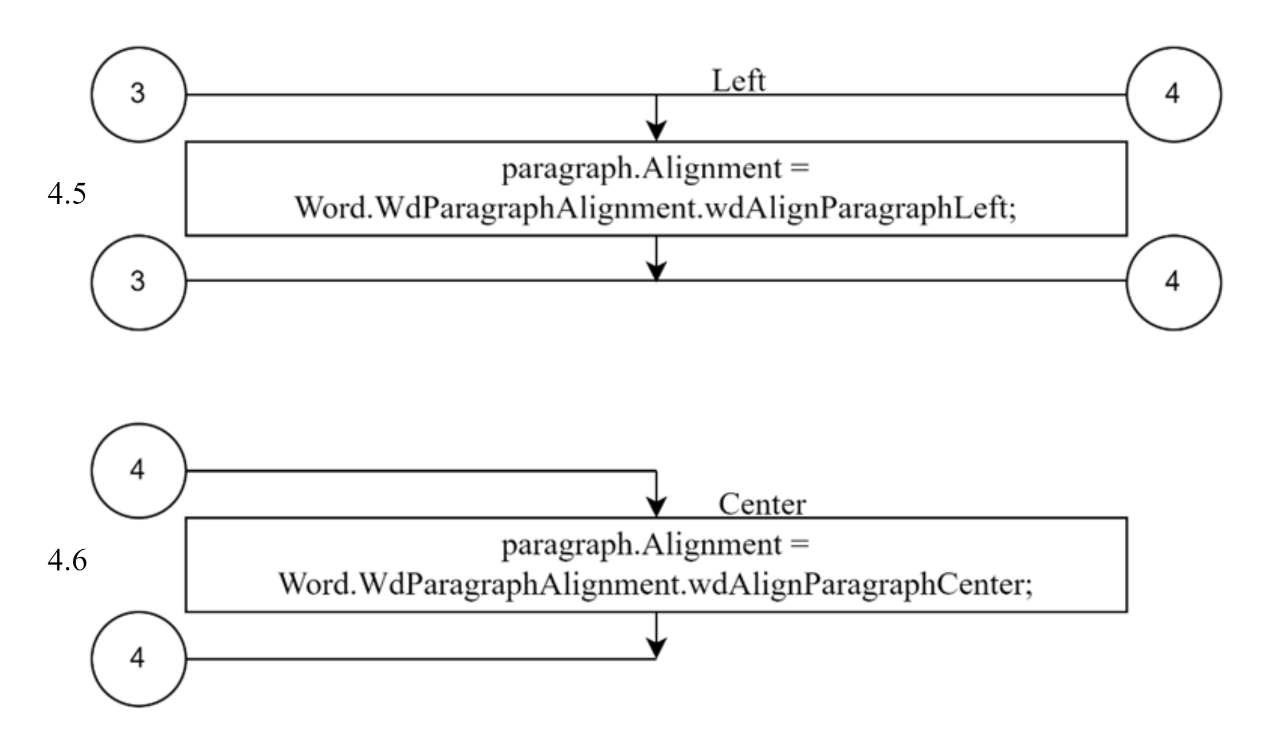


Рисунок 11 – Блок-схема метода *SetParagraphText*

5.8. Разработка функций экспорта сведений из экранной формы в *Excel*

5.8.1 Листинг

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

{

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

app.Visible = false;

Excel.Workbook wb = app.Workbooks.Add(Missing.Value);

Excel.Worksheet ws = (Excel.Worksheet)wb.Sheets.Add();

ws.Activate();

ws.Range[ExcelMethods.ExcelCellTranslator(1, 1), ExcelMethods.ExcelCellTranslator(2, 2)].Merge();

ws.Rows[2].RowHeight = 75;

ws.Columns[1].ColumnWidth = 35;

ws.Columns[2].ColumnWidth = 35;

for(int i = 2; i <= 12; i++)

{

ws.Rows[i].RowHeight = 90;

}

ws.Range[ExcelMethods.ExcelCellTranslator(3, 1), ExcelMethods.ExcelCellTranslator(12, 2)].Borders.LineStyle = Excel.XlLineStyle.xlContinuous;

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

string[] header = lines.Take(Array.IndexOf(lines, "")).ToArray();

ws.Cells[1, 1] = string.Join("\n", header);

int idexOfEmptyString = Array.IndexOf(lines, "");

string[] tableData = lines.Skip(idexOfEmptyString + 1).ToArray();

for (int i = 3 ; i <= 12; i++)

{

string[] row = tableData[i - 3].Split(';');

for (int j = 1; j <= 2; j++)

{

ws.Cells[i, j] = row[j - 1];

ws.Cells[i, j].WrapText = true;

if(j == 1)

{

ws.Cells[i, j].Font.Name = "Times new roman";

ws.Cells[i, j].Font.Size = 14;

ws.Cells[i, j].Font.Bold = 0;

ws.Cells[i, j].HorizontalAlignment = Excel.XlHAlign.xlHAlignLeft;

ws.Cells[i, j].VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

}

else

{

ws.Cells[i, j].Font.Name = "Times new roman";

ws.Cells[i, j].Font.Size = 12;

ws.Cells[i, j].Font.Bold = 1;

ws.Cells[i, j].HorizontalAlignment = Excel.XlHAlign.xlHAlignLeft;

ws.Cells[i, j].VerticalAlignment = Excel.XlVAlign.xlVAlignTop;

}

}

}

ws.Cells[1, 1].Font.Name = "Times new roman";

ws.Cells[1, 1].Font.Size = 14;

ws.Cells[1, 1].Font.Bold = 1;

ws.Cells[1, 1].HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

ws.Cells[1, 1].VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

app.UserControl = true;

wb.SaveCopyAs(System.Windows.Forms.Application.StartupPath + "\\Импортированный Excel-документ.xls");

wb.Close(false);

MessageBox.Show("Импорт в Excel-документ из CSV выполнен");

}

5.8.2 Сеть Петри запрограммированного технологического процесса

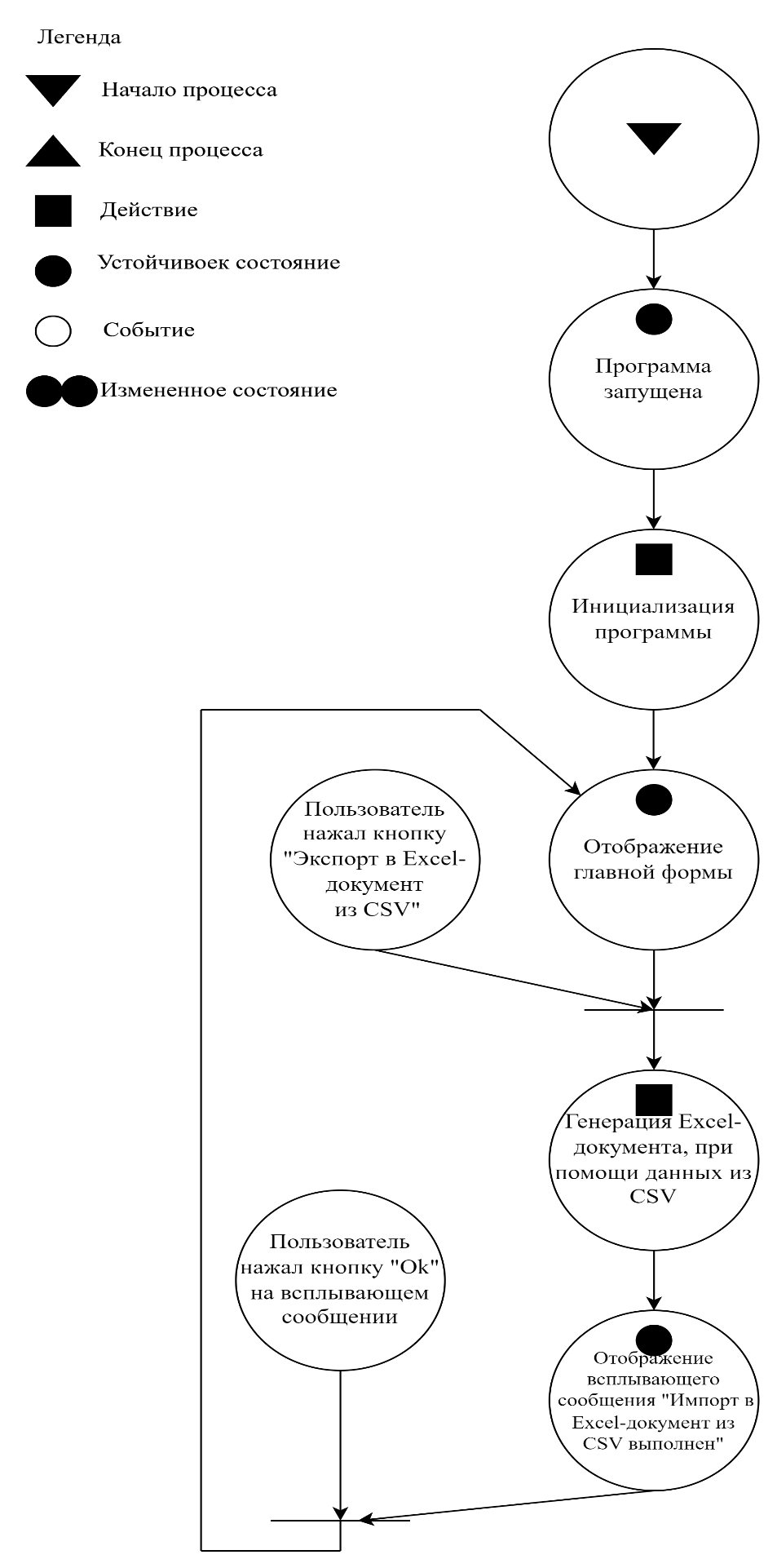


Рисунок 12 – Сеть Петри запрограммированного технологического процесса

5.8.3 Блок-схема алгоритма



Рисунок 13 – Блок-схема метода *exportCSVToExcel\_Click*

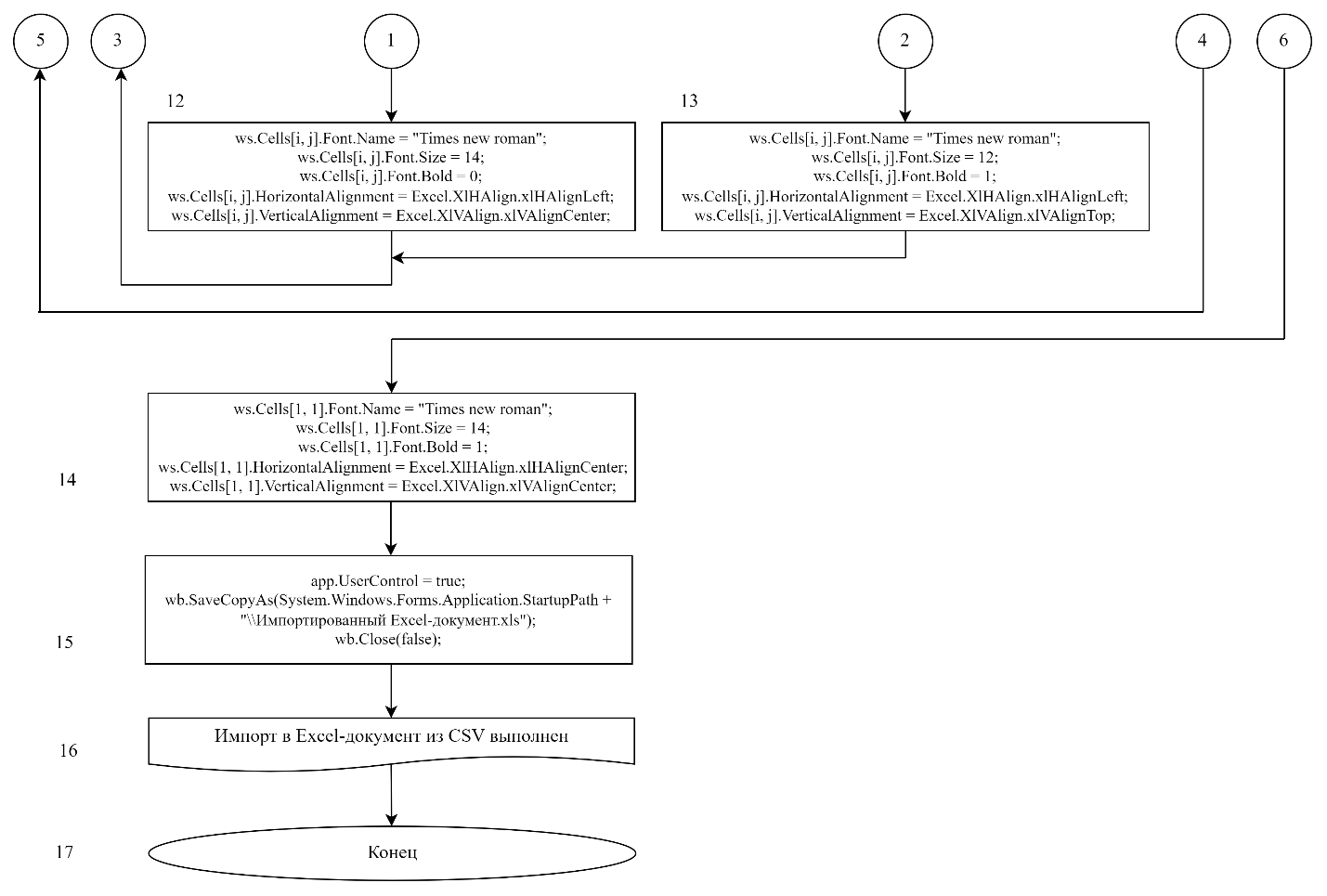


Рисунок 14 – Блок-схема метода *exportCSVToExcel\_Click*

5.9. Графический анализ данных, содержащихся в печатной форме

Режимы диаграмм, отображающихся на элементе экранной формы *Chart*, выбираются при помощи выпадающего списка (*ComboBox*), режимы диаграмм реализованы согласно рабочему заданию:

1. Показывать столбчатую диаграмму количества докладов за всё время только тех авторов, которые указали в названии слово, вводимое в текстовое поле.

2. Показывать столбчатую диаграмму количества докладов за указанный период только тех авторов, которые указали в названии слово, вводимое в текстовое поле.

3. Показывать столбчатую диаграмму распределения докладов по секциям за всё время.

4. Показывать столбчатую диаграмму распределения заявок по секциям за всё время.

5. Показывать столбчатую диаграмму распределения докладов по секциям за указанный период.

6. Показывать столбчатую диаграмму распределения заявок по секциям за указанный период.

7. Показывать столбчатую диаграмму количества уникальных авторов докладов в указанном году.

8. Показывать столбчатую диаграмму количества докладов только от не уникальных авторов за весь период.

9. Показывать столбчатую диаграмму количества докладов, составленных в соавторстве, приходящихся на каждый календарный год всего периода.

10. Показывать столбчатую диаграмму количества заявок за всё время, приходящихся на различные должности авторов-докладчиков.

11. Показывать столбчатую диаграмму количества заявок, поданных студентами за указанный календарный год, с распределением их по учебным группам.

12. Показывать столбчатую диаграмму количества заявок, приходящихся на каждый календарный год всего периода, в которых наименования доклада и статьи в точности совпадают.

13. Показывать столбчатую диаграмму количества докладов, завленных за весь рассматриваемый период от коллективов из 2-х, 3-х, 4-х, 5-ти соавторов.

14. Показывать столбчатую диаграмму количества заявок, приходящихся на конференцию за весь рассматриваемый период.

На экранной форме располагается элемент *TextBox*, обозначенный элементом *Label*, как «Параметр диаграммы» - он призван для получения дополнительных данных при построении диаграммы, если того требует режим работы диаграммы. Если режим работы диаграммы не требует дополнительных данных, то в данное текстовое поле невозможно внести никакие данные.

На экранной форме располагается еще один элемент *TextBox*, обозначенный элементом *Label*, как «Период» - он призван для получения данных о периоде, если того требует режим работы диаграммы. Если режим работы диаграммы не требует периода, то в данное текстовое поле невозможно внести никакие данные.

На экранной форме располагается элемент *Button*, который, если заполнены нужные текстовые поля, если того требует режим диаграммы, инициализирует построение диаграммы.

Данные диаграммы берутся из печатных форм, посредством циклической итерации по каждому документу.

Каждый режим работы диаграммы реализован в отдельном не типизированном методе, которые активируются по событию «*Click*» кнопки «Показать диаграмму», при помощи условной конструкции «*switch-case*».

5.9.1 Пример работы графического анализа данных, содержащихся в печатных формах

Для примера работы графического анализа данных, содержащихся в печатной форме выбрано 3 режима работы диаграмм:

1. Показывать столбчатую диаграмму количества докладов за всё время только тех авторов, которые указали в названии слово, вводимое в текстовое поле.

6. Показывать столбчатую диаграмму распределения заявок по секциям за указанный период.

14. Показывать столбчатую диаграмму количества заявок, приходящихся на конференцию за весь рассматриваемый период.

Пример работы режима диаграммы «1. Показать столбчатую диаграмму количества докладов за всё время только тех авторов, которые указали в названии слово» на Рисунке 15:

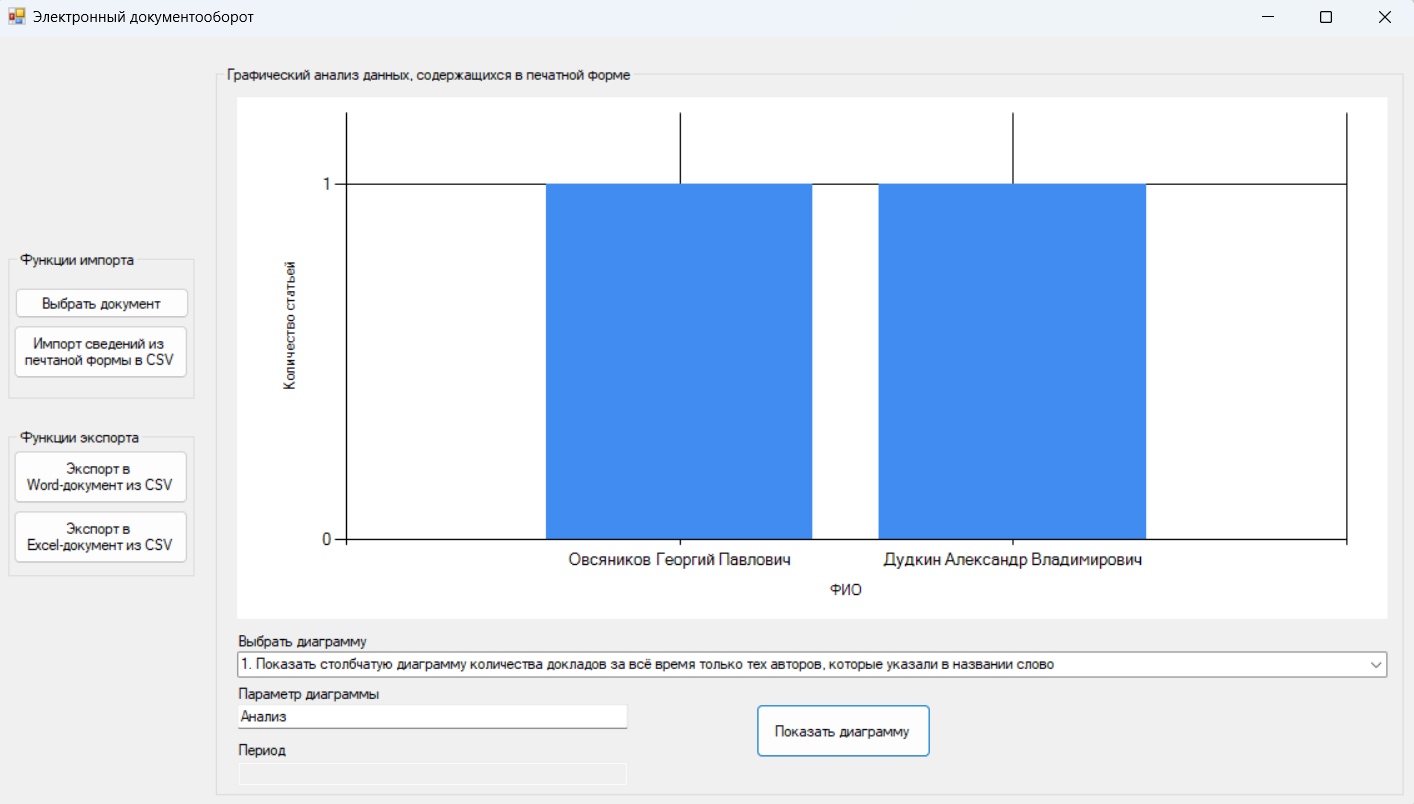


Рисунок 15 – Пример работы первого режима диаграммы

При ручном подсчёте было найдено два автора, имеющих по одному докладу, в названии которого было указано слово «Анализ».

Пример работы режима диаграммы «6. Показать столбчатую диаграмму распределения заявок по секциям за указанный период» на Рисунке 16:

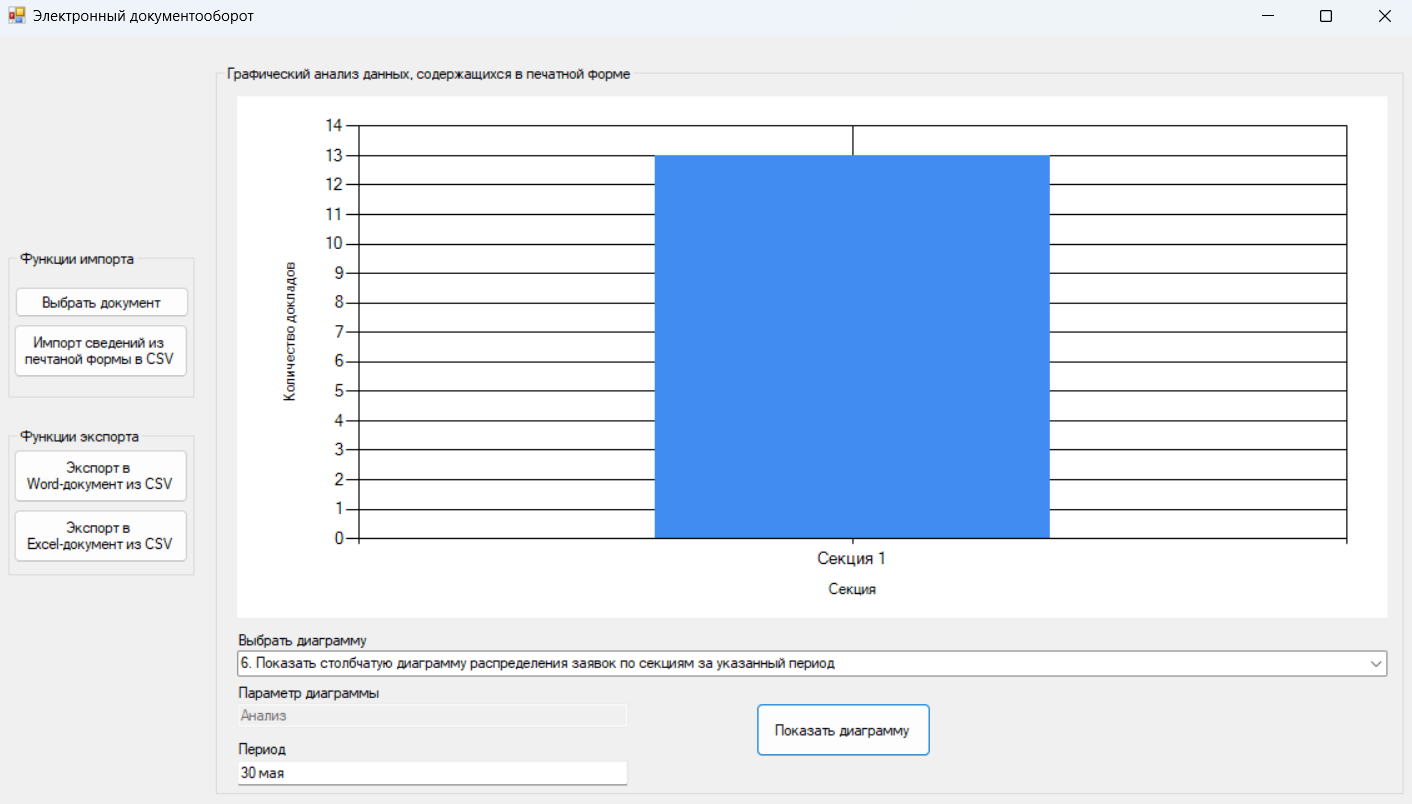


Рисунок 16 – Пример работы шестого режима диаграммы

При ручном подсчете было найдено 13 заявок на выступление с докладом, которые приходятся на 30 мая, все они попадают в первую секцию.

Пример работы режима диаграммы «14. Показать столбчатую диаграмму количества заявок, приходящихся на конференцию за весь рассматриваемый период» на Рисунке 17:

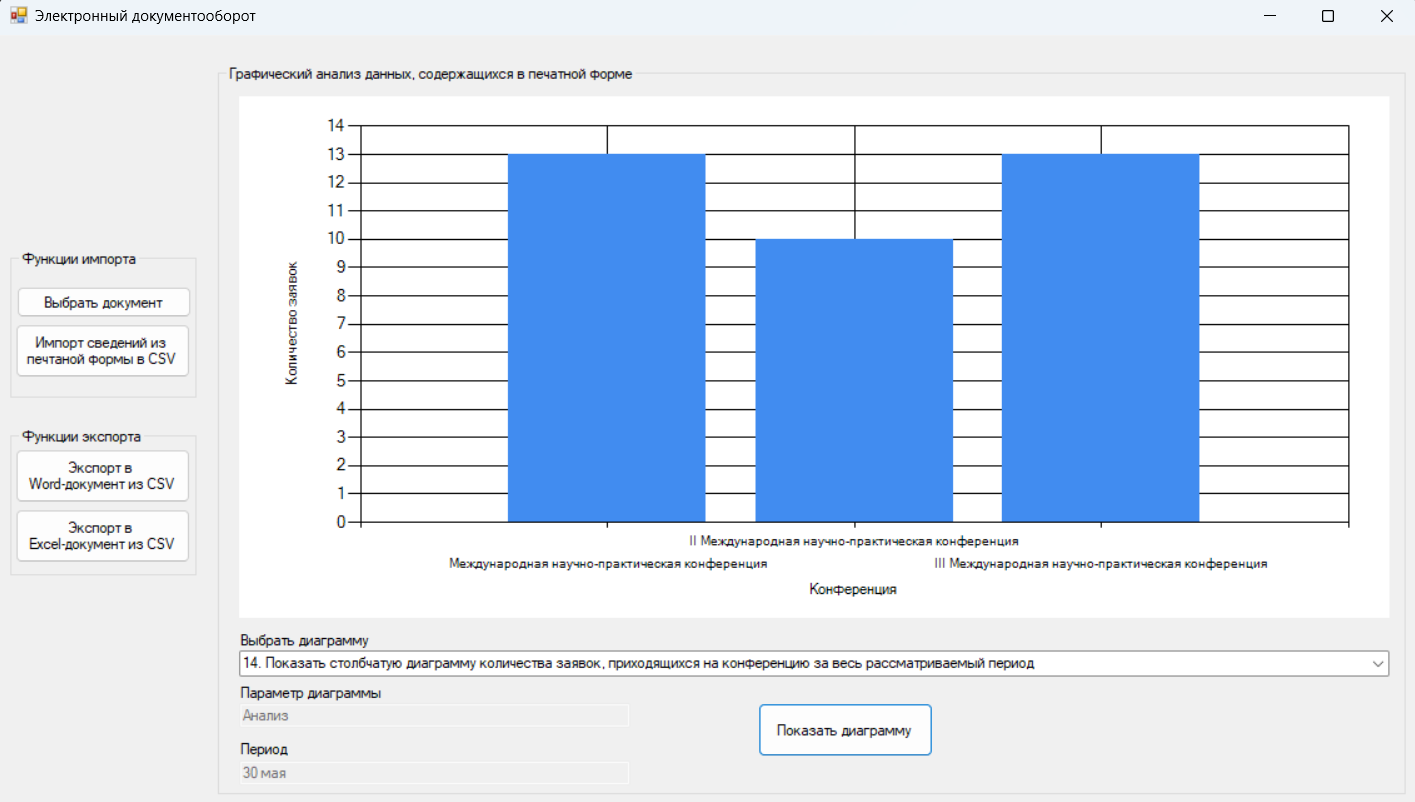


Рисунок 17 – Пример работы четырнадцатого режима диаграммы

При ручном подсчете было найдено 13 заявок на Международную научно-практическую конференцию, 10 заявок на II Международную научно-практическую конференцию и 13 заявок на III Международную конференцию.

6. Инструкция пользователя

Интерфейс главного окна программы на Рисунке 18:

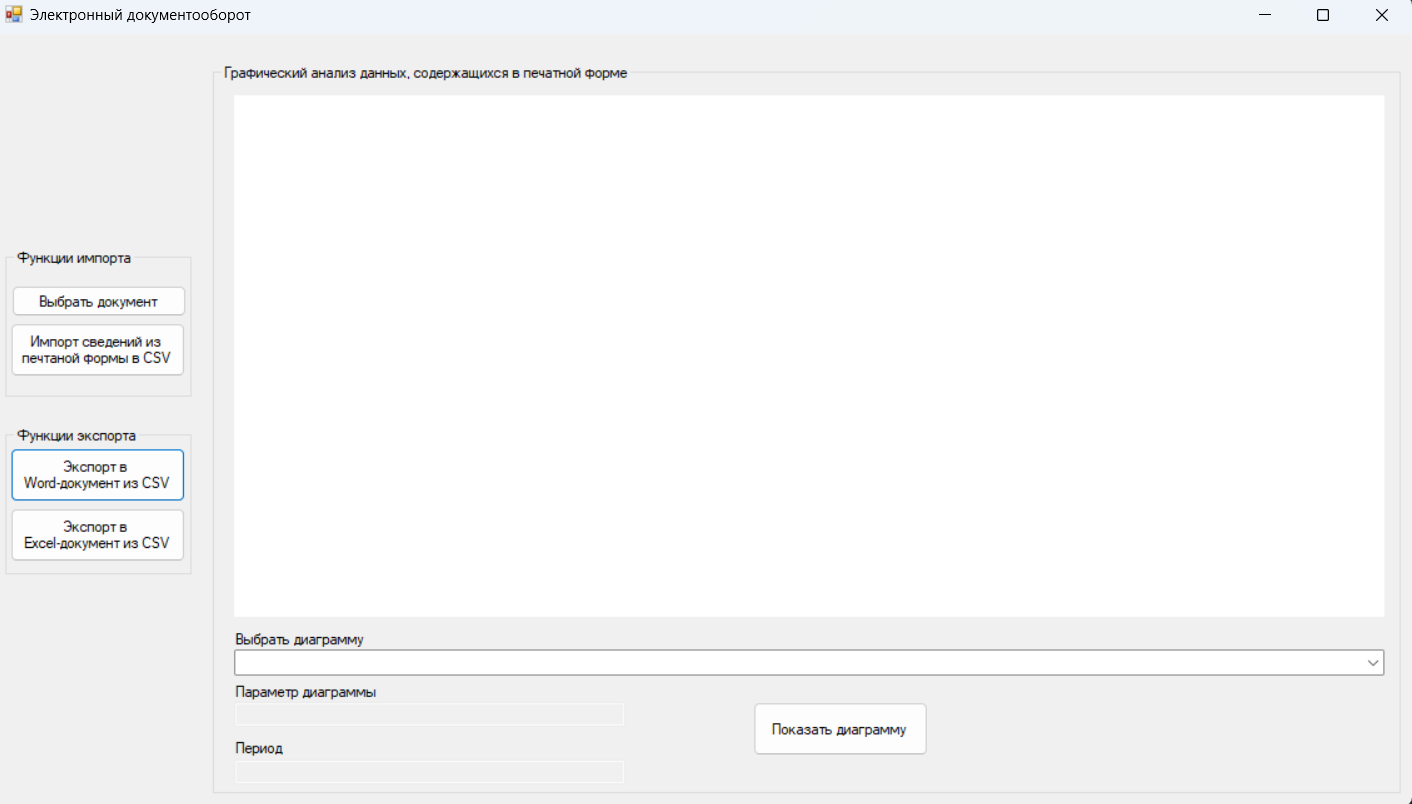


Рисунок 18 – Интерфейс главного окна программы

Группы элементов экранной формы, отвечающих за выбор и генерацию документов:

1. Функции импорта:
   1. Кнопка «Выбрать документ»: позволяет выбрать печатную форму из файловой системы для импорта данных в *CSV*
   2. Кнопка «Импорт сведений из печатной формы в *CSV*»: инициализирует процесс импорта сведений из выбранного файла в *CSV*, если файл не будет предварительно выбран, то будет показано всплывающее сообщение, свидетельствующее об этом (рисунок 19):

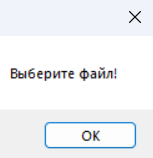


Рисунок 19 – Всплывающее сообщение «Выберите файл»

1. Функции экспорта:
   1. Кнопка «Экспорт в *Word*-документ из *CSV*»: инициализирует процесс генерации *Word*-документа, основываясь на данных из *CSV­*-файла. По завершению этого процесса появится всплывающее сообщение о выполнении процесса экспорта (Рисунок 20):

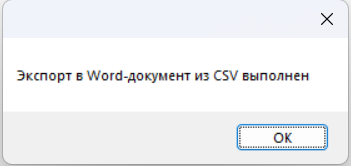


Рисунок 20 – Всплывающее сообщение «Экспорт в *Word*-документ из *CSV* выполнен»

* 1. Кнопка «Экспорт в *Excel*-документ из *CSV*»: инициализирует процесс генерации *Excel*-документа, основываясь на данных из *CSV­*-файла. По завершению этого процесса появится всплывающее сообщение о выполнении процесса экспорта (Рисунок 21):

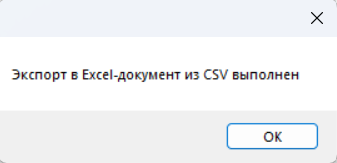


Рисунок 21 – Всплывающее сообщение «Экспорт в *Excel*-документ из *CSV* выполнен»

Группа элементов экранной формы, отвечающих за графический анализ данных, содержащихся в печатной форме:

1. Выпадающий список «Выбрать диаграмму»: позволяет выбрать режим отображаемой диаграммы.
2. Текстовое поле «Параметр диаграммы»: позволяет ввести параметр диаграммы, если выбранный режим того требует. Если параметр диаграммы не был введен, то появится всплывающее сообщение об этом (Рисунок 22):

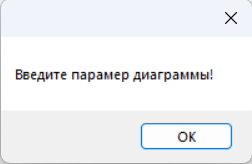


Рисунок 22 – Всплывающее сообщение «Введите параметр диаграммы!»

1. Текстовое поле «Период»: позволяет ввести период, если выбранный режим того требует. Если период не был введен, то появится всплывающее сообщение об этом (Рисунок 23):

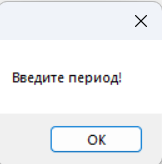


Рисунок 23 – Всплывающее сообщение «период!»

1. Кнопка «Показать диаграмму»: инициализирует отображение диаграммы на экранной форме
2. Элемент экранной формы «*Chart*»: отображает выбранную диаграмму.

7. Заключение

В ходе работы была разработана программа электронного документооборота, которая включает в себя следующие функции: конвертация данных в различные форматы (*docx, xls, CSV*), графический анализ данных, содержащихся в печатной форме, эффективная обработка данных из печатных форм (конвертация и графический анализ).

Разработанное программное обеспечение электронного документооборота автоматизирует работу с набором электронных документов, упрощает их обработку и анализ, а также экономит время пользователя, освобождая его от рутинных задача, что повышает его эффективность при работе с электронными документами.

8. Выводы

В процессе выполнения работы был разработан эргономичный пользовательский интерфейс, который позволяет пользователю легко ориентироваться в элементах и функциях программного обеспечения. В процессе выполнения работы был освоен навык работы с пространством имен *Microsoft.Office.Interop*, использование его для работы с *Excel* и *Word* программным путём (генерация новых документов с использованием данных из *CSV*-файлов). Также был освоен навык работы с *Chart*-элементом экранной формы (настройка осей диаграммы и серий данных диаграммы). Закреплен навык чтения и записи в файлы, а также улучшен навык написания алгоритмов.

9. Библиографический список

1. Электронный документооборот: понятие, виды, плюсы и минусы [Электронный источник]: scloud.ru *URL: https://scloud.ru/blog/elektronnyy\_dokumentooborot\_ponyatie\_vidy\_plyusy\_i\_minusy/* (дата обращения: 30.12.2024).

2. Виды электронного документооборота [Электронный источник]: *nadpo.ru URL: https://nadpo.ru/academy/blog/vidy-i-sut-elektronnogo-dokumentooborota/* (дата обращения: 30.12.2024).

3. Что смогут компании с системой ЭДО? [Электронный источник]: *diadoc.ru URL: https://www.diadoc.ru/articles/20586-plyusy\_i\_minusy\_elektronnogo\_dokumentooborota?utm\_source=chatgpt.com#header\_20586\_3* (дата обращения: 30.12.2024).