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 № 18

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Автоматизация Excel»

Выполнил: ст. гр. ТУУ-211

Грачева Н.С.

Вариант №5

Проверил: к.т.н., доц. Сафронов А.И.

1. **Формулировка цели работы**

Освоить навыки подключения внешних модулей и их использования в разрабатываемых программах; освоить навыки программируемой настройки параметров электронных таблиц Microsoft Office Excel.

1. **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.

1. Продумать и создать эргономичный графический пользовательский интерфейс с кнопками «Предпросмотр» и «Выгрузка в Excel».

2. Нажатие на кнопку инициирует запуск на исполнение метода составления и заполнения электронной таблицы заданной по варианту структуры.

1. **Сеть Петри запрограммированного технологического процесса**

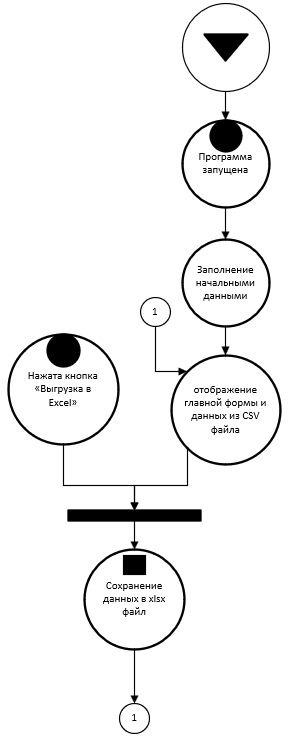
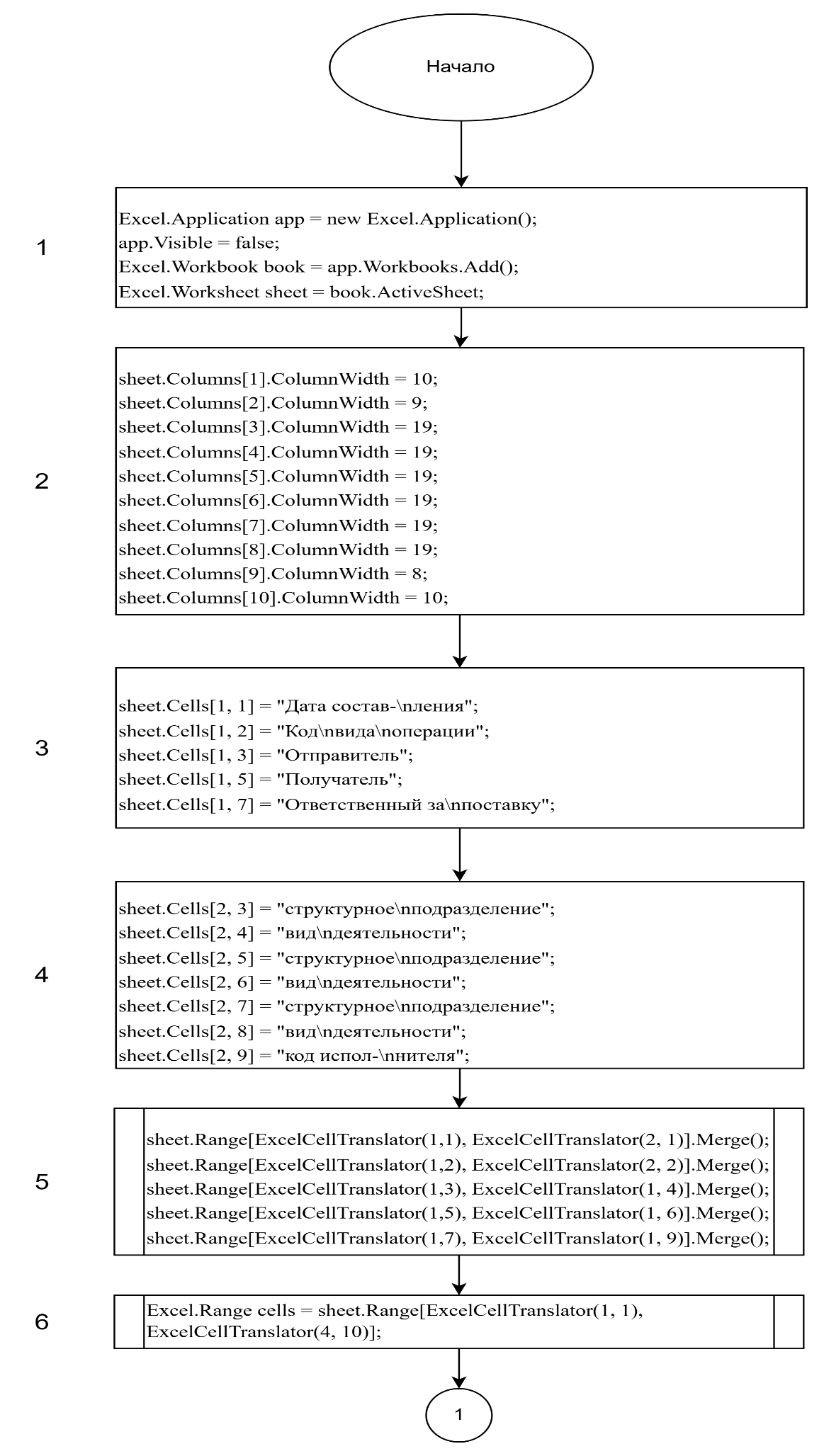
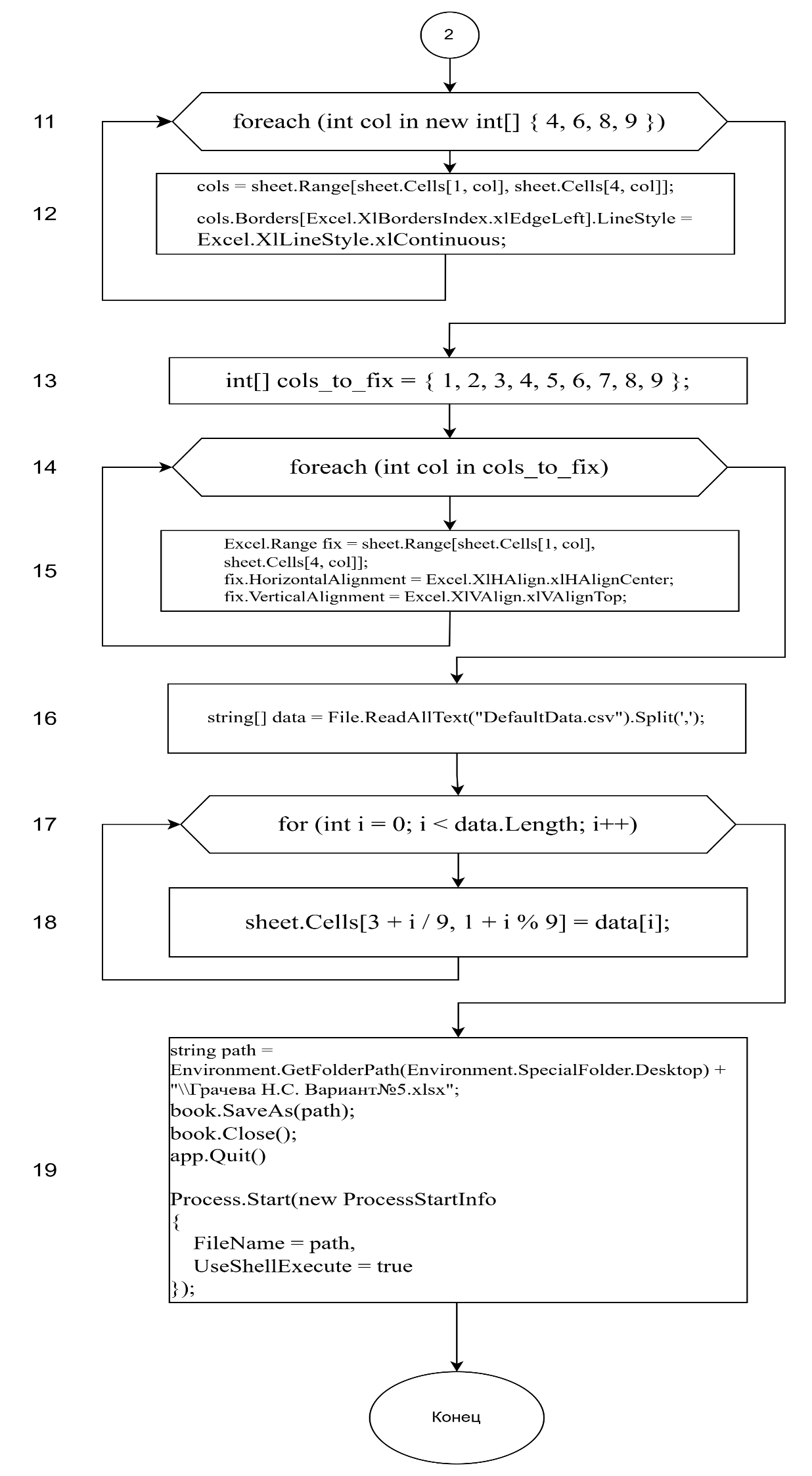


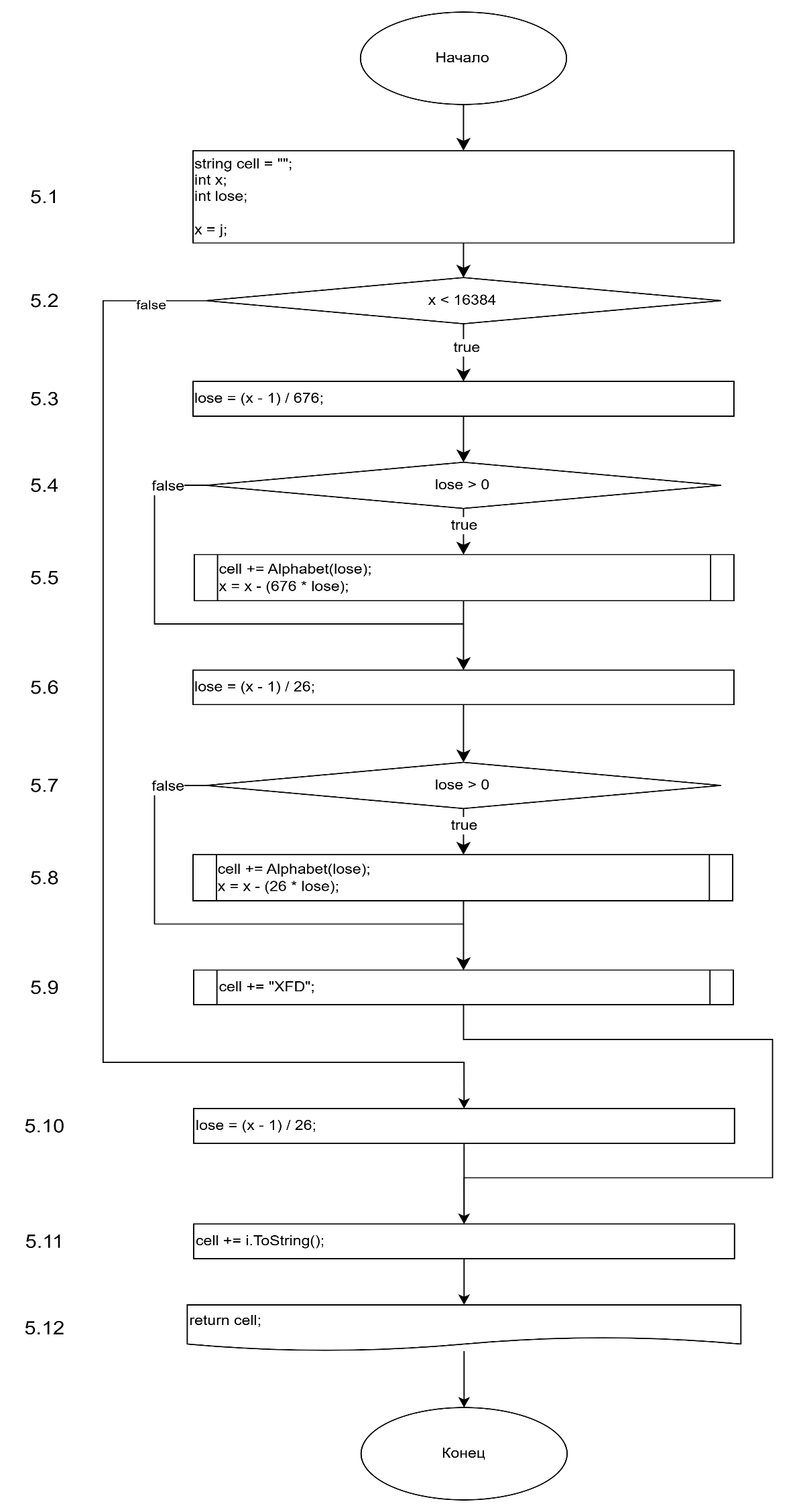
Рисунок 1 – Сеть Петри запрограммированного технологического процесса

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









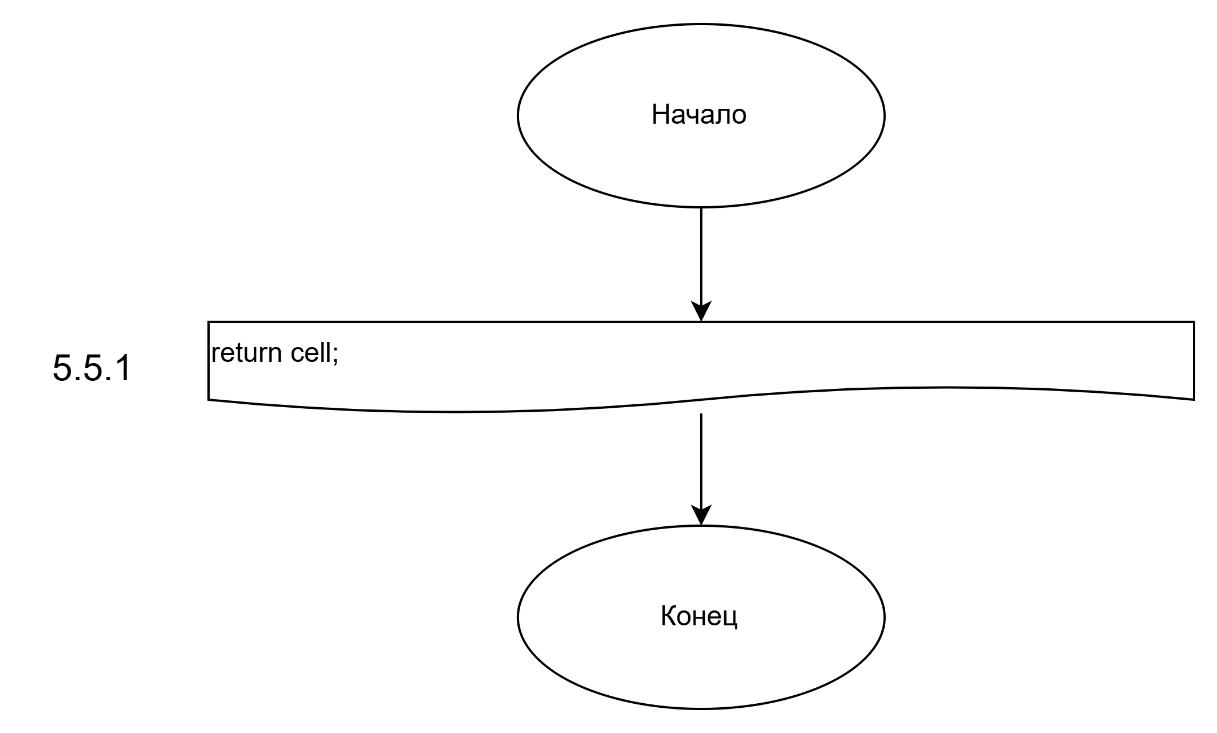


Рисунок 2 – Сохранение данных в xlsx файл

1. **Подбор тестовых примеров**

В качестве тестовых данных созданы два ряда данных:

1) 13.03.2006, 0, Центральный склад материалов, склад, Стройплощадка №8, строительство, Центральный склад материалов, директор, 153

2) 17.03.2006, 0, Центральный склад материалов, склад, Стройплощадка №15, строительство, Центральный склад материалов, зам. директора, 157

1. **Листинг (код) составленного программного обеспечения**

**Файл MyForm.cs**

using System.Diagnostics;

using System;

using System.Windows.Forms;

using Excel = Microsoft.Office.Interop.Excel;

using System.IO;

using System.Linq;

namespace MyExcel

{

public partial class MyForm : Form

{

public MyForm()

{

InitializeComponent();

}

private void buttonPreview\_Click(object sender, System.EventArgs e)

{

string[] data = File.ReadAllText("DefaultData.csv").Split(',');

for (int i = 0; i < data.Length; i += 9)

dataGridViewPreview.Rows.Add(data.Skip(i).Take(i + 9).ToArray());

}

private void buttonExport\_Click(object sender, System.EventArgs e)

{

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

app.Visible = false;

Excel.Workbook book = app.Workbooks.Add();

Excel.Worksheet sheet = book.ActiveSheet;

sheet.Columns[1].ColumnWidth = 10;

sheet.Columns[2].ColumnWidth = 9;

sheet.Columns[3].ColumnWidth = 19;

sheet.Columns[4].ColumnWidth = 19;

sheet.Columns[5].ColumnWidth = 19;

sheet.Columns[6].ColumnWidth = 19;

sheet.Columns[7].ColumnWidth = 19;

sheet.Columns[8].ColumnWidth = 19;

sheet.Columns[9].ColumnWidth = 8;

sheet.Columns[10].ColumnWidth = 10;

sheet.Cells[1, 1] = "Дата состав-\nления";

sheet.Cells[1, 2] = "Код\nвида\nоперации";

sheet.Cells[1, 3] = "Отправитель";

sheet.Cells[1, 5] = "Получатель";

sheet.Cells[1, 7] = "Ответственный за поставку";

sheet.Cells[2, 3] = "структурное\nподразделение";

sheet.Cells[2, 4] = "вид\nдеятельности";

sheet.Cells[2, 5] = "структурное\nподразделение";

sheet.Cells[2, 6] = "вид\nдеятельности";

sheet.Cells[2, 7] = "структурное\nподразделение";

sheet.Cells[2, 8] = "вид\nдеятельности";

sheet.Cells[2, 9] = "код испол-\nнителя";

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

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

sheet.Range[ExcelCellTranslator(1, 3), ExcelCellTranslator(1, 4)].Merge();

sheet.Range[ExcelCellTranslator(1, 5), ExcelCellTranslator(1, 6)].Merge();

sheet.Range[ExcelCellTranslator(1, 7), ExcelCellTranslator(1, 9)].Merge();

Excel.Range cells = sheet.Range[ExcelCellTranslator(1, 1), ExcelCellTranslator(4, 10)];

cells.WrapText = true;

Excel.Range rows;

rows = sheet.Range[ExcelCellTranslator(1, 1), ExcelCellTranslator(1, 9)];

rows.Borders[Excel.XlBordersIndex.xlEdgeTop].LineStyle = Excel.XlLineStyle.xlDouble;

rows = sheet.Range[ExcelCellTranslator(1, 1), ExcelCellTranslator(1, 9)];

rows.Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle = Excel.XlLineStyle.xlContinuous;

rows = sheet.Range[ExcelCellTranslator(2, 1), ExcelCellTranslator(2, 9)];

rows.Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle = Excel.XlLineStyle.xlContinuous;

rows.Borders[Excel.XlBordersIndex.xlEdgeBottom].Weight = Excel.XlBorderWeight.xlThick;

rows = sheet.Range[ExcelCellTranslator(3, 1), ExcelCellTranslator(3, 9)];

rows.Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle = Excel.XlLineStyle.xlContinuous;

rows.Borders[Excel.XlBordersIndex.xlEdgeBottom].Weight = Excel.XlBorderWeight.xlThick;

rows = sheet.Range[ExcelCellTranslator(4, 1), ExcelCellTranslator(4, 9)];

rows.Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle = Excel.XlLineStyle.xlContinuous;

rows.Borders[Excel.XlBordersIndex.xlEdgeBottom].Weight = Excel.XlBorderWeight.xlThick;

Excel.Range cols;

foreach (int col in new int[] { 1, 2, 3, 5, 7, 10 })

{

cols = sheet.Range[sheet.Cells[1, col], sheet.Cells[4, col]];

cols.Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlDouble;

}

foreach (int col in new int[] { 4, 6, 8, 9 })

{

cols = sheet.Range[sheet.Cells[1, col], sheet.Cells[4, col]];

cols.Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlContinuous;

}

int[] cols\_to\_fix = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };

foreach (int col in cols\_to\_fix)

{

Excel.Range fix = sheet.Range[sheet.Cells[1, col], sheet.Cells[4, col]];

fix.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

fix.VerticalAlignment = Excel.XlVAlign.xlVAlignTop;

}

string[] data = File.ReadAllText("DefaultData.csv").Split(',');

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

{

sheet.Cells[3 + i / 9, 1 + i % 9] = data[i];

}

string path = Environment.GetFolderPath(Environment.SpecialFolder.Desktop) + "\\Грачева Н.С. Вариант№5.xlsx";

book.SaveAs(path);

book.Close();

app.Quit();

Process.Start(new ProcessStartInfo

{

FileName = path,

UseShellExecute = true

});

}

private static string ExcelCellTranslator(int i, int j)

{

string cell = "";

int x;

int lose;

x = j;

if (x < 16384)

{

lose = (x - 1) / 676;

if (lose > 0)

{

cell += Alphabet(lose);

x = x - (676 \* lose);

}

lose = (x - 1) / 26;

if (lose > 0)

{

cell += Alphabet(lose);

x = x - (26 \* lose);

}

cell += Alphabet(x);

}

else

{

cell += "XFD";

}

cell += i.ToString();

return cell;

}

private static string Alphabet(int Num)

{

return ((char)('A' + Num - 1)).ToString();

}

}

}

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

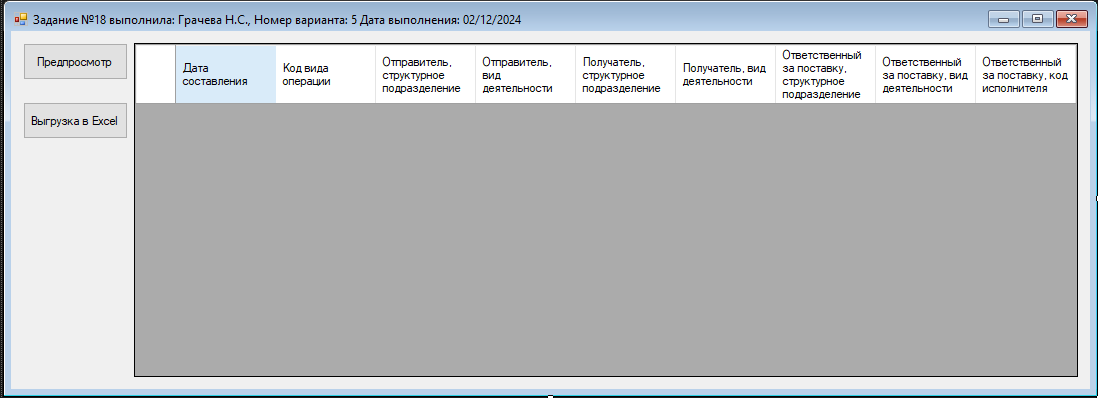
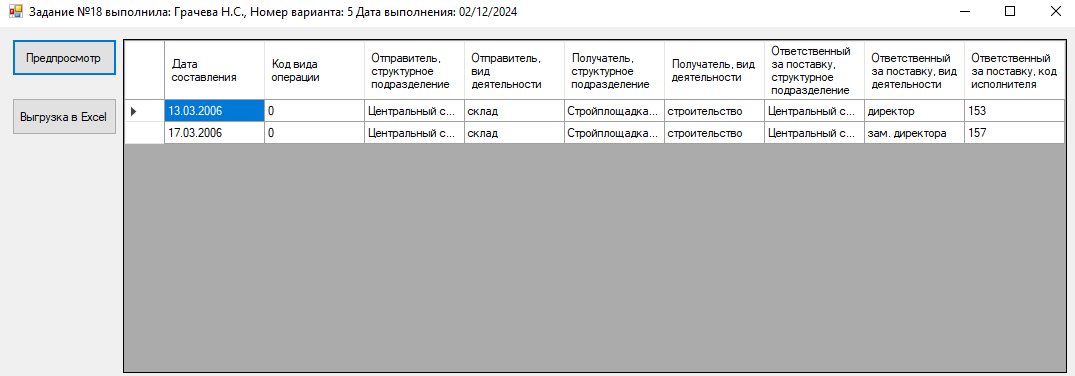


Рисунок 3 – Интерфейс главной формы

На форме присутствует таблица, подписанная «Предпросмотр». В этой таблице отображается загруженная информация из csv файла. По кнопке «Выгрузка в Excel» происходит экспорт данных.

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

Рисунок 4 – Подтверждение соответствия графического пользовательского интерфейса требованиям к оформлению

На форме присутствует возможность предпросмотра данных из csv файла, а также кнопка, позволяющая выгрузки данных в Excel**.**

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



Рисунок 5 – Созданный автоматически файл формата xlsx

1. **Формулировка вывода о проделанной работе**

Освоила навыки подключения внешних модулей и их использования в разрабатываемых программах в интегрированной среде разработки Microsoft Visual Studio. Это позволяет значительно расширить функциональность приложений и интегрировать их с другими системами.

Приобрела навыки программируемой настройки параметров электронных таблиц Microsoft Office Excel, что способствует автоматизации работы с данными и повышению эффективности анализа и обработки информации. Эти знания дают возможность создавать более комплексные решения, оптимизируя процессы и повышая продуктивность работы.