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»

Лабораторная работа №6

по дисциплине:

«Методы программирования»

на тему:  
«Автоматизация Excel»

Выполнил: ст. гр. ТКИ-341

Панаргин В.М.

Вариант №5

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

Москва – 2024 г.

# **1. Цель работы**

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

# **2. Формулировка задачи**

1. Создать графический пользовательский интерфейс с кнопками: «Выгрузка в *Excel*», «Заполнение данными», «Предпросмотр».

2. Нажатие на кнопку «Выгрузка в *Excel*» инициирует запуск на исполнение метода составления электронной таблицы заданной по варианту структуры (обрамление, размеры ячеек, цвета, одноуровневая / многоуровневая «шапка», «чердак», «подвал»).

3. Нажатие на кнопку «Заполнение данными» инициирует запуск на исполнение метода расстановки значений в ячейки из текстового файла. Для вариантов, в схематичной структуре которых присутствуют данные в ячейках таблицы, использовать проиллюстрированные значения, для вариантов, содержащих незаполненные структуры таблиц, необходимо дополнительно продумать возможные наборы данных для заполнения ячеек.

4. Нажатие на кнопку «Предпросмотр» инициирует вызов дочерней экранной формы, содержащей приближенный аналог таблицы, выполненный на базе интерфейсного элемента управления *DataGridView*. «Предпросмотром» можно пользоваться как до подгрузки данных в табличную структуру, так и после.

5. Excel файл должен создаваться в соответсвии с изображением:

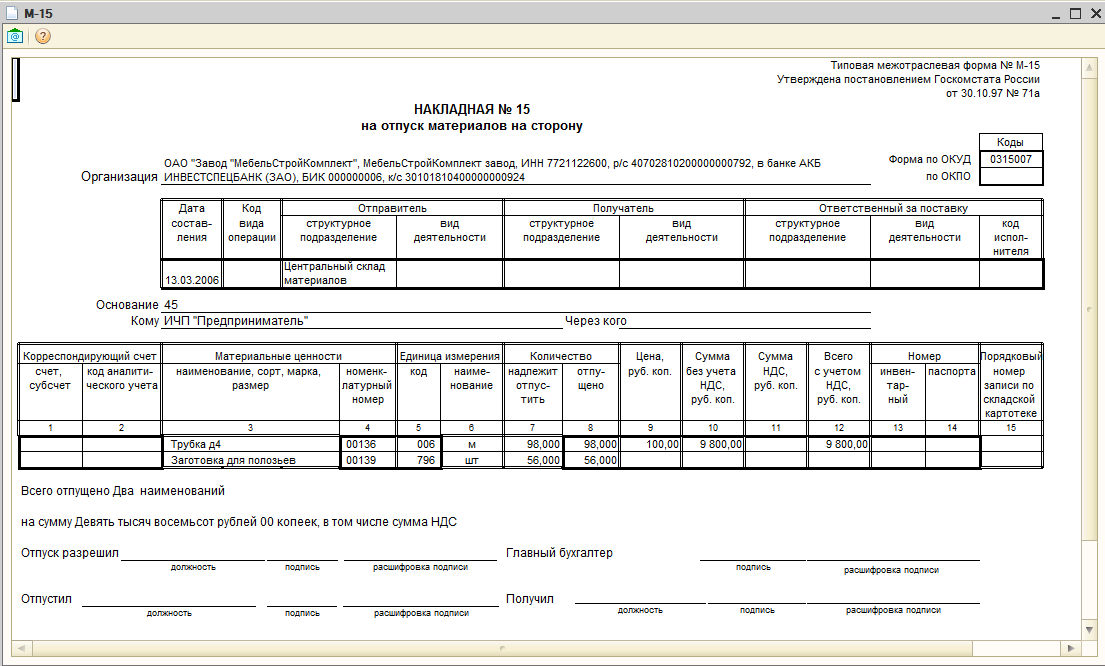


Рисунок 1 - Пример оформления файла

# **3. Составление диаграммы классов, входящих в состав решения.**

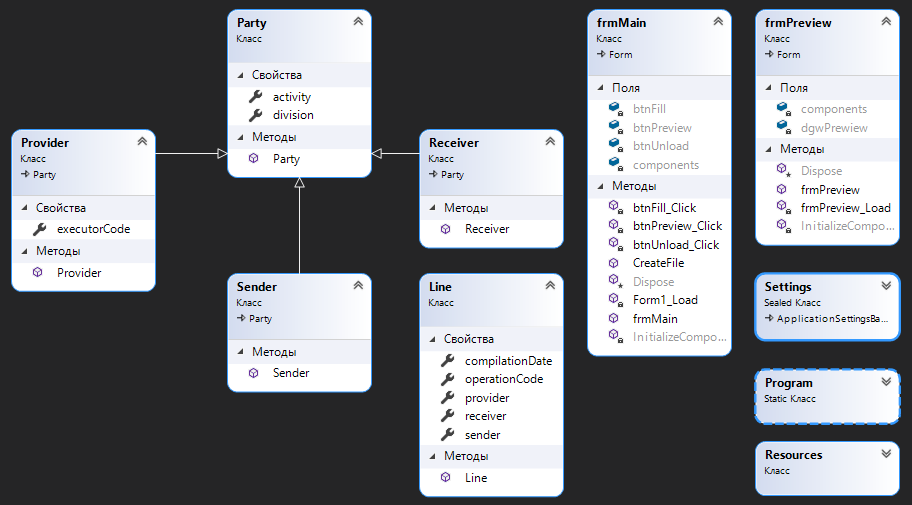


Рисунок 2 - Диаграмма классов

# **4. Составление сети Петри запрограммированного технологического процесса.**

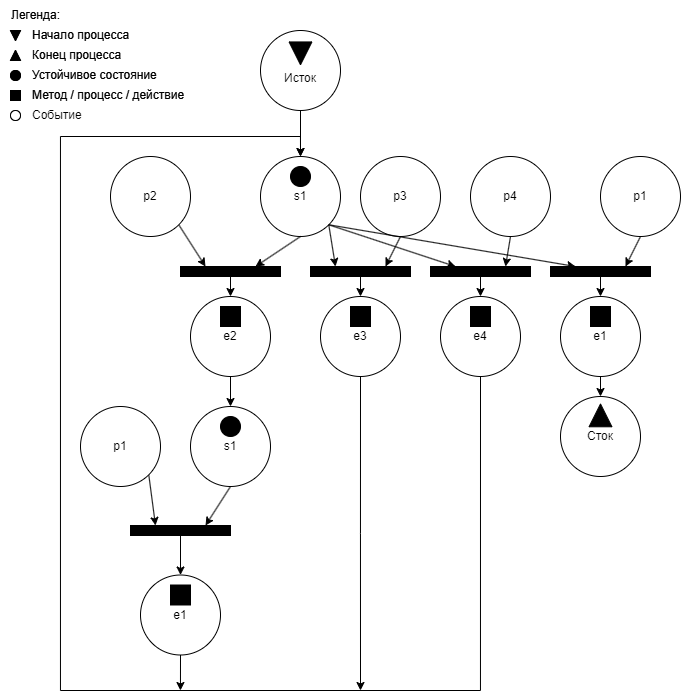


Рисунок 3 - Сеть Петри

**Описание сети Петри**

- состояния (states)

s1 – форма ожидает действий пользователя

- действия (effects)

e1 – закрытие формы

e2 – открытие формы предпросмотра

e3 – запись шапки в файл

e4 – запись данных из текстового файла в таблицу excel

- события (prompts)

p1 – пользователь закрывает форму

p2 – нажата кнопка Предпросмотр

p3 – нажата кнопка Выгрузка в Excel

p4 – нажата кнопка Заполнение данными

# **5. Составление схем алгоритмов методов в составе решения, отмеченных на сети Петри в качестве «эффектов» (метка ).**



Рисунок 4 - Создание шапки

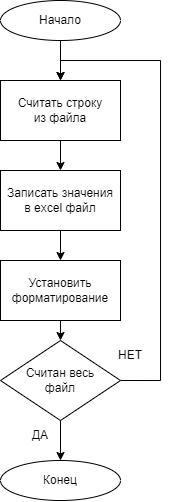


Рисунок 5 - Запись информации из файла

# **6. Подбор тестовых примеров.**

Тестовые примеры:

* Проверить генерируемый файл Excel
* Проверить предпросмотр

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

Основная форма:

public partial class frmMain : Form

{

public frmMain()

{

InitializeComponent();

}

public void CreateFile(ref Excel.Application exapp)

{

Excel.Worksheet workSheet = (Excel.Worksheet)exapp.ActiveSheet;

workSheet.Range["B2"].Value = "Дата составления";

workSheet.Columns[2].ColumnWidth = 20f;

workSheet.Range["B2:B3"].Merge();

workSheet.Range["B2:B3"].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlDouble;

workSheet.Range["C2"].Value = "Код вида операции";

workSheet.Columns[3].ColumnWidth = 20f;

workSheet.Range["C2:C3"].Merge();

workSheet.Range["C2:C3"].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlDouble;

workSheet.Range["D2"].Value = "Отправитель";

workSheet.Range["D2:E2"].Merge();

workSheet.Range["D2:D3"].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlDouble;

workSheet.Range["D3"].Value = "структурное подразделение";

workSheet.Columns[4].ColumnWidth = 30f;

workSheet.Range["E3"].Value = "вид деятельности";

workSheet.Range["E3"].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlContinuous;

workSheet.Range["E3"].Borders[Excel.XlBordersIndex.xlEdgeLeft].Weight = 2;

workSheet.Columns[5].ColumnWidth = 30f;

workSheet.Range["F2"].Value = "Получатель";

workSheet.Range["F2:F3"].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlDouble;

workSheet.Range["F2:G2"].Merge();

workSheet.Range["F3"].Value = "структурное подразделение";

workSheet.Columns[6].ColumnWidth = 30f;

workSheet.Range["G3"].Value = "вид деятельности";

workSheet.Range["G3"].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlContinuous;

workSheet.Range["G3"].Borders[Excel.XlBordersIndex.xlEdgeLeft].Weight = 2;

workSheet.Columns[7].ColumnWidth = 30f;

workSheet.Range["H2"].Value = "Ответственный за поставку";

workSheet.Range["H2:J2"].Merge();

workSheet.Range["H2:H3"].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlDouble;

workSheet.Range["H3"].Value = "структурное подразделение";

workSheet.Columns[8].ColumnWidth = 30f;

workSheet.Range["I3"].Value = "вид деятельности";

workSheet.Range["I3"].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlContinuous;

workSheet.Range["I3"].Borders[Excel.XlBordersIndex.xlEdgeLeft].Weight = 2;

workSheet.Columns[9].ColumnWidth = 30f;

workSheet.Range["J3"].Value = "код исполнителя";

workSheet.Range["J3"].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlContinuous;

workSheet.Range["J3"].Borders[Excel.XlBordersIndex.xlEdgeLeft].Weight = 2;

workSheet.Columns[10].ColumnWidth = 20f;

workSheet.Range["K2:K3"].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlDouble;

workSheet.Range["B2:J2"].Borders[Excel.XlBordersIndex.xlEdgeTop].LineStyle = Excel.XlLineStyle.xlDouble;

workSheet.Range["D2:J2"].Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle = Excel.XlLineStyle.xlContinuous;

workSheet.Range["D2:J2"].Borders[Excel.XlBordersIndex.xlEdgeBottom].Weight = 2;

workSheet.Range["B2:J3"].HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

workSheet.Range["B2:J3"].VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

exapp.ActiveWorkbook.SaveAs(Application.StartupPath + "\\TST.xlsx");

}

private void Form1\_Load(object sender, EventArgs e)

{

}

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

{

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

exapp.Workbooks.Add();

this.CreateFile(ref exapp);

exapp.Quit();

}

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

{

List<Line> existingRecords = new List<Line> { };

string tempStr;

StreamReader sr = new StreamReader(Application.StartupPath + "\\DocInfo.txt");

while ((tempStr = sr.ReadLine()) != null)

{

existingRecords.Add(new Line(tempStr));

}

sr.Close();

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

if (!(File.Exists(Application.StartupPath + "\\TST.xlsx")))

{

exapp.Workbooks.Add();

this.CreateFile(ref exapp);

}

else

{

exapp.Workbooks.Open(Application.StartupPath + "\\TST.xlsx");

}

Excel.Worksheet workSheet = (Excel.Worksheet)exapp.ActiveSheet;

int counter = 1;

int deltar = 3;

int deltac = 1;

foreach (Line line in existingRecords)

{

workSheet.Cells[counter + deltar, 1 + deltac] = line.compilationDate.ToString("d");

workSheet.Cells[counter + deltar, 1 + deltac].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlDouble;

workSheet.Cells[counter + deltar, 2 + deltac] = line.operationCode.ToString();

workSheet.Cells[counter + deltar, 2 + deltac].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlDouble;

workSheet.Cells[counter + deltar, 3 + deltac] = line.sender.division;

workSheet.Cells[counter + deltar, 4 + deltac] = line.sender.activity;

workSheet.Cells[counter + deltar, 4 + deltac].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlContinuous;

workSheet.Cells[counter + deltar, 4 + deltac].Borders[Excel.XlBordersIndex.xlEdgeTop].Weight = 2;

workSheet.Cells[counter + deltar, 3 + deltac].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlDouble;

workSheet.Cells[counter + deltar, 5 + deltac] = line.receiver.division;

workSheet.Cells[counter + deltar, 6 + deltac] = line.receiver.activity;

workSheet.Cells[counter + deltar, 6 + deltac].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlContinuous;

workSheet.Cells[counter + deltar, 6 + deltac].Borders[Excel.XlBordersIndex.xlEdgeTop].Weight = 2;

workSheet.Cells[counter + deltar, 5 + deltac].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlDouble;

workSheet.Cells[counter + deltar, 7 + deltac] = line.provider.division;

workSheet.Cells[counter + deltar, 8 + deltac] = line.provider.activity;

workSheet.Cells[counter + deltar, 8 + deltac].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlContinuous;

workSheet.Cells[counter + deltar, 8 + deltac].Borders[Excel.XlBordersIndex.xlEdgeTop].Weight = 2;

workSheet.Cells[counter + deltar, 9 + deltac] = line.provider.executorCode.ToString();

workSheet.Cells[counter + deltar, 9 + deltac].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlContinuous;

workSheet.Cells[counter + deltar, 9 + deltac].Borders[Excel.XlBordersIndex.xlEdgeTop].Weight = 2;

workSheet.Cells[counter + deltar, 7 + deltac].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlDouble;

workSheet.Cells[counter + deltar, 10 + deltac].Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle = Excel.XlLineStyle.xlDouble;

Excel.Range c1 = workSheet.Cells[counter + deltar, 1 + deltac];

Excel.Range c2 = workSheet.Cells[counter + deltar, 9 + deltac];

workSheet.get\_Range(c1, c2).Borders[Excel.XlBordersIndex.xlEdgeTop].LineStyle = Excel.XlLineStyle.xlContinuous;

workSheet.get\_Range(c1, c2).Borders[Excel.XlBordersIndex.xlEdgeTop].Weight = 4;

workSheet.get\_Range(c1, c2).Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle = Excel.XlLineStyle.xlContinuous;

workSheet.get\_Range(c1, c2).Borders[Excel.XlBordersIndex.xlEdgeBottom].Weight = 4;

workSheet.get\_Range(c1, c2).HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

workSheet.get\_Range(c1, c2).VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

counter++;

}

exapp.ActiveWorkbook.Save();

exapp.Quit();

}

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

{

frmPreview preview = new frmPreview();

preview.Show();

}

}

Форма предпросмотра:

public partial class frmPreview : Form

{

public frmPreview()

{

InitializeComponent();

}

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

{

this.WindowState = FormWindowState.Maximized;

this.dgwPrewiew.ColumnCount = 9;

this.dgwPrewiew.Columns[0].HeaderText = "Дата составления";

this.dgwPrewiew.Columns[0].Width = 150;

this.dgwPrewiew.Columns[1].HeaderText = "Код вида операции";

this.dgwPrewiew.Columns[1].Width = 150;

this.dgwPrewiew.Columns[2].HeaderText = "Отправитель";

this.dgwPrewiew.Columns[2].Width = 200;

this.dgwPrewiew.Columns[3].Width = 200;

this.dgwPrewiew.Columns[4].HeaderText = "Получатель";

this.dgwPrewiew.Columns[4].Width = 200;

this.dgwPrewiew.Columns[5].Width = 200;

this.dgwPrewiew.Columns[6].HeaderText = "Ответственный за поставку";

this.dgwPrewiew.Columns[6].Width = 200;

this.dgwPrewiew.Columns[7].Width = 200;

this.dgwPrewiew.Columns[8].Width = 150;

this.dgwPrewiew.Rows.Add("", "", "структурное подразделение", "вид деятельности", "структурное подразделение", "вид деятельности", "структурное подразделение", "вид деятельности", "код исполнителя");

List<Line> existingRecords = new List<Line> { };

string tempStr;

StreamReader sr = new StreamReader(Application.StartupPath + "\\DocInfo.txt");

while ((tempStr = sr.ReadLine()) != null)

{

existingRecords.Add(new Line(tempStr));

}

sr.Close();

foreach (Line line in existingRecords)

{

this.dgwPrewiew.Rows.Add(line.compilationDate.ToString("d"), line.operationCode.ToString(), line.sender.division, line.sender.activity, line.receiver.division, line.receiver.activity, line.provider.division, line.provider.activity, line.provider.executorCode.ToString());

}

}

}

Дополнительные классы:

public class Party

{

public Party(string partyInfo)

{

string[] strParty = partyInfo.Split(';');

this.division = strParty[0];

this.activity = strParty[1];

}

public string division { get; set; }

public string activity { get; set; }

}

public class Sender : Party

{

public Sender(string senderInfo)

: base(senderInfo) { }

}

public class Receiver : Party

{

public Receiver(string receiverInfo)

: base(receiverInfo) { }

}

public class Provider : Party

{

public Provider(string providerInfo)

: base(providerInfo)

{

string[] strProvider = providerInfo.Split(';');

this.executorCode = Convert.ToInt32(strProvider[2]);

}

public int executorCode { get; set; }

}

public class Line

{

public Line(string lineInfo)

{

string[] temp = lineInfo.Split('!');

string[] strLine = temp[0].Split(';');

this.compilationDate = Convert.ToDateTime(strLine[0]);

this.operationCode = Convert.ToInt32(strLine[1]);

this.sender = new Sender(temp[1]);

this.receiver = new Receiver(temp[2]);

this.provider = new Provider(temp[3]);

}

public DateTime compilationDate { get; set; }

public int operationCode { get; set; }

public Sender sender { get; set; }

public Receiver receiver { get; set; }

public Provider provider { get; set; }

}

# **8. Графический пользовательский интерфейс программного обеспечения и его описание.**

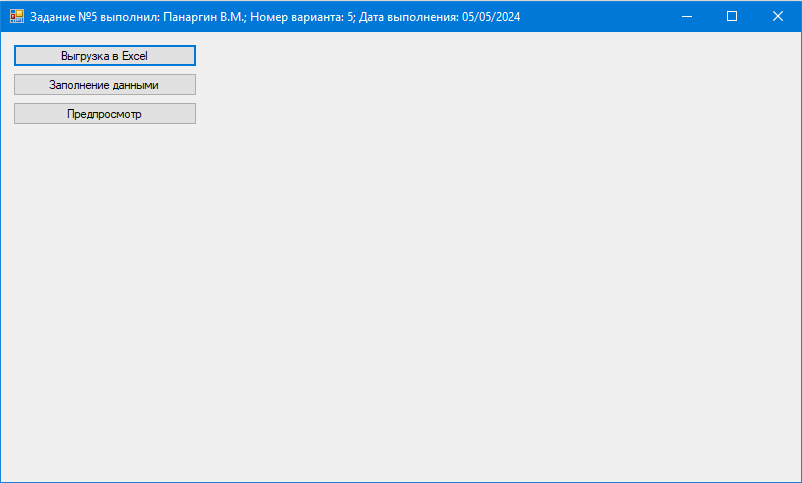


Рисунок 6 - Интерфейс основной формы



Рисунок 7 - Интерфейс формы предпросмотра

# **9. Подтверждение соответствия графического пользовательского интерфейса требованиям к оформлению.**

1. Заголовок главной формы должен содержать надпись вида: «Задание №5 выполнил: [Фамилия И.О. автора]; Номер варианта: [Номер]; Дата выполнения: [дд/мм/гггг]».

См. Рисунок 6

2. Дата выполнения проставляется в момент, когда программа считается законченной и по ней можно готовить итоговый отчёт о выполнении работы. Этот момент должен быть согласован с преподавателем, ведущим учебную дисциплину.



Рисунок 8 - Дата на момент создания отчета

# **10. Расчёт тестовых примеров с использованием составленного программного обеспечения.**

* Проверить генерируемый файл Excel

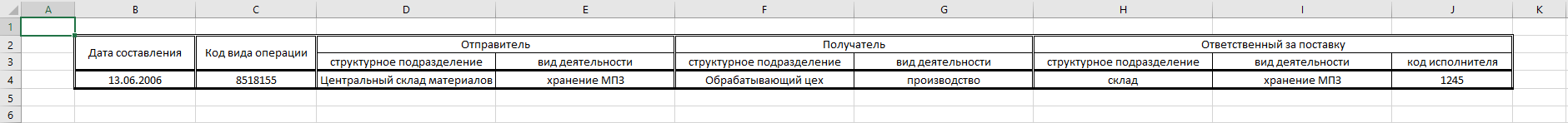


Рисунок 9 - Генерируемый файл Excel

* Проверить предпросмотр

См. Рисунок 7

# **11. Формулировка вывода о проделанной работе**

В рамках данной работы были закреплены навыки подключения внешних модулей (библиотек классов) и их использования в разрабатываемых программах; навыки программируемой настройки параметров документов Microsoft Office Excel.