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

Кругликов Е.А.

Вариант №8

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**

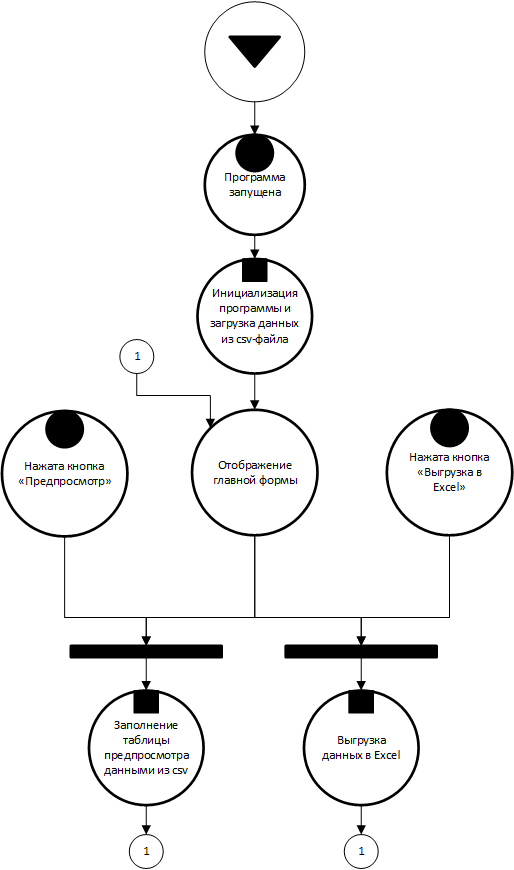


Рисунок 1 – Сеть Петри запрограммированного технологического процесса

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

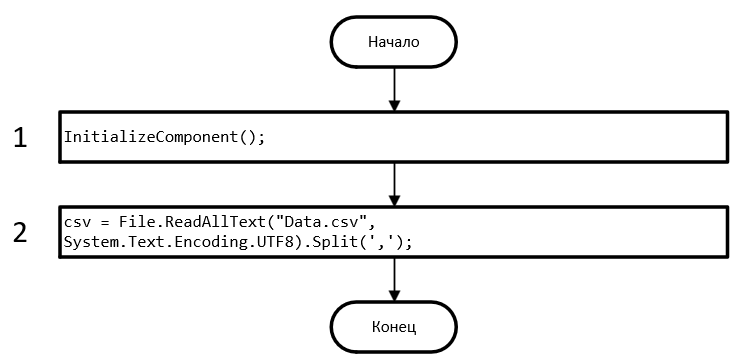


Рисунок 2 – Инициализация программы и загрузка данных из csv-файла

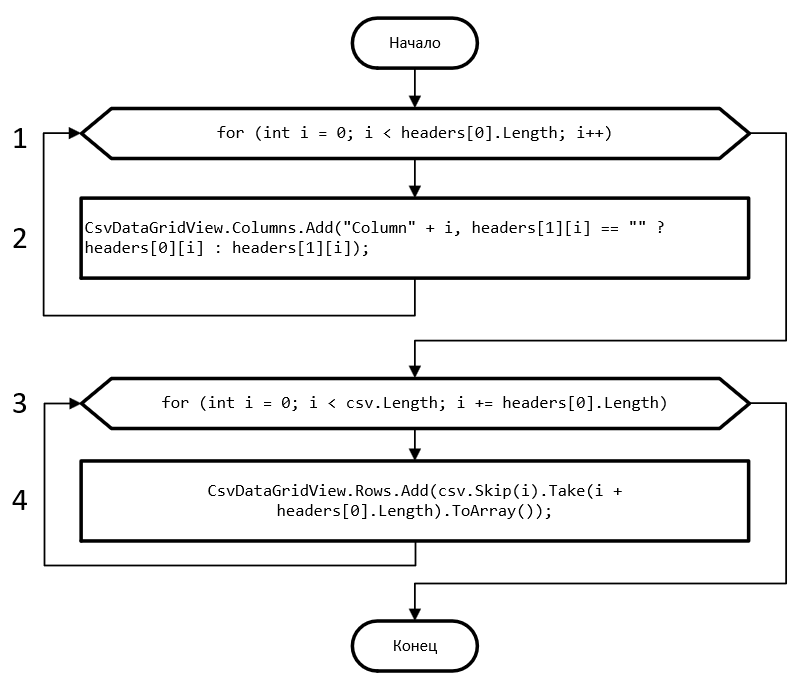


Рисунок 3 – Заполнение таблицы предпросмотра данными из csv

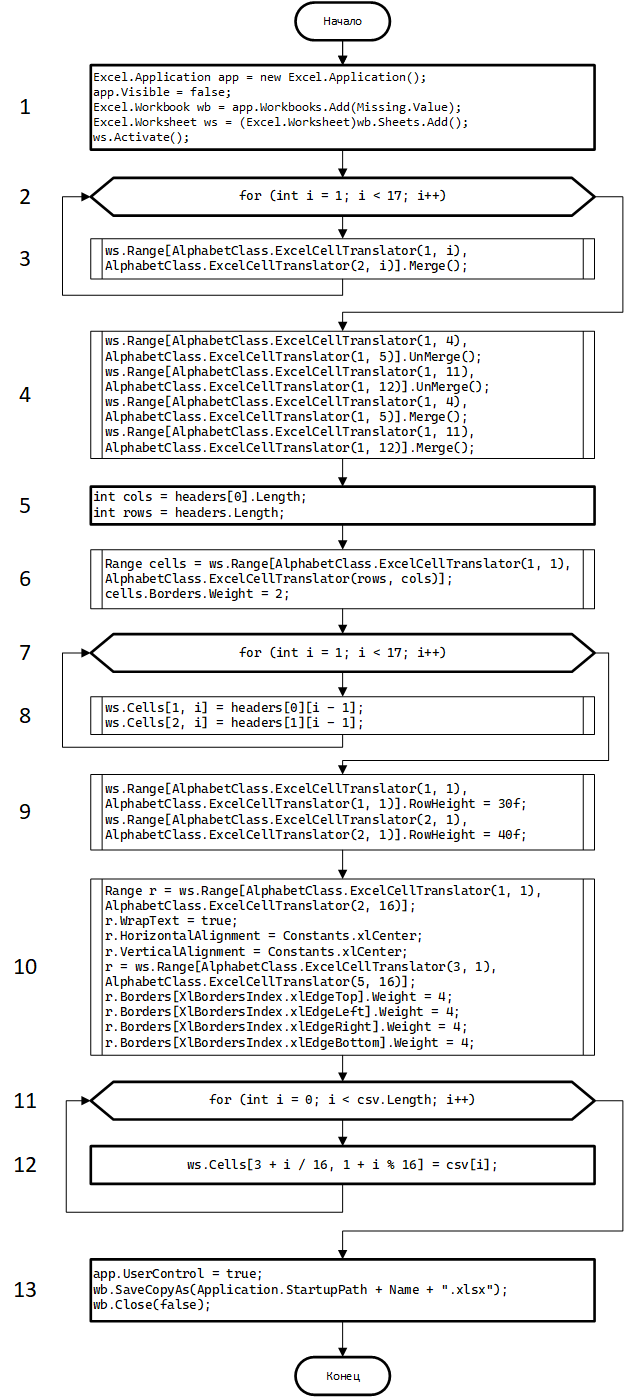


Рисунок 4 – Выгрузка данных в Excel

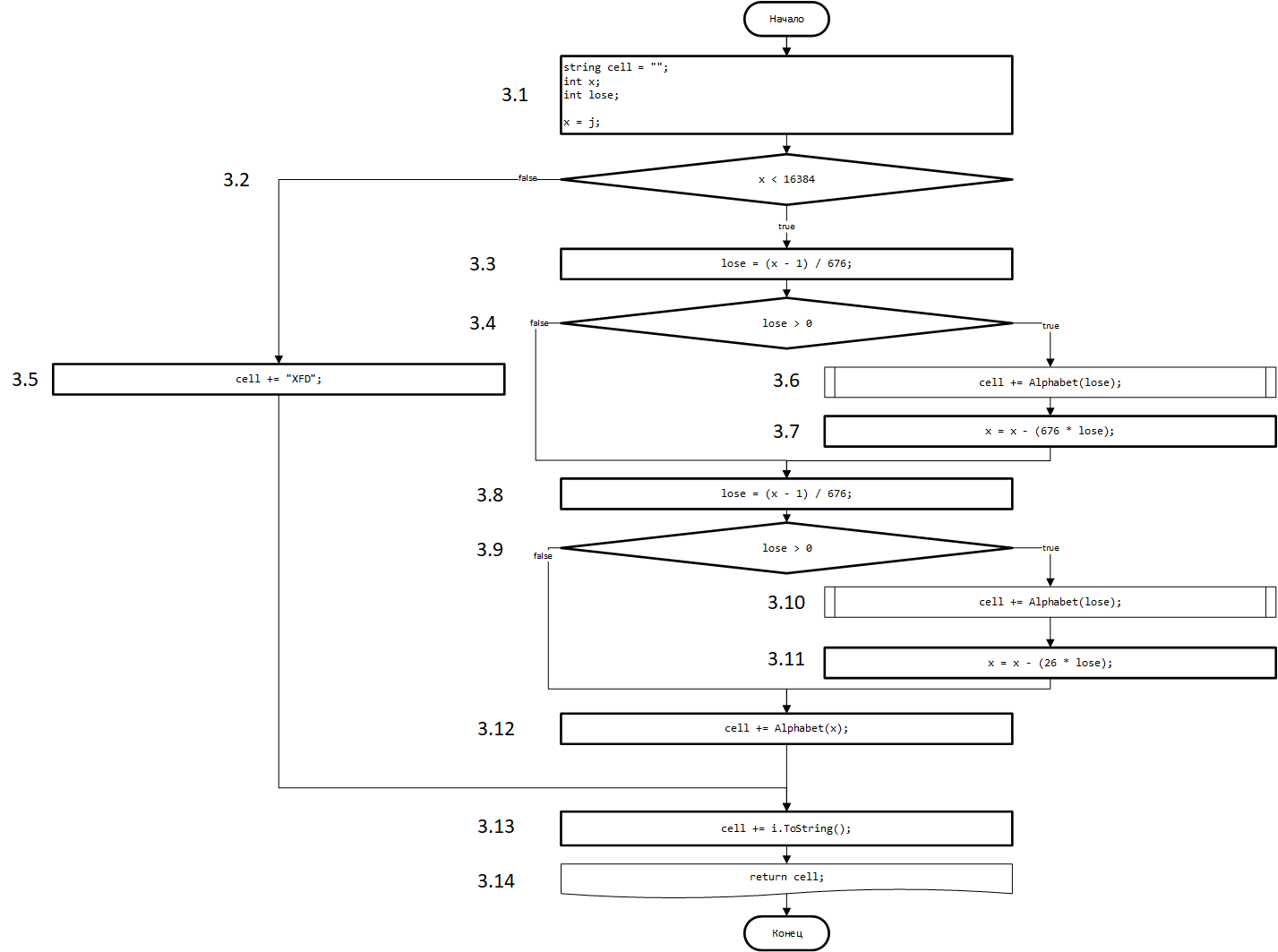


Рисунок 5 – Метод ExcelCellTranslator

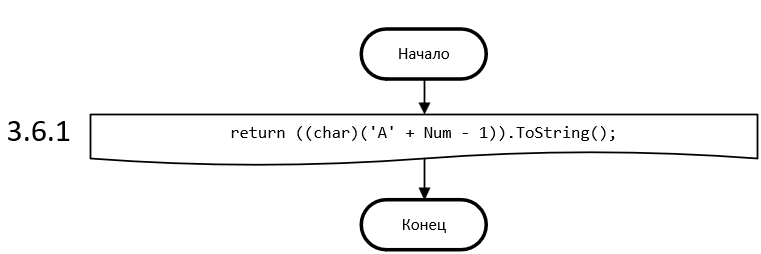


Рисунок 6 – Метод Alphabet

1. **Подбор тестовых примеров**

В качестве тестовых данных создан *csv*-файл, заполненный данными о наличии на двух складах бетона двух марок. Таким образом, *csv*-файл содержит следующий текст:

Самолет-Чертаново,Строительство,Чертаново №1,130,полностью,БСГ М-100 (В7.5),-,-,10 м.к.,156,м.к., метр кубический, 3100.00,70,10 лет,МосБетон,Самолет-Чертаново,Строительство,Чертаново №3,20-70,полностью,БСГ М-100 (В7.5),-,-,150 м.к.,156,м.к., метр кубический, 3100.00,70,10 лет,МосБетон,Самолет-Люблино,Строительство,Люблино №2,5,полностью,БСГ М300 (B22.5),-,-,20 м.к.,159,м.к., метр кубический, 3500.00,20,12 лет,МосБетон

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

**Файл MainForm.cs**

using System;

using System.IO;

using System.Linq;

using System.Reflection;

using System.Windows.Forms;

using Excel = Microsoft.Office.Interop.Excel;

using ExcelAlphabet;

using Microsoft.Office.Interop.Excel;

namespace ExcelTableCreator

{

public partial class MainForm : Form

{

private readonly string[][] headers = new string[][]

{

new string[] { "Структурное подразделение", "Вид деятельности", "Склад", "Место хранения", "", "Марка", "Сорт", "Профиль", "Размер", "Номенклатурный номер", "Единица измерения", "", "Цена, руб. коп.", "Норма запаса", "Срок годности", "Поставщик" },

new string[] { "", "", "", "стеллаж", "ячейка", "", "", "", "", "", "код", "наименование", "", "", "", "" }

};

private string[] csv;

public MainForm()

{

InitializeComponent();

csv = File.ReadAllText("Data.csv", System.Text.Encoding.UTF8).Split(',');

}

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

{

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

app.Visible = true;

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

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

ws.Activate();

for (int i = 1; i < 17; i++)

{

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

}

ws.Range[AlphabetClass.ExcelCellTranslator(1, 4), AlphabetClass.ExcelCellTranslator(1, 5)].UnMerge();

ws.Range[AlphabetClass.ExcelCellTranslator(1, 11), AlphabetClass.ExcelCellTranslator(1, 12)].UnMerge();

ws.Range[AlphabetClass.ExcelCellTranslator(1, 4), AlphabetClass.ExcelCellTranslator(1, 5)].Merge();

ws.Range[AlphabetClass.ExcelCellTranslator(1, 11), AlphabetClass.ExcelCellTranslator(1, 12)].Merge();

int cols = headers[0].Length;

int rows = headers.Length;

Range cells = ws.Range[AlphabetClass.ExcelCellTranslator(1, 1), AlphabetClass.ExcelCellTranslator(rows + 3, cols)];

cells.Borders.Weight = 2;

for (int i = 1; i < 17; i++)

{

ws.Cells[1, i] = headers[0][i - 1];

ws.Cells[2, i] = headers[1][i - 1];

}

ws.Range[AlphabetClass.ExcelCellTranslator(1, 1), AlphabetClass.ExcelCellTranslator(1, 1)].RowHeight = 30f;

ws.Range[AlphabetClass.ExcelCellTranslator(2, 1), AlphabetClass.ExcelCellTranslator(2, 1)].RowHeight = 40f;

Range r = ws.Range[AlphabetClass.ExcelCellTranslator(1, 1), AlphabetClass.ExcelCellTranslator(2, 16)];

r.WrapText = true;

r.HorizontalAlignment = Constants.xlCenter;

r.VerticalAlignment = Constants.xlCenter;

r = ws.Range[AlphabetClass.ExcelCellTranslator(3, 1), AlphabetClass.ExcelCellTranslator(5, 16)];

r.Borders[XlBordersIndex.xlEdgeTop].Weight = 4;

r.Borders[XlBordersIndex.xlEdgeLeft].Weight = 4;

r.Borders[XlBordersIndex.xlEdgeRight].Weight = 4;

r.Borders[XlBordersIndex.xlEdgeBottom].Weight = 4;

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

{

ws.Cells[3 + i / cols, 1 + i % cols] = csv[i];

}

app.UserControl = true;

wb.SaveCopyAs(System.Windows.Forms.Application.StartupPath + Name + ".xlsx");

wb.Close(false);

}

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

{

for (int i = 0; i < headers[0].Length; i++)

CsvDataGridView.Columns.Add("Column" + i, headers[1][i] == "" ? headers[0][i] : headers[1][i]);

for (int i = 0; i < csv.Length; i += headers[0].Length)

CsvDataGridView.Rows.Add(csv.Skip(i).Take(i + headers[0].Length).ToArray());

}

}

}

**Файл AlphabetClass.cs**

namespace ExcelAlphabet

{

public static class AlphabetClass

{

public 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**

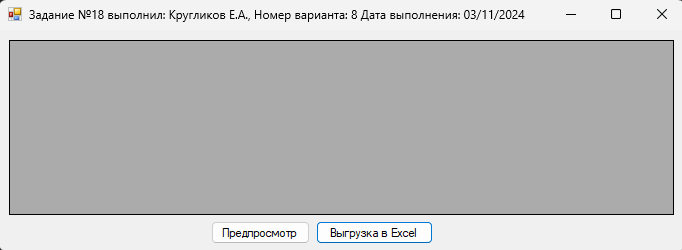


Рисунок 7 – Интерфейс главной формы при запуске

На форме присутствует таблица Предпросмотра. При запуске программы эта таблица пуста.

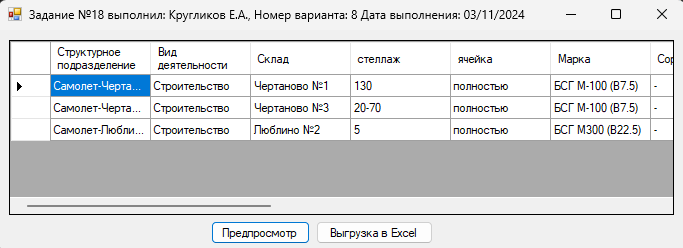


Рисунок 8 – Интерфейс главной формы после нажатия на кнопку «Предпросмотр»

При нажатии на кнопку «Предпросмотр» в эту таблицу отображается загруженная информация из *csv*-файла.

По кнопке «Выгрузка в *Excel*» происходит создание *xlsx*-файла и запись в него данных из *csv*-файла.

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

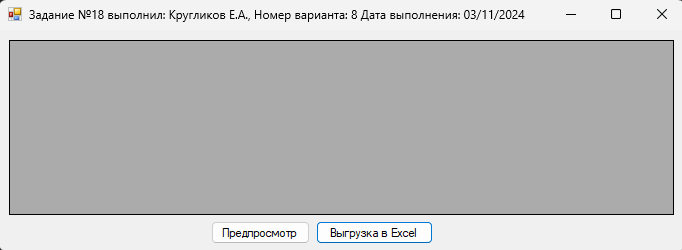


Рисунок 9 – Интерфейс главной формы при запуске

В соответствии с требованиями, на форме присутствует две кнопки: «Предпросмотр» и «Выгрузка в Excel».

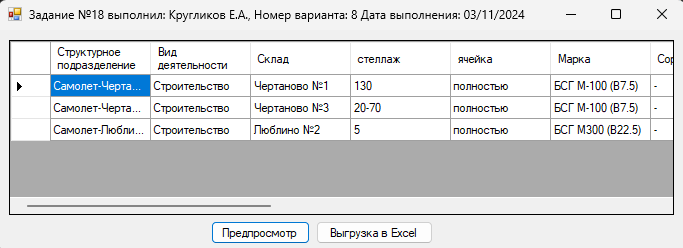


Рисунок 10 – Интерфейс главной формы после нажатия на кнопку «Предпросмотр»

При этом, кнопка «Предпросмотр» при нажатии позволяет отобразить данные, загруженные из csv-файла, на форме, не выгружая их в Excel.

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

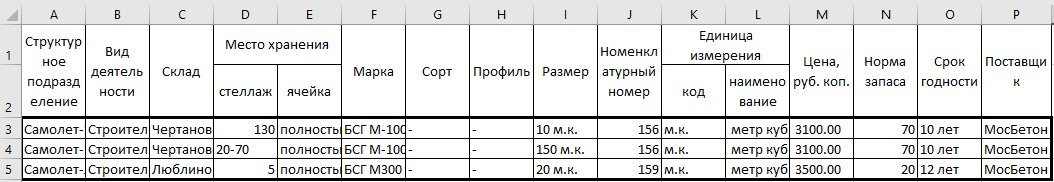


Рисунок 11 – Созданный автоматически файл формата *xlsx*

На данном рисунке мы можем увидеть, что в результате работы программы созданный файл соответствует заложенным требованиям, а также то, что тестовые данные были корректно вставлены в таблицу.

1. **Формулировка вывода о проделанной работе**

В результате выполнения данной работы я освоил навыки подключения внешних модулей и их использования в разрабатываемых программах в интегрированной среде разработки *Microsoft Visual Studio*. Это позволит мне в будущем аналогичным образом создавать приложения для всего пакета офисных программ, а также работать со многими другими внешними модулями, доступными для разработки на платформе .NET Framework.

Также я освоил навыки программируемой настройки параметров электронных таблиц *Microsoft Office Excel*. Это позволит мне автоматизировать работу с упомянутой программой, что не только является полезным навыком для программиста, но и может пригодится в повседневной жизни.

В ходе выполнения лабораторной работы также научился переводить цифровые координаты ячейки в цифро-буквенные, подходящие для таблицы Excel.