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

Токарев В.А.

Макридов А.М.

Вариант №18

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 - Пример оформления файла

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

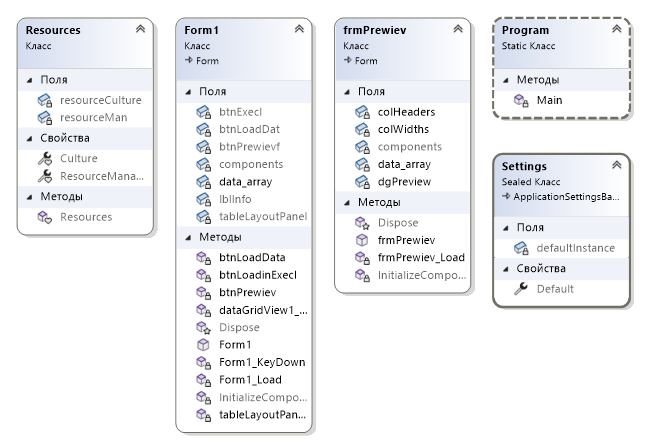


Рисунок - Диаграмма классов

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

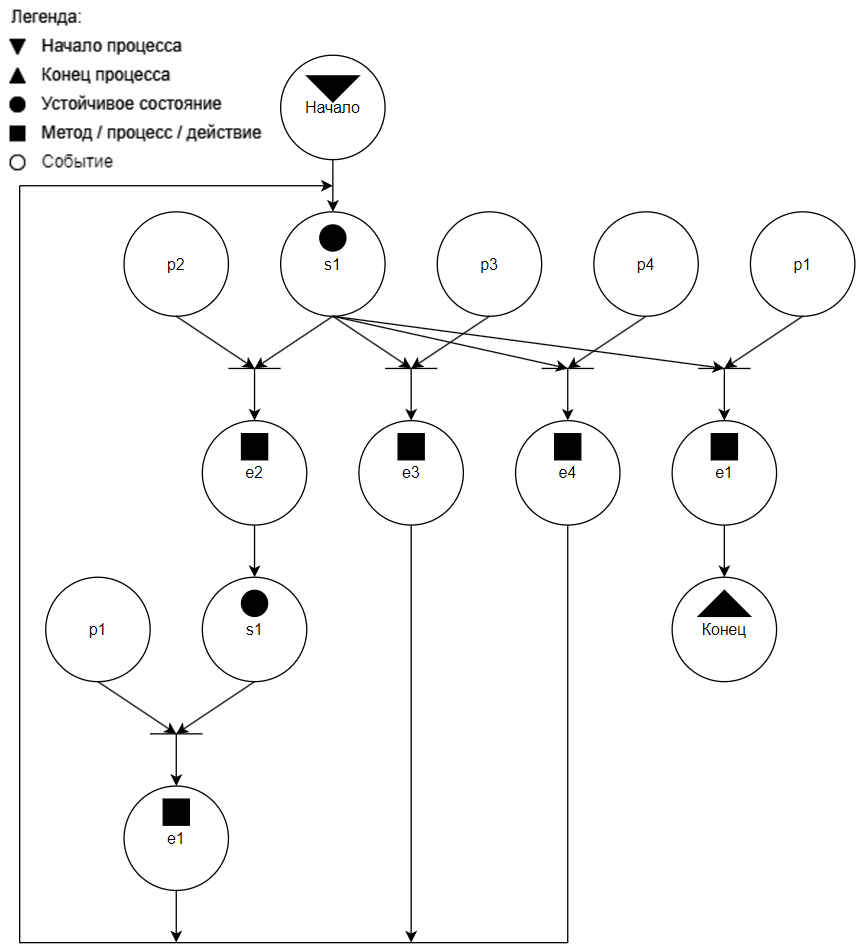


Рисунок - Сеть Петри

**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 - Создание шапки

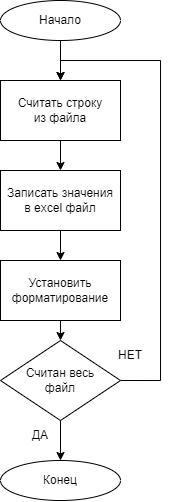


Рисунок - Запись информации из файла

# **6. Подбор тестовых примеров.**

Тестовые примеры:

* Проверить генерируемый файл Excel
* Проверить предпросмотр

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

Основная форма:

using System;

using System.Collections.Generic;

using System.IO;

using System.Windows.Forms;

using WindowsFormsApp5;

using Excel = Microsoft.Office.Interop.Excel;

namespace WindowsFormsApp4

{

public partial class Form1 : Form

{

// Правильно определяем data\_array как список массивов объектов

private List<object[]> data\_array;

public Form1()

{

InitializeComponent();

this.KeyDown += new KeyEventHandler(Form1\_KeyDown);

this.KeyPreview = true;

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void btnLoadData(object sender, EventArgs e)

{

string[] d = File.ReadAllLines("Data.txt");

data\_array = new List<object[]>();

object[] t;

foreach (string c in d)

{

string[] s = c.Split();

t = new object[s.Length];

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

{

t[i] = s[i];

}

data\_array.Add(t);

}

MessageBox.Show("Данные успешно загружены!");

}

private void btnPrewiev(object sender, EventArgs e)

{

frmPrewiev Prewiev = new frmPrewiev(data\_array);

Prewiev.Show();

}

private void btnLoadinExecl(object sender, EventArgs e)

{

// Создаем новый экземпляр приложения Excel

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

// Добавляем новую рабочую книгу

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

// Получаем первый лист

Excel.Worksheet sh = wb.Sheets[1];

sh.Name = "Var18";

sh.Cells.Font.Name = "Times New Roman";

sh.Cells.Font.Size = 12;

// Задаем заголовки и объединяем ячейки

sh.Cells[1, 1].Value = "Производственное подразделение";

sh.Cells[1, 2].Value = "Наименование культуры";

sh.Cells[2, 2].Value = "Код синтетического и аналитического учета";

// Автоподгон ширины столбцов под содержимое для указанных ячеек

sh.Cells[1, 2].EntireColumn.AutoFit();

sh.Cells[2, 2].EntireColumn.AutoFit();

// Задаем высоту строк

sh.Rows[1].RowHeight = 105;

sh.Rows[2].RowHeight = 105;

// Дополнительная настройка заголовков

sh.Cells[1, 3].Value = "Вид работы";

sh.Cells[1, 4].Value = "Наименование работ";

sh.Cells[1, 5].Value = "Объем работ в натуре (га, m^3, тб шеб и др)";

sh.Cells[1, 6].Value = "Объем работ в условных единицах";

// Объединяем и форматируем ячейки

Excel.Range rng = sh.Range[sh.Cells[1, 6], sh.Cells[1, 7]];

rng.MergeCells = true;

sh.Cells[1, 6].EntireColumn.AutoFit();

sh.Cells[1, 7].EntireColumn.AutoFit();

// Объединяем и поворачиваем текст в ячейках

rng = sh.Range[sh.Cells[1, 1], sh.Cells[2, 1]];

rng.MergeCells = true;

rng.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

rng.Orientation = 90;

rng = sh.Range[sh.Cells[1, 3], sh.Cells[2, 3]];

rng.MergeCells = true;

rng.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

rng.Orientation = 90;

rng = sh.Range[sh.Cells[1, 4], sh.Cells[2, 4]];

rng.MergeCells = true;

rng.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

rng.Orientation = 90;

sh.Cells[1, 4].Orientation = 90;

rng = sh.Range[sh.Cells[1, 5], sh.Cells[2, 5]];

rng.MergeCells = true;

rng.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

rng.Orientation = 90;

// Задаем дополнительные значения и ориентацию текста в ячейках

sh.Cells[2, 6].Value = "усл. эт. га";

sh.Cells[2, 6].Orientation = 90;

sh.Cells[2, 7].Orientation = 90;

sh.Cells[2, 7].Value = "мото-машиночасы";

sh.Cells[1, 8].Value = "Человекочасы";

// Объединяем и форматируем ячейки

rng = sh.Range[sh.Cells[1, 8], sh.Cells[2, 8]];

rng.MergeCells = true;

rng.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

rng.Orientation = 90;

// Дополнительные значения и форматирование

sh.Cells[1, 9].Value = "Начисленная сумма оплаты труда";

rng = sh.Range[sh.Cells[1, 9], sh.Cells[1, 13]];

rng.MergeCells = true;

sh.Cells[2, 9].Value = "основная";

sh.Cells[2, 9].Orientation = 90;

sh.Cells[2, 10].Value = "дополнительная";

sh.Cells[2, 10].Orientation = 90;

sh.Cells[2, 11].Value = "натуральная";

sh.Cells[2, 11].Orientation = 90;

sh.Cells[2, 12].Value = "единоврем., поощерит. и др. выплаты";

sh.Cells[2, 12].Orientation = 90;

sh.Cells[2, 13].Value = "ИТОГО";

sh.Cells[2, 13].Orientation = 90;

// Дополнительные заголовки и форматирование

sh.Cells[1, 14].Value = "Горючее";

rng = sh.Range[sh.Cells[1, 14], sh.Cells[1, 15]];

rng.MergeCells = true;

sh.Cells[2, 14].Value = "по норме";

sh.Cells[2, 14].Orientation = 90;

sh.Cells[2, 15].Value = "фактически";

sh.Cells[2, 15].Orientation = 90;

sh.Cells[1, 16].Value = "Тонно-километры";

sh.Cells[2, 16].Value = "автомобилей";

sh.Cells[2, 16].Orientation = 90;

sh.Cells[2, 17].Value = "тракторов";

sh.Cells[2, 17].Orientation = 90;

// Заполняем данными лист из массива data\_array

int y = 3;

int x;

foreach (object[] t in data\_array)

{

x = 1;

foreach (object q in t)

{

sh.Cells[y, x].Value = q;

sh.Cells[y, x].HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

++x;

}

// Устанавливаем формулу для расчета суммы в строке

sh.Cells[y, 13].Formula = $"=SUM(I{y}:L{y})";

++y;

}

// Задаем стиль границ для диапазона ячеек

Excel.Range reng = sh.Range[sh.Cells[1, 1], sh.Cells[y - 1, 17]];

reng.Borders.LineStyle = Excel.XlLineStyle.xlContinuous;

// Сохраняем книгу Excel

SaveFileDialog sfd = new SaveFileDialog();

if (sfd.ShowDialog() == DialogResult.OK)

{

wb.SaveAs(sfd.FileName);

}

// Закрываем книгу и завершаем приложение Excel

wb.Close();

app.Quit();

}

private void tableLayoutPanel\_Paint(object sender, PaintEventArgs e)

{

// Пустой метод для обработки события Paint таблицы tableLayoutPanel

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

// Переключаем видимость lblInfo при нажатии Ctrl+Alt+D4

if (e.Control && e.Alt && e.KeyCode == Keys.D4)

{

lblInfo.Visible = !lblInfo.Visible;

}

}

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

{

}

}

}

Форма предпросмотра:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

namespace WindowsFormsApp5

{

public partial class frmPrewiev : Form

{

private List<object[]> data\_array;

DataGridView dgPreview;

string[] colHeaders = { "Производственное подразделение", "Наименование культуры" , "Код синтетического и аналитического учета",

"Вид работы", "Наименование работ",

"Объем работ в натуре (га, m^3, тб шеб и др)","Объем работ в условных единицах","усл. эт. га","мото-машиночасы",

"Человекочасы",

"Начисленная сумма оплаты труда","основная","дополнительная","натуральная","единоврем., поощерит. и др. выплаты","ИТОГО",

"Горючее","по норме","фактически",

"Тонно-километры","автомобилей"," тракторов"};

double[] colWidths = { 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1};

public frmPrewiev(List<object[]> \_data\_array)

{

InitializeComponent();

data\_array = \_data\_array;

dgPreview = new DataGridView() { Width = 880 };

int n = colHeaders.Length;

dgPreview.ColumnCount = n;

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

{

dgPreview.Columns[i].HeaderText = colHeaders[i];

dgPreview.Columns[i].Width = (int)(colWidths[i]\*50);

}

Controls.Add(dgPreview);

}

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

{

// Убеждаемся, что data\_array не является нулевым

if (data\_array != null)

{

dgPreview.Rows.Clear();

int y = 0;

int x;

foreach (object[] t in data\_array)

{

dgPreview.Rows.Add();

x = 0;

foreach (object o in t)

{

dgPreview.Rows[y].Cells[x++].Value = o.ToString();

}

++y;

}

}

else

{

MessageBox.Show("Нет данных для отображения. Пожалуйста, сначала загрузите данные.");

}

}

}

}

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

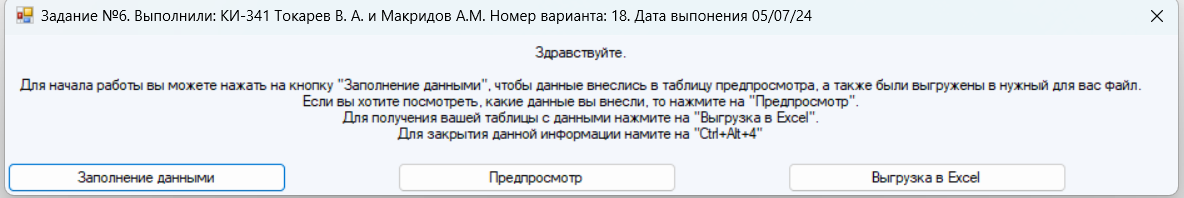
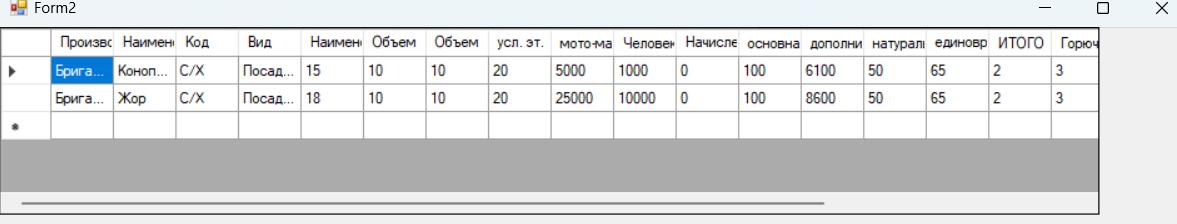


Рисунок - Интерфейс основной формы



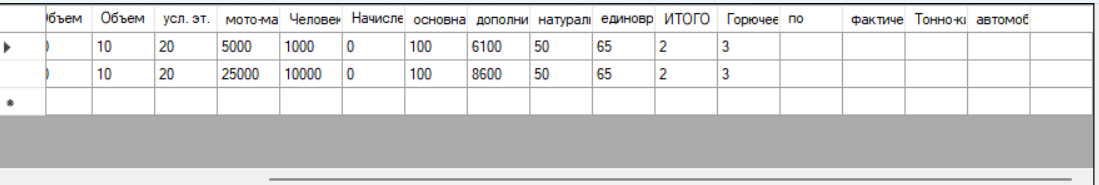


Рисунок - Интерфейс формы предпросмотра

**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 - Дата на момент создания отчета

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

* Проверить генерируемый файл Excel

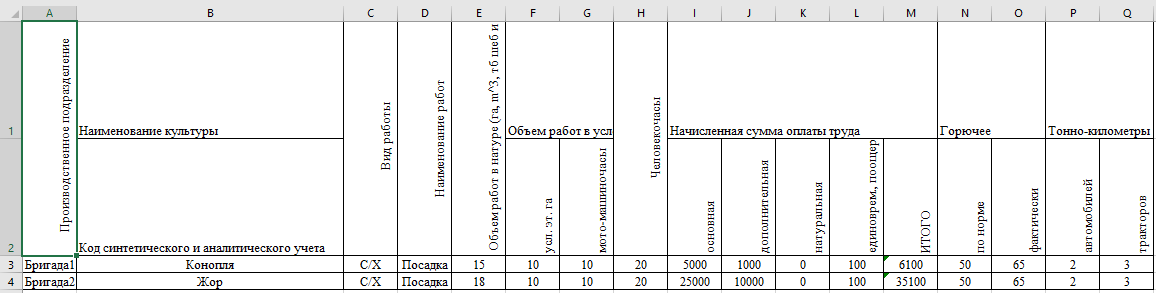


Рисунок - Генерируемый файл Excel

* Проверить предпросмотр

См. Рисунок 7

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

В рамках данной работы были получены навыки работы с электронными таблицами (создание, чтение, форматирование) Microsoft Office Excel с использованием Microsoft Windows Forms C#, а также автоматизации данного процесса и проработкой функций отдельных ячеек.