**МИНИСТЕРСТВО ТРАНСПОРТА**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ**

**АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО**

**ОБРАЗОВАНИЯ**

**“РОССИЙСКИЙ УНИВЕРСИТЕТ**

**ТРАНСПОРТА” (МИИТ)**

**Институт транспортной техники и систем**

**управления**

**(ИТТСУ)**

**Кафедра “Управление и защита информации”**

**Тема: “Лабораторная работа №6”**

**Выполнили студенты группы ТКИ-342**

**Лазухин Максим Сергеевич**

**Макаров Григорий Борисович**

**Специальность 10.05.01 “Компьютерная безопасность”**

**Преподаватель: к.т.н. Сафронов А.И.**

**Москва**

**2024**

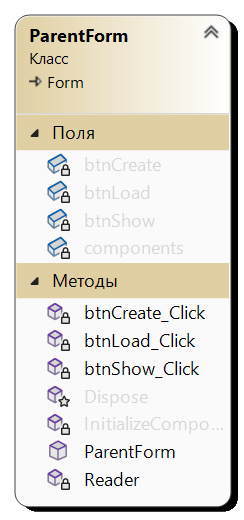
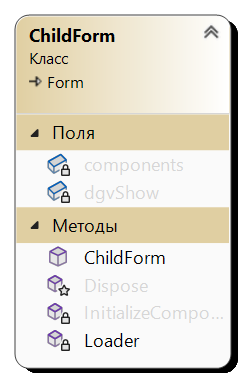
1. Цель работы:

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

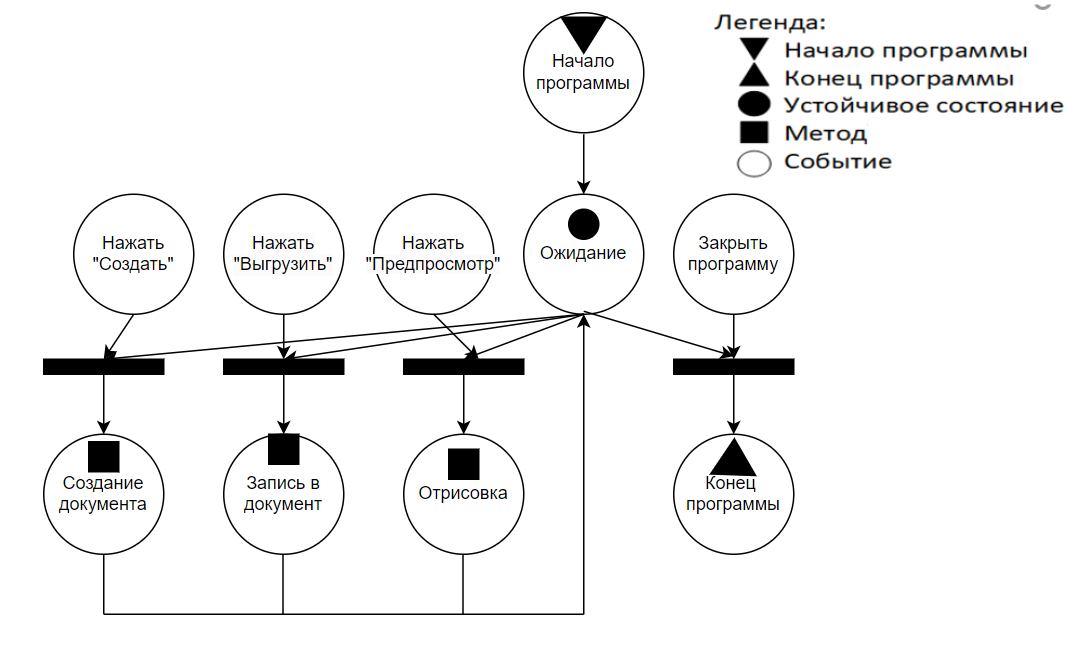
2. Задача:

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

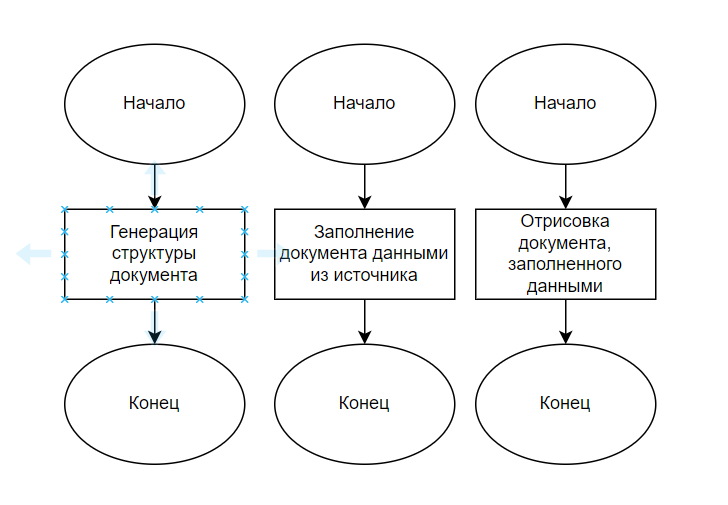
3. Диаграмма классов:

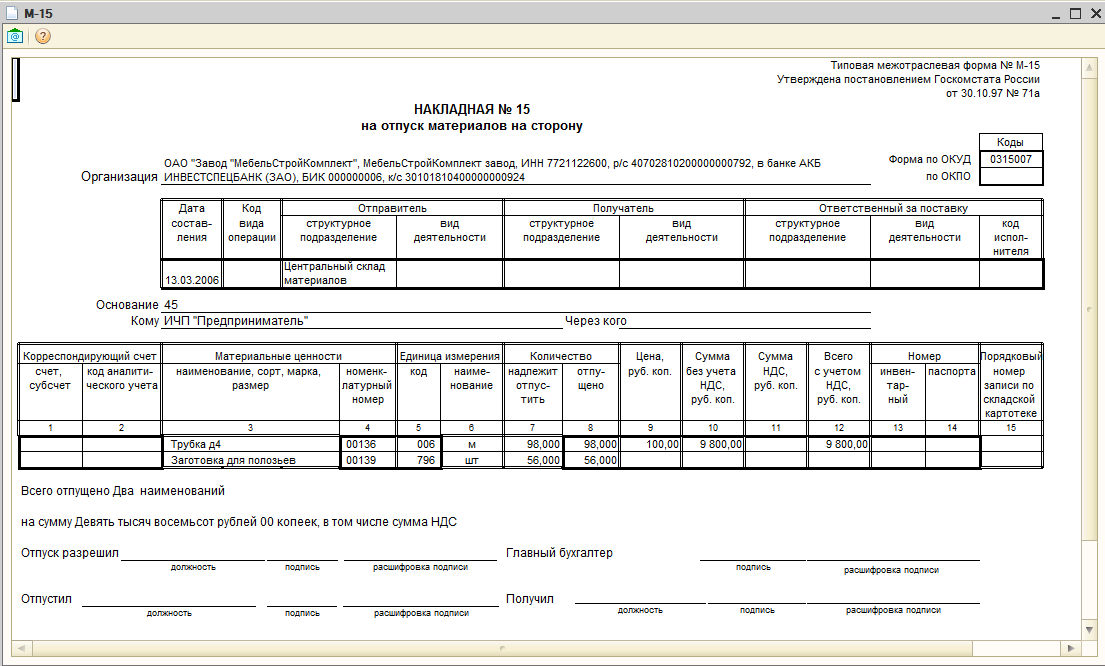
4. Сеть Петри



5. Схемы алгоритмов методов:



6. Подбор тестовых примеров



7. Листинг кода:

using System;

using System.Windows.Forms;

using E = Microsoft.Office.Interop.Excel;

using System.IO;

namespace ExcelLab

{

public partial class ParentForm : Form

{

public ParentForm()

{

InitializeComponent();

}

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

{

E.Application eapp = new E.Application();

eapp.Workbooks.Add();

E.Worksheet worksheet = eapp.Worksheets[1];

worksheet.Range["A1:O5"].WrapText = true;

worksheet.Range["A1:O5"].Font.Size = 8;

worksheet.Range["A1:O3; F4:F5"].HorizontalAlignment = E.XlHAlign.xlHAlignCenter;

worksheet.Range["C4:E5"].HorizontalAlignment = E.XlHAlign.xlHAlignLeft;

worksheet.Range["G4:O5"].HorizontalAlignment = E.XlHAlign.xlHAlignRight;

worksheet.Range["A1:O2"].VerticalAlignment = E.XlHAlign.xlHAlignCenter;

for (int i = 0; i < 15; i++)

{

worksheet.Columns[i+1].ColumnWidth = 12;

}

void Setter()

{

worksheet.Range["A1:O5"].WrapText = true;

worksheet.Range["A1:O5"].Font.Size = 8;

worksheet.Range["A1:O3; F4:F5"].HorizontalAlignment = E.XlHAlign.xlHAlignCenter;

worksheet.Range["C4:E5"].HorizontalAlignment = E.XlHAlign.xlHAlignLeft;

worksheet.Range["G4:O5"].HorizontalAlignment = E.XlHAlign.xlHAlignRight;

worksheet.Range["A1:O2"].VerticalAlignment = E.XlHAlign.xlHAlignCenter;

}

void TripletGenerator(int xcell, int ycell, string toptext, string leftbottom, string rightbottom)

{

E.Range c1 = worksheet.Cells[xcell, ycell];

E.Range c2 = worksheet.Cells[xcell, ycell + 1];

E.Range temprng = (E.Range)worksheet.get\_Range(c1, c2);

temprng.Merge();

temprng.Value = toptext;

temprng = (E.Range)worksheet.Cells[xcell + 1, ycell];

temprng.Value = leftbottom;

temprng = (E.Range)worksheet.Cells[xcell + 1, ycell + 1];

temprng.Value = rightbottom;

c2 = worksheet.Cells[xcell + 4, ycell + 1];

temprng = (E.Range)worksheet.get\_Range(c1, c2);

temprng.Borders[E.XlBordersIndex.xlEdgeBottom].LineStyle = E.XlLineStyle.xlDouble;

temprng.Borders[E.XlBordersIndex.xlEdgeLeft].LineStyle = E.XlLineStyle.xlDouble;

temprng.Borders[E.XlBordersIndex.xlEdgeRight].LineStyle = E.XlLineStyle.xlDouble;

temprng.Borders[E.XlBordersIndex.xlEdgeTop].LineStyle = E.XlLineStyle.xlDouble;

}

void SingleGenerator(int xcell, int ycell, string text)

{

E.Range c1 = worksheet.Cells[xcell, ycell];

E.Range c2 = worksheet.Cells[xcell + 1, ycell];

E.Range temprng = (E.Range)worksheet.get\_Range(c1, c2);

temprng.Merge();

temprng.Value = text;

c2 = worksheet.Cells[xcell + 4, ycell];

temprng = (E.Range)worksheet.get\_Range(c1, c2);

temprng.Borders[E.XlBordersIndex.xlEdgeBottom].LineStyle = E.XlLineStyle.xlDouble;

temprng.Borders[E.XlBordersIndex.xlEdgeLeft].LineStyle = E.XlLineStyle.xlDouble;

temprng.Borders[E.XlBordersIndex.xlEdgeRight].LineStyle = E.XlLineStyle.xlDouble;

temprng.Borders[E.XlBordersIndex.xlEdgeTop].LineStyle = E.XlLineStyle.xlDouble;

}

worksheet.Range["A1:O3"].Borders[E.XlBordersIndex.xlEdgeBottom].LineStyle = E.XlLineStyle.xlDouble;

TripletGenerator(1, 1, "Кореспондирующий счет", "Счет субсчет", "Код аналитического учета");

TripletGenerator(1, 3, "Материальные ценности", "Наименование, сорт, марка, размер ", "Номенклатурный номер");

TripletGenerator(1, 5, "Единица измерения", "код", "наименование");

TripletGenerator(1, 7, "Количество", "надлежит отпустить", "отпущено");

TripletGenerator(1, 13, "Номер", "инвентарный", "паспорта");

SingleGenerator(1, 9, "Цена \n руб. коп.");

SingleGenerator(1, 10, "Сумма. Без учета НДС. \n руб. коп.");

SingleGenerator(1, 11, "Сумма НДС. \n руб. коп.");

SingleGenerator(1, 12, "Всего с учетом НДС. \n руб. коп.");

SingleGenerator(1, 15, "Порядковый номер записи по складской картотеке");

worksheet.Range["A4:B5; D4:E5; H4:N5"].Borders[E.XlBordersIndex.xlEdgeBottom].LineStyle = E.XlLineStyle.xlContinuous;

worksheet.Range["A4:B5; D4:E5; H4:N5"].Borders[E.XlBordersIndex.xlEdgeLeft].LineStyle = E.XlLineStyle.xlContinuous;

worksheet.Range["A4:B5; D4:E5; H4:N5"].Borders[E.XlBordersIndex.xlEdgeRight].LineStyle = E.XlLineStyle.xlContinuous;

worksheet.Range["A4:B5; D4:E5; H4:N5"].Borders[E.XlBordersIndex.xlEdgeTop].LineStyle = E.XlLineStyle.xlContinuous;

eapp.ActiveWorkbook.SaveAs(Application.StartupPath + "\\Test.xlsx");

eapp.Quit();

}

void Reader()

{

E.Application eapp = new E.Application();

eapp.Workbooks.Add();

eapp.Workbooks.Open(Application.StartupPath + "\\Test.xlsx");

E.Worksheet worksheet = eapp.Worksheets[1];

using (StreamReader sr = new StreamReader("info.txt"))

{

for (int i1 = 0; i1 < 3; i1++)

{

string temp = sr.ReadLine();

string[] temp1 = temp.Split(';');

for (int i = 0; i < 15; i++)

{

worksheet.Cells[i1 + 3, i + 1].Value = temp1[i];

}

}

sr.Close();

}

eapp.ActiveWorkbook.Save();

eapp.Quit();

}

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

{

Reader();

}

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

{

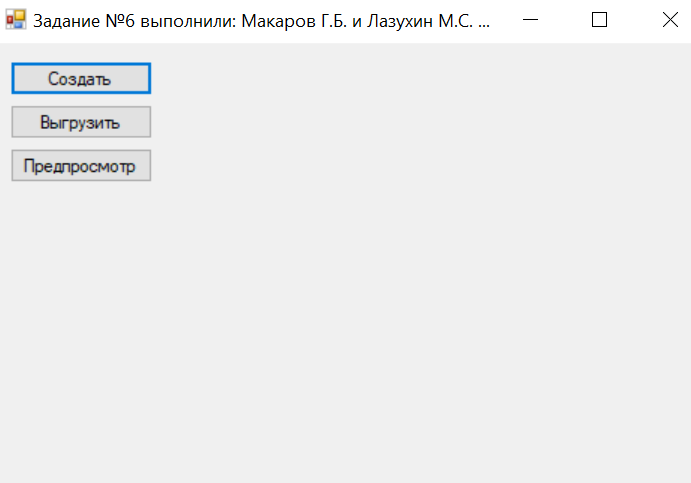
ChildForm Child = new ChildForm();

Child.Show();

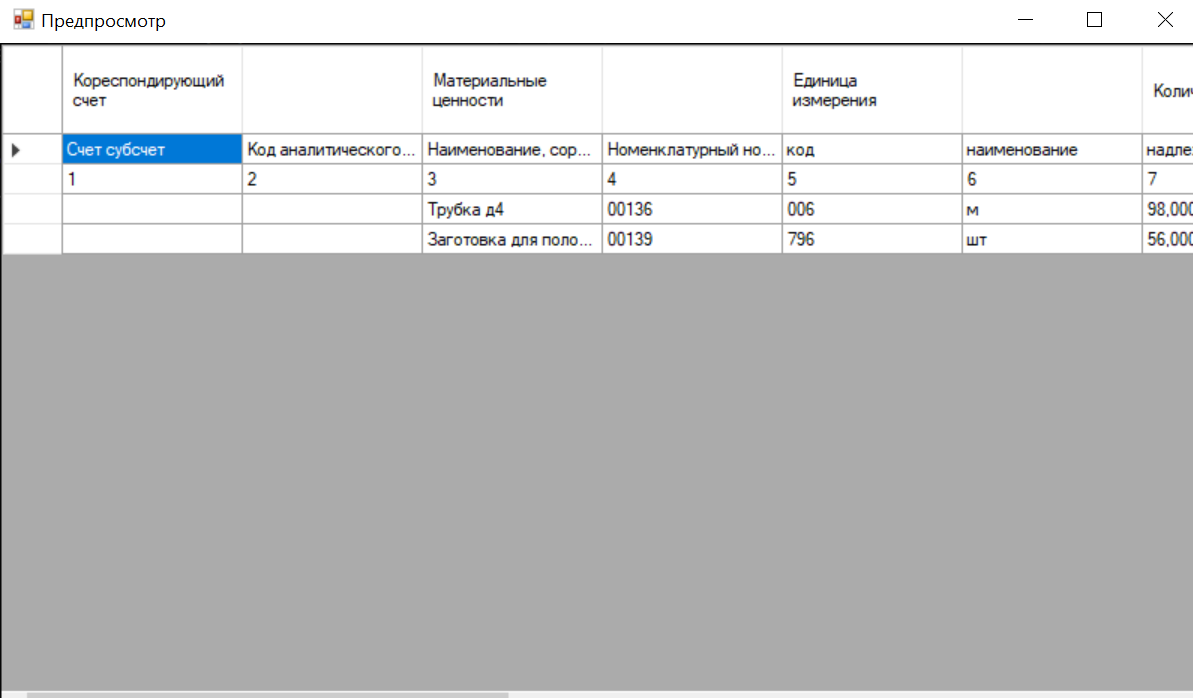
}

}

}

8. Тестирование  
  
Интерфейс  


Предпросмотр



Сгенерированный файл



10. Вывод:

В ходе выполнения задания были освоении навыки подключения внешних модулей и библиотек классов, а также навыки программной настройки параметров электронных таблиц Microsoft Office Excel. Был разработан графический интерфейс с функционалом создания, заполнения и предпросмотра Excel файлов по заданному шаблону.