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

Черкас Н.В., Родин К.П.

Вариант №12

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:

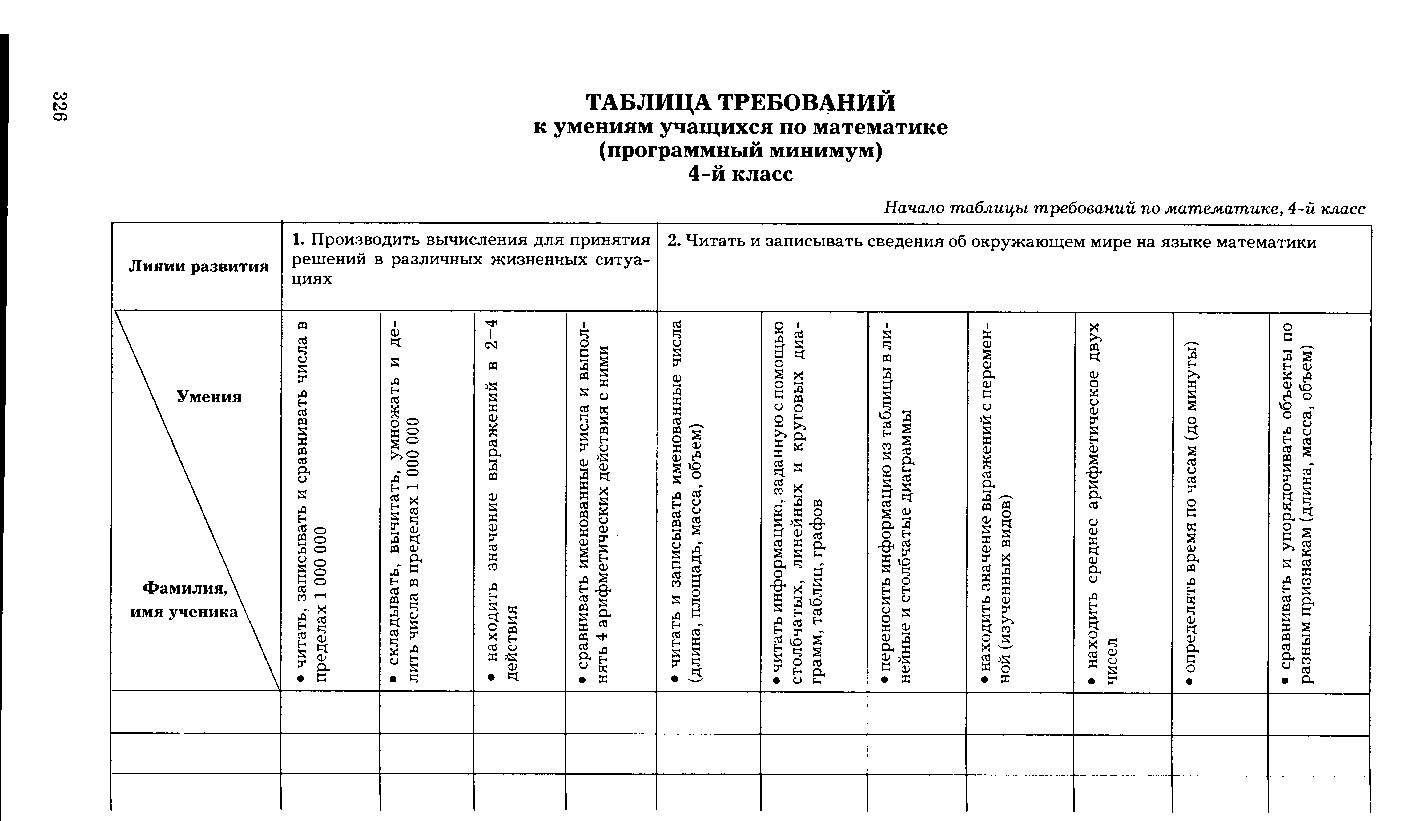


Рисунок 1 - Пример оформления файла

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

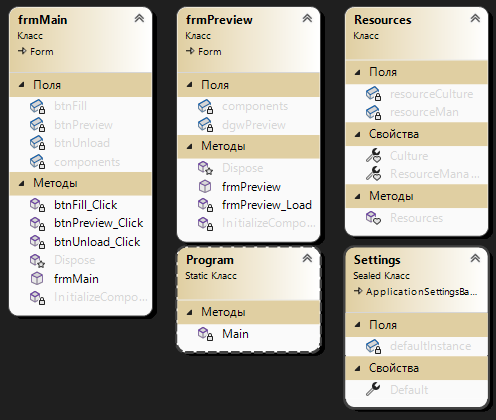


Рисунок 2 - Диаграмма классов

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

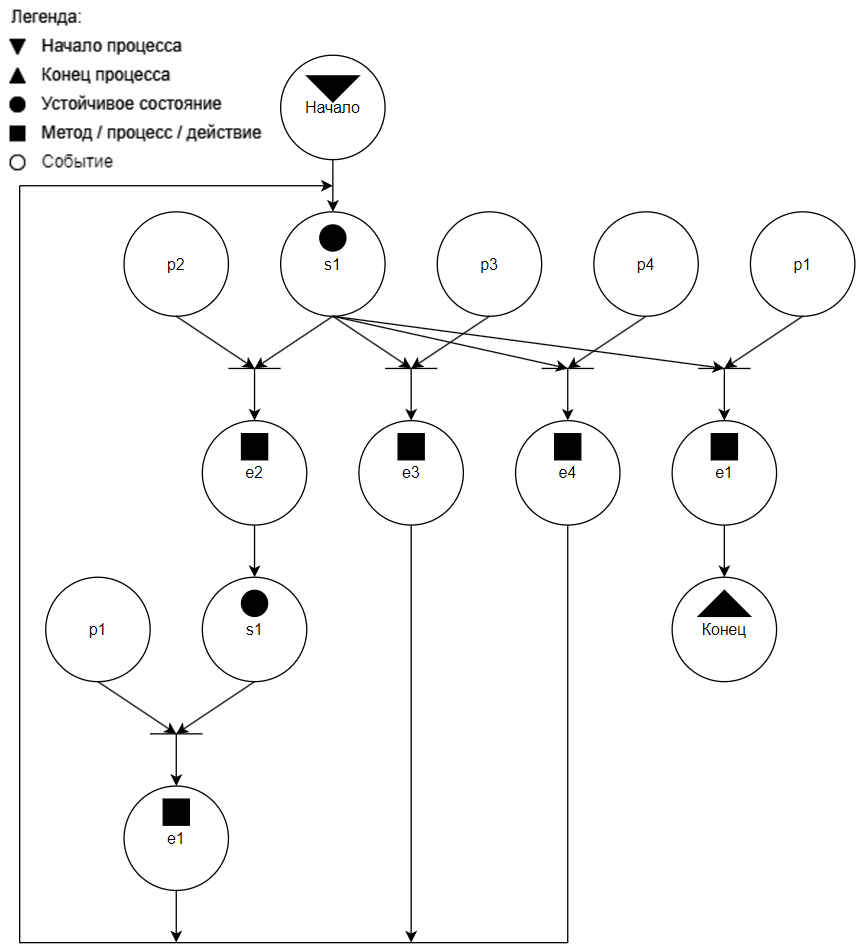


Рисунок 3 - Сеть Петри

**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

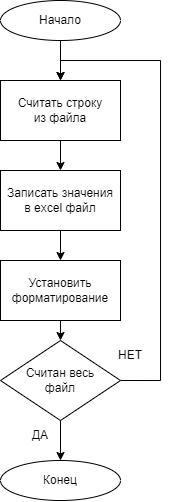


Рисунок 5 - Запись информации из файла

# **6. Подбор тестовых примеров.**

Тестовые примеры:

* Проверить генерируемый файл Excel
* Проверить предпросмотр

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

Основная форма:

using System;

using System.Windows.Forms;

using Excel = Microsoft.Office.Interop.Excel;

using System.IO;

namespace Task6

{

public partial class frmMain : Form

{

public frmMain()

{

InitializeComponent();

}

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

{

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

excelApp.Workbooks.Add();

Excel.Worksheet workSheet = excelApp.Worksheets[1];

workSheet.Range["A1:L13"].WrapText = true;

workSheet.Range["A1:L13"].Font.Name = "Times New Roman";

workSheet.Range["A1:L13"].Font.Size = 12;

workSheet.Range["A1:A13"].Font.Bold = true;

workSheet.Range["A1"].HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

workSheet.Range["A1:F1"].VerticalAlignment = Excel.XlHAlign.xlHAlignCenter;

workSheet.Range["B2:L13"].HorizontalAlignment = Excel.XlHAlign.xlHAlignLeft;

workSheet.Range["A1:L13"].Borders.LineStyle = Excel.XlLineStyle.xlContinuous;

workSheet.Range["A1:L13"].Borders.Weight = Excel.XlBorderWeight.xlThin;

workSheet.Range["A1"].Value = "Линии развития";

workSheet.Columns[1].ColumnWidth = 18f;

workSheet.Range["B1:E1"].Merge();

workSheet.Range["B1:E1"].Value = "1. Производить вычисления для принятия реешений";

workSheet.Range["F1:L1"].Merge();

workSheet.Range["F1:L1"].Value = "2. Читать и записывать сведения об окружающем мире на языке математики";

workSheet.Range["A2:A13"].Merge();

workSheet.Range["A2:A13"].Value = " Умения\n\n\n\n\n\n\n\n\n\nФамилия, имя ученика";

workSheet.Range["A2:A13"].Borders[Excel.XlBordersIndex.xlDiagonalDown].LineStyle = Excel.XlLineStyle.xlContinuous;

workSheet.Range["B2:B13"].Merge();

workSheet.Range["B2:B13"].Value = "• читать, записывать и сравнивать числа в пределах 1 000 000";

workSheet.Range["B2:B13"].Orientation = Excel.XlOrientation.xlUpward;

workSheet.Range["C2:C13"].Merge();

workSheet.Range["C2:C13"].Value = "• складывать, вычитать, умножать и делить числа в пределах 1 000 000";

workSheet.Range["C2:C13"].Orientation = Excel.XlOrientation.xlUpward;

workSheet.Range["D2:D13"].Merge();

workSheet.Range["D2:D13"].Value = "• находить значение выраений в 2-4 действия";

workSheet.Range["D2:D13"].Orientation = Excel.XlOrientation.xlUpward;

workSheet.Range["E2:E13"].Merge();

workSheet.Range["E2:E13"].Value = "• сравнивать именованные числа и выполнять 4 арифметических действия с ними";

workSheet.Range["E2:E13"].Orientation = Excel.XlOrientation.xlUpward;

workSheet.Range["F2:F13"].Merge();

workSheet.Range["F2:F13"].Value = "• читать и записывать именованные числа(длина, площадь, объем)";

workSheet.Range["F2:F13"].Orientation = Excel.XlOrientation.xlUpward;

workSheet.Range["G2:G13"].Merge();

workSheet.Range["G2:G13"].Value = "• читать информацию заданную с помощью столбчатых, линейных и круговых диаграмм, таблиц, графов";

workSheet.Range["G2:G13"].Orientation = Excel.XlOrientation.xlUpward;

workSheet.Range["H2:H13"].Merge();

workSheet.Range["H2:H13"].Value = "• переносить информацию из таблицы в линейные и столбчатые диаграммы";

workSheet.Range["H2:H13"].Orientation = Excel.XlOrientation.xlUpward;

workSheet.Range["I2:I13"].Merge();

workSheet.Range["I2:I13"].Value = "• находить значение выражений с переменной (изученных видов)";

workSheet.Range["I2:I13"].Orientation = Excel.XlOrientation.xlUpward;

workSheet.Range["J2:J13"].Merge();

workSheet.Range["J2:J13"].Value = "• находить среднее арифметическое двух чисел";

workSheet.Range["J2:J13"].Orientation = Excel.XlOrientation.xlUpward;

workSheet.Range["K2:K13"].Merge();

workSheet.Range["K2:K13"].Value = "• определять время по часам (до минуты)";

workSheet.Range["K2:K13"].Orientation = Excel.XlOrientation.xlUpward;

workSheet.Range["L2:L13"].Merge();

workSheet.Range["L2:L13"].Value = "• сравнивать и упорядочивать объекты по разным признакам (длина, масса, объем)";

workSheet.Range["L2:L13"].Orientation = Excel.XlOrientation.xlUpward;

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

excelApp.Quit();

}

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

{

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

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

{

excelApp.Workbooks.Add();

}

else

{

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

}

Excel.Worksheet workSheet = excelApp.Worksheets[1];

string[] lines = File.ReadAllLines(Application.StartupPath + "\\DocInfo.txt");

int row = 14;

foreach (string line in lines)

{

string[] data = line.Split(';');

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

{

workSheet.Cells[row,i + 1].Value = data[i];

workSheet.Cells[row, i + 1].WrapText = true;

workSheet.Cells[row, i + 1].Font.Name = "Times New Roman";

workSheet.Cells[row, i + 1].Font.Size = 12;

workSheet.Cells[row, i + 1].Borders.Weight = Excel.XlBorderWeight.xlThin;

}

row++;

}

excelApp.ActiveWorkbook.Save();

excelApp.Quit();

}

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

{

frmPreview preview = new frmPreview();

preview.Show();

}

}

}

Форма предпросмотра:

using System;

using System.Windows.Forms;

using System.IO;

namespace Task6

{

public partial class frmPreview : Form

{

public frmPreview()

{

InitializeComponent();

}

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

{

this.WindowState = FormWindowState.Maximized;

this.dgwPreview.ColumnCount = 12;

this.dgwPreview.Columns[0].HeaderText = "Линии развития";

this.dgwPreview.Columns[0].Width = 200;

this.dgwPreview.Columns[1].HeaderText = "1. Производить вычисления для принятия реешений";

this.dgwPreview.Columns[1].Width = 250;

this.dgwPreview.Columns[2].Width = 250;

this.dgwPreview.Columns[3].Width = 250;

this.dgwPreview.Columns[4].Width = 250;

this.dgwPreview.Columns[5].Width = 250;

this.dgwPreview.Columns[5].HeaderText = "2. Читать и записывать сведения об окружающем мире на языке математики";

this.dgwPreview.Columns[6].Width = 250;

this.dgwPreview.Columns[7].Width = 250;

this.dgwPreview.Columns[8].Width = 200;

this.dgwPreview.Columns[9].Width = 270;

this.dgwPreview.Columns[10].Width = 250;

this.dgwPreview.Columns[11].Width = 250;

dgwPreview.DefaultCellStyle.WrapMode = DataGridViewTriState.True;

dgwPreview.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;

dgwPreview.AutoResizeRows(DataGridViewAutoSizeRowsMode.AllCells);

this.dgwPreview.Rows.Add("Умения/Фамилия, имя ученика", "• читать, записывать и сравнивать числа в пределах 1 000 000", "• складывать, вычитать, умножать и делить числа в пределах 1 000 000", "• находить значение выраений в 2-4 действия", "• сравнивать именованные числа и выполнять 4 арифметических действия с ними", "• читать и записывать именованные числа(длина, площадь, объем)", "• читать информацию заданную с помощью столбчатых, линейных и круговых диаграмм, таблиц, графов", "• переносить информацию из таблицы в линейные и столбчатые диаграммы","• находить значение выражений с переменной(изученных видов)", "• находить среднее арифметическое двух чисел", "• определять время по часам (до минуты)", "• сравнивать и упорядочивать объекты по разным признакам (длина, масса, объем)");

int height = 0;

foreach (DataGridViewRow row in dgwPreview.Rows)

{

height += row.Height;

}

height += dgwPreview.ColumnHeadersHeight;

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

string[] lines = File.ReadAllLines(Application.StartupPath + "\\DocInfo.txt");

foreach (string line in lines)

{

dgwPreview.Rows.Add(line.Split(';'));

}

sr.Close();

}

}

}

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

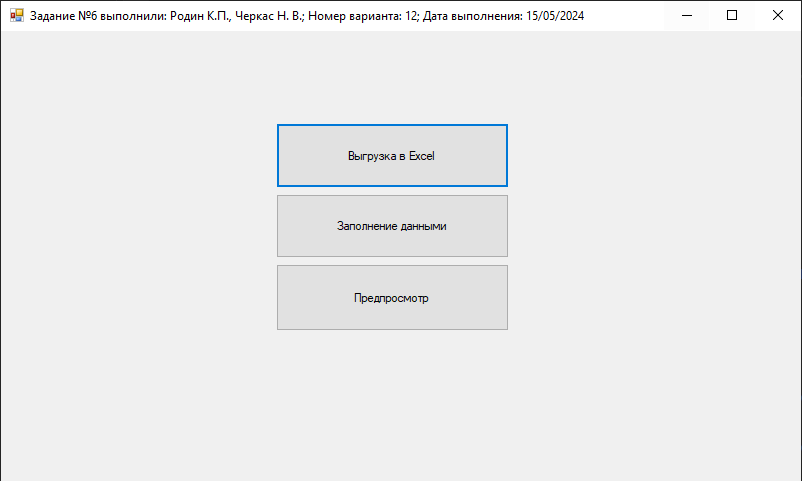


Рисунок 6 - Интерфейс основной формы

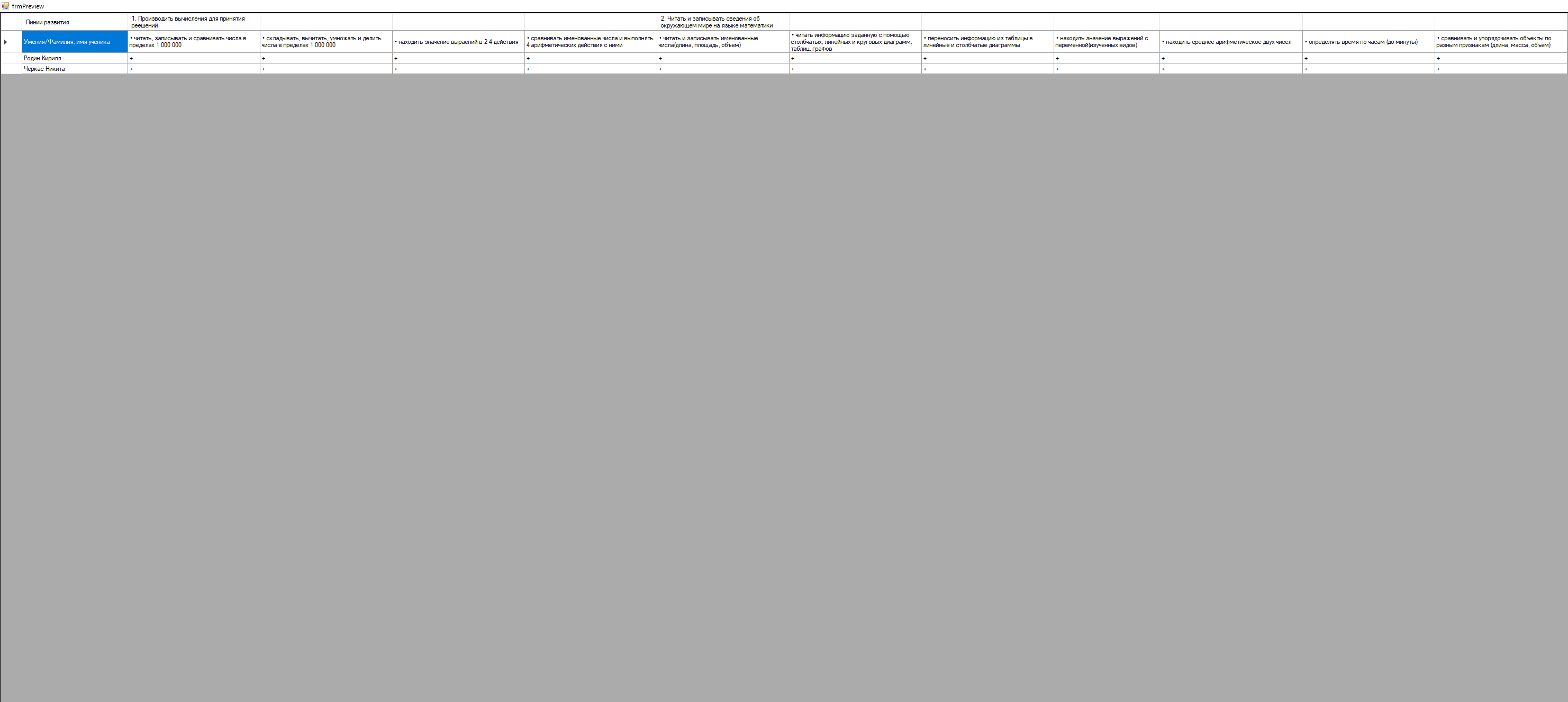


Рисунок 7 - Интерфейс формы предпросмотра

**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

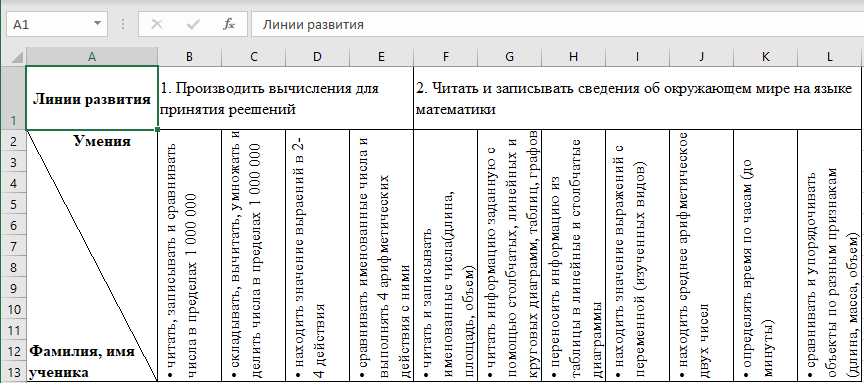


Рисунок 9 - Генерируемый файл Excel

* Проверить предпросмотр

См. Рисунок 7

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

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