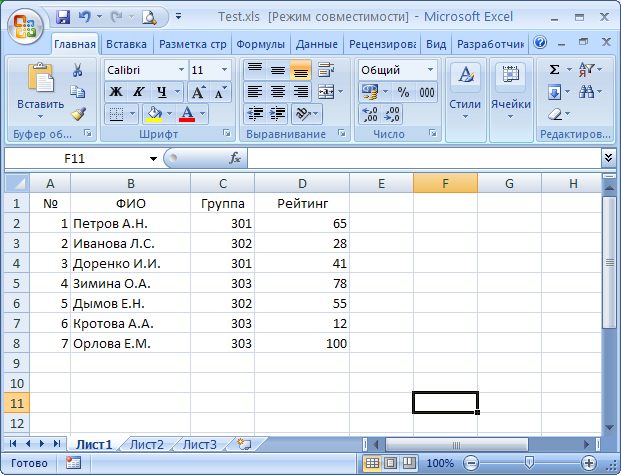
Лабораторная работа 19

**Автоматизация Microsoft Office.**

Рассмотрим пример автоматизации.

Пусть имеется файл «TEST.XLS», содержащий одну таблицу следующего вида:



Количество студентов и групп заранее неизвестно.

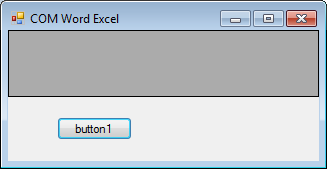
Требуется перенести данные в документ WORD, попутно определив число студентов. В конце документа вывести диаграмму успеваемости по всем студентам.

Реализация будет выполняться с использованием COM технологии.

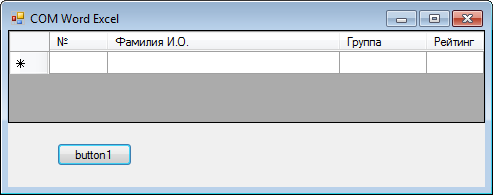
Параллельно, основная форма будет отображать данные, которые будут считаны из файла EXCEL. Этот функционал добавлен исключительно для наглядности. Можно обойтись и другими средствами.

**Шаг 0**.Организуйте, как было показано на лекции, возможность использовать в Вашем проекте функционал COM серверов Word и Excel.

**Шаг 1**. Создадим форму, содержащую кнопку **button1** и компонент для хранения и отображения таблицы **dataGridView1** (в принципе, необязательно) и диалог открытия файла **openDialog1**:



**Шаг 2**. Настроим свойства компонентов. Так, настроим столбцы (свойство **Columns**) компонента **dataGridView1** (4 столбца разной ширины с понятными заголовками).



У компонента **openDialog1** настроим фильтр для файлов **Filter =’ Файлы Excel |\*.xl\*’** (только EXCEL-файлы).

**Шаг 3**. Вся основная работа выполняется в обработчике события **Click** компонента **button1**.

Во-первых, отметим назначение вспомогательных переменных.

E – приложение EXCEL.

Sh – (Sheet) лист в книге EXCEL.

Сh – (Chart) диаграмма EXCEL.

W – приложение WORD.

D – (Document) документ WORD.

t – текущее выделение (место курсора) WORD.

Начинаем с диалога выбора файла с данными активизации сервера EXCEL:

if (openFileDialog1.ShowDialog() != DialogResult.OK) return;

Затем подключаем сервер EXCEL и открываем интересующий наc файл:

var E = new Excel.Application();

E.Visible = true;

E.Workbooks.Open(openFileDialog1.FileName);

Позиционируемся на текущий (первый) лист EXCEL и даём ему имя COM (необязательно):

var Sh = E.ActiveSheet;

Sh.Name = "COM";

Создаём массив строк arr для представления 4-х элементов отдельной строки таблицы и готовим компонент **dataGridView1**:

string[] arr = new string[4];

dataGridView1.ColumnCount = 4;

dataGridView1.Rows.Clear();

Перебираем имеющиеся ячейки с данными на текущем EXCEL листе **Sh.Cells[<строка>,<столбец>]** и переносим их как текст (**.Text** или, если бы нам нужны были данные не строкового, а реального типа, то **.Value**) в таблицу **dataGridView1**:

int r = 0;

do

{

r++;

arr[0] = Sh.Cells[r + 1, 1].Text;

if (arr[0] == "") break;

arr[1] = Sh.Cells[r + 1, 2].Text;

arr[2] = Sh.Cells[r + 1, 3].Text;

arr[3] = Sh.Cells[r + 1, 4].Text;

dataGridView1.Rows.Add(arr[0], arr[1], arr[2], arr[3]);

} while (true);

r--;

Добавляем диаграмму в EXCEL. Она будет создана на отдельном листе, и мы меняем её расположение (.Location) на лист с именем 'COM'.

Т.к. старая ссылка после переноса диаграммы становится неактуальной, мы запоминаем ссылку на текущую (только что перенесённую) диаграмму.

var Ch = E.Charts.Add();

Ch.Location(Excel.XlChartLocation.xlLocationAsObject, "COM");

Ch = E.ActiveChart;

Т.к. диаграмма создана автоматически, то в неё попали все данные из таблицы. Удалим серию, соответствующую номеру группы:

Ch.SeriesCollection(1).Delete();

Для тренировки выполним дополнительное оформление диаграммы (смысл понятен по идентификаторам используемых свойств):

Ch.HasTitle = 1;

Ch.HasLegend = false;

Ch.ChartTitle.Text = "Успеваемость студентов";

Ch.Axes(1).HasTitle = true;

Ch.Axes(1).AxisTitle.Text = "Студенты";

Ch.Axes(2).HasTitle = true;

Ch.Axes(2).AxisTitle.Text = "Рейтинг";

Документ EXCEL подготовлен. Переходим к формированию документа WORD. Начинаем с активизации сервера WORD:

var W = new Word.Application();

W.Visible = false;

Создаём новый документа WORD, запоминая в переменной D ссылку на соответствующий ему объект, а в переменной t – ссылку на объект, отвечающий за текущее выделение в документе (если нет выделения, то это – позиция курсора).

var D = W.Documents.Add();

var t = W.Selection;

В пустом документе набираем текст заголовка, используя другой размер и вид шрифта:

t.Font.Bold = -1;

t.Font.Size = 24;

t.TypeText("Данные о студентах.");

t.TypeParagraph();

Опять меняем шрифт и копируем данные из **Excel**:

t.Font.Bold = 0;

t.Font.Size = 14;

for (int k = 1; k <= r; k++)

{

t.TypeText(k.ToString() + " : ");

t.TypeText(Sh.Cells[k + 1, 2].Value.ToString() + " ");

t.TypeText(Sh.Cells[k + 1, 3].Text + " ");

t.TypeText(Sh.Cells[k+1, 4].Value.ToString());

t.TypeParagraph();

}

Ниже выводим информацию о числе студентов:

t.TypeParagraph();

t.TypeText("Всего " + r.ToString() + " студентов.");

t.TypeParagraph();

А теперь, копируем текущую диаграмму из EXCEL в WORD через буфер обмена:

Ch.ChartArea.Select();

Ch.ChartArea.Copy();

t.TypeParagraph();

t.Paste();

Документ WORD готов. Сохраняем его:

D.SaveAs(Path.GetDirectoryName(openFileDialog1.FileName)+"\\Test");

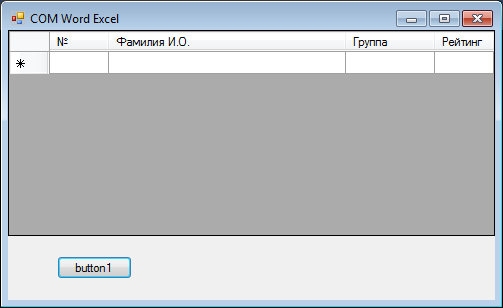
Наконец, освобождаем оба сервера (т.к. книга EXCEL изменилась сервер попросит сохранить её):

W.Quit(); W = null;

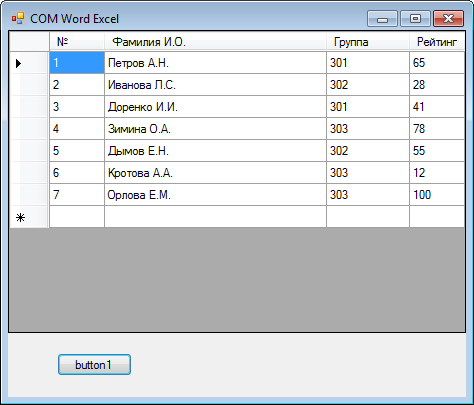
E.Quit(); E = null;

}

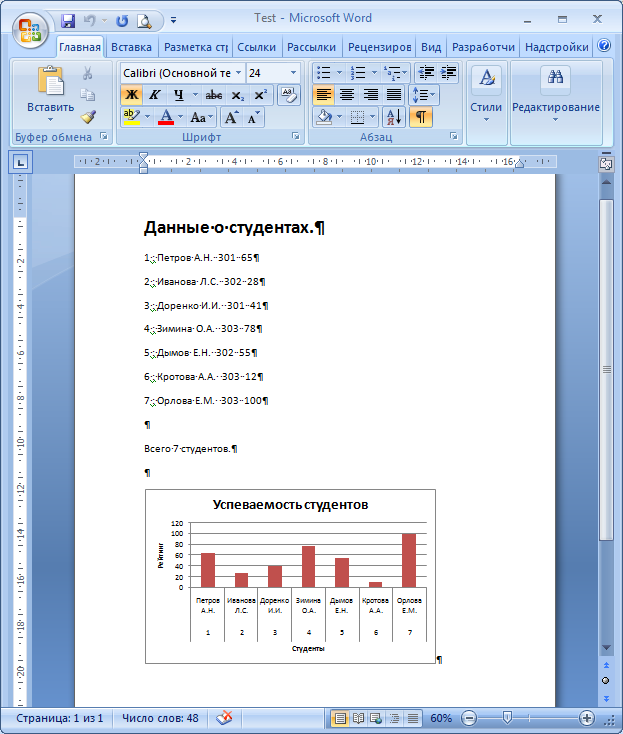
Программу можно запустить. Вначале будет просто пустая таблица:



После выбора кнопки button1 и выбора файла данные появятся на форме:



А в файле «Test.docx» (использовался MS Office 2007) будет записана исходная информация, результаты обработки и таблица:



В классе повторить этот проект (1 балл).

Самостоятельно:

Простое задание (на 3 балла)

1. Расширить список полей исходной таблицы, добавив краткое наименование специальности.
2. При выводе исходных данных в Word, сгруппировать их по специальностям.
3. Найти средний рейтинг по каждой из групп и вывести его в виде таблицы.
4. Сделать дополнительно круговую диаграмму «Численность групп».

Сложное задание (на 7 баллов - много работы)

1. Выполнить свой вариант из файла «Задания COM».

Вам, возможно помогут:

1. Книга **Корняков В.Н. - Программирование документов и приложений MS Office в Delphi**

2. Статья "**Работаем с MS Word из C#"** (и её продолжения)

http://nullpro.info/2012/rabotaem-s-ms-word-iz-c-chast-0-klass-i-testovyj-proekt-primer-winforms/

3. Статья "**Работа с серверами автоматизации Word и Excel в Visual Studio .Net**"

http://wladm.narod.ru/C\_Sharp/componentbegin.html

4. Google