Документация к коду примера.

Содержание

1. Общие слова о взаимодействии

Основным классом, отвечающим за представление (GUI-часть) диалоговых окон приложения, является ConverterWindow. Содержит следующие компоненты:

- кнопка «LoadOldWord»;
- флажок (check-box) «extractToExcel» напротив кнопки «LoadOldWord»;
- кнопка «LoadNewWord»;
- флажок (check-box) «extractToExcel» напротив кнопки «LoadNewWord»;
- кнопка «CompareAndExport»;
- кнопка «Clean»;
- кнопка «LoadPDBExcel»;
- кнопка «CompareWithPDBAndExport».

Класс ConverterWindow отвечает лишь только за интерфейсную часть приложения, вся «бизнес-логика» целиком и полностью описана в классе Handler. Соответственно, нажатие на какую-либо из кнопок класса ConverterWindow вызывает нужный метод класса Handler.

2. Класс ConverterWindow

Класс, отвечающий за представление (графический интерфейс). Основная суть – наличие кнопок, нажатие на которые привязывается к определенному методу внутренней переменной handler_.

Приватные поля класса (внутренние переменные).

- *handler_* – объект класса Handler, отвечающий за всю «бизнес-логику»;

Методы и функции.

- Публичный конструктор класса (метод “New”) – что нужно сделать при создании объекта *ConverterWindow* (то есть при открытии программы).
- Публичный метод *onLoadOldWord* – что нужно сделать при нажатии на кнопку «LoadOldWord».
- Публичный метод *onLoadNewWord* – что нужно сделать при нажатии на кнопку «LoadNewWord».
- Публичный метод *onCompareAndExport* – что нужно сделать при нажатии на кнопку «CompareAndExport».
- Приватный метод *exportToExcel* – выгрузить обработанную информацию, которая хранится в «старом» и «новом» docx-документах в excel-файл.
- Публичный метод *onClean* – что нужно сделать при нажатии на кнопку «Clean».
- Публичный метод *onLoadPDBExcel* – что нужно сделать при нажатии на кнопку «LoadPDBExcel».
- Публичный метод *onCompareWithPDBAndExport* – что нужно сделать при нажатии на кнопку «CompareWithPDBAndExport».
- Приватный метод *tryToEnableComparePDBButton* – попытаться сделать кнопку «CompareWithPDB» активной (изначально она неактивна).
- Приватная функция *getOpenFileName* – открыть диалоговое окно ОС Windows, чтобы пользователь мог выбрать, какой файл открыть. Параметры: *extension* – расширение, файлы с которым будут показаны в диалоговом окне по умолчанию.

3. Класс Handler

Класс, включающий в себя основную бизнес-логику. Зачитываете docx-файлов, их обработка, выгрузка результатов в excel.

Зависимости.

Для корректной работы всех методов необходимо при помощи директивы «Imports» включить следующие зависимости:

- DocumentFormat.OpenXml.Packaging;
- DocumentFormat.OpenXml.Wordprocessing;
- Microsoft.Office.Interop.Excel;

Приватные поля класса (внутренние переменные).

- *pathToOldWord_* - строковая переменная, хранящая в себе путь к «старому» docx-файла;
- *pathToNewWord_* - строковая переменная, хранящая в себе путь к «новому» docx-файлу;
- *dataOldDoc_* - словарь, хранящий в себе все нужные наименования из «старого» docx-документа и соответствующие им номера ревизий;
- *dataNewDoc_* - словарь, хранящий в себе все нужные наименования из «нового» docx-документа и соответствующие им номера ревизий;
- *dataOldHandled_* - хэш-таблица, хранящая в себе пары «наименование - уникальное значение ревизии без повторений» из «старого» docx-документа;
- *dataNewHandled_* - хэш-таблица, хранящая в себе пары «наименование - уникальное значение ревизии без повторений» из «нового» docx-документа;
- *mistakedWithDiffRevInNew_* - словарь, включающий в себя те наименования, ревизии которых в рамках одного документа встречаются несколько раз и с разными значениями (такого не должно быть); соответствует «новому» документу;
- *mistakedWithDiffRevInOld_* - словарь, включающий в себя те наименования, ревизии которых в рамках одного документа встречаются несколько раз и с разными значениями (такого не должно быть); соответствует «старому» документу;
- *names_* - выходной словарь для записи результатов;
- *pathToPDBExcel_* - строковая переменная, хранящая в себе путь к excel-файлу “PDB”;
- *pdbExcelData_* - словарь, в который записываются данные из excel-файла “PDB”;

- *inPDB_* - список тех обработанных «бизнес-логикой» наименований, которые находятся в excel-файле “PDB”;
- *notInPDB_* - список тех обработанных «бизнес-логикой» наименований, которые не находятся в excel-файле “PDB”.

Методы и функции.

- Публичный метод *setPathToOldWord* - задать значение переменой *pathToOldWord_*. Параметры: *path* - путь к «старому» docx-файлу.
- Публичный метод *setPathToNewWord* - задать значение переменой *pathToNewWord_*. Параметры: *path* - путь к «новому» docx-файлу.
- Публичный метод *parseOldWord* - зачитать «старый» docx-документ. Две операции: заполнить переменную *dataOldDoc_* при помощи метода *parseWord* и отыскать ошибочные в документе наименования, поместив их в *mistakedWithDiffRevInOld_*, с помощью метода *handleMistakedInOneDoc*.
- Публичный метод *parseNewWord* - зачитать «новый» docx-документ. Две операции: заполнить переменную *dataNewDoc_* при помощи метода *parseWord* и отыскать ошибочные в документе наименования, поместив их в *mistakedWithDiffRevInNew_*, с помощью метода *handleMistakedInOneDoc*.
- Публичный метод *exportOldWordData* - выгрузить нужные наименования и ревизии из «старого» документа в excel-файл.
- Публичный метод *exportNewWordData* - выгрузить нужные наименования и ревизии из «нового» документа в excel-файл.
- Приватная функция *exportWordData* - выгрузить данные в excel-файл. Параметры: *data* - словарь, содержимое которого выгружается, *path* - место в файловой системе, где нужно сохранить файл excel.
- Приватный метод *parseWord* - зачитать входной docx-файл и заполнить необходимыми данными подданный на вход контейнер. Параметры: *pathToWord* - путь к docx-документу в файловой системе, *dataToFill* - словарь, который необходимо заполнить.
- Приватная функция *isTableIsValid* - ответить на вопрос, является ли рассматриваемая (переданная на вход) таблица «валидной», то есть содержит ли в себе искомые данные. Параметры: *table* - рассматриваемая таблица.
- Приватная функция *getInfoFromTable* - зачитать информацию с «валидной» таблицы (переданной на вход) и поместить ее в необходимый контейнер (переданный на вход). Параметры: *table* - рассматриваемая «валидная» таблица, *dataToFill* - словарь, который необходимо заполнить определенными данными с таблицы.
- Приватный метод *handleMistakedInOneDoc* - отыскать «ошибочные» наименования в рамках одного docx-документа и поместить их в необходимый (переданный на вход) контейнер. Параметры: *dataDoc* - словарь

с вычлененными из docx-документа данными, *withDiffRevInDoc* - словарь, в который необходимо поместить «ошибочные» наименования.

- Приватная функция *nominationIsValidInDoc* - ответить на вопрос, является ли рассматриваемое наименование «валидным» (то есть не «ошибочным») в рамках своего docx-документа. Параметры: *revisions* - список строковых значений, содержащий в себе все номера ревизий данного наименования в пределах своего docx-документа.
- Публичный метод *compareHashTables* - сравнить хэш-таблицы, соответствующие «старому» и «новому» документам, классифицировав информацию в переменной *names_*.
- Приватный метод *findDeletedNominations* - отыскать те наименования, которые были в «старом» документе, но отсутствуют в «новом».
- Публичный метод *exportDataToExcel* - выгрузить результаты сравнения «старого» и «нового» документов в excel-файл. Параметры: *path* - путь, куда нужно сохранить выходной файл.
- Приватный метод *exportNotChanged* - выгрузить те наименования, которые имеют одинаковые значения ревизий в «старом» и «новом» документах (то есть не изменились). Параметры: *workbook* - excel-книга, в которую надо сохранить результат.
- Приватный метод *exportChanged* - выгрузить те наименования, ревизия которых в «новом» документе увеличилась ровно на 1 по сравнению со «старым». Параметры: *workbook* - excel-книга, в которую надо сохранить результат.
- Приватный метод *exportNew* - выгрузить те наименования, которые имеются только в «новом» документе, а в «старом» отсутствуют. Параметры: *workbook* - excel-книга, в которую надо сохранить результат.
- Приватный метод *exportDeleted* - выгрузить те наименование, которые имеются только в «старом» документе, а в «новом» отсутствуют. Параметры: *workbook* - excel-книга, в которую надо сохранить результат.
- Приватный метод *exportMistaked* - выгрузить те наименования, ревизия которых в «новом» документе отличается от значения ревизии в «старом» больше, чем на 1 или -1. Параметры: *workbook* - excel-книга, в которую надо сохранить результат.
- Приватный метод *exportMistakedInNewDoc* - выгрузить те наименования, ревизия которых в рамках «нового» документа повторяется несколько раз, но при этом имеет разные значения. Параметры: *workbook* - excel-книга, в которую надо сохранить результат.
- Публичный метод *setPathToPDBExcel* - установить значение переменной *pathToPDBExcel_*. Параметры: *path* - путь к excel-файлу PDB.

- Публичный метод *parsePDBExcel* - зачитать содержание excel-файла PDB и сохранить результат в переменную *pdbExcelData_*.
- Публичный метод *compareHandledDataWithPDB* - сравнить наименования, которые есть в excel-файле PDB с теми, которые получены в результате обработки docx-файлов «бизнес-логикой». В результате работы метода заполняется две коллекции: *inPDB_* (те наименования, что присутствуют в excel-файле PDB и в результатах обработки docx-файлов) и *notInPDB_* (те наименования, что отсутствуют в excel-файле PDB).
- Публичный метод *exportComparisonPDBToExcel* - выгрузить результаты сравнения excel-PDB файла и «бизнес-логики» в выходной excel-файл. Параметры: *path* - путь, куда сохранить выходной файл.
- Публичный метод *clearData* - почистить все внутренние контейнеры.

Приложение

Публичный конструктор класса ConverterWindow.

Вызывается «внутренний» метод InitializeComponent() для инициализации компонентов GUI, после которого кнопки/флажки появляются на диалоговом окне. Инициализируется внутренняя переменная *handler_* (handler_ = New Handler()).

Приватный метод onLoadOldWord.

1) Переменная *pathToOldWord* приобретает значение с помощью вызова функции *getOpenFileName*.

2) Если пользователь не задал никакое значение (не выбрал ни один из файлов), то метод прерывается (строчка *Return*).

3) Вызывается метод *setPathToOldWord* внутренней переменной *handler_*.

4) В блоке Try/Catch происходит попытка зачитать содержимое «старого» docx-документа и выгрузки (если нужно) в рядом созданный excel.

Приватный метод onLoadNewWord.

Полная аналогия с методом onLoadOldWord, но тут пользователь должен указать правильный путь к «новому» docx-документу (в случае ошибки ответственность за результат ложится на его плечи).

Приватный метод onCompareAndExport.

В блоке Try/Catch происходит попытка при помощи handler_ сравнить содержимое двух только что зачитанных docx-документов.

Приватный метод exportToExcel.

1) Открывается диалоговое окно, пользователь выбирает, где ему сохранить excel-файл.

2) В блоке Try/Catch происходит попытка выгрузки.

Приватный метод onClean.

Зачищаются значения переменные label, в которых до этого отображались пути к загруженным «старому» и «новому» docx-документам. Переменная *handler_* инициализируется заново для обновления данных.

Приватный метод onLoadPDBExcel.

1) Открывается диалоговое окно, пользователь выбирает, какой excel-файл загрузить.

2) В блоке Try/Catch происходит попытка зачитать содержимое PDB excel-файла при помощи метода *parsePDBExcel* переменной *handler_*.

Приватный метод onCompareWithPDBAndExport.

В блоке Try/Catch происходит попытка сравнения содержимого PDB excel-файла с информацией, обработанной «бизнес-логикой» и хранимой

ранее в «старом» и «новом» docx-документах, а также выгрузка результатов в выходной excel-файл.

Приватная функция getOpenFileName.

В функции происходит системный вызов диалогового окна ОС Windows, при взаимодействии с которым пользователь выбирает, какой файл ему открыть.

Приватная функция exportWordData.

Функция экспортирует данные словаря в книгу Excel. Принимает два параметра: «*data*» — словарь, содержащий строковый ключ и список строковых значений, и «*path*» — путь к файлу для сохранения книги Excel.

Создается экземпляр класса Microsoft.Office.Interop.Excel.Application и добавляется новая книга. Затем выбирается первый рабочий лист в рабочей книге и записываются заголовки столбцов «NAME» и «REV.» в первой строке.

Затем функция перебирает каждую пару ключ-значение в словаре *data* и извлекает ключ и значение. Далее перебирается каждое значение в списке и записывается ключ и значение в ячейки рабочего листа, увеличивая номер строки для каждого значения.

Наконец, функция задает имя листа, сохраняет книгу по указанному пути к файлу, закрывает книгу и закрывает приложение Excel.

Приватный метод parseWord.

Метод извлекает данные из таблиц в документе Microsoft Word и заполняет словарь извлеченными данными. Два параметра: «*pathToWord*», который представляет собой путь к файлу документа Word для анализа, и «*dataToFill*», который представляет собой словарь для заполнения извлеченными данными.

Функция использует класс WordprocessingDocument из Open XML SDK, чтобы открыть документ Word и получить доступ к его содержимому. Затем перебираются все таблицы в основном теле документа и, если таблица «валидна» (определяется функцией *isTableIsValid*), извлекает данные из таблицы с помощью функции *getInfoFromTable* и добавляет их в словарь *dataToFill*.

Приватная функция isTableIsValid.

Функция принимает WordTable в качестве входного параметра и возвращает логическое значение. Функция проверяет правильность таблицы, исследуя содержимое ее первой строки. Если первая строка не соответствует определенным условиям, функция возвращает False.

Сначала извлекаются все строки в таблице с помощью метода Elements(Of T), который возвращает набор элементов типа T. Затем выбирается первая строка с помощью метода First. Далее проверяется содержимое

последней ячейки первой строки, чтобы увидеть, содержит ли оно слово «Weight» или «(kg)». Если это не так, функция возвращает False.

Затем функция проверяет содержимое второй ячейки первой строки, чтобы увидеть, содержит ли оно какое-либо из следующих слов: «Workpack», «Block» или «Assembly». Если это так, функция возвращает False.

Наконец, функция проверяет содержимое второй ячейки первой строки, чтобы увидеть, содержит ли оно какие-либо из следующих слов: «Subassembly», «node», «Mark», «DETAILS», «Details», «Single», или «part». Если это так, функция возвращает True.

Приватная функция getInfoFromTable.

Функция извлекает данные из таблицы Microsoft Word и заполняет входной словарь. Функция принимает два параметра: WordTable, т. е. таблица, из которой нужно извлечь данные, и объект Dictionary(Of String, List(Of String)), который будет заполнен извлеченными данными.

Функция сначала получает все строки в таблице, затем перебирает каждую строку (исключая первую строку) и извлекает данные из третьей и четвертой ячеек каждой строки. Затем он проверяет, содержит ли уже словарь значение наименования, извлеченное из третьей ячейки, и, если нет, добавляет его в словарь вместе с пустым списком для хранения номеров ревизий.

Наконец, функция добавляет номер ревизии, извлеченный из четвертой ячейки, в список, соответствующий значению номинации в словаре. Если в строке меньше трех ячеек, функция пропускает эту строку и переходит к следующей.

Приватный метод handleMistakedInOneDoc.

Метод принимает три параметра: словарь с именем *dataDoc*, который содержит сопоставление строкового ключа со списком строковых значений, другой словарь с именем *withDiffRevInDoc*, который также содержит сопоставление строкового ключа со списком строковых значений, и хеш-таблицу с именем *dataHanled*.

Далее перебирается каждый ключ в *dataDoc* и проверяется, является ли значение, связанное с ключом, «валидным», используя функцию *nominationIsValidInDoc*. Если значение «валидно», первое значение в списке добавляется в хэш-таблицу *dataHandled* с ключом, являющимся наименованием.

Если значение «невалидно», метод перебирает каждое уникальное значение в списке и добавляет его в словарь *withDiffRevInDoc* с ключом, являющимся наименованием.

Приватная функция nominationIsValidInDoc.

Функция проверяет, «валидны» ли наименования в документе. В качестве входных данных он принимает список строк, содержащих ревизии

наименования. Если количество ревизий в списке равно 1, возвращается True. В противном случае выполняется итерация по списку и проверяется, равна ли предыдущая версия текущей версии. Если любые две последовательные ревизии не равны, возвращается False. Если все ревизии равны, возвращается True.

Публичный метод compareHashTables.

Сравнивается две хеш-таблицы: *dataOldDoc_* и *dataNewHandled_*. Перебирается каждый ключ в таблице *dataNewHandled_* и проверяется, существует ли он в таблице *dataOldDoc_*.

Если ключ существует в обеих таблицах, он сравнивает значения соответствующих элементов в каждой таблице. Если значения совпадают, ключ добавляется в список *notChanged*. Если значения отличаются, проверяется, равно ли новое значение старому значению плюс один. Если это так, ключ добавляется в список changed. В противном случае он добавляется в список *mistaked*.

Если ключ не существует в старой таблице, он добавляется в список *new*.

Наконец, подпрограмма вызывает функцию *findDeletedNominations*, которая занимается поиском тех наименований, которые были в «старом» документе, но отсутствуют в «новом».

Приватный метод findDeletedNominations.

Цель метода — найти удаленные наименования в словаре.

Сначала извлекаются все ключи из словаря с именем *dataOldHandled_* и сохраняются в переменной с именем *oldNominations*. Затем метод перебирает каждый ключ в *oldNominations*. Если ключ не найден в другом словаре с именем *dataNewHandled_* и не найден в третьем словаре с именем *errordWithDiffRevInNew_*, то он считается удаленным наименованием и его ключ добавляется в *names_*("Deleted").

Публичный метод exportDataToExcel.

Этот код экспортирует данные в файл Excel. Экспортированные данные разделены на шесть категорий, которые экспортируются на разные рабочие листы в одной книге. Категории: «*Not changed*», «*Changed*», «*New*», «*Deleted*», «*Mistaked*» и «*MistakedInNewDoc*».

Параметры: *path* - строковый параметр, указывающий путь к файлу Excel, в который будут экспортированы данные.

Метод использует Microsoft.Office.Interop.Excel для создания нового приложения Excel.

Создается новая книга, выбирается первый лист в книге. Затем вызывается шесть различных функций экспорта для выгрузки данных на выбранный рабочий лист, каждая для своей категории. Имя выбранного рабочего листа изменяется на «*Data*», файл сохраняется.

Приватный метод exportNotChanged.

В качестве параметра метод принимает объект с именем «*workSheet*». Целью метода является экспорт данных на рабочий лист Excel. Он начинается с установки заголовков первой строки на листе: «Not changed» и «REV.» (1-ый и 2-ой столбцы соответственно). Затем перебирается набор наименований и устанавливается значение текущей строки в столбце 1 (наименование) и значение текущей строки в столбце 2 (ревизия).

Приватный метод exportChanged.

В качестве параметра метод принимает объект с именем «*workSheet*». Целью метода является экспорт данных на рабочий лист Excel. Он начинается с установки заголовков первой строки на листе: «Changed» и «NEW REV.» (4-ый и 5-ый столбцы соответственно). Затем перебирается набор наименований и устанавливается значение текущей строки в столбце 4 (наименование) и значение текущей строки в столбце 5 (ревизия).

Приватный метод exportNew.

В качестве параметра метод принимает объект с именем «*workSheet*». Целью метода является экспорт данных на рабочий лист Excel. Он начинается с установки заголовков первой строки на листе: «New» и «NEW REV.» (7-й и 8-й столбцы соответственно). Затем перебирается набор наименований и устанавливается значение текущей строки в столбце 7 (наименование) и значение текущей строки в столбце 8 (ревизия).

Приватный метод exportDeleted.

В качестве параметра метод принимает объект с именем «*workSheet*». Целью метода является экспорт данных на рабочий лист Excel. Он начинается с установки заголовков первой строки на листе: «Deleted» и «OLD REV.» (10-й и 11-й столбцы соответственно). Затем перебирается набор наименований и устанавливается значение текущей строки в столбце 10 (наименование) и значение текущей строки в столбце 11 (ревизия).

Приватный метод exportMistaked.

В качестве параметра метод принимает объект с именем «*workSheet*». Целью метода является экспорт данных на рабочий лист Excel. Он начинается с установки заголовков первой строки на листе: «Mistaked» и «OLD REV.»/ «NEW REV.» (13-й, 14-й и 15-й столбцы соответственно). Затем перебирается набор наименований и устанавливается значение текущей строки в столбце 13 (наименование) и значение текущей строки в столбце 14 и 15 (старая и новая ревизии).

Приватный метод exportMistakedInNewDoc.

В качестве параметра метод принимает объект с именем «*workSheet*». Целью метода является экспорт данных на рабочий лист Excel. Он начинается

с установки заголовков первой строки на листе: «Mistaked with diff REV.» и «REV.» (17-й, 18-й столбцы соответственно). Затем перебирается набор наименований. Для каждого наименования происходит перебор значений соответствующих ему ревизий. Данные помещаются в 17-й (наименование) и 18-й (ревизия) столбцы.

Публичный метод parsePDBExcel.

Метод анализирует файл Excel в «формате PDB». Он создает новый экземпляр Microsoft Excel, открывает файл PDB Excel и считывает значения из столбцов «O» и «V» первого рабочего листа. Затем он сохраняет эти значения в двух списках, *nominations* и *revisions*. Затем проверяется наличие повторяющихся записей в списке *nominations* и выдается исключение, если найден дубликат. Если дубликатов нет, добавляются значения наименования (из *nominations*) и ревизии (из *revisions*) в словарь *pdbExcelData_*.

Публичный метод compareHandledDataWithPDB.

Метод сравнивает данные в двух списках, чтобы определить, существуют ли определенные наименования в обоих контейнерах (*pdbExcelData_* и *names_*). Метод создает список рассматриваемых наименований и перебирает их. Для каждого наименования проверяется, содержит ли контейнер имен в PDB (*pdbExcelData_*) это наименование. Если это так, имя добавляется в список *inPDB_*. Если нет, наименование добавляется в список *notInPDB_*.

Публичный метод exportComparisonPDBToExcel.

Метод экспортирует данные в файл Excel. Метод создает новый объект приложения Excel, новый объект рабочей книги и новый объект рабочего листа. Затем он объединяет ячейки («A1:D1» и «F1:I1») и устанавливает их значения («In PDB» и «Not in PDB»). После этого он перебирает два списка данных (*inPDB_* и *notInPDB_*) и заполняет рабочий лист данными. Наконец, он устанавливает имя листа, сохраняет книгу, закрывает ее и закрывает приложение Excel.