**Документация к коду примера.**

**Содержание**

[1. Общие слова о взаимодействии 2](#_Toc136289428)

[2. Класс ConverterWindow 3](#_Toc136289429)

[3. Класс Handler 4](#_Toc136289430)

[Приложение. 8](#_Toc136289431)

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.

1. **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.

1. **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 («с чистого листа»).