

Класс ConvertText

Класс предназначен для обработки текстов и символьных строк. Основная задача, решаемая классом — это поиск в тексте заданных слов (словосочетаний) с различными словоформами.

Создание объекта — экземпляра класса:

```
oCnv = createobject('ConvertText' [, lLang])
```

где lLang — необязательный параметр, указывающий параметры перекодировки символов. Если параметр опущен (или False), то все символы латинского алфавита, имеющие одинаковое начертание с кириллицей, преобразуются к кириллице; если этот параметр — True, то выполняется преобразование к латинице.

Класс предоставляет следующие методы:

Имя метода	Назначение
Convert	Преобразует текст, выполняя замену «нечитаемых» символов на допустимые, и выполняя предварительное форматирование (удаление повторных пробелов, «склеивание» переносов слов, и ряд других)
GetTitle	Возвращает первый абзац текста (заголовок статьи)
WordParser	Разбивает текст на слова и возвращает полученный набор слов в массиве
WordForms	Преобразует слово (словосочетание) с различными словоформами в массив, содержащий все возможные варианты написания слов
FindWordForm	Выполняет поиск слов (словосочетаний) в массиве слов

Ниже приводится подробное описание этих методов.

Метод Convert

Метод выполняет следующие действия:

- Замена «нечитаемых» символов на допустимые; замена символов, имеющих одинаковое смысловое значение, на единый символ (например, символы « и » заменяются на кавычки (")). Если символ невозможно заменить, то он отбрасывается. Символы табуляции заменяются на пробелы.
- Удаление повторяющихся пробелов
- Склеивание переносов слов
- Если слово в начале следующей строки начинается с заглавной буквы, то выполняется его склеивание со словом, завершающим предыдущую строку, с сохранением дефиса. Например, следующая конструкция:
 - ... происходило в Санкт-Петербурге
 - будет заменена на
 - ... происходило в Санкт-Петербурге
- Во всех остальных случаях слово «склеивается», символ переноса строки отбрасывается.
- Обработка дефиса. При необходимости добавляются ведущий и конечный пробелы
- Например, следующая конструкция:
 - Этот вариант меня устраивает,-заметил клиент.
 - будет заменена на
 - Этот вариант меня устраивает, - заметил клиент.
- Вставка пробелов перед заглавной буквой, если она следует за знаком препинания, завершающим предложение.

Формат вызова:

```
cFormattingText = oCnv.Convert(cText)
```

где:

cText — исходный (преобразуемый) текст

cFormattingText — текст после обработки

Все строки в *cFormattingText* завершаются символами CR и LF.

Метод **GetTitle**

Метод выделяет из текста первый абзац и возвращает его.

Формат вызова:

```
cTitle = oCnv.GetTitle()
```

Метод **WordParser**

Метод разбивает текст на слова и возвращает полученный набор слов в массиве. При вызове метода вы можете указать минимальную длину включаемого в массив слова; все слова, длина которых меньше указанной, будут проигнорированы.

Замечание

Все слова в результирующем массиве приводятся к нижнему регистру.

Формат вызова:

```
lResult = oCnv.WordParser(cText, @aResult [, nMinLength])
```

где:

cText — исходный текст;

aResult — одномерный массив, в который помещаются выбранные из текста слова (без ведущих и конечных пробелов).

nMinLength — минимально допустимая длина слова (необязательный параметр; по умолчанию — 3 символа);

При успешном выполнении метод возвращает True.

Метод **WordForms**

Метод преобразует слово (словосочетание) с различными словоформами в массив, содержащий все возможные варианты написания слов.

Формат для словоформ предполагает наличие неизменяемой части слова и перечисленные через вертикальную черту окончания, например:

Компани|я|и|ю|ей

В результате выполнения метода будет создан массив из 4-х элементов со следующим содержанием:

```
компания  
компании  
компанию  
компанией
```

Замечание

Все слова в результирующем массиве приводятся к нижнему регистру.

Если преобразуется словосочетание, то для каждого слова в этом словосочетании могут быть указаны словоформы, например:

Это известн|ая|ую|ой компани|я|и|ю

В результате выполнения метода будет создан массив из девяти элементов со следующим содержанием:

```
это известная компания  
это известная компании  
это известная компанию  
это известную компания  
это известную компании  
это известную компанию  
это известной компания  
это известной компании
```

это известной компании

Замечание

Передаваемые методу слова могут не иметь словоформ (типов окончаний), как это показано выше: слово «Это» в примере не имеет словоформ.

Если неизменяемая часть слова имеет смысл (например, для слова «город»), то вы можете указать два последовательно расположенных символа «|» для указания того, что при формировании вариантов написания должна учитываться и неизменяемая часть слова, например:

Город| |а|у|е|ах

Замечание

В одном слове должно быть не более одного вхождения конструкции «||».

Формат вызова:

```
lReturn = oCnv.WordForms(cWord, @aResult)
```

где:

cWord — строка, содержащая слово либо словосочетание (со словоформами)

aResult — одномерный массив, в который помещаются возможные написания слов (в словосочетании)

При успешном выполнении метод возвращает True.

Метод FindWordForm

Метод выполняет поиск слов (словосочетаний) в массиве слов. Он получает два параметра:

- Массив возможных написаний слов (словосочетаний); этот массив должен быть сформирован методом **WordForms** класса;
- Массив слов текста, в котором нужно выполнить поиск; этот массив должен быть сформирован методом **WordParser** класса.

Помимо рассмотренного выше (в описании метода **WordForms**) синтаксиса вы можете указывать символ «*» как завершающий неизменяемую часть слова. Если этот символ не указан, то требуется точное совпадение слова (слов) в тексте со словами для поиска. Если символ указан, то поиск выполняется по совпадению первых символов.

Вы можете в словосочетании для одних слов указывать как окончание символа «*», а для других — использовать словоформы, например:

Строительн* компани|я|ю|и|ей|ями

При выполнении этого примера как правильные будут признаны все возможные словоформы для слова «Строительн», например, такие, как «Строительный», «Строительному» и т.д., и словоформы «компания», «компанию», «компаний», «компанией» и «компаниями» — для второго слова в словосочетании.

Слова ищутся в той последовательности, в которой они указаны. В случае, если вам необходимо найти слова (словосочетания), расположенные в разных частях текста, используйте символ «+», например:

Санкт-Петербург+банк| |а|е|у|Ом

Результат поиска будет положительным только тогда, когда в тексте одновременно (но в любом месте) присутствуют слова «Санкт-Петербург» и указанные словоформы слова «банк».

Формат вызова:

```
lReturn = oCnv.FindWordForm(@aWords, @aText)
```

где:

aWords — массив, содержащий все возможные написания слов

aText — массив слов текста, в котором выполняется поиск.

При успешном выполнении метод возвращает True.

Замечание

Для корректной работы метода SET EXACT должно быть установлено в OFF