

ООО «ЭЛТЕК»

Руководство по установке и эксплуатации программного обеспечения
«ЛАН.Текстовый процессор»

2024

1 Общие сведения

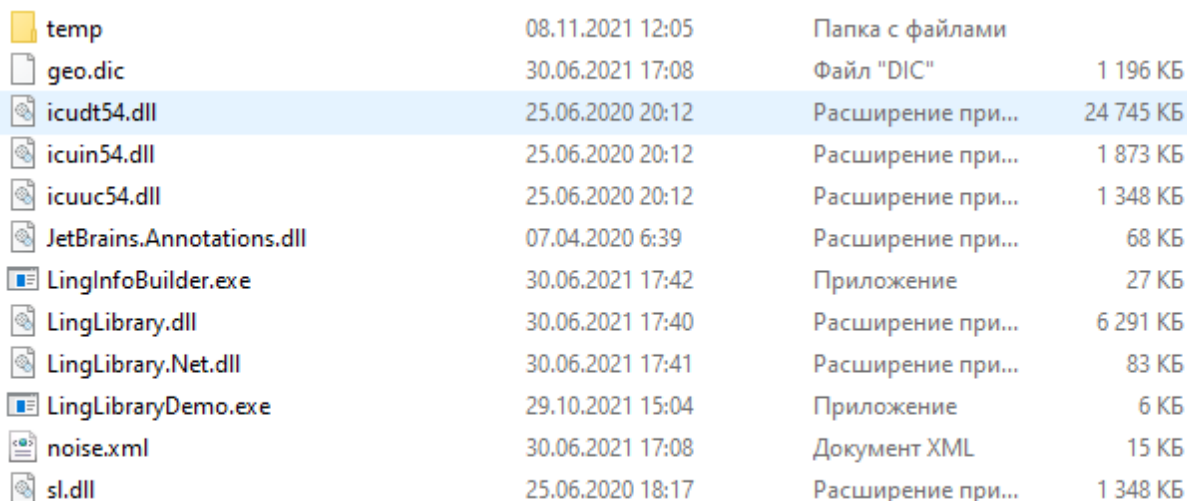
Программное обеспечение «ЛАН.Текстовый процессор» разработано для извлечения, обработки и анализа текстовых данных из различных источников и на различных языках.

«ЛАН.Текстовый процессор» является программной библиотекой и не имеет графического интерфейса. Далее приводится описание библиотеки.

2 Установка

Для установки программы «ЛАН.Текстовый процессор» требуется разархивировать архив с программой.

В результате программа будет размещена в директории разархивации. Содержимое директории с программой представлено на рисунке 1.



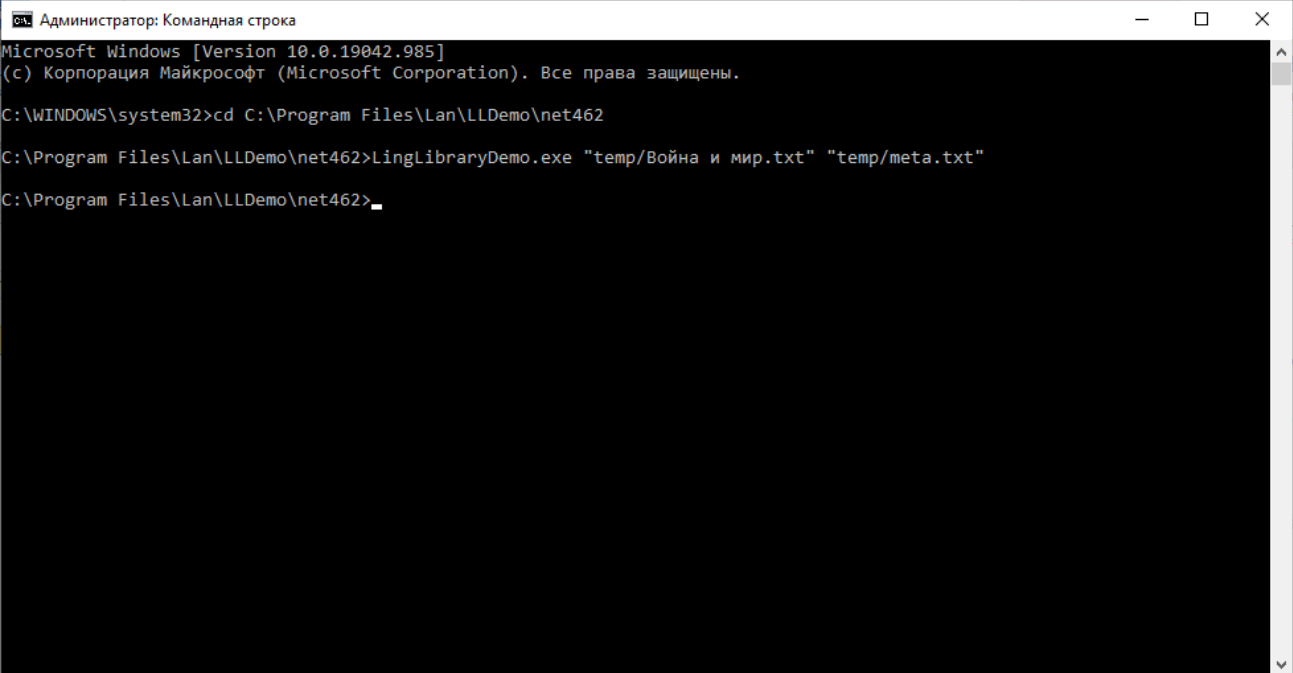
temp	08.11.2021 12:05	Папка с файлами	
geo.dic	30.06.2021 17:08	Файл "DIC"	1 196 КБ
icudt54.dll	25.06.2020 20:12	Расширение при...	24 745 КБ
icuin54.dll	25.06.2020 20:12	Расширение при...	1 873 КБ
icuuc54.dll	25.06.2020 20:12	Расширение при...	1 348 КБ
JetBrains.Annotations.dll	07.04.2020 6:39	Расширение при...	68 КБ
LingInfoBuilder.exe	30.06.2021 17:42	Приложение	27 КБ
LingLibrary.dll	30.06.2021 17:40	Расширение при...	6 291 КБ
LingLibrary.Net.dll	30.06.2021 17:41	Расширение при...	83 КБ
LingLibraryDemo.exe	29.10.2021 15:04	Приложение	6 КБ
noise.xml	30.06.2021 17:08	Документ XML	15 КБ
sl.dll	25.06.2020 18:17	Расширение при...	1 348 КБ

Рисунок 1 – Директория с программой «ЛАН.Текстовый процессор»

3 Эксплуатация

3.1 Запуск

«ЛАН.Текстовый процессор» запускается с помощью консоли cmd. В консоли производится переход в директорию программы, а затем исполняется команда «LingLibraryDemo.exe "temp/Война и мир.txt" "temp/meta.txt"».



```
Администратор: Командная строка
Microsoft Windows [Version 10.0.19042.985]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\WINDOWS\system32>cd C:\Program Files\Lan\LLDemo\net462
C:\Program Files\Lan\LLDemo\net462>LingLibraryDemo.exe "temp/Война и мир.txt" "temp/meta.txt"
C:\Program Files\Lan\LLDemo\net462>_
```

Рисунок 2 – Запуск программы «ЛАН.Текстовый процессор»

Результатом работы программы является файл «meta.txt», находящийся в директории программы в поддиректории «temp».

3.2 Интерфейсы

IClsDocMetaInfo

Интерфейс для получения метаинформации в кэше классификаторов.

Поля:

IClsDocMetaInfo.MetaInfoId – Получает Id метаинформации в кэше классификаторов.

IGetWordsForm

Интерфейс для трансформации и получения форм слова

Методы:

`IGetWordsForm.Metaphone(string word)` – Преобразование слова по правилам метафона

- `word` – слово.

`IGetWordsForm.GetAllWordForms(string word, bool ex)` – Получение всех форм слова

- `word` – слово;
- `ex` – значение, показывающее, использовать ли алгоритм предсказания формы.

`IGetWordsForm.GetWordForm(string word, WordGrammarType wordType, WordGrammarCase resultCase, WordGrammarNumber resultNumber, WordGrammarCase sourceCase)` – Получение заданной формы слова.

- `word` – Исходное слово;
- `wordType` – Тип исходного слова;
- `resultCase` – Падеж, в который необходимо установить слово;
- `resultNumber` – Число, в которое необходимо поставить слово;
- `sourceCase` – Исходный падеж слова.

`IGetWordsForm.GetWordFormFromNorm(string word, WordGrammarType wordType, WordGrammarCase resultCase, WordGrammarNumber resultNumber, WordGrammarCase sourceCase)` – Получение заданной формы слова из нормальной формы слова.

- `word` – Исходное слово в нормальной форме;
- `wordType` – Тип исходного слова;
- `resultCase` – Падеж, в который необходимо установить слово;
- `resultNumber` – Число, в которое необходимо поставить слово;
- `sourceCase` – Исходный падеж слова.

`IGetWordsForm.GetPhraseForm(string phrase, WordGrammarCase resultCase, WordGrammarNumber resultNumber)` – Получение формы фразы.

- `phrase` – Исходная фраза в именительном падеже;
- `resultCase` – Падеж, в который необходимо установить фразу;
- `resultNumber` – Число, в которое необходимо поставить фразу.

`IGetWordsForm.GetAllForms(FullNameGrammarCase sourceName)` – Получение всех падежей полного имени.

- `sourceName` – Исходное полное имя.

`IGetWordsForm.GetAllFormsFromNorm(FullNameGrammarCase sourceName)` – Получение всех падежей полного имени из нормальной исходной формы.

- `sourceName` – Исходное полное имя.

`IGetWordsForm.GetAllForms(PhraseGrammarCase sourcePhrase)` – Получение всех падежей фразы.

- `sourcePhrase` – Исходная фраза.

`IGetWordsForm.GetFullNameForm(FullNameGrammarCase sourceName, WordGrammarCase grammarCase, WordGrammarNumber grammarNumber)` – Получение формы полного имени.

- `sourceName` – Исходное полное имя;
- `grammarCase` – Падеж, в который необходимо установить имя;
- `grammarNumber` – Число, в которое необходимо поставить имя.

`IGetWordsForm.GetFullNameFormFromNorm(FullNameGrammarCase sourceName, WordGrammarCase grammarCase, WordGrammarNumber grammarNumber)` – Получение формы полного имени из нормальной формы.

- `sourceName` – Исходное полное имя;
- `grammarCase` – Падеж, в который необходимо установить имя;
- `grammarNumber` – Число, в которое необходимо поставить имя.

`IGetWordsForm.GetFullNameForm(FullNameGrammarCase sourceName, WordGrammarCase grammarCase)` – Получение формы полного имени в единственном числе.

- `sourceName` – Исходное полное имя;
- `grammarCase` – Падеж, в который необходимо установить имя.

`IGetWordsForm.GetFullNameFormFromNorm(FullNameGrammarCase sourceName, WordGrammarCase grammarCase)` – Получение формы полного имени в единственном числе из нормальной формы.

- `sourceName` – Исходное полное имя;
- `grammarCase` – Падеж, в который необходимо установить имя.

`IGetWordsForm.GetPhraseForm(PhraseGrammarCase phrase, WordGrammarCase grammarCase, WordGrammarNumber grammarNumber)` – Получение формы фразы.

- `phrase` – Исходная фраза;
- `grammarCase` – Падеж, в который необходимо установить фразу;
- `grammarNumber` – Число, в которую необходимо установить фразу.

`IGetWordsForm.GetPhraseForm(PhraseGrammarCase phrase, WordGrammarCase grammarCase)` – Получение формы фразы в единственном числе.

- `phrase` – Исходная фраза;
- `grammarCase` – Падеж, в который необходимо установить фразу.

IQueryAnalyzer

Интерфейс проверки запроса на ошибки.

Методы:

`IQueryAnalyzer.Check(string query)` – Проверить запрос

- `query` – Запрос на проверку.

`IQueryAnalyzer.FullCheck(string localQuery, string globalQuery, System.Threading.CancellationToken token)` – проверить запрос на наличие грамматических ошибок и заикливания

- `localQuery` – Локальный запрос на проверку;
- `globalQuery` – Глобальный запрос на проверку;
- `token` – Механизм прерывания.

`IQueryAnalyzer.GetErrorMessage()` – Вернуть описание ошибок в виде строки

`IQueryAnalyzer.GetMessagesList()` – Вернуть описание ошибок в виде списка

`IQueryAnalyzer.InitializeProgressCallBack(IProgressCallback callBack)` – Привязать к процессу механизм обратного вызова

- `callBack` – реализация механизма обратного вызова.

IQueryExecuteOnText

Интерфейс для класса, производящего разбор запроса, и выполняющего разметку текста по запросу, выделение атрибутов и объектов из текста

Методы:

`IQueryExecuteOnText.ReInit(string localQuery, string globalQuery, IProgressCallback progress)` – Инициализировать запрос

- `localQuery` – локальный запрос;
- `globalQuery` – глобальный запрос;
- `progress` – механизм обратного вызова.

`IQueryExecuteOnText.MarkupText(string projectPath, string shortFileName, string text)`
– Получить размеченный текст

- `projectPath` – путь к папке проекта;
- `shortFileName` – имя файла;
- `text` – текст.

`IQueryExecuteOnText.GetSemanticDescription(string projectPath, string shortFileName, string text)` – Получить описание объектов и атрибутов текста по запросу

- `projectPath` – путь к папке проекта;
- `shortFileName` – имя файла;
- `text` – текст.

IClassifier

Интерфейс классификатора.

Поля:

`IClassifier.Name` – Получает название классификатора

`IClassifier.Guid` – Получает GUID Классификатора

`IClassifier.Rubrics` – Получает словарь рубрик, где ключ – Guid рубрики, значение – название рубрики

Методы:

`IClassifier.Classify(string text)` – Классифицировать текст

- `text` – Текст для классификации.

`IClassifier.Classify(string text, string metaInfo)` – Классифицировать текст
Функция работает быстрее предыдущей, так как внутри не строится метаинформация.
ВАЖНО! Текст должен соответствовать метаинформации. Внутри проверки на соответствие не делается

- `text` – Текст для классификации;
- `metaInfo` – Метаинформация по тексту.

`IClassifier.Classify(string text, IClDocMetaInfo metaInfo)` – Классифицировать текст
Функция работает быстрее предыдущей, так как внутри не строится метаинформация.
ВАЖНО! Текст должен соответствовать метаинформации. Внутри проверки на соответствие не делается

- `text` – Текст для классификации;
- `metaInfo` – Метаинформация по тексту.

`IClassifier.ClassifyPack(string text, IClDocMetaInfo metaInfo, string sectionSeparator)` – Классифицировать нескольких текстов, предварительно скомбинированный с разделителями. Используется разделитель вида `SECTION_separator`.

- `text` – Скомбинированный текст с разделителями;
- `metaInfo` – Метаинформация для комбинированного текста;
- `sectionSeparator` – Имя разделителя.

3.3 Перечисления

DataModel.ClusterIdErrorStatus

Статус ID кластера, к которому отнесен документ.

`DataModel.ClusterIdErrorStatus.IsOk` – ID кластера, к которому отнесен документ, валидный.

`DataModel.ClusterIdErrorStatus.IsError` – ID кластера, к которому отнесен документ, соответствует неизвестной ошибке.

`DataModel.ClusterIdErrorStatus.IsDuplicate` – ID кластера, к которому отнесен документ, соответствует документу-дубликату.

DataModel.DateTimeDirection

Указание для определения даты в будущем, настоящем или прошедшем времени.

DataModel.DateTimeDirection.Undefined – Дата на текущий момент.

DataModel.DateTimeDirection.Backward – Дата в прошедшем времени.

DataModel.DateTimeDirection.Forward – Дата в будущем времени.

DataModel.WordGrammarCase

Падеж слова.

DataModel.WordGrammarCase.Nominative – RusNominative – именительный

DataModel.WordGrammarCase.Genitive – RusGenitive – родительный

DataModel.WordGrammarCase.Dative – RusDative – дательный

DataModel.WordGrammarCase.Accusative – RusAccusative – винительный

DataModel.WordGrammarCase.Instrumental – RusInstrumental – орудительный

DataModel.WordGrammarCase.Prepositional – RusPrepositional – предложный

DataModel.WordGrammarCase.Vocative – RusVocative – изъявительный

DataModel.WordGrammarNumber

Число слова.

DataModel.WordGrammarNumber.Plural – Множественное число.

DataModel.WordGrammarNumber.Singular – Единственное число.

DataModel.WordGrammarType

Тип слова.

DataModel.WordGrammarType.RusAny – тип не известен (выбирается первый из списка омонимов).

DataModel.WordGrammarType.RusInitialism – инициалы.

DataModel.WordGrammarType.RusPatronymic – отчество.

DataModel.WordGrammarType.RusToponym – географическое название.

DataModel.WordGrammarType.RusOrganisation – название организации.

DataModel.WordGrammarType.RusQualitative – числительное.

DataModel.WordGrammarType.RusName – имя.

DataModel.WordGrammarType.RusSurName – фамилия.

DataModel.WordGrammarType.RusImpersonal – неличное.

DataModel.WordGrammarType.RusSlang – сленг.

DataModel.WordGrammarType.RusProfession – профессия.

3.4 Классы

AdvancedClusterizator

Класс для создания файлов статистики подборки для последующего обучения.

Методы:

AdvancedClusterizator(string docPath, string cachePath, string configPath, int blockSize) – Конструктор

- docPath – путь к документам
- cachePath – путь к кэшу
- configPath – путь к конфигурационному файлу
- blockSize – размер блока документов

AdvancedClusterizator.CreateJsonTable(string outputPath) – Запись в файл JSON–результата

- outputPath – путь к файлу для записи результата

AdvancedClusterizator.CreateEtalonFile(string outputPath) – Запись в файл эталонного разбиения

- outputPath – путь к файлу для записи результата

Classifier

Класс для выполнения классификации документов.

Поля:

Classifier.Name – Получает название классификатора

Classifier.Guid – Получает GUID Классификатора

Classifier.Rubrics – Получает словарь рубрик, где ключ - Guid рубрики, значение - название рубрики

Методы:

Classifier.Init(byte[] binaryClsData) – Инициализация классификатора из бинарных данных.

- binaryClsData – Бинарные данные cls-файла.

Classifier.Init(string name, string guid, byte[] binaryClsData) – Инициализация классификатора из бинарных данных, именем и GUID

- name – Имя классификатора.
- guid – Уникальный идентификатор классификатора.
- binaryClsData – Бинарные данные классификатора.

`Classifier.Init(string clsFile)` – Инициализация классификатора из CLS-файла.

- `clsFile` – Путь к CLS-файлу.

`Classifier.Classify(string text)` – Классифицировать текст

- `text` – Текст для классификации

`Classifier.Classify(string text, string metaInfo)` – Классифицировать текст. Функция работает быстрее предыдущей, так как внутри не строится метаинформация. ВАЖНО! Текст должен соответствовать метаинформации. Внутри проверки на соответствие не делается.

- `text` – Текст для классификации.
- `metaInfo` – Метаинформация по тексту.

`Classifier.Classify(string text, IClDocMetaInfo metaInfo)` –

Классифицировать текст Функция работает быстрее предыдущей, так как внутри не строится метаинформация. ВАЖНО! Текст должен соответствовать метаинформации. Внутри проверки на соответствие не делается

- `text` – Текст для классификации
- `metaInfo` – Метаинформация по тексту

`Classifier.ClassifyPack(string text, IClDocMetaInfo metaInfo, string sectionSeparator)` – Классифицировать нескольких текстов, предварительно скомбинированный с разделителями. Используется разделитель вида `SECTION_separator`.

- `text` – Скомбинированный текст с разделителями.
- `metaInfo` – Метаинформация для комбинированного текста.
- `sectionSeparator` – Имя разделителя.

`Classifier.Dispose` – Завершить работу классификатора.

ClDocMetaInfo

Класс для получения метаинформации в кэше классификатора.

Поля:

`ClDocMetaInfo.MetaInfoId` – Получает Id метаинформации в кэше классификаторов.

Методы:

`ClDocMetaInfo.BuildClMetaInfo(string text)` – Создать метаинформацию из текста в кэше классификаторов.

- `text` – Текст, для которого создается метаинформация.

Clusterizator

Класс, проводящий кластеризацию подборки документов

Поля:

Clusterizator.ClusterizatorPtr – Получает или задает указатель на память, содержащую объект, проводящий кластеризацию подборки документов

Методы:

Clusterizator(string cachePath, string configPath, int blockSize) – Создание объекта класса, проводящего кластеризацию подборки документов

- cachePath – путь к кэшу
- configPath – путь к конфигурационному файлу
- blockSize – размер блока документов

Clusterizator(int blockSize) – Создание объекта класса, проводящего кластеризацию подборки документов

- blockSize – размер блока документов

Clusterizator.RunClusterization(List<string> filesList, IProgressCallback callback, CancellationToken token) – Запускает кластеризацию с возвратом списка ID

- filesList – Список путей к файлам
- callback – Механизм обратного вызова для прогресса
- token – Механизм корректного останова

Clusterizator.RunClusterizationMultiThread(List<string> filesList, uint numOfThreads, IProgressCallback callback, System.Threading.CancellationToken token) – Запускает кластеризацию с возвратом списка ID

- filesList – Список путей к файлам
- numOfThreads – Количество потоков, в которых выполнять кластеризацию.
- callback – Механизм обратного вызова для прогресса
- token – Механизм корректного останова

Clusterizator.GetMesuareForDocumentPair(int ,int) – Получение меры близости для пары документов

- i – Порядковый индекс документа 1
- j – Порядковый индекс документа 2

Clusterizator.GetMesuareForDocumentPair(int ,int ,System.Int64) – Получение меры близости для пары документов

- i – Порядковый индекс документа 1
- j – Порядковый индекс документа 2
- type – тип блока

Clusterizator.Dispose – Освобождение ресурсов

DataModel.DateTimeExtracted

Класс извлеченных даты из строки.

Поля:

DataModel.DateTimeExtracted.FirstDate – Получает дату начала события.

DataModel.DateTimeExtracted.LastDate – Получает дату конца события.

DataModel.DateTimeExtracted.IsRange – Показывает, является ли извлеченная дата диапазоном дат.

Методы:

DataModel.DateTimeExtracted(DateTime firstDate, DateTime lastDate) – Конструктор класса.

- firstDate – Дата начала события.
- lastDate – Дата конца события.

DataModel.DetectedLanguage

Класс результата детектирования языка и набора символов.

Поля:

DataModel.DetectedLanguage.CharacterSet – Получает набор символов текста.

DataModel.DetectedLanguage.LanguageCode – Получает код языка текста.

DataModel.FullNameGrammarCase

Класс ФИО человека в заданном падеже.

Поля:

DataModel.FullNameGrammarCase.GrammarCase – Получает или задает падеж ФИО.

DataModel.FullNameGrammarCase.FirstName – Получает или задает Имя.

DataModel.FullNameGrammarCase.MiddleName – Получает или задает Отчество.

DataModel.FullNameGrammarCase.LastName – Получает или задает Фамилию.

Методы:

DataModel.FullNameGrammarCase – Конструктор по умолчанию устанавливает падеж в .

DataModel.LingConvertResult

Класс результата работы конвертеров из SCATQL-запроса.

Поля:

DataModel.LingConvertResult.ConvertResult – Получает результат преобразования из SCATQL.

`DataModel.LingConvertResult.ErrorInfo` – Получает информацию об ошибке во время преобразования.

`DataModel.LingConvertResult.IsSimple` – Получает значение, показывающее, что SCATQL–запрос является простым.

`DataModel.LingConvertResult.IsError` – Получает значение, показывающее, что во время преобразования произошла ошибка.

DataModel.PhraseGrammarCase

Класс фразы в заданном падеже.

Поля:

`DataModel.PhraseGrammarCase.GrammarCase` – Получает или задает падеж фразы.

`DataModel.PhraseGrammarCase.Phrase` – Получает или задает фразу.

Методы:

`DataModel.PhraseGrammarCase` – Конструктор по умолчанию устанавливает падеж в .

DateTimeExtractor

Класс извлечения даты и времени из строки текста.

Методы:

`DateTimeExtractor.GetIsoDateTime(string sourceDateTime, DateTime publicationDate, DateTimeDirection direction)` – Получить дату и время в ISO формате из строки текста с учетом прошедшего или будущего времени.

Направление используется в случае, например, строки с текстом "в прошлый четверг" или "в следующий вторник".

- `sourceDateTime` – Исходная строка текста с датой и временем.
- `publicationDate` – Дата публикации документа, относительно которой определяется требуемая дата.
- `direction` – Направление преобразования строки текста в дату.

Utils.SectionBuilder

Класс построения текста, разделенного на секции

Методы:

`Utils.SectionBuilder.SetTitle(string title)` – Задает содержимое секции `SECTION_TITLE`.

- `title` – Содержимое секции `SECTION_TITLE`.

`Utils.SectionBuilder.SetText(string text)` – Задает содержимое секции `SECTION_TEXT`.

- `text` – Содержимое секции `SECTION_TEXT`.

`Utils.SectionBuilder.SetComment(string comment)` – Задает содержимое секции `SECTION_COMMENT`.

- `comment` – Содержимое секции `SECTION_COMMENT`.

`Utils.SectionBuilder.SetUrl(string url)` – Задает содержимое секции `SECTION_URL`.

- `url` – Содержимое секции `SECTION_URL`.

`Utils.SectionBuilder.SetAttribute(string name, string value)` – Задает содержимое секции `SECTION_ATTRIBUTES`.

- `name` – Имя атрибута секции `SECTION_ATTRIBUTES`.
- `value` – Значение атрибута секции `SECTION_ATTRIBUTES`.

`Utils.SectionBuilder.RemoveFirstAttribute(string name, string value)` – Удаляет первый атрибут с заданным именем и значением из секции `SECTION_ATTRIBUTES`.

- `name` – Имя атрибута секции `SECTION_ATTRIBUTES` для удаления.
- `value` – Значение атрибута секции `SECTION_ATTRIBUTES` для удаления.

`Utils.SectionBuilder.ToWrappedText()` – Возвращает текст с установленными секциями `SECTION_...`

Utils.SimpleProgressCallback

Класс простого callback'a для прогресс-бара.

Поля:

`Utils.SimpleProgressCallback.ProgressDone` – Получает текущее значение прогресса.

`Utils.SimpleProgressCallback.Warning` – Получает сообщение об ошибке.

Методы:

`Utils.SimpleProgressCallback()` – Создает новый экземпляр класса

`Utils.SimpleProgressCallback.OnProgress(long processedData, long totalData, string additionalInfo)` – Метод обратного вызова, устанавливающий текущий прогресс операции.

- `processedData` – Количество обработанных данных.
- `totalData` – Количество данных всего.
- `additionalInfo` – Дополнительная информация.

`Utils.SimpleProgressCallback.OnWarning(string warning)` – Метод обратного вызова, устанавливающий сообщение об ошибке.

- `warning` – Сообщение об ошибке.

Extensions.LongExtensions

Методы расширения для типа long, которым представлены ID кластеров.

Extensions.LongExtensions.GetClusterIdErrorStatus(long id) – Получить статус ошибки ID кластера, к которому отнесен документ.

- id – ID кластера, к которому отнесен документ.

GetWordsForm

Класс для трансформации и получения форм слова

Методы:

GetWordsForm.GetAllWordForms(string word, bool ex) – Получение всех форм слова

- word – слово
- ex – значение, показывающее, использовать ли алгоритм предсказания формы

GetWordsForm.GetWordForm(string word, WordGrammarType wordType, WordGrammarCase resultCase, WordGrammarNumber resultNumber, WordGrammarCase sourceCase) – Получение заданной формы слова.

- word – Исходное слово.
- wordType – Тип исходного слова.
- resultCase – Падеж, в который необходимо установить слово.
- resultNumber – Число, в которое необходимо поставить слово.
- sourceCase – Исходный падеж слова.

GetWordsForm.GetWordFormFromNorm(string word, WordGrammarType wordType, WordGrammarCase resultCase, WordGrammarNumber resultNumber, WordGrammarCase sourceCase) – Получение заданной формы слова из нормальной формы слова.

- word – Исходное слово в нормальной форме.
- wordType – Тип исходного слова.
- resultCase – Падеж, в который необходимо установить слово.
- resultNumber – Число, в которое необходимо поставить слово.
- sourceCase – Исходный падеж слова.

GetWordsForm.GetPhraseForm(string phrase, WordGrammarCase resultCase, WordGrammarNumber resultNumber) – Получение формы фразы.

- phrase – Исходная фраза в именительном падеже.
- resultCase – Падеж, в который необходимо установить фразу.
- resultNumber – Число, в которое необходимо поставить фразу.

GetWordsForm.GetAllForms(FullNameGrammarCase sourceName) – Получение всех падежей полного имени.

- sourceName – Исходное полное имя.

`GetWordsForm.GetAllFormsFromNorm(FullNameGrammarCase)` – Получение всех падежей полного имени из нормальной исходной формы.

- `sourceName` – Исходное полное имя.

`GetWordsForm.GetAllForms(PhraseGrammarCase)` – Получение всех падежей фразы.

- `sourcePhrase` – Исходная фраза.

`GetWordsForm.GetFullNameForm(FullNameGrammarCase sourceName, WordGrammarCase grammarCase, WordGrammarNumber grammarNumber)` – Получение формы полного имени.

- `sourceName` – Исходное полное имя.
- `grammarCase` – Падеж, в который необходимо установить имя.
- `grammarNumber` – Число, в которое необходимо поставить имя.

`GetWordsForm.GetFullNameFormFromNorm(FullNameGrammarCase sourceName, WordGrammarCase grammarCase, WordGrammarNumber grammarNumber)` – Получение формы полного имени из нормальной формы.

- `sourceName` – Исходное полное имя.
- `grammarCase` – Падеж, в который необходимо установить имя.
- `grammarNumber` – Число, в которое необходимо поставить имя.

`GetWordsForm.GetFullNameForm(FullNameGrammarCase sourceName, WordGrammarCase grammarCase)` – Получение формы полного имени в единственном числе.

- `sourceName` – Исходное полное имя.
- `grammarCase` – Падеж, в который необходимо установить имя.

`GetWordsForm.GetFullNameFormFromNorm(FullNameGrammarCase sourceName, WordGrammarCase grammarCase)` – Получение формы полного имени в единственном числе из нормальной формы.

- `sourceName` – Исходное полное имя.
- `grammarCase` – Падеж, в который необходимо установить имя.

`GetWordsForm.GetPhraseForm(PhraseGrammarCase phrase, WordGrammarCase grammarCase, WordGrammarNumber grammarNumber)` – Получение формы фразы.

- `phrase` – Исходная фраза.
- `grammarCase` – Падеж, в который необходимо установить фразу.
- `grammarNumber` – Число, в которую необходимо установить фразу.

`GetWordsForm.GetPhraseForm(PhraseGrammarCase phrase, WordGrammarCase grammarCase)` – Получение формы фразы в единственном числе.

- `phrase` – Исходная фраза.

- `grammarCase` – Падеж, в который необходимо установить фразу.

`GetWordsForm.Metaphone(string word)` – Преобразование слова по правилам метафона

- `word` – слово

LanguageDetection

Методы по определению языка и набора символов по тексту.

Методы:

`LanguageDetection.DetectLanguage(string text)` – Определение набора символов и языка для заданного текста.

- `text` – Текст, для которого необходимо определить язык.

QueryAnalyzer

Проверщик запроса на наличие ошибок.

Методы:

`QueryAnalyzer.InitializeProgressCallBack(IProgressCallback callBack)` – инициализировать механизм обратного вызова

- `callBack` – реализация механизма обратного вызова

`QueryAnalyzer.Check(string query)` – проверить запрос на наличие грамматических ошибок

- `query` – запрос

`QueryAnalyzer.FullCheck(string localQuery, string globalQuery, System.Threading.CancellationToken token)` – проверить запрос на наличие грамматических ошибок и заикливания

- `localQuery` – локальный запрос
- `globalQuery` – глобальный запрос
- `token` – механизм прерывания

`QueryAnalyzer.GetErrorMessage()` – получить строку с описанием ошибок

`QueryAnalyzer.GetMessagesList()` – получить список ошибок

QueryFuzzy2SQL

Класс получения xml-структуры слов с возможными опечатками из БД.

Методы:

`QueryFuzzy2SQL(string connectionString)` – Конструктор класса поиска подбора слов из базы данных по заданным опечаткам

- `connectionString` – Строка подключения к базе данных

`QueryFuzzy2SQL.PreprocessQueryForFuzzyQuery(string xmlQuery, string`

databaseName, string databaseName) – Заполнение xml строки словами с возможными опечатками

- xmlQuery – Строка xml со словами, заданных оператором с опечатками
- databaseName – Имя базы данных
- tableName – Имя таблицы, в которой подбираются слова

TextModificationMethodsFactory

Методы по модификации и конвертации текстов.

Методы:

TextModificationMethodsFactory.GetDuplicateDetector(string config) –
Получение объекта для детектирования дубликатов

- config – конфигурационный файл

TextModificationMethodsFactory.FragmentToRegex(string fragment) –
преобразование текста в регулярное выражение

- fragment – текст

TextModificationMethodsFactory.ConvertQuery2SqlQueryC(string query) –
Преобразует текст в строку для использования в функциях полнотекстового поиска

- query – Текст

TextModificationMethodsFactory.ConvertQuery2SqlQueryMistakes(string query) –
Преобразует текст с ошибочным запросом в строку для использования в функциях полнотекстового поиска

- query – Текст

TextModificationMethodsFactory.GetFuzzyXmlVariants(string query) –
Преобразует текст в строку формата "XML" для поиска слов с опечатками

- query – Текст запроса

TextModificationMethodsFactory.ConvertQuery2SQLQueryFuzzy(string query, string wordsVariants) – Преобразует текст с опечатками в строку для использования в функциях полнотекстового поиска

- query – Текст запроса
- wordsVariants – Варианты слов с опечатками в XML

TextModificationMethodsFactory.ConvertQuery2OracleQueryC(string query) –
Преобразует текст в строку для использования в функциях полнотекстового поиска

- query – Текст

`TextModificationMethodsFactory.ConvertQuery2ElasticQueryC(string query)` – Преобразует текст в строку для использования в функциях полнотекстового поиска

- `query` – Текст

`TextModificationMethodsFactory.AttributeHash(string word)` – Получить идентификатор от значения атрибута

- `word` – Значение атрибута

`TextModificationMethodsFactory.AttributeHashSimple(string word)` – Получить идентификатор от значения атрибута без нормализации и сортировки

- `word` – Значение атрибута

`TextModificationMethodsFactory.AttributeHashCastedToInt32(string word)` – Получить идентификатор от значения атрибута

- `word` – Значение атрибута

`TextModificationMethodsFactory.AttributeHashSimpleCastedToInt32(string word)` – Получить идентификатор от значения атрибута без нормализации и сортировки

- `word` – Значение атрибута

`TextModificationMethodsFactory.ConvertQuery2RegexQuery3C(string query, string globalDef)` – Преобразует поисковой запрос в регулярную строку для поиска

- `query` – локальный запрос
- `globalDef` – глобальный запрос

`TextModificationMethodsFactory.DocQuerySimilarQuery(string text, int len)` – Получение запроса для текста

- `text` – текст
- `len` – длина

`TextModificationMethodsFactory.GetTerms(string text)` – Получение списка терминов из текста

- `text` – текст

`TextModificationMethodsFactory.GetNormForms(string word, bool predict)` – Получение нормальных форм заданного слова на русском или английском языке

- `word` – слово
- `predict` – использовать ли алгоритм предсказаний

`TextModificationMethodsFactory.GetNormText(string sourceText)` – Получение текста с преобразованными к нормальной форме словами.

- `sourceText` – Исходный текст для преобразования.

`TextModificationMethodsFactory.GetNormFormsEx(string word)` – Получение нормальных форм заданного слова на русском или английском языке с учетом возможных омонимов для его словоформ

- `word` – слово

`TextModificationMethodsFactory.TextToCollocList(string text)` – Конвертирует строку текста в список словосочетаний (словосочетания отделяются друг от друга при помощи '\n'), экранируя недопустимые символы языка запросов

- `text` – текст

`TextModificationMethodsFactory.TextToTermList(string text)` – Конвертирует строку текста в список терминов, экранируя недопустимые символы языка запросов

- `text` – Текст

`TextModificationMethodsFactory.DocQueryMarkupAbstract(string text, string query, string def, string tag, int maxLength)` – Получение реферата документа с разметкой.

- `text` – Текст.
- `query` – Локальный запрос.
- `def` – Глобальный запрос.
- `tag` – Класс тега `span`, который обрамляет размеченный текст.
- `maxLength` – Длина реферата.

`TextModificationMethodsFactory.DocQueryMarkupText(string text, string query, string def, string tag)` – Получение разметки текста по запросу.

- `text` – Текст для разметки.
- `query` – Запрос, по которому должен быть размечен текст.
- `def` – Определения переменных (если есть).
- `tag` – Тег для выделения найденных фрагментов

`TextModificationMethodsFactory.DocQueryMarkupMetaInfoFromText(string projectPath, string shortFileName, string localQuery, string globalQuery, string tag, string text, string model)` – Получение разметки для документа из проекта классификатора

- `projectPath` – путь к проекту
- `shortFileName` – имя файла
- `localQuery` – локальный запрос
- `globalQuery` – глобальный запрос

- tag – каким тэгом помечаем найденные фрагменты
- text – входной текст на разметку
- model – модель языка запросов

TextModificationMethodsFactory.IsValidScatqlQuery(string query, string model)

– Проверяет, является ли запрос корректным SCATQL-запросом.

- query – Полученный запрос.
- model – Модель языка SCATQL.

TextModificationMethodsFactory.ConvertFragment2ScatQL(string fragment) –

Преобразует текстовый фрагмент в запрос ScatQL.

- fragment – Текстовый фрагмент.

LingFactory

Класс, возвращающий интерфейсы доступа

Методы:

LingFactory.CreateClassifier(string clsFile) – Создать классификатор на основе CLS файла

- clsFile – Путь к CLS файлу

LingFactory.CreateClassifier(byte[] clsMetaData) – Создать классификатор на основе бинарных данных классификатора

- clsMetaData – Бинарные данные классификатора

LingFactory.CreateClassifier(string name, string guid, byte[] clsMetaData) – Создать классификатор на основе бинарных данных классификатора

- name – Имя классификатора
- guid – Уникальный идентификатор классификатора
- clsMetaData – Бинарные данные классификатора

LingFactory.CreateClsDocMetaInfo(string text) – Создать метаинформацию из текста в кэше классификаторов.

- text – Текст, для которого создается метаинформация.

LingFactory.CreateMetainfo(string text) – Создать метаинформацию для текста

- text – текст

LingFactory.CreateMetainfo(string text, uint maxSize) – Создать метаинформацию для текста

- text – Текст.
- maxSize – Максимальный размер текста, при котором не вызывается внешнее извлечение метаинформации.

`LingFactory.CombineTextsForCls(List<string> texts, string sectionName)` – Объединяет несколько текстов в один, чтобы пачкой передать классификатору. Тексты разделяются тегами `SECTION_name`, где `name` задается в параметре.

- `texts` – Список текстов для объединения.
- `sectionName` – Имя тега `SECTION_` для разделения текстов.

`LingFactory.ParseMetaInfo(string metaInfo)` – Получить интерфейс метаинформации документа

- `metaInfo` – строка метаинформации документа

`LingFactory.CreateFilterRelevanceQuery(string query, string globalQuery)` – Получить вычислитель релевантности запроса

- `query` – локальный запрос
- `globalQuery` – глобальный запрос

`LingFactory.CreateIStatisticPhraseByQuery(string query)` – Получить объект, извлекающий фразы из текста по запросу с их контекстом

- `query` – запрос

`LingFactory.CreateQueryAnalyzer` – Получить анализатор запроса на предмет ошибок

`LingFactory.CreateClusterizator(string docPath, string cachePath, string configPath, int blockSize)` – Получить объект, проводящий кластерный анализ документов

- `docPath` – путь к документам
- `cachePath` – путь к кэшу
- `configPath` – путь к файлу конфигурации
- `blockSize` – размер обрабатываемого блока

`LingFactory.CreateClusterizator(string cachePath, string configPath, int blockSize)` – Создать объект, проводящий кластерный анализ документов

- `cachePath` – путь к кэшу
- `configPath` – путь к файлу конфигурации
- `blockSize` – размер обрабатываемого блока

`LingFactory.CreateGrammarLogger()` – Создать объект, возвращающий дерево разбора локального запроса в XML-виде в соответствии с правилами грамматики

`LingFactory.GenerateClusterNumber(string fullDocName, long loadDate)` – Возвращает сгенерированный номер кластера

- `fullDocName` – Полный путь к файлу, для которого генерируем номер кластера
- `loadDate` – Время загрузки или создания файла, если времени загрузки нет (с точностью до 100 мс)

`LingFactory.GenerateQueryExecute(string localQuery, string globalQuery, IProgressCallback progress)` – Возвращает исполнитель запросов, умеющий получать разметку документа, объекты и атрибуты

- `localQuery` – локальный запрос
- `globalQuery` – глобальный запрос
- `progress` – механизм обратного вызова

`LingFactory.CreateGetWordsForm()` – Возвращает интерфейс для трансформации и получения форм слова

3.5 Исключения

Exceptions.BadQueryException

Исключение, генерируемое в ситуации, когда невозможно разобрать запрос

Методы:

`Exceptions.BadQueryException(string error)` – Конструктор

- `error` – текст ошибки

Exceptions.ClsBadVersionException

Класс ошибок неправильной версии проекта классификатора.

Методы:

`Exceptions.ClsBadVersionException(string clsFile, string clsFile)` – Создает экземпляр с сообщением о том, какой файл проекта классификатора вызвал ошибку, и какая версия классификатора в нем используется.

- `clsFile` – Путь к файлу проекта классификации.
- `versionWarning` – Версия файла проекта классификатора.

Exceptions.ClsDocMetaInfoAlreadyBuiltException

Класс исключения, которое возникает, если метаинформация уже построена.

Методы:

`Exceptions.ClsDocMetaInfoAlreadyBuiltException()` – Создает экземпляр с сообщением о том, что метаинформация для данного экземпляра класса метаинформации уже была построена.

Exceptions.ClsDocMetaInfoNotInitializedException

Класс исключения, которое возникает, если метаинформация не была построена и внесена в кэш.

Методы:

`Exceptions.ClsDocMetaInfoNotInitializedException()` – Создает экземпляр с сообщением о том, что метаинформация не была построена и внесена в кэш классификаторов.

Exceptions.ClusterizatorConfigNotFoundExeption

Исключение, генерируемое при отсутствии конфигурационного файла кластеризации при создании экземпляра класса, проводящего кластеризацию подборки документов

Методы:

`Exceptions.ClusterizatorConfigNotFoundExeption(string configPath)` –
Конструктор

- `configPath` – путь к отсутствующему конфигурационному файлу

Exceptions.MarkupExeption

Исключение при разметке

Методы:

`Exceptions.MarkupExeption(string errorDescription)` – Исключение при разметке

- `errorDescription` – описание проблемы