# Описание проекта

**Критерии**

- Класс Text_information выдаёт информацию про предложенный текст, например: процент значимых слов, самые часто используемые фразы, сколько процентов текста занимают первые n процентов слов.
- Сравниваются 6 текстов разных размеров и написанные с использованием разных литературных принципов.
- Делаются некоторые выводы о возможной связи статистических параметров и литературных аспектов текста.

**Библиотеки**

- nltk для работы с текстом
- re для очистки текста от лишнего
- numpy для расчёта статистических параметров
- wordcloud - красивое облачко со словами

**Описание обработчика текста.** Он отдельно лежит в файле "Только код".
В проекте есть несколько вспомогательных классов

1. meaningless_words

   Его можно создать от любой пары наборов слов на русском и английском языке. При пустом вводе он будет состоять из стоп-слов из библиотеки nltk. Его атрибуты: universal - все стоп-слова, russian - русские стоп-слова, english - английские.

2. LanguageException

   Вспомогательная ошибка, чтобы люди не искали латинские буквы, где их нет, ну или не пытались спросить про китайский язык.

3. Language

   Просто содержит названия языков по названию алфавита, чтобы не путаться. russian = cyrillic, english = latin

Есть несколько вспомогательных функций

1. get_the_words - выкидывает пунктуацию и цифры из текста
2. get_punctuation_marks - достаёт всю пунктуацию
3. separate_languages - разделяет слова с кириллическими и латинскими буквами

4. calculate_emphasis - считает количества восклицательных и вопросительных знаков. В итоге не пригодилось.

Основной класс - Text_information
Он создается от текста и содержит его как атрибут.
Атрибуты класса:

- text - сам текст
- words - отдельно слова
- marks - отдельно знаки препинания
- is_cyrillic - есть ли кириллические буквы
- is_latin - есть ли латинские буквы
- cyrillic - все слова, содержащие кириллические буквы
- latin - все слова, содержащие латинские буквы
- emphasis - количество вопросительных и восклицательных знаков
- marks_statistics - словарь со знаками препинания и их количеством в тексте

Методы класса:
(language в параметрах - это язык, слова которого будут обрабатываться. Принимает как значение атрибуты класса language, meaningful в параметрах отвечает за то, какие слова будут обрабатываться: все или только значимые. Принимает True и False)

- word_statistics(meaningful, language) - возвращает словарь со всеми словами и их количесвом в тексте
- plot_word_statistic(number,meaningful, language) - рисует график number самых популярных слов и их количеств
- plot_with_Zipf(meaningful, language) - строит график зависимости количества раз которое слово встречается от его номера по встречаемости (порядка слова) и график закона Ципфа на этой же картинке.
- deviation_from_Zipf(meaningful, language) - возвращает разброс отклонений от закона Ципфа
- only_meaningful(language) - возвращает список значащих слов.
- percent_of_unique_words(meaningful, language) - процент слов, встречающихся один раз
- percent_of_meaningful_words(language) - процент значащих слов

- number_of_words_occurring_n_times(meaningful, language) - возвращает словарь, по числу выдающий все слова встречающиеся столько раз.
- plot_the_number_of_words_by_occurrence(meaningful, language) - строит график предыдущей зависимости в обычной или логарифмической шкале в зависимости от параметра logarithmic
- some_statistic(meaningful, language) - возвращает словарь с ключами
  - most_frequent - количество раз, которое встречается самое частое слово
  - least_frequent - количество раз, которое встречается самое редкое слово
  - number_of_words - количество слов в тексте
  - number_of_different_words - количество различных слов в тексте
  - mean - среднее частот слов
  - dispersion - дисперсия частот слов
  - median - медиана частот слов
  - number_of_unique_words - количество слов, встречающихся один раз
- percent_of_most_common_words(k, meaningful, language) - возвращает процент текста который занимают первые k процентов слов.
- phrases(length, language) - возвращает список фраз длины k (то есть просто напборов последовательоных k слов)
- phrases_statistics(length, language) - возвращает словарь со всеми фразами длины k и их количесвом в тексте
- some_phrases_statistic(length, language) - то же, что и со словами но для фраз длины k
- plot_phrases_statistic(number, length, language) - то же, что и со словами но для фраз длины k
- number_of_phrases_occurring_n_times(length, language) - то же, что и со словами но для фраз длины k
- plot_the_number_of_phrases_by_occurrence(length, language)- то же, что и со словами но для фраз длины k
- cloud(meaningful, language) - рисует красивое облако самых частых слов

**Описание анализа текстов.** Файл с ни называется "Код и текст"

Разобраны 6 текстов, 3 прозаических: "Война и мир"(первые 2 тома), "Мёртвые души"(первый том),"Мастер и Маргарита"и 3 стихотворных: "Кому на Руси жить хорошо", "Евгений Онегин"и "Мцыри".

Всё что с ними сделано и выводы описаны в самом файле.