**Токенизация текста**

Токенизация — это процесс разбиения заданного текста на элементарные блоки, называемые токенами. Токены — это отдельные слова, знаки препинания, предложения. Текст состоит из предложений. Предложение состоит из слов, разделенных пробелами или знаками препинания.

На входе вашей программы - произвольный текст (наример, текст учебника по программированию). Текст необходимо представить коллекцией токенов. Каждый токен - это объект типа **Word** или **Sentence** или **Punctuation**. Поэтому необходимо определить как минимум такие классы, как **Text**, **Sentence**, **Word**, **Punctuation**. **Text -** это коллекция предложений. **Sentence -** это коллекция слов и знаков препинания**. Word -** это неразрывная последовательность символов, за которой может следовать знак препинания **Punctuation**. **Punctuation -** это один из возможных знаков препинания.

Исходный текст может быть текстом, как на английском, так и на русском языке.

Можно реализовать один из вариантов парсинга исходного текста (TextParser):

**Парсер 1**. Читаем посимвольно содержимое файла. Из символов складываем токены. Если собрали токен **Word,** тодобавляем его в **Sentence.** Если собрали **Sentence,** то добавляем его в объект **Text.** В итоге парсер возвращает объект **Text**, который является коллекцией предложений.

**Парсер 2**. Читаем содержимое файла полностью или построчно. Разбиваем прочитанные фрагменты на токены (можно использовать методы класса String или Regular Expressions). Объединяем токены **Word** и токены **Punctuation** в предложения, предложения в текст. В итоге парсер возвращает объект Text, который является коллекцией предложений.

.

Далее для объекта Text необходимо реализовать следующую функциональность:

1. Вывести все предложения заданного текста в порядке возрастания количества слов в предложениях.

2. Вывести все предложения заданного текста в порядке возрастания длины предложения.

3. Во всех вопросительных предложениях текста найти слова заданной длины (не повторять одни и те же слова).

4. Удалить из текста все слова заданной длины, начинающиеся с согласной буквы.

5. В некотором предложении текста заменить слова заданной длины на указанную подстроку, длина которой может не совпадать с длиной слова.

6. В текстах часто встречаются стоп-слова. К стоп-словам (или словам-шумам) относятся предлоги, союзы, междометия, частицы и другие части, которые часто встречаются в тексте, но не несут смысловой нагрузки — они избыточны. Удалить стоп-слова из текста. Вам даны файлы с русскими и английскими стоп-словами.

7. Экспортировать текстовый объект в XML-документ (с помощью механизма сериализации, System.Xml.Serialization)

