

Coding Skills Test

Your task is to implement part of a text processing library.

The following are assumptions and definitions that limit the scope of the task:

- **Word:** To simplify, a word is represented by a sequence of one or more characters between 'a' and 'z' or between 'A' and 'Z'). For example "agdfBh".
- **Letter Case:** When counting frequencies, we are interested in the case insensitive frequency (i.e. in the text "The sun shines over the lake", the library should count 2 occurrences for any of the words "the" or "The" or "tHE" etc).
- **Input Text:** The input text contains words separated by various separator characters. Note that the characters from 'a' and 'z' and from 'A' and 'Z' can only appear within words.
- **Available Memory:** There is enough memory to store the whole input text.

Implementation

- Create a class which implements the following interface (or the equivalent of an interface in the language that you are using).

```
Name: WordFrequency
Attributes:
    str word { get; }
    int frequency { get; }
```

- Create a class which implements the following interface

```
Name: WordFrequencyAnalyzer
Attributes:
    int calculate_highest_frequency(str text)
    int calculate_frequency_for_word(str text, str word);
    List[WordFrequency] calculate_most_frequent_n_words(str text, int n);
```

Implement the three methods defined in this interface

- `calculate_highest_frequency` should return the highest frequency in the text (several words might actually have this frequency)
- `calculate_frequency_for_word` should return the frequency of the specified word
- `calculate_most_frequent_n_words` should return a list of the most frequent 'n' words in the input text, all the words returned in lower case. If several words have the same frequency, this method should return them in ascendant alphabetical order (for input text "The sun shines over the lake" and $n = 3$, it should return the list { ("the", 2), ("lake", 1), ("over", 1) }

Test Cases

- Implement test cases for this class. You may use 3rd party frameworks to support your tests.

Submission

- Return deliverables as electronic source code files. Please do not include any binary files.
- Please submit all files you think are required to do your solution justice.
- Ensure that you include the names (and versions) of all 3rd party libraries/frameworks used.
- The source code file(s) delivered should be ready to be executed once inserted into a project containing the interface defined in the same way (i.e. into the one we have waiting for the code already).

As a final note, please **follow the instructions** and follow the **industry standards** of the programming language that you use ☺.