

The case: counting words.

Introduction

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 (Front-end Typescript version)

```
//Create a class which implements the following interface:
```

```
interface WordFrequency {  
    getWord(): string;  
    getFrequency(): number;  
};
```

```
//Create a class which implements the following interface:
```

```
interface WordFrequencyAnalyzer {  
    calculateHighestFrequency(text: string): number;  
    calculateFrequencyForWord(text: string, word: string): number;  
    calculateMostFrequentNWords(text: string, n: number): WordFrequency[];  
};
```

Part 1

Implement the three methods defined in this interface

- CalculateHighestFrequency should return the highest frequency in the text (several words might actually have this frequency)
- CalculateFrequencyForWord should return the frequency of the specified word
- CalculateMostFrequentNWords 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}])

Requirements

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

Part 2

Next part of the implementation is to make it visual. It is up to you how you want to do that.

There is no restriction on the front-end framework you choose as long as Typescript is being used.

What you make visual is up to you. How do you want to show us your implementation works as described in the assignment?

Requirements

It should be responsive

Submission

Submit the assignment using Github, Gitlab or any other public repository.

As a final note, please follow the instructions
and perform the tasks on an industry standard level.

Good luck and happy coding!