

## Programming task: Text statistics

### Introduction

In the document there is a programming task that you solve at home in peace and quiet. The solution should be in Java, Node or C#. If you have questions about the task, please contact: [mikael.tennhammar@additude.se](mailto:mikael.tennhammar@additude.se) or the manager who sent you the test.

### Task

Using two interfaces, you will create an application that counts statistics about a text. The program should count instances of words and be able to display the twenty most frequent and the ten longest words in a literary text. Implement it as you would in code to be put into production. We would like to see the solution wire-proof and reasonably fast.

Here you can find literary texts in a variety of formats.

<https://www.gutenberg.org/>

(ex <https://www.gutenberg.org/files/45839/45839.txt>, Dracula by Bram Stoker)

The interfaces are on the next page, they are only the basis, if you need to, just expand them.

You should generate statistics for (at least) two different texts and list the results for each text and the total for all texts. We would like to see the program run against different sources without having to change the code.

## Interface

```
/**
 * Represents a word and its frequency.
 */
public interface WordFrequency {

    /**
     * The word.
     * @return the word as a string.
     */
    String word();

    /**
     * The frequency.
     * @return a long representing the frequency of the word.
     */
    long frequency();
}

public interface TextStatistics {

    /**
     * Returns a list of the most frequented words of the text.
     * @param n how many items of the list
     * @return a list representing the top n frequent words of the text.
     */
    List<WordFrequency> topWords(int n);

    /**
     * Returns a list of the longest words of the text.
     * @param n how many items to return.
     * @return a list with the n longest words of the text.
     */
    List<String> longestWords(int n);

    /**
     * @return total number of words in the text.
     */
    long numberOfWords();

    /**
     * @return total number of line of the text.
     */
    long numberOfLines();
}
```