BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
K. K. BIRLA GOA CAMPUS
FIRST SEMESTER 2023-2024

CS F213 (Object Oriented Programming) - Lab 4, Date: 28/08/24

Lab Problem Statement

You have a class **TextAnalyzer**. It has the following attributes:

- **String paragraph**: Stores the input text
- **String[] words**: Stores unique words from the paragraph
- **int[] wordFrequency**: wordFrequency[i] stores frequency of the ith word in words, i.e., words[i]
- **String[] palindromicWords**: Stores palindromic[1] words
- **String longestWord**: Stores the longest word in the paragraph

You need to implement the following methods:

- You must create the **getters** for all class attributes.
- **void populateWords():** Processes the paragraph for the unique words and populates the words array. Also ensure that the length of the words array is exactly equal to the number of unique words in the paragraph.
- **void populateWordFrequency():** Populates the wordFrequency array with wordFrequency[i] being the frequency of the word words[i] in the paragraph.
- **private boolean isPalindrome(String word):** Helper function that returns true if the word passed as argument is a palindrome.
- **void populatePalindromicWords():** Populates the palindromicWords array with words from the paragraph that are palindromes. The words in this array must also be unique and it must be ensured that the length of the palindromicWords array is exactly equal to the number of unique palindromic words in the paragraph.
- **String populateLongestWord():** Sets the value of the class attribute longestWord as the longest word in the paragraph. If there are two or more words with the same length (that is the maximum), then the longest word should be the one which appears first.
- **void displayResults():** Display the following:
    - Count of unique words
    - Count of unique palindromes
    - Longest word

(Read the instructions at the end of this document.)

---

[1] A palindrome is a word, phrase, number, or other sequence of characters that reads the same forward and backward (ignoring spaces, punctuation, and capitalization).

Example:

Input 1:

Madam Anna walked to the park and the racecar was at the park

Output 1:

Count of unique words: 10
Count of unique palindromes: 1
Longest word: racecar

Input 2:

The radar picked up a level signal just as the racecar zoomed past the control tower at noon

Output 2:

Count of unique words: 17
Count of unique palindromes: 5
Longest word: racecar

Instructions:

Follow the steps given below to complete the OOP lab problem.

Step 1: Read the Lab Question

Read and understand the Lab problem given above.

Step 2: Edit the Solution Java file

Edit the solution java file to solve the given problem. For Lab 4, the solution java file is named: `TextAnalyzer.java`.

Step 3: See the input and the expected output

Use the following command to see the input and the expected output for test case `T1`. Note that `L4` refers to Lab 4, `Q1` refers to Question 1 and `T1` refers to test case 1.

```
:~$ ./RunTestCase L4 Q1 YourBITSId T1
```

Use your own (13 character) BITS Id in place of `YourBITSId` in the above command. Type your BITS Id in upper case (capital letters). Ensure that you enter your BITS Id in 202XA7PSXXXXG format.

Run the command from within the folder containing the executables and the java files.

Step 4: Modify the solution java file

Modify the solution Java file and repeat Step 3 until all the test cases are passed. The test cases are numbered T1, T2, etc.

Step 5: Passing evaluative test cases

There are evaluative test cases whose expected output is masked. The evaluative test cases are numbered ET1, ET2, etc. The lab marks will be based on the evaluative test cases. Use the following command to check whether your solution passes the evaluative test cases.

```
:~$ ./RunTestCase L4 Q1 YourBITSId ET1
```

Ensure that your solution passes all the evaluative test cases ET1, ET2, etc. The expected output of the evaluative test cases are *hidden*.

Step 6: Create submission zip file

Use the following command to create the submission zip file.

```
:~$ ./CreateSubmission L4 YourBITSId
```

The first command line argument must correspond to the lab number and the second command line argument must be your 13 character BITS id.

The above command will create a zip file. The command will also list the evaluative test cases that your solution program has passed.

Step 7: Upload submission zip file

Upload the submission zip file created in Step 6. Do not change the name of the zip file or modify any file inside the zip file. You will not be awarded marks if the zip file is tampered with in any manner.