Computer Science 3A

Practical Assignment 1

9 February 2017

Time: 9 February 2017 13:45 – 17:00 Marks: 50

Practical assignments must be uploaded to `eve.uj.ac.za` **before** 17h00 in the practical session.

Late submissions **will not be accepted**, and will therefore not be marked. You are **not allowed to collaborate** with any other student. You **must** upload your assignment to Eve **before** it will be marked. You **must** include your Javadoc for each question in your submission. If you do not include it, **you will get a zero for your entire question**.

One problem experienced on the Internet is garbled or random text generated by computers, which are then posted on social media, forums or are submitted in assignments. It not only wastes bandwidth, but it hinders the experience of legitimate users on certain online platforms and frustrates lecturers. In order to address this problem the University of Johannesburg has tasked you with the job of building a Java application that can detect fake, garbled or randomly generated text in text files.

You are required to implement the following text-based functions:

- **readTextfileFully** – which reads a text file fully and returns its content as a String.
- **get and set methods** – which provides accessor and modifier functions for the source text found in the *UJRubbishIndex* class.
- **getWordCount** – which counts the amount of words found in the source text.
- **getLongWordCount** – which counts the amount of long words found in the source text.
- **getSentenceCount** – which counts the amount of sentences found in the source text.
- **calculateUJRubbishIndex** – which takes the sentence, long word and word counts in order to calculate the UJ rubbish index. The formula is for the index is: $ujRubbishIndex(sentenceCount, longWordCount, wordCount) = \frac{sentenceCount}{wordCount} * 0.8 + \frac{LongWordCount}{wordCount} * 0.2$
- **isFake** – which takes in the index and returns whether the it represents fake ($ujRubbishIndex > 0.1$) or real text.

You are required to implement a Java Program that realises the above operations. You must implement the following items:

1. `UJRubbishIndex - 25 Marks` — a class that contains all the required functions to calculate the UJ Rubbish Index that hints at whether text has been generated or if it is natural language.
   (a) Your class should have a private text-based data member.
   (b) Methods to access and modify the text-based data member.
   (c) Appropriate constructors and an overloaded toString.
   (d) The operations listed above.
2. `Main - 25 Marks` — A main function class that can be used to demonstrate all the functionality of your application using the provided 4 text files (*Computer_Science.txt*, *Random.txt*, *Lorem_ipsum* and *IT.txt*). Your class should implement all of the indicated Exceptions in the appropriate functions. All aspects of your UJRubbishIndex class should be displayed. An example of your output should look like:

```
Text: ÃŕÂżÂ£Computer ScienceComputer science is the study of the theory, experimentat
Word count: 1322
Long Word count: 164
Sentence count: 60
UJ Rubbish Index: 0.06111951588502269
Fake?: false

Text: ÃŕÂżÂ£Dashwood contempt on mr unlocked resolved provided of of. Stanhill wonder
Word count: 623
Long Word count: 9
Sentence count: 83
UJ Rubbish Index: 0.10947030497592297
Fake?: true

Text: ÃŕÂżÂ£Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean ut nulla
Word count: 890
Long Word count: 40
Sentence count: 112
UJ Rubbish Index: 0.10966292134831462
Fake?: true

Text: Information TechnologyInformation technology (IT) is the application of compute
Word count: 537
Long Word count: 76
Sentence count: 25
UJ Rubbish Index: 0.06554934823091248
Fake?: false
```

The following files must be submitted to EVE:

1. *studentnumber*_p1.zip

Marksheet

1.	Main: readTextfileFully	[5]
2.	Main: main function	[5]
3.	UJRubbishIndex: get and set source text	[2]
4.	UJRubbishIndex: getWordCount	[5]
5.	UJRubbishIndex: getSentenceCount	[5]
6.	UJRubbishIndex: getLongSentenceCount	[5]
7.	UJRubbishIndex: calculateUJRubbishIndex	[5]
8.	UJRubbishIndex: isFake	[5]
9.	UJRubbishIndex: toString	[3]
10.	Compilation and Correct execution.	[10]