CS102 – Algorithms and Programming II
Lab Programming Assignment 3
Fall 2016

ATTENTION:

- Feel free to ask questions on Moodle on the Lab Assignment Forum.
- Compress all of the Java program source files (.java) files into a single zip file.
- The name of the zip file should follow the below convention:

  CS102_SecX_Asgn3_YourSurname_YourName.zip
- Replace the variables "YourSurname" and "YourName" with your actual surname and name and X with your Section id (1, 2 or 3).
- Upload the above zip file to Moodle by the deadline before the lab (if not significant points will be taken off). You will get a chance to update and improve your solution by consulting to the TA during the lab. You will resubmit your code once you demo your work to the TA.

GRADING WARNING:

- Please read the grading criteria provided on Moodle.

Q1 [100 p.] In this lab, you will implement a program for analyzing documents in terms of their word frequencies. You will implement the following classes:

1. You will implement a **Document** class to represent documents. The class should have at least the following methods and instance variables (you may include new methods and variables if you feel the necessity):
   - The Document objects should store the filename of the Document being created and a private array of **Term** objects (which is explained in the second part). You will create a Term object for each unique word in the document and their count and store these Term objects in this array. Do not use an arraylist. You may assume the documents will contain at most 10.000 terms. You may include other instance variables or static variables if you feel the necessity.
   - A default constructor
   - A constructor which takes a string *fileName* and sets the filename of the document. The array is initialized.
   - **void processDocument()**
     This method should read the document and fill the array of Term objects based the words in the document and their number of appearance.
   - **int getCount(String word)**
     This method should return the number of appearance of a given word.
   - **int getFrequency(String word)**
     This method should take a word as a parameter, calculate and return the term frequency (*tf*) of that word. Term frequency is the number of times the given word appeared in the document divided by the number of words in the document.
   - **Term getMostFrequentTerm()**
     This method should return the most frequent term in the document as Term object.

- Getter and setter methods.

2. To store terms, you will implement a class **Term**. The purpose of this class is to store the word and its total number of appearance in the document.

The class should have the following:

- a constructor which takes a string and initialize the number of appearance of the word with 0.
- a private string ***word***
- a private integer ***count***
- **void incrementCount()**
  This method should increase the value of *count* by 1 whenever it is called.
- Getter and setter methods.

3. To test your program, implement a class **DocumentAnalyzer**. In this class, you first take the number of documents that the user will enter. Then, you will take the filenames of these documents. You should store these document in an array as Document objects. To analyze these documents with different approaches, you will implement a menu. In this menu different options should be provided to the user for:

- displaying the frequency of a user entered word in a given document (option 1).
- finding the most frequent terms in each document (option 2). It should print out the most frequent term for each document and the number of appearance in the document.
- for calculating tf-idf of user entered word (see below) for each document (option 3). It should print out the result of these calculation in a proper way.

DocumentAnalyzer class should include public static method named **calculateTfidf(Document[ ] docs, Document doc, String word)**. This method will calculate the tf-idf of a given word in a document with respect to the all documents and returns the result. The term frequency–inverse document frequency (tf–idf), is a numerical statistic that reflects how important a word is to a document in a collection of documents. It is used widely in information retrieval systems.

Tf-idf is calculated as follows:

$$tfidf(t,d,D) = tf(t,d) * idf(t,D)$$

where t is the word, d is the current document for which we are calculating tf-idf and D is the set of all documents. tf, i.e. term frequency, is the frequency of the term. idf, i.e. inverse document frequency, is used to distinguish meaningful and discriminative terms. The inverse document frequency is a measure of how much information the word carries. For example “the” is a common across all documents, but “cpu” will appear on subset of documents related to computers.

To calculate $tf(t,d)$:

$$tf(t,d) = (number\ of\ times\ t\ appeared\ in\ d) / (total\ number\ of\ words\ in\ d)$$

To calculate $idf(t, D)$:

$idf(t, D) = log$ ((*number of total documents in* D) / (*number of documents including* t *in* D))

There are ways to smooth the counts and so on but we will not use this. To read more about tf-idf and other tf and idf calculations, see the Wiki page.

You will need a second method to calculate tf-idfs for all documents in the collection. Use proper method decomposition in your programs.

Sample Output

```
Enter the number of documents:
3
Enter the name of the document 1:
text1.txt
Enter the name of the document 2:
text2.txt
Enter the name of the document 3:
text3.txt
Enter an option (enter 4 to quit):
3
Enter the word which you want to calculate tf-idf:
hut
Document 1
 word: hut
 tf-idf: 0.003526
Document 2
 word: hut
 tf-idf: 0.000000
Document 3
 word: hut
 tf-idf: 0.002917
Enter an option (enter 4 to quit):
2
Document 1
 word: and
 number of appearance: 8
Document 2
 word: to
 number of appearance: 6
Document 3
 word: the
 number of appearance: 10
Enter the option (enter 4 to quit):
1
Enter the word which you want to find the frequency:
```

```
simon

Document 1
 word: simon
 term frequency: 0.017391
Document 2
 word: simon
 term frequency: 0.000000
Document 3
 word: simon
 Term frequency: 0.021583
Enter the option (enter 4 to quit):
4
Done!
```

Texts are taken from the book “What Men Live By” written by Lev Tolstoy.