Instructions for Program WordCount

This project consists of writing a program to read in a file (provided) and count the occurrences of all of the words in the file.

You are expected to use the **Map<String, Integer>** interface. You will use the two different implementations of **Map**: **HashMap** and **TreeMap**. We will discuss the details of their implementations later in the semester, but you should understand how to use these implementations. One of the fundamental differences is that the **TreeMap** implementation produces a sorted collection based on the **key** field of the **Map** while the **HashMap** produces a random collection. However, the **HashMap** implementation is more efficient than the **TreeMap** implementation. The choice of which to use depends on the need for performance versus the need for an order **Set** of **key**s.

The input file (**"DoI.txt"**) contains the text of the Declaration of Independence. It contains long paragraphs as well as short paragraphs. You will need to edit the contents of what you read in to facilitate the counting process. These edits consist of:

- converting all letters to lower case so that the word list is case insensitive
- removing all digits (only words are processed)
- removing all punctuation marks (e.g., periods, commas, etc.)
- hyphens (dashes) are replaced with spaces making the two parts of a hyphenation word into separate words.

You will need to do some research into the use of the following classes and concepts:

- **class File**
- **class FileInputStream**
- **class IOException**
- **interface Map**
    - the inner **interface Map.Entry**
- **class HashMap**
- **class TreeMap**
- **class Arrays**
- **interface Comparator**
- the use of generics

I would suggest that you generate the following methods for your program:

- **private static long buildMap(String[] words, Map<String, Integer> map)**
- **private static long displayMap(Map<String, Integer> map)**
- **private static String[] processInputFile(String fileName)**

See the included JavaDoc for the **WordCount** class for more details of these methods and include your own JavaDoc comments as appropriate.

The **main()** method will use the above methods to assist it.

I am including two other files for you to look at:

- **WordCount.out** – the display of the **hashMap** and **treeMap** objects and the display of a list of words ordered by usage from high to low.
- **WordCount.err** – the display of the performance statistics that you should gather and display.

Your task is to recreate these files as output files (using the **java** commands redirection operators: "**`java WordCount > WordCount.out 2> WordCount.err`**").

Note: I had trouble with getting the usage ordered list of words and had to create a new, non-generic, inner class **Pair** which contains two instance fields: **String key** and **Integer value**, and the appropriate constructor.