**Problem**

Write a Java program that reads a collection of files and prints the top N words ordered by their occurrences across all files.

**Input**

Reads the following from command line arguments:

1. An integer representing the top N words.
2. A list of paths that represent the directories and/or files to traverse.

**Input errors**

The program must print a descriptive usage upon receiving wrong number of parameters or badly formatted parameters.

**Output**

The program prints the N words that have the most occurrences in descending occurrence order in the following format:

<word 1> occurred <x> times  
<word 2> occurred <y> times

**Deliverable**

Please provide the code, with any testing examples you run.

**Requirements**

**Functional**

* If a specified path is a directory, that directory should be scanned for files, and subdirectories are also scanned recursively.
* Words are separated with whitespace characters. Other characters cannot be considered as delimiters. "re-design" is a single word, not two.
* Words comparison is case insensitive. These words are the same: “Bank”, “BANK”, “bank”, “baNk”.
* When two words share the same count, they are sub-sorted in alphabetical order, ignoring case.

**Performance**

Parallel processing should be used to decrease the amount of time it will take to parse all the files.

**Assumptions**

* All files are valid UTF-8 text files.
* All other assumptions made should be documented with your reasoning.