```java
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.TreeMap;

import components.queue.Queue;
import components.queue.Queue1L;
import components.simplereader.SimpleReader;
import components.simplereader.SimpleReader1L;
import components.simplewriter.SimpleWriter;
import components.simplewriter.SimpleWriter1L;

/**
 * Takes input file and outputs number of occurrences for each word into an HTML
 * file.
 *
 * @author Gage Farmer
 *
 */
public final class WordCounter {

     * whitespace.
    private static String[] WHITESPACE = { "-", " ", ".", ",", "!", "?", ";",
            "$", "%" };

     * No argument constructor--private to prevent instantiation.
    private WordCounter() {

    /**
     * Scans through the input to get all words.
     *
     * @param words
     * @param input
     *
     */
    private static void getWords(SimpleReader input, Queue<String> words) {

        /*
         * Scan through the file and return only space-less lines
         */
        while (!input.atEOS()) {
            String temp = input.nextLine();

            if (!(temp.contains(" ") && temp.isEmpty())) {
                String[] subList = temp.split(" ");

                for (int idx = 0; idx < subList.length; idx++) {
                    String[] cleanedWords = cleanWord(subList[idx]);
                    for (String clean : cleanedWords) {
                        words.enqueue(clean);
                    }

                }
            }

        }
    }

    /**
```

```java
     * helper method for getWords, just cleans up the list by removing any
     * special characters, and splitting any combined words.
     *
     * @param word
     *            words
     * @return cleaned word
     */
    private static String[] cleanWord(String word) {

        // gee i sure hope there isn't an edge case where the array of 10 is too
small!
        Boolean t = true;
        String[] result = null;
        word = word.toLowerCase();

        for (String special : WHITESPACE) {
            if (word.contains(special) && t == true) {
                result = word.split(special);
            }
        }
        if (result == null) {
            result = word.split("SUPERDUPERTOPSECRETPHRASE!!!!DON'TTYPEMEEE");
        }

        return result;
    }

     * Converts two queues to one sorted treemap
    private static TreeMap queueToTreeMap(Queue<String> word) {

    /**
     * Formats and prints the graph into html.
     *
     * @param list
     * @param output
     * @throws IOException
     */
    private static void printGraph(TreeMap list, FileWriter output)
            throws IOException {

        /*
         * Intentionally skipping the first key and value (it is whitespace)
         */
        String key = (String) list.firstKey();
        int value = (int) list.get(key);
        list.remove(key);
        int idx = 0;

        output.write(
                "<table style=margin-left:auto;margin-right:auto;> \n <tr> <th>"
                        + "Word</th> <th>Occurances</th> </tr> \n");

        while (list.size() > 0) {
            key = (String) list.firstKey();
            value = (int) list.get(key);
            output.write(
                    "<tr> <th>" + key + "</th><th>" + value + "</th> </tr>\n");
            list.remove(key);
        }

    }

    /**
     * creates and outputs html header.
     *
     * @param output
     *            file to print to
     * @param fileName
     *            name of de file
     * @throws IOException
     */
    private static void printHeader(FileWriter output, String fileName)
            throws IOException {

        output.write("<!DOCTYPE html>\n");
        output.write("<html>\n" + "<style>\n" + "table, th, td {\n"
                + "     border:2px solid red;color:#FFFFFF;\n"

                + "}\n" + "#grad1 {\n" + "  height: 55px;\n"
                + "  background-color: red;\n"
                + "  background-image: linear-gradient(to right, red, orange, "
                + "yellow, violet);\n" + "}\n" + "</style>\n" + "<head>\n"
                + "<div id=\"grad1\" style=\"text-align:center;margin:auto;"
                + "color:#FFFFFF;font-size:40px;font-weight:bold\">\n"
                + "Words Counted in " + fileName + "\n" + "</div>" + "</head>\n"
                + "<body style=\"background-color:#353535;\">\n");

    }

    /**
     * Main method.
     *
     * @param args
     *            the command line arguments
     * @throws IOException
     */
    public static void main(String[] args) throws IOException {
        SimpleReader in = new SimpleReader1L();
        SimpleWriter out = new SimpleWriter1L();

        /*
         * Gonna try using queues to keep track of these things
         */
        Queue<String> words = new Queue1L<>();

        /*
         * Gets name of file and opens it
         */
        out.println("Enter file name: ");
        String name = in.nextLine();
        SimpleReader input = new SimpleReader1L(name);

        /*
         * Gets name of output folder
         */
        out.println("Enter the location of the output folder: ");
        String name2 = in.nextLine();

        /*
         * Creates html file
         */
        File file = new File(name2 + "/wordcount.html");
        FileWriter output = new FileWriter(name2 + "/wordcount.html");
        printHeader(output, name);

        /*
         * Gets word data + occurrences to a list
         */
        getWords(input, words);
        TreeMap list = queueToTreeMap(words);

        /*
         * Turn that list into an HTML graph Print html footer
         */
        printGraph(list, output);

        /*
         * Close input and output streams
         */
        in.close();
        input.close();
        out.close();
        output.close();

    }

}

```