```java
package proj5;

/**
 * class for computing word frequencies from a text file
 * author: Son Nguyen (Kyrie)
 * version: 6/3/2020
 */
public class WordCounter {

    // instance variables
    private LineReader paragraph;
    private String[] currentLine;
    private BinarySearchTree<Counter> myWordCounter;

    /**
     * Default constructor
     */
    public WordCounter() {
        paragraph = null;
        currentLine = null;
        myWordCounter = new BinarySearchTree<Counter>();
    }

    /**
     * helper method to slice a word including punctuation to its content only
     * @param word
     * @return word's content
     */
    private String splitWord(String word) {
        String toReturn = word;
        char firstChar = word.charAt(0);
        if (!Character.isLetter(firstChar)) {
            if (toReturn.length() == 1) {
                return "";
            }
            toReturn = toReturn.substring(1);
        }

        int lastChar = word.charAt(word.length() - 1);
        if (!Character.isLetter(lastChar)) {
            if (toReturn.length() == 1) {
                return "";
            }
            toReturn = toReturn.substring(0, toReturn.length() - 1);
        }
        return toReturn;
    }

    /**
     * Computes frequency of each word in given file
     * @param file path to file, such as "src/input.txt"
     */
    public void findFrequencies(String file) {
        paragraph = new LineReader(file, " ");
        currentLine = paragraph.getNextLine();
        myWordCounter = new BinarySearchTree<Counter>();
        while (currentLine != null) {
            for (String rawWord: currentLine) {
                String currentWord = splitWord(rawWord.toLowerCase());
                if (currentWord != "") {
                    Counter current = new Counter(currentWord);
                    Counter counterpart = myWordCounter.search(current);
                    if (counterpart == null) {
                        counterpart = current;
                        myWordCounter.insert(counterpart);
                    }
                    counterpart.increment();
                }
            }
            currentLine = paragraph.getNextLine();
        }
        paragraph.close();
    }

    /**
     * returns the frequency of the given word
     * @param word word - string to get the frequency
```

```
 of
     * @return the number of times word appears in the input file
     */
    public int getFrequency(String word) {
        Counter toReturn = new Counter(word);
        Counter counterpart = myWordCounter.search(toReturn);
        if (counterpart != null) {
            return counterpart.getCount();
        }
        return 0;
    }

    /**
     * @return words and their frequencies as a printable String.
     * Each word/frequency pair should be on a separate line,
     * and the format of each line should be <word> : <frequency>
     * For example,
     * are: 3
     * bacon: 2
     *
     * Words should be in alphabetical order.
     */
    public String toString() {
        return myWordCounter.toString();
    }
}

```