WORKED EXAMPLE 15.1 Word Frequency

Problem Statement Write a program that reads a text file and prints a list of all words in the file in alphabetical order, together with a count that indicates how often each word occurred in the file.

For example, the following is the beginning of the output that results from processing the book *Alice in Wonderland*:

```
a                                653
abide                              1
able                               1
about                             97
above                              4
absence                            1
absurd                             2
```

Step 1 Determine how you access the values.

In our case, the values are the word frequencies. We have a frequency value for every word. That is, we want to use a map that maps words to frequencies.

Step 2 Determine the element types of keys and values.

Each word is a `String` and each frequency is an `Integer`. (You cannot use an `int` as a type parameter because it is a primitive type.) Therefore, we need a `Map<String, Integer>`.

Step 3 Determine whether element or key order matters.

We are supposed to print the words in sorted order, so we will use a `TreeMap`.

Step 4 For a collection, determine which operations must be efficient.

We skip this step because we use a map, not a collection.

Step 5 For hash sets and maps, decide what to do about the `equals` and `hashCode` methods.

We skip this step because we use a tree map.

Step 6 If you use a tree, decide whether to supply a comparator.

The key type for our tree map is `String`, which implements the `Comparable` interface. Therefore, we need to do nothing further.

We have now chosen our collection. The program for completing our task is fairly simple. Here is the pseudocode:

```
For each word in the input file
   Remove non-letters (such as punctuation marks) from the word.
   If the word is already present in the frequencies map
      Increment the frequency.
   Else
      Set the frequency to 1.
```

Here is the program code:

worked_example_1/WordFrequency.java

```
1  import java.util.Map;
2  import java.util.Scanner;
3  import java.util.TreeMap;
4  import java.io.File;
5  import java.io.FileNotFoundException;
```

```
6
7  /**
8     This program prints the frequencies of all words in "Alice in Wonderland".
9  */
10 public class WordFrequency
11 {
12    public static void main(String[] args)
13       throws FileNotFoundException
14    {
15       Map<String, Integer> frequencies = new TreeMap<>();
16       Scanner in = new Scanner(new File("alice30.txt"));
17       while (in.hasNext())
18       {
19          String word = clean(in.next());
20
21          // Get the old frequency count
22
23          Integer count = frequencies.get(word);
24
25          // If there was none, put 1; otherwise, increment the count
26
27          if (count == null) { count = 1; }
28          else { count = count + 1; }
29
30          frequencies.put(word, count);
31       }
32
33       // Print all words and counts
34
35       for (String key : frequencies.keySet())
36       {
37          System.out.printf("%-20s%10d\n", key, frequencies.get(key));
38       }
39    }
40
41    /**
42       Removes characters from a string that are not letters.
43       @param s a string
44       @return a string with all the letters from s
45    */
46    public static String clean(String s)
47    {
48       String r = "";
49       for (int i = 0; i < s.length(); i++)
50       {
51          char c = s.charAt(i);
52          if (Character.isLetter(c))
53          {
54             r = r + c;
55          }
56       }
57       return r.toLowerCase();
58    }
59 }
```