

```

1 // foj/4/dictionary/Dictionary.java
2 //
3 //
4 // Copyright 2003 Bill Campbell and Ethan Bolker
5
6 import java.util.*;
7
8 /**
9  * Model a dictionary with a TreeMap of (word, Definition) pairs.
10  *
11  * @see Definition
12  *
13  * @version 4
14  */
15
16 public class Dictionary
17 {
18     private TreeMap entries;
19
20     /**
21      * Construct an empty Dictionary.
22      */
23     public Dictionary()
24     {
25         entries = new TreeMap();
26     }
27
28     /**
29      * Add an entry to this Dictionary.
30      *
31      * @param word the word being defined.
32      * @param definition the Definition of that word.
33      */
34     public void addEntry( String word, Definition definition )
35     {
36         entries.put( word, definition );
37     }
38
39     /**
40      * Look up an entry in this Dictionary.
41      *
42      * @param word the word whose definition is sought
43      * @return the Definition of that word, null if none.
44      */
45     public Definition getEntry( String word )
46     {
47         return (Definition)entries.get(word);
48     }
49
50     /**
51      * Get the size of this Dictionary.
52      *
53      * @return the number of words.
54
55
56

```

```

57     */
58     public int getSize()
59     {
60         return entries.size();
61     }
62
63     /**
64      * Construct a String representation of this Dictionary.
65      *
66      * @return a multiline String representation.
67      */
68     public String toString()
69     {
70         String str = "";
71         String word;
72         Definition definition;
73         Set allWords = entries.keySet();
74         Iterator wordIterator = allWords.iterator();
75         while ( wordIterator.hasNext() ) {
76             word = (String)wordIterator.next();
77             definition = this.getEntry( word );
78             str += word + ":\n" + definition.toString() + "\n";
79         }
80         return str;
81     }
82
83
84 }

```