```
1   // joi/4/textfiles/Directory.java
2   //
3   //
4   // Copyright 2003 Ethan Bolker and Bill Campbell
5   //
6   // This draft contains just stubs for the methods.
7   // You can invoke them all, but none will do anything.
8
9   /**
10   * Directory of TextFiles.
11   *
12   * @version 4
13   */
14
15  public class Directory
16  {
17      /**
18       * Construct a Directory.
19       */
20
21      public Directory( )
22      {
23      }
24
25      /**
26       * The size of a directory is the number of TextFiles it contains.
27       *
28       * @return the number of TextFiles.
29       */
30
31      public int getSize()
32      {
33          return 0;
34      }
35
36      /**
37       * Add a TextFile to this Directory. Overwrite if a TextFile
38       * of that name already exists.
39       *
40       * @param name the name under which this TextFile is added.
41       * @param afile the TextFile to add.
42       */
43
44      public void addTextFile(String name, TextFile afile)
45      {
46      }
47
48      /**
49       * Get a TextFile in this Directory, by name .
50       *
51       * @param filename the name of the TextFile to find.
52       * @return the TextFile found, null if none.
53       */
54
55      public TextFile retrieveTextFile( String filename )
56      {
```

```
57          return null;
58      }
59
60      /**
61       * Get the contents of this Directory as an array of
62       * the file names, each of which is a String.
63       *
64       * @return the array of names.
65       */
66
67      public String[] getFileNames()
68      {
69          // pseudocode for an implementation:
70          // declare an array of String
71          // create that array with as many spaces as there
72          // are TextFile's in this Directory
73          // loop through the keys of the TreeMap of TextFiles,
74          // adding each String key to the array
75          // return the array
76
77          // the next line is there because we have to return
78          // _something_ in order to satisfy the compiler
79          return new String[0];
80      }
81
82      /**
83       * main, for unit testing.
84       *
85       * The command
86       * <pre>
87       * java Directory
88       * </pre>
89       * should produce output
90       * <pre>
91       * bill    17    Sun Jan 06 19:40:13 EST 2003    diary
92       * eb      12    Sun Jan 06 19:40:13 EST 2003    greeting
93       * </pre>
94       * (with current dates, of course).
95       */
96
97      public static void main( String[] args )
98      {
99          Directory dir = new Directory();
100         dir.addTextFile("greeting", new TextFile("eb", "Hello, world"));
101         dir.addTextFile("diary", new TextFile("bill", "Writing Directory"
102         // now list TextFiles in dir to get output specified
103     }
104 }
```