```
 1  // joi/6/jfiles/TextFile.java
 2  //
 3  //
 4  // Copyright 2003 Ethan Bolker and Bill Campbell
 5
 6  /**
 7   * A TextFile is a JFile that holds text.
 8   *
 9   * @version 6
10   */
11
12  public class TextFile extends JFile
13  {
14      private String contents;  // The text itself
15
16      /**
17       * Construct a TextFile with initial contents.
18       *
19       * @param name    the name for this TextFile (in its parent Directory
20       * @param creator the owner of this new TextFile
21       * @param parent  the Directory in which this TextFile lives.
22       * @param initialContents the initial text
23       */
24
25      public TextFile( String name, User creator, Directory parent,
26                       String initialContents )
27      {
28          super( name, creator, parent );
29          setContents( initialContents );
30      }
31
32      /**
33       * Construct an empty TextFile.
34       *
35       * @param name    the name for this TextFile (in its parent Directory
36       * @param creator the owner of this new TextFile
37       * @param parent  the Directory in which this TextFile lives
38       */
39
40      TextFile( String name, User creator, Directory parent )
41      {
42          this( name, creator, parent, "" );
43      }
44
45      /**
46       * The size of a text file is the number of characters stored.
47       *
48       * @return the file's size.
49       */
50
51      public int getSize()
52      {
53          return contents.length();
54      }
55
56      /**
```



```
57       * Suffix used for printing text file names is "".
58       *
59       * @return an empty suffix (for TextFiles).
60       */
61
62      public String getSuffix()
63      {
64          return "";
65      }
66
67      /**
68       * Replace the contents of the file.
69       *
70       * @param contents the new contents.
71       */
72
73      public void setContents( String contents )
74      {
75          this.contents = contents;
76          setModDate();
77      }
78
79      /**
80       * The contents of a text file.
81       *
82       * @return String contents of the file.
83       */
84
85      public String getContents()
86      {
87          return contents;
88      }
89
90      /**
91       * Append text to the end of the file.
92       *
93       * @param  text the text to be appended.
94       */
95
96      public void append( String text )
97      {
98          setContents( contents + text );
99      }
100
101
102     /**
103      * Append a new line of text to the end of the file.
104      *
105      * @param  text the text to be appended.
106      */
107
108     public void appendLine( String text )
109     {
110         this.setContents(contents + '\n' + text);
111     }
112 }
```