File.java

```
 1  package caltool.model.file;
 2
 3  import mvp.*;
 4  import caltool.model.caldb.*;
 5
 6  /****
 7   *
 8   * Class File is the model class for the Calendar Tool file handling.  It
 9   * contains methods for all of the operations defined on the File menu, which
10   * constitute the functional command group for file handling.
11   *
12   * @author Gene Fisher (gfisher@calpoly.edu)
13   * @version 13apr15
14   *
15   */
16  public class File extends Model {
17
18      /**
19       * Construct this with the given companion view and the parent CalendarDB
20       * model.  The CalendarDB is provided for its service methods that access
21       * the Calendar Tool workspace.
22       */
23      public File(View view, CalendarDB calDB) {
24          super(view);
25          this.calDB = calDB;
26      }
27
28      /*-*
29       * Derived methods
30       */
31
32      /**
33       * Add a new empty calendar to the workspace and make it current.
34       */
35      public void fileNew() {
36          System.out.println("In File.fileNew");
37      }
38
39      /**
40       * Open an existing calendar file of the given name and put the data from
41       * that file in the workspace.
42       */
43      public void open(String filename) {
44          System.out.println("In File.open");
45      }
46
47      /**
48       * Close the current calendar if it does not require saving.  If saving is
49       * required, ask the user what to do.
50       */
51      public void close() {
52          System.out.println("In File.close");
53      }
54
55      /**
56       * Close the all open calendars if they do not require saving.  If saving
57       * is required, ask the user what to do.
58       */
59      public void closeAll() {
60          System.out.println("In File.closeAll");
61      }
62
63      /**
64       * If the current calendar in the workspace requires saving, save it.
65       */
66      public void save() {
67          System.out.println("In File.save");
68      }
69
70      /**
71       * Save the current calendar in a file of the given name.
72       */
73      public void saveAs(String filename) {
74          System.out.println("In File.saveAs");
75      }
76
77      /**
78       * For each open calendar in the workspace, save it if it requires saving.
79       */
80      public void saveAll() {
81          System.out.println("In File.saveAll");
82      }
83
84      /**
85       * Save the current workspace configuration, including the positions of all
86       * open view windows.
87       */
88      public void saveConfig() {
89          System.out.println("In File.saveConfig");
90      }
91
92      /**
93       * Print the current calendar per the given print specs.
94       */
95      public void print(PrintSpecs printSpecs) {
96          System.out.println("In File.print");
97      }
98
99      /**
100      * Exit the Calendar Tool.  If saving is required for any open calendars,
101      * ask the user what to do.
102      */
103     public void exit() {
104         System.out.println("In File.exit.");
105         System.exit(0);
106     }
107
108     /**
109      * Temporary system test method to dump out the current user calendar to
110      * stdout.
111      */
112     public void dumpUserCal() {
```

```
113          System.out.println(calDB.getCurrentCalendar().toString());
114      }
115
116
117      /*-*
118       * Data fields
119       */
120
121      /** The CalendarDB, containing the data to be stored onto files and into
122          which file data are read. */
123      CalendarDB calDB;
124
125  }
```