

```

1  package caltool.view.file;
2
3  import caltool.model.file.*;
4  import javax.swing.*;
5  import java.awt.*;
6  import java.awt.event.*;
7  import java.lang.reflect.*;
8  import mvp.*;
9
10 /**
11  *
12  * Class FileMenu is the pulldown menu view of the <a href =
13  * "../file/File.html" File </a> model class. The FileMenu widget is a Java
14  * JMenu. Anonymous instances of JMenuItem are defined for each item in the
15  * menu.
16  *
17  * @author Gene Fisher (gfisher@calpoly.edu)
18  * @version 13apr15
19  *
20  */
21 public class FileMenu extends View {
22
23     /**
24      * Construct this with the given File model.
25      */
26     public FileMenu(Screen screen, File file, FileUI fileUI) {
27
28         /**
29          * Call the parent constructor.
30          */
31         super(screen, file);
32
33         /**
34          * Create a local reference to super.model for non-casting convenience.
35          * This way, the model can be referenced as "file" instead of
36          * "(File)model".
37          */
38         this.file = file;
39
40     }
41
42     /**
43      * Compose this by inserting each of its nine menu items into the pulldown
44      * menu. The items are New, Open, Close, Close All, Save, Save As, Save
45      * All, Print, and Exit. JSeparators are placed after the 'Close All',
46      * 'Save All', and 'Print' items.
47      */
48     public Component compose() {
49
50         /**
51          * Make the widget of this the JMenu.
52          */
53         widget = new JMenu("File");
54
55         /**
56          * Create a local convenience ref to super.widget.
57
58          */
59         menu = (JMenu) widget;
60
61         /**
62          * Add each of the items to the menu, with separators in the
63          * appropriate places (per the requirements).
64          */
65         addNewItem();
66         addOpenItem();
67         addCloseItem();
68         addCloseAllItem();
69         menu.add(new JSeparator());
70
71         addSaveItem();
72         addSaveAsItem();
73         addSaveAllItem();
74         menu.add(new JSeparator());
75
76         addSaveConfigItem();
77         menu.add(new JSeparator());
78
79         addPrintItem();
80         menu.add(new JSeparator());
81
82         addExitItem();
83         menu.add(new JSeparator());
84         addSystemTestItems();
85
86         return widget;
87     }
88
89     /**-
90      * Protected methods
91      */
92
93     /**
94      * Add the 'New' menu item. Its action listener invokes the File.fileNew
95      * method.
96      *
97      * This and all other menu items are created using the following pattern:
98      *
99      *
100      * JMenu.add(new JMenuItem("<em>Item name</em>")).addActionListener(
101      *     new ActionListener() {
102      *         public void actionPerformed(ActionEvent e) {
103      *             <em>Model.method()</em>
104      *         }
105      *     }
106      * );
107
108     /**
109      * Use the standard menu item pattern for the item.
110      */
111     menu.add(new JMenuItem("New")).addActionListener(
112         new ActionListener() {

```

```

113         public void actionPerformed(ActionEvent e) {
114             /*
115              * Call the stubbed-out model method.
116              */
117             file.fileNew();
118         }
119     };
120 };
121 }
122
123 /**
124  * Add the 'Open ...' menu item. Its action listener invokes the File.open
125  * method. See the description of the <a href= FileMenu.html#addNewItem>
126  * addNewItem </a> method for further info.
127  */
128 protected void addOpenItem() {
129
130     final JMenuItem item;          // Used by JFileChooser for placement
131
132     menu.add(
133         item = new JMenuItem("Open ...").addActionListener(
134             new ActionListener() {
135                 public void actionPerformed(ActionEvent e) {
136                     /*
137                      * For the stubbed design, just display a JFileChooser.
138                      * When things are refined, we'll actually use the file
139                      * chooser to select a file.
140                      */
141                     (new JFileChooser()).showOpenDialog(item);
142
143                     /*
144                      * Call the stubbed-out model method.
145                      */
146                     file.open(null);
147                 }
148             }
149         );
150 }
151
152 /**
153  * Add the 'Close' menu item. Its action listener invokes the File.close
154  * method. See the description of the <a href= FileMenu.html#addNewItem>
155  * addNewItem </a> method for further info.
156  */
157 protected void addCloseItem() {
158     menu.add(new JMenuItem("Close").addActionListener(
159         new ActionListener() {
160             public void actionPerformed(ActionEvent e) {
161                 file.close();
162             }
163         }
164     ));
165 }
166
167 /**
168  * Add the 'Close All menu item. Its action listener invokes the
169
170     * File.closeAll method. See the description of the <a href=
171     * FileMenu.html#addNewItem> addNewItem </a> method for further info.
172     */
173     protected void addCloseAllItem() {
174         menu.add(new JMenuItem("Close All").addActionListener(
175             new ActionListener() {
176                 public void actionPerformed(ActionEvent e) {
177                     file.closeAll();
178                 }
179             }
180         ));
181     }
182
183     /**
184     * Add the 'Save' menu item. Its action listener invokes the File.save
185     * method. See the description of the <a href= FileMenu.html#addNewItem>
186     * addNewItem </a> method for further info.
187     */
188     protected void addSaveItem() {
189         menu.add(new JMenuItem("Save").addActionListener(
190             new ActionListener() {
191                 public void actionPerformed(ActionEvent e) {
192                     file.save();
193                 }
194             }
195         ));
196     }
197
198     /**
199     * Add the 'Save As ...' menu item. Its action listener invokes the
200     * File.saveAs method. See the description of the <a href=
201     * FileMenu.html#addNewItem> addNewItem </a> method for further info.
202     */
203     protected void addSaveAsItem() {
204         final JMenuItem item;          // Used by JFileChooser for placement
205
206         menu.add(
207             item = new JMenuItem("Save As ...").addActionListener(
208                 new ActionListener() {
209                     public void actionPerformed(ActionEvent e) {
210                         (new JFileChooser()).showSaveDialog(item);
211                         file.saveAs(null);
212                     }
213                 }
214             ));
215     }
216
217     /**
218     * Add the 'Save All' menu item. Its action listener invokes the
219     * File.saveAll method. See the description of the <a href=
220     * FileMenu.html#addNewItem> addNewItem </a> method for further info.
221     */
222     protected void addSaveAllItem() {
223         menu.add(new JMenuItem("Save All").addActionListener(
224             new ActionListener() {
225                 public void actionPerformed(ActionEvent e) {

```

```

225         file.saveAll();
226     }
227 }
228 );
229 }
230
231 /**
232  * Add the 'Save Config' menu item. Its action listener invokes the
233  * File.saveConfig method. See the description of the <a href=
234  * FileMenu.html#addNewItem> addNewItem </a> method for further info.
235  */
236 protected void addSaveConfigItem() {
237     final JMenuItem item; // Used by JFileChooser for placement
238
239     menu.add(
240         item = new JMenuItem("Save Config ...").addActionListener(
241             new ActionListener() {
242                 public void actionPerformed(ActionEvent e) {
243                     (new JFileChooser()).showSaveDialog(item);
244                     file.saveConfig();
245                 }
246             }
247         );
248 }
249
250 /**
251  * Add the 'Print ...' menu item. Its action listener invokes the
252  * File.print method. See the description of the <a href=
253  * FileMenu.html#addNewItem> addNewItem </a> method for further info.
254  */
255 protected void addPrintItem() {
256     final JMenuItem item; // Used by JFileChooser for placement
257
258     menu.add(new JMenuItem("Print ...").addActionListener(
259         new ActionListener() {
260             public void actionPerformed(ActionEvent e) {
261                 /*
262                  * For the stubbed design, just display a quickie dialog
263                  * that will be refined later.
264                  */
265                 JOptionPane.showInputDialog("Choose a printer: ");
266                 file.print(null);
267             }
268         }
269     );
270 }
271
272 /**
273  * Add the 'Exit' menu item. Its action listener invokes the File.exit
274  * method. See the description of the <a href= FileMenu.html#addNewItem>
275  * addNewItem </a> method for further info.
276  */
277 protected void addExitItem() {
278     menu.add(new JMenuItem("Exit").addActionListener(
279         new ActionListener() {
280             public void actionPerformed(ActionEvent e) {
281                 file.exit();
282             }
283         }
284     );
285 }
286
287 /**
288  * Add menu items to perform any system tests that may be useful during
289  * development. This submenu goes away in the production version of the
290  * system.
291  */
292 protected void addSystemTestItems() {
293     JMenu smenu = new JMenu("System Tests");
294     menu.add(smenu);
295
296     /*
297      * Add the 'Dump User Cal' menu item.
298      */
299     smenu.add(new JMenuItem("Dump User Cal").addActionListener(
300         new ActionListener() {
301             public void actionPerformed(ActionEvent ev) {
302                 file.dumpUserCal();
303             }
304         }
305     );
306 }
307
308 /**-
309  * Data fields
310  */
311
312 /** Local reference to super.model for convenience to avoid casting
313  * everywhere; i.e., file is set equal to (File)super.model once in the
314  * constructor.
315  */
316 File file;
317
318 /** Local reference to (JMenu)super.widget for convenience to avoid
319  * casting; i.e., menu is set equal to (File)super.widget once in the
320  * constructor. */
321 JMenu menu;
322 }

```