

```

Loading vc-cvs...
1 package caltool.file_ui;
2
3 import caltool.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 6feb04
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         menu = (JMenu) widget;
59
60         /**
61          * Add each of the items to the menu, with separators in the
62          * appropriate places (per the requirements).
63          */
64         addNewItem();
65         addOpenItem();
66         menu.add(new JSeparator());
67
68         addCloseItem();
69         addCloseAllItem();
70         menu.add(new JSeparator());
71
72         addSaveItem();
73         addSaveAsItem();
74         addSaveAllItem();
75         menu.add(new JSeparator());
76
77         addLoadSettingsItem();
78         addSaveSettingsItem();
79         menu.add(new JSeparator());
80
81         addPageSetupItem();
82         addPrintItem();
83         menu.add(new JSeparator());
84
85         addExitItem();
86         menu.add(new JSeparator());
87         addSystemTestItems();
88
89         return widget;
90     }
91 }
92
93 /**-
94  * Protected methods
95  */
96
97 /**
98  * Add the 'New' menu item. Its action listener invokes the File.fileNew
99  * method.
100  *
101  * This and all other menu items are created using the following pattern:
102  *
103  *
104  * JMenu.add(new JMenuItem("<em>Item name</em>")).addActionListener(
105  *     new ActionListener() {
106  *         public void actionPerformed(ActionEvent e) {
107  *             <em>Model.method()</em>
108  *         }
109  *     }
110  * );
111  *
112  * protected void addNewItem() {

```

```

112     /*
113     * Use the standard menu item pattern for the item.
114     */
115     menu.add(new JMenuItem("New")).addActionListener(
116         new ActionListener() {
117             public void actionPerformed(ActionEvent e) {
118                 /*
119                 * Call the stubbed-out model method.
120                 */
121                 file.fileNew();
122             }
123         }
124     );
125 }
126
127 /**
128 * Add the 'Open ...' menu item. Its action listener invokes the File.open
129 * method. See the description of the <a href= FileMenu.html#addNewItem>
130 * addNewItem </a> method for further info.
131 */
132 protected void addOpenItem() {
133
134     final JMenuItem item;          // Used by JFileChooser for placement
135
136     menu.add(
137         item = new JMenuItem("Open ...").addActionListener(
138             new ActionListener() {
139                 public void actionPerformed(ActionEvent e) {
140                     JFileChooser chooser = new JFileChooser();
141                     java.io.File f;
142                     chooser.showOpenDialog(item);
143                     if ((f = chooser.getSelectedFile()) != null) {
144                         file.open(f.getName());
145                     }
146                 }
147             }
148         );
149 }
150
151 /**
152 * Add the 'Close' menu item. Its action listener invokes the File.close
153 * method. See the description of the <a href= FileMenu.html#addNewItem>
154 * addNewItem </a> method for further info.
155 */
156 protected void addCloseItem() {
157     menu.add(new JMenuItem("Close").addActionListener(
158         new ActionListener() {
159             public void actionPerformed(ActionEvent e) {
160                 file.close();
161             }
162         }
163     );
164 }
165
166 /**
167 * Add the 'Close All menu item. Its action listener invokes the
168 * File.closeAll method. See the description of the <a href=
169 * FileMenu.html#addNewItem> addNewItem </a> method for further info.
170 */
171 protected void addCloseAllItem() {
172     menu.add(new JMenuItem("Close All").addActionListener(
173         new ActionListener() {
174             public void actionPerformed(ActionEvent e) {
175                 file.closeAll();
176             }
177         }
178     );
179 }
180
181 /**
182 * Add the 'Save' menu item. Its action listener invokes the File.save
183 * method. See the description of the <a href= FileMenu.html#addNewItem>
184 * addNewItem </a> method for further info.
185 */
186 protected void addSaveItem() {
187     menu.add(new JMenuItem("Save").addActionListener(
188         new ActionListener() {
189             public void actionPerformed(ActionEvent e) {
190                 file.save();
191             }
192         }
193     );
194 }
195
196 /**
197 * Add the 'Save As ...' menu item. Its action listener displays a file
198 * chooser to select the file on which to save, then invokes the
199 * File.saveAs method. See the description of the <a href=
200 * FileMenu.html#addNewItem> addNewItem </a> method for further info.
201 */
202 protected void addSaveAsItem() {
203     final JMenuItem item;          // Used by JFileChooser for placement
204
205     menu.add(
206         item = new JMenuItem("Save As ...").addActionListener(
207             new ActionListener() {
208                 public void actionPerformed(ActionEvent e) {
209                     (new JFileChooser()).showSaveDialog(item);
210                     file.saveAs(null);
211                 }
212             }
213         );
214 }
215
216 /**
217 * Add the 'Save All' menu item. Its action listener invokes the
218 * File.saveAll method. See the description of the <a href=
219 * FileMenu.html#addNewItem> addNewItem </a> method for further info.
220 */
221 protected void addSaveAllItem() {
222     menu.add(new JMenuItem("Save All").addActionListener(
223         new ActionListener() {

```

```

224         public void actionPerformed(ActionEvent e) {
225             file.saveAll();
226         }
227     };
228 };
229 }
230
231 /**
232  * Add the 'Load Settings' menu item. Its action listener displays a load
233  * settings dialog, from which the File.loadSettings method is invoked.
234  * See the description of the <a href= FileMenu.html#addNewItem> addNewItem
235  * </a> method for further info.
236  */
237 protected void addLoadSettingsItem() {
238     final JMenuItem item; // Used by JFileChooser for placement
239
240     menu.add(
241         item = new JMenuItem("Load Settings ...").addActionListener(
242             new ActionListener() {
243                 public void actionPerformed(ActionEvent e) {
244                     /*
245                      * For the stubbed design, just display a quickie dialog
246                      * that will be refined later.
247                      */
248                     JOptionPane.showInputDialog("... Load Settings stuff goes here 304");
249                 }
250             }
251         );
252 }
253
254 /**
255  * Add the 'Load Settings' menu item. Its action listener displays a save
256  * settings dialog, from which the File.saveSettings method is invoked.
257  * FileMenu.html#addNewItem> addNewItem </a> method for further info.
258  */
259 protected void addSaveSettingsItem() {
260     final JMenuItem item; // Used by JFileChooser for placement
261
262     menu.add(
263         item = new JMenuItem("Save Settings ...").addActionListener(
264             new ActionListener() {
265                 public void actionPerformed(ActionEvent e) {
266                     /*
267                      * For the stubbed design, just display a quickie dialog
268                      * that will be refined later.
269                      */
270                     JOptionPane.showInputDialog("... Save Settings stuff goes here 326");
271                 }
272             }
273         );
274 }
275
276 /**
277  * Add the 'Page Setup ...' menu item. Its action listener displays a Page
278  * Setup dialog, from which the File.pageSetup method is invoked. See the
279  * description of the <a href= FileMenu.html#addNewItem> addNewItem </a>
280  * method for further info.
281  */
282 protected void addPageSetupItem() {
283     final JMenuItem item; // Used by JFileChooser for placement
284
285     menu.add(new JMenuItem("Page Setup ...").addActionListener(
286         new ActionListener() {
287             public void actionPerformed(ActionEvent e) {
288                 /*
289                  * For the stubbed design, just display a quickie dialog
290                  * that will be refined later.
291                  */
292                 JOptionPane.showInputDialog("... Page Setup stuff goes here ...");
293             }
294         }
295     );
296 }
297
298 /**
299  * Add the 'Print ...' menu item. Its action listener invokes the
300  * File.print method. See the description of the <a href=
301  * FileMenu.html#addNewItem> addNewItem </a> method for further info.
302  */
303 protected void addPrintItem() {
304     final JMenuItem item; // Used by JFileChooser for placement
305
306     menu.add(new JMenuItem("Print ...").addActionListener(
307         new ActionListener() {
308             public void actionPerformed(ActionEvent e) {
309                 /*
310                  * For the stubbed design, just display a quickie dialog
311                  * that will be refined later.
312                  */
313                 JOptionPane.showInputDialog("Choose a printer: ");
314                 file.print(null);
315             }
316         }
317     );
318 }
319
320 /**
321  * Add the 'Exit' menu item. Its action listener invokes the File.exit
322  * method. See the description of the <a href= FileMenu.html#addNewItem>
323  * addNewItem </a> method for further info.
324  */
325 protected void addExitItem() {
326     menu.add(new JMenuItem("Exit").addActionListener(
327         new ActionListener() {
328             public void actionPerformed(ActionEvent e) {
329                 file.exit();
330             }
331         }
332     );
333 }
334
335 }

```

```
336     /**
337     * Add menu items to perform any system tests that may be useful during
338     * development. This submenu goes away in the production version of the
339     * system.
340     */
341     protected void addSystemTestItems() {
342         JMenu menu = new JMenu("System Tests");
343         menu.add(menu);
344
345         /*
346         * Add the 'Dump User Cal' menu item.
347         */
348         menu.add(new JMenuItem("Dump User Cal")).addActionListener(
349             new ActionListener() {
350                 public void actionPerformed(ActionEvent ev) {
351                     file.dumpUserCal();
352                 }
353             }
354         );
355     }
356 }
357
358
359
360     /*-
361     * Data fields
362     */
363
364     /** Local reference to super.model for convenience to avoid casting
365     * everywhere; i.e., file is set equal to (File)super.model once in the
366     * constructor.
367     */
368     File file;
369
370     /** Local reference to (JMenu)super.widget for convenience to avoid
371     * casting; i.e., menu is set equal to (File)super.widget once in the
372     * constructor. */
373     JMenu menu;
374
375 }
```