CalendarToolUI.java

```
 1  package caltool.view;
 2
 3  import caltool.model.*;
 4  import javax.swing.*;
 5  import java.awt.*;
 6  import mvp.*;
 7
 8  /****
 9   *
10   * Class CalendarToolUI is a companion view to the <a href =
11   * "caltool/model/CalendarTool.html"> CalendarTool </a> model class.  The
12   * window component of the CalendarToolUI is a free-floating JMenuBar,
13   * containing JMenus for the pulldown command menus.
14   *                                                                    <p>
15   * CalendarToolUI contains an instance of the view classes that are companion
16   * to each of the CalendarTool Model classes.  Here is the correspondence:
17   *                                                                  <pre>
18   *     <strong>Model Class              Companion View Class</strong>
19   *     ---------------------------------------------
20   *     file.File                file_ui.FileUI
21   *     edit.Edit                edit_ui.EditUI
22   *     schedule.Schedule        schedule_ui.ScheduleUI
23   *     view.View                view_ui.ViewUI
24   *     admin.Admin              admin_ui.AdminUI
25   *     options.Options          options_ui.OptionsUI
26   *     help.Help                help_ui.HelpUI
27   *                                                                 </pre>
28   *
29   * Each of these view classes contains all of the interface components needed
30   * for UI access to their companion model's methods and data.
31   *                                                                    <p>
32   * CalendarToolUI extends the abstract View class as follows.  First, the
33   * constructor is sent a Screen and CalendarTool model.  The screen and model
34   * inputs come from the top-level Main function, which calls the CalendarToolUI
35   * constructor.  These inputs are passed up to the parent View constructor,
36   * which initializes the inherited screen and model data fields, respectively.
37   * The CalendarToolUI constructor then creates its own JMenuBar and calls the
38   * constructors for each of the component views.  These component view
39   * constructors in turn call the constructors for their components, and so on
40   * until all interface components are constructed.
41   *                                                                    <p>
42   * CalendarToolUI specializes the View.compose method to compose its own
43   * top-level window with the menubar.  CalendarToolUI.compose then calls the
44   * compose methods for each of its component views, which in turn do their own
45   * composition, call their components' compose methods, and so on until all
46   * interface components are composed.
47   *                                                                    <p>
48   * Displaywise, all of the subviews are autonomous units that are not
49   * controlled by the top-level CalendarToolUI.  All that this top-level class
50   * does is construct and compose the menubar, construct and compose the
51   * subviews, and set up the initial display state.
52   *                                                                    <p>
53   * See also the companion model class <a href= "../CalendarTool.html">
54   * CalendarTool. </a>
55   *
56   * @author Gene Fisher (gfisher@calpoly.edu)
57   * @version 13apr15
58   *
59   */
60
61  public class CalendarToolUI extends View {
62
63      /**
64       * Construct this with the given UI screen and CalendarTool model.  Also
65       * construct the pulldown menu subviews and display the initial calendar
66       * view(s) based on the tool option settings.
67       */
68      public CalendarToolUI(Screen screen, CalendarTool calTool) {
69
70          /*
71           * Call the parent constructor.
72           */
73          super(screen, calTool);
74
75          /*
76           * Construct the menubar.
77           */
78          menuBar = new JMenuBar();
79
80          /*
81           * Construct the component views
82           */
83          constructSubviews(this);
84
85          /*
86           * Display initial display windows based on tool option settings.
87           */
88          initialize();
89
90      }
91
92      /**
93       * Compose this by (1) creating a new window, (2) setting the window's
94       * menubar to this' menubar, (3) populating the menubar with the menus, (4)
95       * calling compose in turn for each menu, and (5) setting the window
96       * title.
97       *                                                                  <p>
98       * Since this is the top-level window, call setExitOnClose(true) to have a
99       * close of this window exit the entire application.  This means that the
100      * companion model must implement the exit method to do the right thing.
101      */
102     public Component compose() {
103
104         /*
105          * Make a new window for this, which in Java will be a JFrame -- the
106          * standard outermost container for a Swing window.
107          */
108         window = new mvp.Window();
109
110         /*
111          * Add the menus to the menubar.  This will in turn call compose on
112          * each menu.
```

```
113          */
114         composeMenuBar();
115
116         /*
117          * Set the JFrame window's built-in menu bar to this' menubar.
118          */
119         window.setJMenuBar(menuBar);
120
121         /*
122          * Prevent the menubar window from being resizable.
123          */
124         window.setResizable(false);
125
126         /*
127          * Set the window title, which will appear in the banner of the window.
128          */
129         window.setTitle("Calendar Tool");
130
131         /*
132          * Call JFrame.pack to have Java size up the window properly.
133          */
134         window.pack();
135
136         /*
137          * Install a exit-on-close listener.
138          */
139         setExitOnClose(true);
140
141         /*
142          * Return value is unused for this top-level view.
143          */
144         return widget;
145
146     }
147
148
149     /*-*
150      * Protected methods.
151      */
152
153     /**
154      * Call the constructor for each of the component views.  Pass each
155      * constructor its companion model.
156      */
157     protected void constructSubviews(CalendarToolUI this2) {
158
159         /*
160          * Set a local pointer of type CalendarTool to this' model.  This is
161          * simply a convenience to avoid having to cast the inherited model
162          * field in more than one place.
163          */
164         CalendarTool calTool = (CalendarTool) model;
165
166         /*
167          * Build each subview.
168          */
169         fileUI = new caltool.view.file.FileUI(screen, calTool.getFile());
170         editUI = new caltool.view.edit.EditUI(screen, calTool.getEdit());
171         scheduleUI = new caltool.view.schedule.ScheduleUI(screen,
172             calTool.getSchedule(), this);
173         viewUI = new caltool.view.view.ViewUI(screen, calTool.getCalView(),
174             calTool.getSchedule(), this);
175         adminUI = new caltool.view.admin.AdminUI(screen, calTool.getAdmin());
176         optionsUI = new caltool.view.options.OptionsUI(screen,
177             calTool.getOptions(), this);
178         helpUI = new caltool.view.help.HelpUI(screen, calTool.getHelp());
179
180     }
181
182     /**
183      * Compose this' menubar by adding each composed menu to it.
184      */
185     protected void composeMenuBar() {
186         menuBar.add((JMenu) fileUI.compose());
187         menuBar.add((JMenu) editUI.compose());
188         menuBar.add((JMenu) scheduleUI.compose());
189         menuBar.add((JMenu) viewUI.compose());
190         menuBar.add((JMenu) adminUI.compose());
191         menuBar.add((JMenu) optionsUI.compose());
192         menuBar.add(composeHelpSpacing());
193         menuBar.add((JMenu) helpUI.compose());
194     }
195
196     /**
197      * Compose a piece of horizontal spacing to be inserted between the options
198      * menu and the help menu in the menubar.
199      */
200     protected Box composeHelpSpacing() {
201
202         Box spacing = Box.createHorizontalBox();
203         spacing.add(Box.createHorizontalStrut(12 /*points*/ * 12 /*chars*/));
204
205         return spacing;
206     }
207
208     /**
209      * Configure the initial display based on the tool's option settings.  The
210      * standard default is to display a monthly view immediately below the
211      * menubar.  This standard default can be changed by the user to other
212      * display windows, including no windows at all.  Details TBD.
213      */
214     protected void initialize() {}
215
216
217     /*-*
218      * Data fields
219      */
220
221     /** The top-level menu bar. */
222     protected JMenuBar menuBar;
223
224     /** The pulldown File menu. */
```

```
225     protected caltool.view.file.FileUI fileUI;
226
227     /** The pulldown Edit menu. */
228     protected caltool.view.edit.EditUI editUI;
229
230     /** The pulldown Schedule menu. */
231     protected caltool.view.schedule.ScheduleUI scheduleUI;
232
233     /** The pulldown View menu. */
234     protected caltool.view.view.ViewUI viewUI;
235
236     /** The pulldown Admin menu. */
237     protected caltool.view.admin.AdminUI adminUI;
238
239     /** The pulldown Options menu. */
240     protected caltool.view.options.OptionsUI optionsUI;
241
242     /** The pulldown Help menu. */
243     protected caltool.view.help.HelpUI helpUI;
244
245 }
```