

```

Loading vc-cvs...
1 package caltool.options_ui;
2
3 import caltool.options.*;
4 import javax.swing.*;
5 import java.awt.*;
6 import java.awt.event.*;
7 import mvp.*;
8
9 /**
10 *
11 * Class OptionsMenu is the pulldown menu view of the <a href = "Options.html"
12 * Options </a> model class. The OptionsMenu widget is a Java JMenu.
13 * Anonymous instances of JMenuItem are defined for each item in the menu.
14 *
15 */
16 public class OptionsMenu extends mvp.View {
17
18     /**
19     * Construct this with the given Options model.
20     */
21     public OptionsMenu(Screen screen, Options options, OptionsUI optionsUI) {
22         super(screen, options);
23         this.optionsUI = optionsUI;
24     }
25
26     /**
27     * Compose this by inserting each of its three menu items into the pulldown
28     * menu. The items are Times and Dates, Scheduling, Viewing, and
29     * Administrative.
30     */
31     public Component compose() {
32
33         /**
34         * Make the widget of this the JMenu.
35         */
36         widget = new JMenu("Options");
37
38         /**
39         * Create a constant reference to the Options model for use in the
40         * menu item action listeners. This is necessary due to the subtle
41         * rules of anonymous inner classes in Java.
42         */
43         final Options options = (Options) model;
44
45         /**
46         * Create the menu items using the following pattern:
47         *
48         *         JMenuItem.add(new JMenuItem("<em>Item name</em>").addActionListener(
49         *             new ActionListener() {
50         *                 public void actionPerformed(ActionEvent e) {
51         *                     <em>Model.method()</em>
52         *                 }
53         *             }
54         *         );
55
56         /**
57         * Add the 'Times and Dates' menu item.
58         */
59         ((JMenu) widget).add(new JMenuItem("Times and Dates ...").addActionListener(
60             new ActionListener() {
61                 public void actionPerformed(ActionEvent e) {
62                     optionsUI.show();
63                     optionsUI.selectTab("Times and Dates");
64                     System.out.println("In Options.timesAndDates.");
65                 }
66             }
67         ));
68
69         /**
70         * add the 'Fonts ...' menu item.););
71         */
72         ((JMenu) widget).add(new JMenuItem("Fonts ...").addActionListener(
73             new ActionListener() {
74                 public void actionPerformed(ActionEvent e) {
75                     optionsUI.show();
76                     optionsUI.selectTab("Fonts");
77                     System.out.println("In Options.viewing.");
78                 }
79             }
80         ));
81
82         /**
83         * Add the 'Scheduling ...' menu item.););
84         */
85         ((JMenu) widget).add(new JMenuItem("Scheduling ...").addActionListener(
86             new ActionListener() {
87                 public void actionPerformed(ActionEvent e) {
88                     optionsUI.show();
89                     optionsUI.selectTab("Scheduling");
90                     System.out.println("In Options.scheduling.");
91                 }
92             }
93         ));
94
95         /**
96         * add the 'Viewing ...' menu item.););
97         */
98         ((JMenu) widget).add(new JMenuItem("Viewing ...").addActionListener(
99             new ActionListener() {
100                 public void actionPerformed(ActionEvent e) {
101                     optionsUI.show();
102                     optionsUI.selectTab("Viewing");
103                     System.out.println("In Options.viewing.");
104                 }
105             }
106         ));
107
108         /**
109         * add the 'Administrative ...' menu item.););
110         */
111         ((JMenu) widget).add(new JMenuItem("Administrative ...").addActionListener(

```

```
112         new ActionListener() {
113             public void actionPerformed(ActionEvent e) {
114                 optionsUI.show();
115                 optionsUI.selectTab("Administrative");
116 //                 System.out.println("In Options.administrative.");
117             }
118         }
119     );
120
121     /*
122     * Add a separator.
123     */
124     ((JMenu) widget).add(new JSeparator());
125
126     /*
127     * add the 'Global ...' menu item.););
128     */
129     ((JMenu) widget).add(new JMenuItem("Global ...")).addActionListener(
130         new ActionListener() {
131             public void actionPerformed(ActionEvent e) {
132                 System.out.println("In OptionsMenu 'Global' listener.");
133             }
134         }
135     );
136
137     /*
138     * Add a separator.
139     */
140     ((JMenu) widget).add(new JSeparator());
141
142     /*
143     * add the 'Restore Defaults' menu item.););
144     */
145     ((JMenu) widget).add(new JMenuItem("Restore Defaults")).addActionListener(
146         new ActionListener() {
147             public void actionPerformed(ActionEvent e) {
148                 System.out.println("In OptionsMenu 'Restore Defaults' listener.");
149             }
150         }
151     );
152
153     return widget;
154 }
155
156 /** Parent view */
157 protected OptionsUI optionsUI;
158
159 }
```