

```

1  package caltool.view.schedule;
2
3  import caltool.model.schedule.*;
4  import javax.swing.*;
5  import java.awt.*;
6  import java.awt.event.*;
7  import.mvp.*;
8
9  /****
10 *
11 * Class ScheduleMenu is the pulldown menu view of the <a href =
12 * "Schedule.html" Schedule </a> model class. The first four items of the menu
13 * correspond to the public Schedule methods. The last item corresponds to the
14 * Categories component of Schedule, which is the editable list of scheduled
15 * item categories.
16 *
17 * The four items that correspond to methods each launch a method input dialog.
18 * The last menu item, that corresponds to the categories list, launches a
19 * non-modal category editing dialog.
20 *
21 * In terms of Java Swing components, the ScheduleMenu widget is a Java JMenu.
22 * Anonymous instances of JMenuItem are defined for each menu item. Details of
23 * the dialog structures are given in their respective class definitions.
24 *
25 * @author Gene Fisher (gfisher@calpoly.edu)
26 * @version 13apr15
27 *
28 */
29 public class ScheduleMenu extends.mvp.View {
30
31     /**
32     * Construct this with the given Schedule model and parent ScheduleUI view.
33     */
34     public ScheduleMenu(Screen screen, Schedule schedule,
35         ScheduleUI scheduleUI) {
36
37         /**
38         * Invoke the parent constructor.
39         */
40         super(screen, schedule);
41
42         /**
43         * Store the local reference to the parent view.
44         */
45         this.scheduleUI = scheduleUI;
46     }
47
48     /**
49     * Compose this by inserting each of its five menu items into the pulldown
50     * menu. The items are Appointment, Meeting, Task, Event, and Categories.
51     * A JSeparator is placed after the 'Event ...' item.
52     */
53     public Component compose() {
54
55         /**
56
57         * Make the widget of this the JMenu.
58         */
59         widget = new JMenu("Schedule");
60
61         /**
62         * Add the items.
63         */
64         addAppointmentItem();
65         addMeetingItem();
66         addTaskItem();
67         addEventItem();
68         ((JMenu) widget).add(new JSeparator());
69         addCategoriesItem();
70
71         return widget;
72     }
73
74     /**-
75     * Protected methods
76     */
77
78     /**
79     * Add the 'Appointment ...' menu item. Its action listener invokes the
80     * scheduleUI.getScheduleAppointmentDialog().show() method.
81     */
82     void addAppointmentItem() {
83         ((JMenu) widget).add(new JMenuItem("Appointment ...").addActionListener(
84             new ActionListener() {
85                 public void actionPerformed(ActionEvent e) {
86                     scheduleUI.getScheduleAppointmentDialog().show();
87                 }
88             }
89         ));
90     }
91
92     /**
93     * Add the 'Meeting ...' menu item. Its action listener invokes the
94     * scheduleUI.getScheduleMeetingDialog().show() method.
95     */
96     void addMeetingItem() {
97         ((JMenu) widget).add(new JMenuItem("Meeting ...").addActionListener(
98             new ActionListener() {
99                 public void actionPerformed(ActionEvent e) {
100                     scheduleUI.getScheduleMeetingDialog().show();
101                 }
102             }
103         ));
104     }
105
106     /**
107     * Add the 'Task ...' menu item. Its action listener invokes the
108     * scheduleUI.getScheduleTaskDialog().show() method.
109     */
110     void addTaskItem() {
111         ((JMenu) widget).add(new JMenuItem("Task ...").addActionListener(

```

```
113         new ActionListener() {
114             public void actionPerformed(ActionEvent e) {
115                 scheduleUI.getScheduleTaskDialog().show();
116             }
117         }
118     );
119 }
120
121 /**
122  * Add the 'Event ...' menu item. Its action listener invokes the
123  * scheduleUI.getScheduleTaskDialog().show() method.
124  */
125 void addEventItem() {
126     ((JMenu) widget).add(new JMenuItem("Event ...")).addActionListener(
127         new ActionListener() {
128             public void actionPerformed(ActionEvent e) {
129                 scheduleUI.getScheduleEventDialog().show();
130             }
131         }
132     );
133 }
134
135 /**
136  * Add the 'Categories ...' menu item. Its action listener invokes the
137  * scheduleUI.getCategoriesEditor().show() method.
138  */
139 void addCategoriesItem() {
140     ((JMenu) widget).add(new JMenuItem("Categories ...")).addActionListener(
141         new ActionListener() {
142             public void actionPerformed(ActionEvent e) {
143                 scheduleUI.getCategoriesEditor().show();
144             }
145         }
146     );
147 }
148
149
150 /*-
151  * Data fields
152  */
153
154 /** The parent view. */
155 protected ScheduleUI scheduleUI;
156
157 }
```