```
Loading vc-cvs...
package caltool.schedule_ui;

import caltool.schedule.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import mvp.*;

/****
 *
 * Class ScheduleMenu is the pulldown menu view of the <a href =
 * "Schedule.html" Schedule </a> model class.  The first four items of the menu
 * correspond to the public Schedule methods.  The last item corresponds to the
 * Categories component of Schedule, which is the editable list of scheduled
 * item categories.
 *                                                                          <p>
 * The four items that correspond to methods each launch a method input dialog.
 * The last menu item, that corresponds to the categories list, launches a
 * non-modal category editing dialog.
 *                                                                          <p>
 * In terms of Java Swing components, the ScheduleMenu widget is a Java JMenu.
 * Anonymous instances of JMenuItem are defined for each menu item.  Details of
 * the dialog structures are given in their respective class definitions.
 *
 * @author Gene Fisher (gfisher@calpoly.edu)
 * @version 6feb04
 *
 */
public class ScheduleMenu extends mvp.View {

    /**
     * Construct this with the given Schedule model and parent ScheduleUI view.
     */
    public ScheduleMenu(Screen screen, Schedule schedule,
            ScheduleUI scheduleUI) {

        /*
         * Invoke the parent constructor.
         */
        super(screen, schedule);

        /*
         * Store the local reference to the parent view.
         */
        this.scheduleUI = scheduleUI;

    }

    /**
     * Compose this by inserting each of its five menu items into the pulldown
     * menu.  The items are Appointment, Meeting, Task, Event, and Categories.
     * A JSeparator is placed after the 'Event ...' item.
     */
    public Component compose() {

        /*
         * Make the widget of this the JMenu.
         */
        widget = new JMenu("Schedule");

        /*
         * Add the items.
         */
        addAppointmentItem();
        addMeetingItem();
        addTaskItem();
        addEventItem();
        ((JMenu) widget).add(new JSeparator());
        addCategoriesItem();

        return widget;
    }


    /*-*
     * Protected methods
     */

    /**
     * Add the 'Appointment ...' menu item.  Its action listener invokes the
     * scheduleUI.getScheduleAppointmentDialog().show() method.
     */
    void addAppointmentItem() {
        ((JMenu) widget).add(new JMenuItem("Appointment ...")).addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    scheduleUI.getScheduleAppointmentDialog().show();
                }
            }
        );
    }

    /**
     * Add the 'Meeting ...' menu item.  Its action listener invokes the
     * scheduleUI.getScheduleMeetingDialog().show() method.
     */
    void addMeetingItem() {
        ((JMenu) widget).add(new JMenuItem("Meeting ...")).addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    scheduleUI.getScheduleMeetingDialog().show();
                }
            }
        );
    }

    /**
     * Add the 'Task ...' menu item.  Its action listener invokes the
     * scheduleUI.getScheduleTaskDialog().show() method.
     */
    void addTaskItem() {
```

```
112         ((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 }
```