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

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

/****
 *
 * Class CalendarToolUI is a companion view to the <a href =
 * "../CalendarTool.html"> CalendarTool </a> model class.  The window component
 * of the CalendarToolUI is a free-floating JMenuBar, containing JMenus for the
 * pulldown command menus.
 *                                                                          <p>
 * CalendarToolUI contains an instance of the view classes that are companion
 * to each of the CalendarTool Model classes.  Here is the correspondence:
 *                                                                        <pre>
 *     <strong>Model Class              Companion View Class</strong>
 *     ---------------------------------------------
 *     file.File                file_ui.FileUI
 *     edit.Edit                edit_ui.EditUI
 *     schedule.Schedule        schedule_ui.ScheduleUI
 *     view.View                view_ui.ViewUI
 *     admin.Admin              admin_ui.AdminUI
 *     options.Options          options_ui.OptionsUI
 *     help.Help                help_ui.HelpUI
 *                                                                       </pre>
 *
 * Each of these view classes contains all of the interface components needed
 * for UI access to their companion model's methods and data.
 *                                                                          <p>
 * CalendarToolUI extends the abstract View class as follows.  First, the
 * constructor is sent a Screen and CalendarTool model.  The screen and model
 * inputs come from the top-level Main function, which calls the CalendarToolUI
 * constructor.  These inputs are passed up to the parent View constructor,
 * which initializes the inherited screen and model data fields, respectively.
 * The CalendarToolUI constructor then creates its own JMenuBar and calls the
 * constructors for each of the component views.  These component view
 * constructors in turn call the constructors for their components, and so on
 * until all interface components are constructed.
 *                                                                          <p>
 * CalendarToolUI specializes the View.compose method to compose its own
 * top-level window with the menubar.  CalendarToolUI.compose then calls the
 * compose methods for each of its component views, which in turn do their own
 * composition, call their components' compose methods, and so on until all
 * interface components are composed.
 *                                                                          <p>
 * Displaywise, all of the subviews are autonomous units that are not
 * controlled by the top-level CalendarToolUI.  All that this top-level class
 * does is construct and compose the menubar, construct and compose the
 * subviews, and set up the initial display state.
 *                                                                          <p>
 * See also the companion model class <a href= "../CalendarTool.html">
 * CalendarTool. </a>
 *
 * @author Gene Fisher (gfisher@calpoly.edu)
 * @version 6feb04
 *
 */

public class CalendarToolUI extends View {

    /**
     * Construct this with the given UI screen and CalendarTool model.  Also
     * construct the pulldown menu subviews and display the initial calendar
     * view(s) based on the tool option settings.
     */
    public CalendarToolUI(Screen screen, CalendarTool calTool) {

        /*
         * Call the parent constructor.
         */
        super(screen, calTool);

        /*
         * Construct the menubar.
         */
        menuBar = new JMenuBar();

        /*
         * Construct the component views
         */
        constructSubviews(this);

        /*
         * Display initial display windows based on tool option settings.
         */
        initialize();

    }

    /**
     * Compose this by (1) creating a new window, (2) setting the window's
     * menubar to this' menubar, (3) populating the menubar with the menus, (4)
     * calling compose in turn for each menu, and (5) setting the window
     * title.
     *                                                                      <p>
     * Since this is the top-level window, call setExitOnClose(true) to have a
     * close of this window exit the entire application.  This means that the
     * companion model must implement the exit method to do the right thing.
     */
    public Component compose() {

        /*
         * Make a new window for this, which in Java will be a JFrame -- the
         * standard outermost container for a Swing window.
         */
        window = new mvp.Window();

        /*
         * Add the menus to the menubar.  This will in turn call compose on
```

```
         * each menu.
         */
        composeMenuBar();

        /*
         * Set the JFrame window’s built-in menu bar to this’ menubar.
         */
        window.setJMenuBar(menuBar);

        /*
         * Prevent the menubar window from being resizable.
         */
        window.setResizable(false);

        /*
         * Set the window title, which will appear in the banner of the window.
         */
        window.setTitle("Calendar Tool");

        /*
         * Call JFrame.pack to have Java size up the window properly.
         */
        window.pack();

        /*
         * Install a exit-on-close listener.
         */
        setExitOnClose(true);

        /*
         * Return value is unused for this top-level view.
         */
        return widget;

    }


    /*-*
     * Protected methods.
     */

    /**
     * Call the constructor for each of the component views.  Pass each
     * constructor its companion model.
     */
    protected void constructSubviews(CalendarToolUI this2) {

        /*
         * Set a local pointer of type CalendarTool to this’ model.  This is
         * simply a convenience to avoid having to cast the inherited model
         * field in more than one place.
         */
        CalendarTool calTool = (CalendarTool) model;

        /*
         * Build each subview.
         */
        fileUI = new caltool.file_ui.FileUI(screen, calTool.getFile());
        editUI = new caltool.edit_ui.EditUI(screen, calTool.getEdit());
        scheduleUI = new caltool.schedule_ui.ScheduleUI(screen,
            calTool.getSchedule(), this);
        viewUI = new caltool.view_ui.ViewUI(screen, calTool.getCalView(),
            calTool.getSchedule(), this);
        adminUI = new caltool.admin_ui.AdminUI(screen, calTool.getAdmin());
        optionsUI = new caltool.options_ui.OptionsUI(screen,
            calTool.getOptions(), this);
        helpUI = new caltool.help_ui.HelpUI(screen, calTool.getHelp());

    }

    /**
     * Compose this’ menubar by adding each composed menu to it.
     */
    protected void composeMenuBar() {
        menuBar.add((JMenu) fileUI.compose());
        menuBar.add((JMenu) editUI.compose());
        menuBar.add((JMenu) scheduleUI.compose());
        menuBar.add((JMenu) viewUI.compose());
        menuBar.add((JMenu) adminUI.compose());
        menuBar.add((JMenu) optionsUI.compose());
        menuBar.add(composeHelpSpacing());
        menuBar.add((JMenu) helpUI.compose());
    }

    /**
     * Compose a piece of horizontal spacing to be inserted between the options
     * menu and the help menu in the menubar.
     */
    protected Box composeHelpSpacing() {

        Box spacing = Box.createHorizontalBox();
        spacing.add(Box.createHorizontalStrut(12 /*points*/ * 12 /*chars*/));

        return spacing;
    }

    /**
     * Configure the initial display based on the tool’s option settings.  The
     * standard default is to display a monthly view immediately below the
     * menubar.  This standard default can be changed by the user to other
     * display windows, including no windows at all.  Details TBD.
     */
    protected void initialize() {}


    /*-*
     * Data fields
     */

    /** The top-level menu bar. */
    protected JMenuBar menuBar;

```

```
    /** The pulldown File menu. */
    protected caltool.file_ui.FileUI fileUI;

    /** The pulldown Edit menu. */
    protected caltool.edit_ui.EditUI editUI;

    /** The pulldown Schedule menu. */
    protected caltool.schedule_ui.ScheduleUI scheduleUI;

    /** The pulldown View menu. */
    protected caltool.view_ui.ViewUI viewUI;

    /** The pulldown Admin menu. */
    protected caltool.admin_ui.AdminUI adminUI;

    /** The pulldown Options menu. */
    protected caltool.options_ui.OptionsUI optionsUI;

    /** The pulldown Help menu. */
    protected caltool.help_ui.HelpUI helpUI;

}
```