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

import caltool.options.*;
import caltool.caltool_ui.*;
import mvp.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

/****
 *
 * Initial version of OptionsUI class that constructs and composes its pulldown
 * menu and tabbed dialog.  Each item in the menu is a category of options,
 * with a corresponding pane in the tabbed dialog.  Selection of a menu item
 * makes the tabbed dialog active and selects the pane that corresponds to the
 * menu item.
 *
 */
public class OptionsUI extends CalendarToolWindow {

    /**
     * Construct this with the given screen, model, and parent view.  Construct
     * the panes for the tabbed dialog.  The parent view is used for setting
     * the initial position of this' window relative to the top-level menubar.
     */
    public OptionsUI(Screen screen, Options options,
            CalendarToolUI calToolUI) {
        super(screen, options, calToolUI);

        /*
         * Construct the menu.
         */
        optionsMenu = new OptionsMenu(screen, options, this);

        /*
         * Construct the tabbed pane and add its kids.
         */
        tabs = new JTabbedPane();
        timesAndDatesPanel = new TimesAndDatesOptionsPanel(screen, options);
        fontsPanel = new FontsOptionsPanel(screen, options);
        schedulingPanel = new SchedulingOptionsPanel(screen, options);
        viewingPanel = new ViewingOptionsPanel(screen, options);
        administrativePanel = new AdministrativeOptionsPanel(screen, options);
    }

    /*
     * Compose this by adding the panes to the dialog, adding the button row on
     * the bottom, and composing the menu.
     */
    public Component compose() {

        /*
         * Make the base layer a JPanel, as usual.
         */
        panel = new JPanel();
        window.add(panel);

        /*
         * Set the layout style of the panel to be a vertical box.
         */
        panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));

        /*
         * Set the window title.
         */
        window.setTitle("Options");

        /*
         * Compose the tabbed dialog, button row, and pack things up.
         */
        composeTabs();
        composeButtonRow();
        window.pack();

        /*
         * Compose and return the menu.
         */
        return optionsMenu.compose();

    }

    protected void composeTabs() {

        Box hbox = Box.createHorizontalBox();

        hbox.add(Box.createHorizontalStrut(20));
        hbox.add(tabs);
        hbox.add(Box.createHorizontalStrut(20));

        panel.add(Box.createVerticalStrut(20));
        panel.add(hbox);
        panel.add(Box.createVerticalStrut(20));

        tabs.setPreferredSize(new Dimension(575,475));

        tabs.addTab("Times and Dates", timesAndDatesPanel.compose());
        tabs.addTab("Fonts", fontsPanel.compose());
        tabs.addTab("Scheduling", schedulingPanel.compose());
        tabs.addTab("Viewing", viewingPanel.compose());
        tabs.addTab("Administrative", administrativePanel.compose());

    }

    protected void composeButtonRow() {

        Box hbox = Box.createHorizontalBox();

        /*
         * Construct the buttons.
         */
        JButton applyButton = new JButton("Apply");
```

```
        JButton saveAsButton = new JButton("Save As ...");
        JButton loadButton = new JButton("Load ...");
        JButton clearButton = new JButton("Clear");
        JButton cancelButton = new JButton("Cancel");

        /*
         * Attach the appropriate action listeners to each button.
         */
        cancelButton.addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    hide();
                }
            }
        );
        // ...

        /*
         * Set the initial states of the buttons (dummy processing until we
         * connect with the model).
         */
        applyButton.setEnabled(false);
        clearButton.setEnabled(false);

        /*
         * Add everything to the hbox and stick it in the panel.
         */
        hbox.add(applyButton);
        hbox.add(Box.createHorizontalStrut(12));
        hbox.add(saveAsButton);
        hbox.add(Box.createHorizontalStrut(12));
        hbox.add(loadButton);
        hbox.add(Box.createHorizontalStrut(12));
        hbox.add(clearButton);
        hbox.add(Box.createHorizontalStrut(12));
        hbox.add(cancelButton);

//      panel.add(Box.createVerticalStrut(20));
        panel.add(hbox);
        panel.add(Box.createVerticalStrut(20));

    }

    /**
     * Select the tabbed with the name of the given string.
     */
    public void selectTab(String selection) {
        tabs.setSelectedIndex(tabs.indexOfTab(selection));
    }


    /** The background panel */
    protected JPanel panel;

    /** The tabbed dialog */
    protected JTabbedPane tabs;

    /** The times and dates pane */
    protected TimesAndDatesOptionsPanel timesAndDatesPanel;

    /** The fonts pane */
    protected FontsOptionsPanel fontsPanel;

    /** The scheduling pane */
    protected SchedulingOptionsPanel schedulingPanel;

    /** The viewing pane */
    protected ViewingOptionsPanel viewingPanel;

    /** The administrative pane */
    protected AdministrativeOptionsPanel administrativePanel;

    /** The menu */
    protected OptionsMenu optionsMenu;

}
```