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

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

/****
 *
 * Class EditMenu is the pulldown menu view of the <a href = "Edit.html" Edit
 * </a> model class.  The EditMenu widget is a Java JMenu.  Anonymous instances
 * of JMenuItem are defined for each item in the menu.
 *
 */
public class EditMenu extends View {

    /**
     * Construct this with the given name as the pulldown label.  The given
     * Edit is the companion model.
     */
    public EditMenu(Screen screen, Edit edit, EditUI editUI) {
        super(screen, edit);
    }

    /**
     * Compose this by inserting each of its ten menu items into the pulldown
     * menu.  The items are Undo, Redo, Cut, copy, Past, Delete, Select All,
     * and Find.  JSeparators are placed after the 'Redo' and 'Select All'
     * items.
     */
    public Component compose() {

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

        /*
         * Create a constant reference to the File model for use in the menu
         * item action listeners.  This is necessary due to the subtle rules of
         * anonymous inner classes in Java.
         */
        final Edit edit = (Edit) model;

        /*
         * Create the menu items using the following pattern:
         *                                                                 <pre>
         *     JMenu.add(new JMenuItem("<em>Item name</em>").addActionListener(
         *         new ActionListener() {
         *             public void actionPerformed(ActionEvent e) {
         *                 <em>Model.method()</em>
         *                                                                 </pre>
         */


        /*
         * Add the 'Undo' menu item.
         */
        ((JMenu) widget).add(new JMenuItem("Undo")).addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent ev) {
                    edit.undo();
                }

            }
        );

        /*
         * Add the 'Redo' menu item.
         */
        ((JMenu) widget).add(new JMenuItem("Redo")).addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent ev) {
                    edit.redo();
                }

            }
        );

        /*
         * Add a separator.
         */
        ((JMenu) widget).add(new JSeparator());

        /*
         * Add the 'Cut' menu item.
         */
        ((JMenu) widget).add(new JMenuItem("Cut")).addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent ev) {
                    edit.cut();
                }
            }
        );

        /*
         * Add the 'Copy' menu item.
         */
        ((JMenu) widget).add(new JMenuItem("Copy")).addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent ev) {
                    edit.copy();
                }
            }
        );

        /*
         * Add the 'Paste' menu item.
         */
        ((JMenu) widget).add(new JMenuItem("Paste")).addActionListener(
```

```
112             new ActionListener() {
113                 public void actionPerformed(ActionEvent ev) {
114                     edit.paste();
115                 }
116             }
117         );
118 
119         /*
120          * Add the 'Delete' menu item.
121          */
122         ((JMenu) widget).add(new JMenuItem("Delete")).addActionListener(
123             new ActionListener() {
124                 public void actionPerformed(ActionEvent ev) {
125                     edit.delete();
126                 }
127             }
128         );
129 
130         /*
131          * Add the 'Select All' menu item.
132          */
133         ((JMenu) widget).add(new JMenuItem("Select All")).addActionListener(
134             new ActionListener() {
135                 public void actionPerformed(ActionEvent ev) {
136                     edit.selectAll();
137                 }
138             }
139         );
140 
141         /*
142          * Add a separator.
143          */
144         ((JMenu) widget).add(new JSeparator());
145 
146         /*
147          * Add the 'Find ...' menu item.
148          */
149         ((JMenu) widget).add(new JMenuItem("Find ...")).addActionListener(
150             new ActionListener() {
151                 public void actionPerformed(ActionEvent ev) {
152                     JOptionPane.showInputDialog("Search for: ");
153                     ((Edit) model).find(null);
154                 }
155             }
156         );
157 
158         return widget;
159     }
160 
161 }
```