```java
package rolodex;

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

/****
 *
 * Class RolodexMenu is an operational view of the <a href = "Rolodex.html">
 * Rolodex model </a> class in the menu-based GUI.  The RolodexMenu window is a
 * Java JMenu menu.  Anonymous instances of JMenuItem are defined for each item
 * in the menu.
 *                                                                          <p>
 * To say that RolodesMenu is an "operational" view of the rolodex model means
 * that it is a view of the model operations but not of its data.  Other views
 * of rolodex show some or all of its data.  See <a href =
 * "AddCardDialog.html"> AddCardDialog </a> for further discussion of the
 * correspondence between the rolodex model and its views.
 *
 */

public class RolodexMenu extends View {

    /**
     * Construct this with the given name as the pulldown label.  The given
     * Rolodex is the companion model.  The given RolodexUI is the parent view.
     * Note that in the current non-modal implementation, the model is not
     * necessary here, since the input dialogs will invoke the model operations
     * directly.  The model is passed in as a matter of "good form" in case
     * implementation details change such that one or more menu operations
     * invoke model functions directly.
     */
    public RolodexMenu(String name, Rolodex r, RolodexMenuUI rmui) {

        super(null, r);

        /*
         * For convenience, save a copy of the model as a downcast value.
         */
        this.r = r;

        /*
         * Store the parent view.
         */
        this.rmui = rmui;

        /*
         * Make the widget of this an IV pulldown menu.
         */
        widget = jm = new JMenu(name);

    }

    /**
     * Compose this by inserting each of its four menu items -- Add, Delete,
     * Change, and Find -- into the pulldown menu window.  Each item has its
     * own action listener.
     */
    public void compose() {

        jm.add(new JMenuItem("Add ...")).addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    /*
                     * Bring up the add-card dialog, and pass the buck to
                     * there.  Since this is a non-model dialog interface, we
                     * won't come back here to call the model to do the actual
                     * add.  Rather, the OK button in the add-card dialog will
                     * call the model add function.
                     */
                    rmui.showAddCardDialog();
                }
            }
        );

        jm.add(new JMenuItem("Delete ...")).addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    rmui.showDeleteCardDialog();
                }
            }
        );

        jm.add(new JMenuItem("Change ..")).addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    rmui.showChangeCardDialog();
                }
            }
        );

        jm.add(new JMenuItem("Find ...")).addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    rmui.showFindCardDialog();
                }
            }
        );


    }

    public JMenu getMenu() {
        return jm;
    }

    /** Companion model, downcast version of this.m */
    protected Rolodex r;

    /** The parent view */
    protected RolodexMenuUI rmui;
```

```

    /** For convenience, downcast version of this.w */
    protected JMenu jm;

    /** */
    protected AddCardDialog acd;

}
```