```
package rolodex;

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

/****
 *
 * Class AddCardDialog is a partial rolodex view containing JTextFields for
 * entering the values of the fields in a rolodex card.  Above the text fields
 * is a prompting message directed to the user (i.e., "Enter information
 * ...:").  Below the text fields are OK, Clear, and Cancel buttons.
 *                                                                          <p>
 * That AddCardDialog is a "partial" rolodex view means that the companion view
 * for the model is a rolodex, however the view does not show all of the
 * rolodex contents.  What precisely AddCardDialog shows is a single card that
 * will be used as the input parameter to Rolodex::add.  There is no model data
 * member to which AddCardDialog corresponds.  Rather, it is a view of the
 * input to a Rolodex member function.
 *                                                                          <p>
 * This is an example of a general MVP principle that views may correspond to
 * model data independent of where the model data are allocated.  Frequently a
 * view corresponds to a model datum that is allocated as a data member in some
 * larger containing class.  However, it is perfectly acceptable to have a
 * view correspond to a model datum that is allocated as a function parameter.
 *                                                                          <p>
 * One might consider the model for AddCardDialog view to be a single card,
 * rather than the larger rolodex.  However, in addition to the view of a card,
 * AddCardDialog has an OK button that is a view of the Rolodex.add operation.
 * Hence, the dialog is larger than just a card itself and hence the term
 * "partial" rolodex view is used.  Such terminology is not really that
 * important.  What *is* important is that each view has a companion model that
 * represents some or all of the data and functions depicted in the view.  It's
 * OK if the model has more data than are depicted in the view.
 *
 */

public class AddCardDialog extends View {

    /**
     * Construct the dialog to enter the fields of a card and that will be
     * added to the given rolodex.  The given Rolodex is the companion model.
     * The given RolodexMenuUI is the parent view.
     */
    public AddCardDialog(Rolodex r, RolodexMenuUI rmui) {
        super(null, r);

        /*
         * Construct the five text fields.
         */
        name_jtf = new JTextField(TEXT_FIELD_WIDTH);
        id_jtf = new JTextField(TEXT_FIELD_WIDTH);
        age_jtf = new JTextField(TEXT_FIELD_WIDTH);
        sex_jtf = new JTextField(TEXT_FIELD_WIDTH);
        addr_jtf = new JTextField(TEXT_FIELD_WIDTH);

        /*
         * Construct the class that holds the OK, Clear, and Cancel buttons.
         */
        acb = new AddCardButtons(r, this, rmui);

        /*
         * Construct the boxes for the labels, text fields, plus outer box
         * layers.
         */
        labels_box = new Box(BoxLayout.Y_AXIS);         // Labels V Box
        fields_box = new Box(BoxLayout.Y_AXIS);         // Text fields V Box
        labeled_fields_box = new Box(BoxLayout.X_AXIS); // Labels+fields H Box
        outer_box = new Box(BoxLayout.Y_AXIS);          // Label+HBox+ButtonBox

    }

    /**
     * Compose this by adding a prompting message, the labeled text fields, and
     * the OK, Cancel, and Clear buttons.
     */
    public void compose() {

        /*
         * Make the widget of this the outer vertical box, wrapped in an opaque
         * JPanel.
         */
        widget = new JPanel();
        ((JPanel) widget).setOpaque(true);
        ((JPanel) widget).add(outer_box);

        /*
         * Stick some spacing above the prompting message.
         */
        outer_box.add(Box.createVerticalStrut(VERTICAL_SPACING));

        /*
         * Insert the prompting message into the top of the box.  Set its font
         * to Bold 16 point.  Set its alignment to centered.
         */
        top_label = new JLabel("Enter information for a rolodex card:");
        top_label.setAlignmentX(JFrame.CENTER_ALIGNMENT);
        top_label.setFont(new Font(RolodexMenuUI.LABEL_FONT_NAME,
            Font.BOLD, 16));
        top_label.setForeground(Color.black);
        outer_box.add(top_label);

        /*
         * Stick some spacing between the message and the text fields below.
         */
        outer_box.add(Box.createVerticalStrut(VERTICAL_SPACING));

        /*
         * Compose the parallel label and text field vertical boxes and stick
         * them in a horizontal box.
```

```
113          */
114         composeLabels();
115         composeTextFields();
116         composePair();
117
118         /*
119          * Insert the labeled fields horizontal box into the outer box.
120          */
121         outer_box.add(labeled_fields_box);
122
123         /*
124          * Compose the dialog buttons and insert into the outer box.
125          */
126         acb.compose();
127         outer_box.add(Box.createVerticalStrut(VERTICAL_SPACING));
128         outer_box.add(acb.getWidget());
129         outer_box.add(Box.createVerticalStrut(VERTICAL_SPACING));
130     }
131
132     /**
133      * Return the string value entered by the user in the Name text field.
134      */
135     public String getName() {
136         return name_jtf.getText();
137     }
138
139     /**
140      * Return the Integer value entered by the user in the Id text field.  If
141      * an empty or non-integer string is entered, return the Integer 0.
142      */
143     public Integer getId() {
144         String text;                    // Raw text value from text field
145         Integer rtn;                    // Text value converted to Integer
146
147         /*
148          * Snag the string value from the text field.
149          */
150         text = id_jtf.getText();
151
152         /*
153          * Convert the raw text string to an Integer.  If not a legal integer
154          * value, convert to 0.
155          */
156         try {
157             rtn = new Integer(text);
158         }
159         catch (NumberFormatException nfe) {
160             rtn = new Integer(0);
161         }
162
163         return rtn;
164     }
165
166     /**
167      * Return the Integer value entered by the user in the Age text field.  If
168      * an empty or non-integer string is entered, return the Integer -1.
169      */
170     public Integer getAge() {
171         String text;                    // Raw text value from text field
172         Integer rtn;                    // Text value converted to Integer
173
174         /*
175          * Snag the string value from the text field.
176          */
177         text = age_jtf.getText();
178
179         /*
180          * Convert the raw text string to an Integer.  If not a legal integer
181          * value, convert to 0.
182          */
183         try {
184             rtn = new Integer(text);
185         }
186         catch (NumberFormatException nfe) {
187             return new Integer(-1);
188         }
189
190         return rtn;
191     }
192
193     /**
194      * Return the string value entered by the user in the Sex text field.
195      */
196     public String getSex() {
197         return sex_jtf.getText();
198     }
199
200     /**
201      * Return the string value entered by the user in the Address text field.
202      */
203     public String getAddr() {
204         return addr_jtf.getText();
205     }
206
207     /**
208      * Clear the text in each of the text fields.
209      */
210     public void clear() {
211         name_jtf.setText("");
212         id_jtf.setText("");
213         age_jtf.setText("");
214         sex_jtf.setText("");
215         addr_jtf.setText("");
216     }
217
218     /**
219      * Insert each field label into the labels vertical box.
220      */
221     protected void composeLabels() {
222
223         /*
224          * Add some spacing at the top of the labels box.
```

```
         */
        labels_box.add(Box.createVerticalStrut(VERTICAL_SPACING));

        /*
         * Add the labels to the box, with some stretchy spacing in between.
         */
        addLabel("Name:");
        addLabel("Id:");
        addLabel("Age:");
        addLabel("Sex:");
        addLabel("Address:");

    }

    /**
     * Add a label with the given string value to the labels vertical box,
     * including some glue spacing above and below.
     */
    protected void addLabel(String s) {

        JLabel jl;                      // Temp label var

        /*
         * Put some stretchable glue above the label.  */
        labels_box.add(Box.createVerticalGlue());

        /*
         * Add the label itself.
         */
        labels_box.add(jl = new JLabel(s + " "));

        /*
         * Put some stretchable glue below the label.
         */
        labels_box.add(Box.createVerticalGlue());


        /*
         * Set the horizontal alignment of the label to the right (per the
         * layout in the requirements).
         */
        jl.setAlignmentX(JFrame.RIGHT_ALIGNMENT);


        /*
         * Set the font to Bold 14 point (also per the
         * requirements).
         */
        jl.setFont(new Font(RolodexMenuUI.LABEL_FONT_NAME, Font.PLAIN, 14));

        /*
         * Set the foreground color of the text to black (the Java default is
         * is a rather unattractive blue.
         */
        jl.setForeground(Color.black);

        /*
         * Add some spacing below the label.
         */
        labels_box.add(Box.createVerticalStrut(VERTICAL_SPACING));

    }

    /**
     * Insert each field label into the labels vertical box.
     */
    protected void composeTextFields() {
        /*
         * Add the text fields to their vertical box, with some spacing in
         * between.  Note that the glue around each label will cause the labels
         * to be properly centered around the text fields.
         */
        fields_box.add(Box.createVerticalStrut(VERTICAL_SPACING));
        fields_box.add(name_jtf);
        fields_box.add(Box.createVerticalStrut(VERTICAL_SPACING));
        fields_box.add(id_jtf);
        fields_box.add(Box.createVerticalStrut(VERTICAL_SPACING));
        fields_box.add(age_jtf);
        fields_box.add(Box.createVerticalStrut(VERTICAL_SPACING));
        fields_box.add(sex_jtf);
        fields_box.add(Box.createVerticalStrut(VERTICAL_SPACING));
        fields_box.add(addr_jtf);
        fields_box.add(Box.createVerticalStrut(VERTICAL_SPACING));
    }

    /**
     * Insert the label and text field vertical boxes box into an outer
     * horizontal box.
     */
    protected void composePair() {
        /*
         * Put the label and text field vertical boxes into a horizontal box.
         */
        labeled_fields_box.add(Box.createHorizontalStrut(25));
        labeled_fields_box.add(labels_box);
        labeled_fields_box.add(fields_box);
        labeled_fields_box.add(Box.createHorizontalStrut(25));
    }


    /*-*
     * Data members.
     */

    /** Default width of the text fields (in characters) */
    public static final int TEXT_FIELD_WIDTH = 25;

    /** Size of vertical spacing between UI elements (in pixels) */
    public static final int VERTICAL_SPACING = 10;

    /** Label at top of dialog */
    JLabel top_label;
```

```
337
338      /** Text field for card name */
339      protected JTextField name_jtf;
340
341      /** Text field for card id */
342      protected JTextField id_jtf;
343
344      /** Text field for card age */
345      protected JTextField age_jtf;
346
347      /** Text field for card sex */
348      protected JTextField sex_jtf;
349
350      /** Text field for card address */
351      protected JTextField addr_jtf;
352
353      /** Vertical box for holding text field labels */
354      protected Box labels_box;
355
356      /** Vertical box for holding text fields */
357      protected Box fields_box;
358
359      /** Horizontal box for holding the label and field boxes */
360      protected Box labeled_fields_box;
361
362      /** Outer most vertical box for holding all of the components */
363      protected Box outer_box;
364
365      /** Subview containing add card buttons OK, Clear, and Cancel */
366      protected AddCardButtons acb;
367
368  }
```