```
package rolodex;

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

/****
 *
 * Class FoundCardsDialog contains components for displaying the cards found
 * by Rolodex.find.  The cards are displayed one at a time in the dialog, with
 * buttons to move back and forth between the cards.
 *                                                                        <p>
 * The top of the dialog contains a two-line message that says how many cards
 * were found and what card is currently displayed.  Below the two messages are
 * labeled read-only text fields that display the fields of a card.  Below the
 * text fields are Next and Previous buttons, which are active if more
 * than one card was found.
 *                                                                        <p>
 * It is interesting to note that FoundCardsDialog has no persistent model
 * object.  Rather, this dialog is a view of a transient model object produced
 * as the output of Rolodex.find.  Furthermore, this view need not consult with
 * the model for data access, since the list is treated generically, using
 * List.get for data access.  The transient model object is sent in as a
 * parameter to FoundCardsDialog.showResults and deleted in FoundCardsDialog.
 * done.
 *
 */

public class FoundCardsDialog extends View {

    /*
     * Construct this with the given model and top-level view.
     */
    public FoundCardsDialog(Rolodex r, RolodexMenuUI rmui) {
        super(null, r);

        /*
         * Construct the two labels that show how many cards found and which
         * card is currently displayed.
         */
        n_label = new JLabel();
        n_label.setAlignmentX(JFrame.RIGHT_ALIGNMENT);
        i_label = new JLabel();

        /*
         * Construct the five text fields, disabling typing in each.  Note how
         * disabling typing is performed.  One way to do this would be simply to
         * call the JTextField.setEnabled() method with a parameter value of
         * false.  This has the effect of disabling all typing, but it shades
         * the displayed text grey.  To avoid this shading, the disabling
         * method used here is to reimplement the processComponentKeyEvent and
         * processMouseEvent methods to do nothing.  This disables typing and
         * clicking in the text fields, but leaves the text colored normally.
         */
        name_jtf = new JTextField(AddCardDialog.TEXT_FIELD_WIDTH) {
            protected void processComponentKeyEvent(KeyEvent e) {}
            protected void processMouseEvent(MouseEvent e) {}
        };

        id_jtf = new JTextField(AddCardDialog.TEXT_FIELD_WIDTH) {
            protected void processComponentKeyEvent(KeyEvent e) {}
            protected void processMouseEvent(MouseEvent e) {}
        };

        age_jtf = new JTextField(AddCardDialog.TEXT_FIELD_WIDTH) {
            protected void processComponentKeyEvent(KeyEvent e) {}
            protected void processMouseEvent(MouseEvent e) {}
        };

        sex_jtf = new JTextField(AddCardDialog.TEXT_FIELD_WIDTH) {
            protected void processComponentKeyEvent(KeyEvent e) {}
            protected void processMouseEvent(MouseEvent e) {}
        };

        addr_jtf = new JTextField(AddCardDialog.TEXT_FIELD_WIDTH) {
            protected void processComponentKeyEvent(KeyEvent e) {}
            protected void processMouseEvent(MouseEvent e) {}
        };

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

        /*
         * Construct the boxes for the labels, text fields, plus outer box
         * layers.
         */
        b1 = new Box(BoxLayout.X_AXIS);                 // Top msg line 1
        b2 = new Box(BoxLayout.X_AXIS);                 // Top msg line 2
        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 the top two message lines, the labeled text
     * fields, and the OK, Cancel, and Clear buttons.
     */
    public void compose() {

        JLabel jl;                      // Temporary label var

        /*
         * 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(AddCardDialog.VERTICAL_SPACING));

        /*
         * Compose and insert the two-line top message.
         */
        composeTopMessages();

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

        /*
         * Compose the parallel label and text field vertical boxes and stick
         * them in a horizontal box.
         */
        composeLabels();
        composeTextFields();
        composePair();

        /*
         * Insert the labeled fields horizontal box into the outer box.
         */
        outer_box.add(labeled_fields_box);

        /*
         * Compose the dialog buttons and insert into the outer box.
         */
        fcb.compose();
        outer_box.add(Box.createVerticalStrut(AddCardDialog.VERTICAL_SPACING));
        outer_box.add(fcb.getWidget());
        outer_box.add(Box.createVerticalStrut(AddCardDialog.VERTICAL_SPACING));
    }

    /**
     * Display the given card list (the results of the Rolodex::Find command)
     * in a browseable sequence of screens.  Set the internal next-card index to
     * 1.
     */
    public void showResults(CardList cl) {

        /*
         * Store the transient model data for display purposes.
         */
        this.cl = cl;

        /*
         * Initialize the current card index to 0.
         */
        cur_index = 0;

        /*
         * If just one card found, disable the Next and Prev buttons, otherwise
         * enable them.
         */
        if (cl.size() == 1)
            fcb.setNextPrevEnabled(false);
        else
            fcb.setNextPrevEnabled(true);

        /*
         * Set the value of N in the " N card(s) found ..." message.
         */
        showNumberOfCards(cl.size());

        /*
         * Call Next (with a 0 current index) to display the first card.
         */
        next();

    }

    /**
     * Show the next card, if any, in the results list.  Increment the internal
     * next-card index to 1.
     */
    public void next() {
        if (cur_index < cl.size()) {
            showWhichCard(++cur_index);
            showCardData((Card) cl.get(cur_index-1));
        }
    }

    /**
     * Show the previous card, if any, in the results list.  Decrement the
     * internal next-card index to 1.
     */
    public void previous() {
        if (cur_index > 1) {
            showWhichCard(--cur_index);
            showCardData((Card) cl.get(cur_index-1));
        }
    }

    /**
     * Compose the two-line message at the top of the dialog.
     */
    protected void composeTopMessages() {

        JLabel jl;                      // Temporary label var

        b1.add(n_label);
        setLabelProperties(n_label);

        b1.add(jl = new JLabel(" card(s) of the given name found."));
        setLabelProperties(jl);

```

```
225         b2.add(jl = new JLabel("Here is card "));
226         setLabelProperties(jl);
227
228         b2.add(i_label);
229         setLabelProperties(i_label);
230
231         b2.add(jl = new JLabel(":"));
232         setLabelProperties(jl);
233
234         outer_box.add(b1);
235         outer_box.add(b2);
236     }
237
238     /**
239      * Set the following properties of the given label: center alignment, bold
240      * 16 point font, pen color black.
241      */
242     protected void setLabelProperties(JLabel jl) {
243         jl.setAlignmentX(JFrame.CENTER_ALIGNMENT);
244         jl.setFont(new Font(RolodexMenuUI.LABEL_FONT_NAME, Font.BOLD, 16));
245         jl.setForeground(Color.black);
246     }
247
248     /**
249      * Insert each field label into the labels vertical box.
250      */
251     protected void composeLabels() {
252
253         /*
254          * Add some spacing at the top of the labels box.
255          */
256         labels_box.add(Box.createVerticalStrut(
257             AddCardDialog.VERTICAL_SPACING));
258
259         /*
260          * Add the labels to the box, with some stretchy spacing in between.
261          */
262         addLabel("Name:");
263         addLabel("Id:");
264         addLabel("Age:");
265         addLabel("Sex:");
266         addLabel("Address:");
267
268     }
269
270     /**
271      * Add a label with the given string value to the labels vertical box,
272      * including some glue spacing above and below.
273      */
274     protected void addLabel(String s) {
275
276         JLabel jl;                      // Temp label var
277
278         /*
279          * Put some stretchable glue above the label.
280          */
281         labels_box.add(Box.createVerticalGlue());
282
283         /*
284          * Add the label itself.
285          */
286         labels_box.add(jl = new JLabel(s + " "));
287
288         /*
289          * Put some stretchable glue below the label.
290          */
291         labels_box.add(Box.createVerticalGlue());
292
293
294         /*
295          * Set the horizontal alignment of the label to the right (per the
296          * layout in the requirements).
297          */
298         jl.setAlignmentX(JFrame.RIGHT_ALIGNMENT);
299
300
301         /*
302          * Set the font to Bold 14 point (also per the requirements).
303          */
304         jl.setFont(new Font(RolodexMenuUI.LABEL_FONT_NAME, Font.PLAIN, 14));
305
306         /*
307          * Set the foreground color of the text to black (the Java default is
308          * is a rather unattractive blue.
309          */
310         jl.setForeground(Color.black);
311
312         /*
313          * Add some spacing below the label.
314          */
315         labels_box.add(Box.createVerticalStrut(AddCardDialog.VERTICAL_SPACING));
316
317     }
318
319     /**
320      * Insert each field label into the labels vertical box.
321      */
322     protected void composeTextFields() {
323         /*
324          * Add the text fields to their vertical box, with some spacing in
325          * between.  Note that the glue around each label will cause the labels
326          * to be properly centered around the text fields.
327          */
328         fields_box.add(Box.createVerticalStrut(
329             AddCardDialog.VERTICAL_SPACING));
330         fields_box.add(name_jtf);
331
332         fields_box.add(Box.createVerticalStrut(
333             AddCardDialog.VERTICAL_SPACING));
334         fields_box.add(id_jtf);
335
336         fields_box.add(Box.createVerticalStrut(
```

```
            AddCardDialog.VERTICAL_SPACING));
        fields_box.add(age_jtf);

        fields_box.add(Box.createVerticalStrut(
            AddCardDialog.VERTICAL_SPACING));
        fields_box.add(sex_jtf);

        fields_box.add(Box.createVerticalStrut(
            AddCardDialog.VERTICAL_SPACING));
        fields_box.add(addr_jtf);

        fields_box.add(Box.createVerticalStrut(
            AddCardDialog.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));
    }

    /**
     * Insert the label and string editor boxes into an outer horizontal box
     * with some stretchable glue.
     */
    protected void composeOuter() {}

    /**
     * Display the given integer in the string editor that holds the value of
     * the "N" in the "N cards ... found" message.
     */
    protected void showNumberOfCards(int n) {
        n_label.setText(Integer.toString(n));
    }

    /**
     * Display the given integer in the string editor that holds the value of
     * the "i" in the "Here is card i:" message.
     */
    protected void showWhichCard(int i) {
        i_label.setText(Integer.toString(i));
    }

    /**
     * Display each of the five fields of the given card in the five read-only
     * string editors.
     */
    protected void showCardData(Card c) {
        name_jtf.setText(c.getName().stringValue());
        id_jtf.setText(c.getId().integerValue().toString());
        age_jtf.setText(c.getAge().integerValue().toString());
        sex_jtf.setText(c.getSex().stringValue());
        addr_jtf.setText(c.getAddr().stringValue());
    }


    /** Companion model. */
    Rolodex r;

    /** Message line 1 -- "N cards ... found." */
    protected Box b1;

    /** Message line 2 -- "Here is card i:" */
    protected Box b2;

    /** Label holding the value of N in message line 1*/
    protected JLabel n_label;

    /** Label holding the value of i in message line 2*/
    protected JLabel i_label;

    /** Text field for name field */
    protected JTextField name_jtf;

    /** Text field for id field */
    protected JTextField id_jtf;

    /** Text field for age field */
    protected JTextField age_jtf;

    /** Text field for sex field */
    protected JTextField sex_jtf;

    /** Text field for address field */
    protected JTextField addr_jtf;

    /** Vertical box for holding text field labels */
    protected Box labels_box;

    /** Vertical box for holding text fields */
    protected Box fields_box;

    /** Horizontal box for holding the label and field boxes */
    protected Box labeled_fields_box;

    /** Horizontal box for holding all of the components */
    protected Box outer_box;

    /** Subview containing add card buttons OK, Clear, and Cancel */
    protected FoundCardsButtons fcb;

    /** Transient list of cards to display */
    protected CardList cl;

```

```
449      /** Current index in displayed card list. */
450      protected int cur_index;
451
452  }
```