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

import caltool.schedule.*;
import mvp.*;
import javax.swing.*;
import java.awt.*;

/****
 *
 * Class ScheduleEventDialog provides a view of Event as an input to the
 * scheduleEvent method.  Hence, the dialog is a view of both an Event object
 * as well as the scheduleEvent method.  The data-entry components of the
 * dialog are the Event view.  The 'OK' button is the view of the ScheduleEvent
 * method.
 *
 * The data components consist of JLabels, JTextFields, and a JComboBox.  The
 * 'OK', 'Clear', and 'Cancel' buttons are JButtons.  The description of the <a
 * href= "#compose()"> compose </a> method has details of how the components
 * are laid out in the dialog window.
 *
 * The companion model for ScheduleEventDialog is the Schedule class, since
 * Schedule has the method that must be invoked from the 'OK' button action
 * listener.  See class <a href= "OKScheduleEventButtonListener">
 * OKScheduleEventButtonListener.html </a> for details of how the
 * Schedule.scheduleEvent method is invoked.
 *
 */

public class ScheduleEventDialogGridBag extends View {

    public ScheduleEventDialogGridBag(Screen screen, Schedule schedule) {

        /*
         * Call the parent constructor.
         */
        super(screen, schedule);

        /*
         * Allocate a constraints object for use with the gridbag layout.
         */
        constraints = new GridBagConstraints();
    }

    public Component compose() {

        /*
         * Make a new window for this, which in Java will be a JFrame -- the
         * standard outermost container for a Swing window.
         */
        window = new mvp.Window();

        /*
         * Add a JPanel to the window.  JPanel is the standard background
         * container for holding Swing components.
         */
        panel = new JPanel();
        window.add(panel);

        /*
         * Control the panel layout using a GridBagLayout.
         */
        layout = new GridBagLayout();
        panel.setLayout(layout);

        /*
         * Compose each of the rows in turn by adding them to the grid.
         */
        composeTitleRow();
        composeStartAndEndDateRow();
        composeCategoryAndLocationRow();
        composeButtonRow();

        /*
         * Set the window titlebar.
         */
        window.setTitle("Schedule an Event");

        /*
         * Call JFrame.pack to have Java size up the window properly.
         */
        window.pack();

        /*
         * Return the window to the caller.
         */
        return window;
    }

    protected void composeTitleRow() {

        /*
         * Construct the label and text field for the title component of the
         * dialog.
         */
        JLabel label = new JLabel("Title:");
        JTextField textField = new JTextField();

        /*
         * Insert the label as the first component of a new grid row.
         */
        il(label, 0, 0);
//      insertLabel(label, 0.0, true, 0);

        /*
         * Insert the text field as the second component of the row.
         */
        itf(textField, 1, 0, 300);
//      insertLabeledItem(textField, 1.0, 400, true, 0);

    }

```

```java
    protected void composeStartAndEndDateRow() {

        /*
         * Construct the labels and text fields.
         */
        JLabel startLabel = new JLabel("Start Date:");
        JTextField startTextField = new JTextField();
        JLabel endLabel = new JLabel("End Date:");
        JTextField endTextField = new JTextField();

        /*
         * Insert all four components.
         */
        il(startLabel, 0, 1);
        itf(startTextField, 1, 1, 200);
        il(endLabel, 2, 1);
        itf(startTextField, 3, 1, 200);

        /*
        insertLabel(startLabel, 0.0, false, 4);
        insertLabeledItem(startTextField, 1.0, 200, false, 4);
        insertLabel(endLabel, 1.0, true, 0);
        insertLabeledItem(endTextField, 1.0, 200, true, 0);
         */
    }

    protected void composeCategoryAndLocationRow() {

    }

    protected void composeButtonRow() {

    }


    protected void il(JLabel l, int x, int y) {
        constraints.gridx = x;
        constraints.gridx = y;
        constraints.fill = GridBagConstraints.NONE;
        constraints.anchor = GridBagConstraints.EAST;
        constraints.insets = new Insets(10, 10, 10, 2);
        constraints.ipadx = 0;
        layout.setConstraints(l, constraints);
        panel.add(l);
    }

    protected void itf(JTextField tf, int x, int y, int pad) {
        constraints.gridx = x;
        constraints.gridx = y;
        constraints.fill = GridBagConstraints.BOTH;
        constraints.anchor = GridBagConstraints.WEST;
        constraints.insets = new Insets(10, 2, 10, 10);
        constraints.ipadx = pad;
        layout.setConstraints(tf, constraints);
        panel.add(tf);
    }

    protected void insertLabel(JLabel label, double weightx,
            boolean relative, int width) {
        constraints.weightx = weightx;
        constraints.fill = GridBagConstraints.NONE;
        constraints.anchor = GridBagConstraints.EAST;
        constraints.insets = new Insets(10, 10, 10, 2);
        constraints.ipadx = 0;
        if (relative) constraints.gridwidth = GridBagConstraints.RELATIVE;
        else constraints.gridwidth = width;
        layout.setConstraints(label, constraints);
        panel.add(label);
    }

    protected void insertLabeledItem(Component item, double weightx, int ipadx,
            boolean remainder, int width) {
        constraints.weightx = weightx;
        constraints.fill = GridBagConstraints.BOTH;
        constraints.insets = new Insets(10, 2, 10, 10);
        constraints.ipadx = ipadx;
        if (remainder) constraints.gridwidth = GridBagConstraints.REMAINDER;
        else constraints.gridwidth = 1;
        layout.setConstraints(item, constraints);
        panel.add(item);
    }

    /** The grid bag that controls component layout in the panel */
    protected GridBagLayout layout;

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

    /** Gridbag constraints used to control the layout of each dialog
        component. */
    GridBagConstraints constraints;

}
```