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

import caltool.caltool_ui.*;
import caltool.file.*;
import caltool.edit.*;
import caltool.schedule.*;
import caltool.view.*;
import caltool.admin.*;
import caltool.options.*;
import caltool.help.*;
import caltool.caldb.*;
import mvp.Model;

/****
 *
 * Class CalendarTool is the top-level model class for the regular-user
 * Calendar Tool program.  CalendarTool has references to the functional model
 * classes of the tool: File, Edit, Schedule, View, Admin, Options, and Help.
 * There is also a reference to the CalendarDB class that houses the tool's
 * major data bases.
 *                                                                          <p>
 * The CalendarTool class also has the main method for the program.  This
 * method contructs the top-level model, view, and process classes.  It then
 * shows the top-level UI and let's the Java event loop take it from there.
 *                                                                          <p>
 * Functionalitywise, all of the model classes are autonomous units.  They each
 * do their own work as invoked by the user.  All that this top-level class
 * does is to construct the work-doing model classes and set up the initial
 * state of the tool when it is invoked from the outside operating system.
 *                                                                          <p>
 * See also the companion view class <a href= "caltool_ui/CalendarToolUI.html">
 * CalendarToolUI. </a>
 *                                                                          <p>
 * @author Gene Fisher (gfisher@calpoly.edu)
 * @version 6feb05
 *
 */
public class CalendarTool extends Model {

    /**
     * Construct this with the given companion view.  Call the submodel
     * constructors.  Initialize the start-up state based on default options
     * and command-line arguments.
     */
    public CalendarTool(CalendarToolUI calToolUI) {

        /*
         * Call the parent constructor.
         */
        super(calToolUI);

        /*
         * Construct and store the submodel instances.  Note that the
         * CalendarDB is constructed before the File so the latter can observe
         * the former for changes.
         */
        caldb = new CalendarDB();
        file = new File(null, caldb, this);
        edit = new Edit(null);
        schedule  = new Schedule(null, caldb);
        calView = new View(null, caldb);

        /*
         * Set up the initial state of the tool.
         */
        initialize();
    }

    /**
     * Implement the exit method to pass the buck to file.exit().  Per set up
     * performed in the companion CalendarToolUI view, this method is called
     * when the user closes the top-level menubar window, e.g., via the window
     * manager close button.
     */
    public void exit() {
        file.exit();
    }

    /**
     * Construct models, construct views, compose views, and fire the puppy up.
     */
    public static void main(String[] args) {
        mvp.Screen s;                        // The GUI screen
        CalendarTool calTool;            // The top-level Calendar model
        CalendarToolUI calToolUI;        // The top-level Calendar view

        /*
         * Construct the GUI screen, thereby initializing the GUI system.  In
         * this Java-based implementation, the GUI screen is managed entirely
         * by the Java runtime environment, so the only thing to do in the way
         * of screen construction is to set its look and feel, if desired.  In
         * other GUI toolkits, screen construction may involve more substantial
         * work.
         */
        s = new mvp.Screen();

        /*
         * Construct the top-level Calendar model.  It will in turn construct
         * all subsidiary model classes and the data objects they require.
         * Note that the view parameter is null, since the view has not yet
         * been constructed.  After it is, we call the calTool.setView method,
         * which is inherited from class Model.
         */
        calTool = new CalendarTool(null);

        /*
         * Construct and compose the Calendar Tool view.  Compose will lay out
         * all GUI components, including dialogs that appear during the course
         * of user interaction.
         */
        calToolUI = new CalendarToolUI(s, calTool);
```

```java
        calToolUI.compose();

        /*
         * Store the view in the model to enable two-way communication between
         * the model and the view.
         */
        calTool.setView(calToolUI);

        /*
         * Display the top-level view window on the UI screen.
         */
        calToolUI.show(10,10);

        /*
         * In Java, no call to View.run is necessary, since showing any window
         * on the screen automatically starts the event loop.  The program
         * subsequently exits when the System.exit function is called, e.g.,
         * in response to the user choosing the File Quit menu item.
         */

    }

    /** Return the File model. */
    public File getFile() { return file; }

    /** Return the Edit model. */
    public Edit getEdit() { return edit; }

    /** Return the Schedule model. */
    public Schedule getSchedule() { return schedule; }

    /** Return the View model. */
    public View getCalView() { return calView; }

    /** Return the Admin model. */
    public Admin getAdmin() { return admin; }

    /** Return the Options model. */
    public Options getOptions() { return options; }

    /** Return the Help model. */
    public Help getHelp() { return help; }


    /*-*
     * Protected methods
     */

    /**
     * Set up the initial state of the tool, based on default option values and
     * command-line arguments, if any.  Details TBD.
     */
    void initialize() {}


    /*-*
     * Data fields
     */

    /** File-handling module */
    protected File file;

    /** Basic editing module */
    protected Edit edit;

    /** Scheduling module */
    protected Schedule schedule;

    /** Calendar viewing module */
    protected View calView;

    /** Calendar administration module */
    protected Admin admin;

    /** Tool options module */
    protected Options options;

    /** Tool help module */
    protected Help help;

    /** Calendar database */
    protected CalendarDB caldb;

}
```