```java
package caltool.model;

import caltool.view.*;
import caltool.model.file.*;
import caltool.model.edit.*;
import caltool.model.schedule.*;
import caltool.model.view.*;
import caltool.model.admin.*;
import caltool.model.options.*;
import caltool.model.help.*;
import caltool.model.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>
 * 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 13apr15
 *
 */
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(null);
        file = new File(null, caldb);
        edit = new Edit(null);
        schedule  = new Schedule(null, caldb);
        calView = new View(null, caldb);
        admin = new Admin(null);
        options = new Options(null);

        /*
         * 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();
    }

    /** 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;

}
```