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

import caltool.caldb.*;

/****
 *
 * Like an Appointment, a Task adds a number of components to a generic
 * ScheduledItem.  A Task differs from an Appointment as follows: (1)
 * Appointments have Duration and Location, Tasks do not.  (2) For
 * Appointments, the priority is either 'Must' or 'Optional'; for Tasks,
 * priority is a positive integer indicating the relative priority of a task
 * compared to other tasks.  (3) Tasks have a CompletedFlag and CompletionDate
 * components; Appointments do not.
 *
 */

public class Task extends ScheduledItem {

    /**
     * Construct an empty task.
     */
    public Task() {
    }

    /**
     * Construct a task with the given field values.  Generate and store the
     * unique key for this task.
     */
    public Task(String title, Date startOrDueDate, Date endDate, Category
            category, Time dueTime, RecurringInfo recurringInfo, Security
            security, int priority, RemindInfo remindInfo, String details,
            boolean completedFlag, Date completionDate) {

        this.title = title;
        this.startOrDueDate = startOrDueDate;
        this.endDate = endDate;
        this.category = category;
        this.dueTime = dueTime;
        this.recurringInfo = recurringInfo;
        this.security = security;
        this.priority = priority;
        this.remindInfo = remindInfo;
        this.details = details;
        this.completedFlag = completedFlag;
        this.completionDate = completionDate;

        itemKey = new ItemKey(startOrDueDate, dueTime, null, title, priority);
    }


    /*-*
     * Process methods
     */

    /**
     * Return the unique key for this, consisting of date, time, title, and
     * priority.  Duration is unsed.
     */
    public ItemKey getKey() {
        return itemKey;
    }

    /*-*
     * Derived data fields
     */
    /** Due time of the task */
    protected Time dueTime;

    /** Defines if and how an task recurs */
    protected RecurringInfo recurringInfo;

    /** Indicates who can see that the task is scheduled */
    protected Security security;

    /** Defines the relative priority of this task compared to others */
    protected int priority;

    /** Indicates if and how user is reminded */
    protected RemindInfo remindInfo;

    /** Free-form text describing any specific appointment details */
    protected String details;

    /** CompletedFlag is true if a Task has been completed, false if not.  The
        system does not enforce any specific constraints on the setting of a
        task's CompletedFlag.  That is, the user may set or clear it at will.
        Hence the meaning of the CompletedFlag is up to user interpretation,
        particularly for recurring tasks.
     */
    protected boolean completedFlag;

    /** CompletionDate is date on which as task is completed.  The system does
        not enforce any specific constraints on the setting of a task's
        CompletionDate (other than it being a legal Date value).  The meaning
        of the CompletionDate value is up to user interpretation, particularly
        for recurring tasks.
     */
    protected Date completionDate;


    /*-*
     * Process data field
     */

    /** The uniqe key for storing this in the UserCalendar items list */
    protected ItemKey itemKey;

}
```