

```

Loading vc-cvs...
1 package caltool.schedule;
2
3 import caltool.caldb.*;
4 import mvp.*;
5
6 /**
7  *
8  * Class Appointment adds a number of components to a generic ScheduledItem.
9  * The StartTime and Duration indicate when the appointment starts and how long
10 * it lasts. The RecurringInfo defines if and how an appointment recurs. The
11 * Location is where it is held. The Security indicates who can see that the
12 * appointment is scheduled. Priority is how important the appointment is.
13 * RemindInfo indicates if and how the user is reminded of the appointment.
14 * Details are free form text describing any specific appointment details.
15 *
16 */
17
18 public class Appointment extends ScheduledItem {
19
20     /**
21      * Construct an empty appointment.
22      */
23     public Appointment() {
24         super();
25     }
26
27     /**
28      * Construct an appointment with the given field values. Generate and
29      * store the unique key for this appointment.
30      */
31     public Appointment(String title, Date startOrDueDate, Date endDate,
32                        Time startTime, Duration duration, RecurringInfo recurringInfo,
33                        Category category, String location, Security security,
34                        Priority priority, RemindInfo remindInfo, String details) {
35
36         this.title = title;
37         this.startOrDueDate = startOrDueDate;
38         this.endDate = endDate;
39         this.startTime = startTime;
40         this.duration = duration;
41         this.recurringInfo = recurringInfo;
42         this.category = category;
43         this.location = location;
44         this.security = security;
45         this.priority = priority;
46         this.remindInfo = remindInfo;
47         this.details = details;
48
49         itemKey = new ItemKey(startOrDueDate, startTime, duration, title, 0);
50     }
51 }
52
53 /*-*
54  * Access methods.
55
56     */
57
58     /**
59      * Return the start date.
60      */
61     public Date getStartDate() {
62         return startOrDueDate;
63     }
64
65     /**
66      * Return the end date.
67      */
68     public Date getEndDate() {
69         return endDate;
70     }
71
72     /**
73      * Return the start time.
74      */
75     public Time getStartTime() {
76         return startTime;
77     }
78
79     /**
80      * Return the duration.
81      */
82     public Duration getDuration() {
83         return duration;
84     }
85
86     /**
87      * Return the recurring info.
88      */
89     public RecurringInfo getRecurringInfo() {
90         return recurringInfo;
91     }
92
93     /**
94      * Return the location.
95      */
96     public String getLocation() {
97         return location;
98     }
99
100    /**
101     * Return the security.
102     */
103    public Security getSecurity() {
104        return security;
105    }
106
107    /**
108     * Return the priority.
109     */
110    public Priority getPriority() {
111        return priority;

```

```
112     }
113
114     /**
115      * Return the remind info.
116      */
117     public RemindInfo getRemindInfo() {
118         return remindInfo;
119     }
120
121     /**
122      * Return the details.
123      */
124     public String getDetails() {
125         return details;
126     }
127
128
129     /**-
130      * Process methods
131      */
132
133     /**
134      * Return the unique key for this, consisting of date, time, duration, and
135      * title. Priority is unused at 0. Note that this method need not be
136      * specialized in Meeting, since appointments and meetings have the same
137      * key formats.
138      */
139     public ItemKey getKey() {
140         return itemKey;
141     }
142
143
144     /**-
145      * Derived data fields, in addition to those inherited from ScheduledItem.
146      */
147     /** Starting time of the appointment */
148     protected Time startTime;
149
150     /** How long the appointment lasts */
151     protected Duration duration;
152
153     /** Defines if and how an appointment recurs */
154     protected RecurringInfo recurringInfo;
155
156     /** Where the appointment is held */
157     protected String location;
158
159     /** Indicates who can see that the appointment is scheduled */
160     protected Security security;
161
162     /** How important the appointment is */
163     protected Priority priority;
164
165     /** Indicates if and how user is reminded */
166     protected RemindInfo remindInfo;
167
168     /** Free-form text describing any specific appointment details */
169     protected String details;
170
171
172     /**-
173      * Process data field
174      */
175
176     /** The unique key for storing this in the UserCalendar items list */
177     protected ItemKey itemKey;
178 }
```