

```

Loading vc-cvs...
1 package caltool.edit;
2
3 import mvp.*;
4
5 /****
6 *
7 * Class Edit is the model class for the Calendar Tool edit handling. It
8 * contains methods for all of the operations defined on the Edit menu, which
9 * constitute the functional command group for basic editing.
10 *
11 */
12 public class Edit extends Model {
13
14     /**
15      * Construct this.
16      */
17     public Edit(View v) {
18         super(v);
19     }
20
21     /**
22      * Undo the most recent scheduling operation. Only one level of undo/redo
23      * is specified here. Successive invocations of undo toggle between the
24      * current and previous states of the calendar. Note that the calendar
25      * editing operations support undo by saving a snapshot of the input
26      * calendar in the previous_state component of the workspace.
27      */
28     public void undo() {
29         System.out.println("In Edit.undo");
30     }
31
32     /**
33      * Redo the last undo.
34      */
35     public void redo() {
36         System.out.println("In Edit.redo");
37     }
38
39     /**
40      * Redo the given operation the given number of times.
41      */
42     public void repeat(CalendarToolOperation operation, int repeatTimes) {
43         System.out.println("In Edit.repeat");
44     }
45
46     /**
47      * The currently selected text segment is copied into the clipboard and
48      * removed from its context. The workspace selection is set to empty.
49      */
50     public void cut() {
51         System.out.println("In Edit.cut");
52     }
53
54     /**
55      * The currently selected text segment is copied into the clipboard;
56
57      */
58     public void copy() {
59         System.out.println("In Edit.copy");
60     }
61
62     /**
63      * Paste the contents of the clipboard into the currently selected start
64      * position, replacing any selected text from start to end position.
65      */
66     public void paste() {
67         System.out.println("In Edit.paste");
68     }
69
70     /**
71      * Delete the currently selected text. This function can be specialized by
72      * subclasses to delete a tool-specific object. */
73     public void delete() {
74         System.out.println("In Edit.delete");
75     }
76
77     /**
78      * Select all text in the current workspace window.
79      */
80     public void selectAll() {
81         System.out.println("In Edit.selectAll");
82     }
83
84     /**
85      * Find the given text string in all active windows.
86      */
87     public void find(String text) {
88         System.out.println("In Edit.find");
89     }
90
91     /** Data representation forthcoming */
92 }

```