

```

1 // fo1/7/juno/ShellCommand.java
2 //
3 //
4 // Copyright 2003 Bill Campbell and Ethan Bolker
5 import java.util.*;
6
7 /**
8  * Model those features common to all ShellCommands.
9  *
10 * Each concrete extension of this class provides a constructor
11 * and an implementation for method doit().
12 *
13 * @version 7
14 */
15
16 public abstract class ShellCommand
17 {
18     private String helpString; // documents the command
19     private String argString; // any args to the command
20
21     /**
22      * A constructor, always called (as super()) by the subclass.
23      * Used only for commands that have arguments.
24      *
25      * @param helpString a brief description of what the command does.
26      * @param argString a prototype illustrating the required arguments.
27      */
28
29     protected ShellCommand( String helpString, String argString )
30     {
31         this.argString = argString;
32         this.helpString = helpString;
33     }
34
35     /**
36      * A constructor for commands having no arguments.
37      *
38      * @param helpString a brief description of what the command does.
39      */
40
41     protected ShellCommand( String helpString )
42     {
43         this( helpString, "" );
44     }
45
46     /**
47      * Execute the command.
48      *
49      * @param args the remainder of the command line.
50      * @param sh the current shell
51      *
52      * @exception JunoException for reporting errors
53      */
54
55     public abstract void doit( StringTokenizer args, Shell sh )
56

```

```

57     throws JunoException;
58
59     /**
60      * Help for this command.
61      *
62      * @return the help string.
63      */
64
65     public String getHelpString()
66     {
67         return helpString;
68     }
69
70     /**
71      * The argument string prototype.
72      *
73      * @return the argument string prototype.
74      */
75
76     public String getArgString()
77     {
78         return argString;
79     }
80 }

```