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

```
57        public abstract void doIt( StringTokenizer args, Shell sh )
58            throws JunoException;
59
60        /**
61         * Help for this command.
62         *
63         * @return the help string.
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        public String getArgString()
76        {
77            return argString;
78        }
79    }
80
81
```