```
 1  // joi/10/juno/User.java
 2  //
 3  //
 4  // Copyright 2003 Ethan Bolker and Bill Campbell
 5
 6  /**
 7   * Model a juno user.  Each User has a login name, password,
 8   * a home directory, and a real name.
 9   * name.
10   *
11   * @version 10
12   */
13
14  public class User
15      implements java.io.Serializable
16  {
17      private String name;     // the User's login name
18      private String password; // The user's login password.
19      private Directory home;  // her home Directory
20      private String realName; // her real name
21
22      /**
23       * Construct a new User.
24       *
25       * @param name      the User's login name.
26       * @param password  the user's login password.
27       * @param home      her home Directory.
28       * @param realName  her real name.
29       */
30
31      public User( String name, String password,
32                   Directory home, String realName )
33      {
34          this.name     = name;
35          this.password = password;
36          this.home     = home;
37          this.realName = realName;
38      }
39
40      /**
41       *   Confirm password. Throw a JunoException on failure.
42       *
43       *   @param guess the string to test against the password.
44       *
45       *   @exception JunoException
46       *      if password fails to match
47       */
48
49      public void matchPassword( String guess ) throws JunoException
50      {
51          if (!guess.equals( password )) {
52              throw new JunoException( "bad password" );
53          }
54      }
55
56      /**
```



```
57       * Get the User's login name.
58       *
59       * @return the name.
60       */
61
62      public String getName()
63      {
64          return name;
65      }
66
67      /**
68       * Convert the User to a String.
69       * The String representation for a User is her
70       * login name.
71       *
72       * @return the User's name.
73       */
74
75      public String toString()
76      {
77          return getName();
78      }
79
80      /**
81       * Get the User's home Directory.
82       *
83       * @return the home Directory.
84       */
85
86      public Directory getHome()
87      {
88          return home;
89      }
90
91      /**
92       * Get the user's real name.
93       *
94       * @return the real name.
95       */
96
97      public String getRealName()
98      {
99          return realName;
100     }
101 }
```