

```
1  /****
2  *
3  * LeafNode extends TreeNode by adding an Object value field, but no subtree
4  * references.
5  *
6  */
7  public class LeafNode extends TreeNode {
8
9      /**
10     * Construct with the given id and value.
11     */
12     public LeafNode(int id, Object value) {
13         super(id);
14         this.value = value;
15     }
16
17     /**
18     * A la the other constructor, but with line and column numbers.
19     */
20     public LeafNode(int id, Object value, int line, int column) {
21         super(id, line, column);
22         this.value = value;
23     }
24
25     /**
26     * Return the string representation of this leaf, which starts with the
27     * String value of its ID.  If the value field of this leaf is not null,
28     * then the return value has appneded a colon, followed by the toString of
29     * the value.
30     */
31     public String toString(int level) {
32         return symPrint(id) +
33             (value == null ? "" : (": " + value.toString())) +
34             toStringLineAndColumn(" ");
35     }
36
37     /** The generic value of this node, used for identifier names and literal
38     * values. */
39     public Object value;
40
41 }
```