

```
1  /**
2  *
3  * TreeNode2 extends TreeNode by adding two child components, which are
4  * references other TreeNodes. Hence, TreeNode2 is used to represent binary
5  * syntactic constructs in a parse tree.
6  *
7  */
8  public class TreeNode2 extends TreeNode {
9
10     /**
11     * Construct this with the given id and child TreeNode references.
12     */
13     public TreeNode2(int id, TreeNode child1, TreeNode child2) {
14         super(id);
15         this.child1 = child1;
16         this.child2 = child2;
17     }
18
19     /**
20     * Return the String representation of this subtree, which is the String
21     * value of its ID, followed on the next two indented lines by the
22     * recursive toString of its two children. See the documentation for <a
23     * href= "TreeNode.html#toString()"> TreeNode.toString() </a> for a general
24     * description the way trees are represented as strings.
25     */
26     public String toString(int level) {
27         String indent = "";
28         for (int i = 0; i < level; i++) {
29             indent += " ";
30         }
31         return symPrint(id) + "\n" +
32             indent + " " + (child1 == null ? "null" : child1.toString(level+1)) + "\n" +
33             indent + " " + (child2 == null ? "null" : child2.toString(level+1));
34     }
35
36     /** Reference to the left child of this node. */
37     public TreeNode child1;
38
39     /** Reference to the right child of this node. */
40     public TreeNode child2;
41
42 }
```