

```

1  /****
2  *
3  * TreeNode3 extends TreeNode by adding three child components, which are
4  * references other TreeNodes. Hence, TreeNode3 is used to represent ternary
5  * syntactic constructs in a parse tree.
6  *
7  */
8  public class TreeNode3 extends TreeNode {
9
10     /**
11     * Construct this with the given id and child TreeNode references.
12     */
13     public TreeNode3(int id, TreeNode child1, TreeNode child2,
14         TreeNode child3) {
15         super(id);
16         this.child1 = child1;
17         this.child2 = child2;
18         this.child3 = child3;
19     }
20
21     /**
22     * A la the other constructor, but with line and column numbers.
23     */
24     public TreeNode3(int id, TreeNode child1, TreeNode child2,
25         TreeNode child3, int line, int column) {
26         super(id, line, column);
27         this.child1 = child1;
28         this.child2 = child2;
29         this.child3 = child3;
30     }
31
32     /**
33     * Return the String representation of this subtree, which is the String
34     * value of its ID, followed on the next three indented lines by the
35     * recursive toString() of its three children. See the documentation for <a
36     * href= "TreeNode.html#toString()"> TreeNode.toString() </a> for a general
37     * description the way trees are represented as strings.
38     */
39     public String toString(int level) {
40         String indent = "";
41         for (int i = 0; i < level; i++) {
42             indent += " ";
43         }
44         return symPrint(id) + toStringLineAndColumn(" ") + "\n" +
45             indent + " " +
46             (child1 == null ? "null" : child1.toString(level+1)) + "\n" +
47             indent + " " +
48             (child2 == null ? "null" : child2.toString(level+1)) + "\n" +
49             indent + " " +
50             (child3 == null ? "null" : child3.toString(level+1));
51     }
52
53     /** Reference to the left child of this node. */
54     public TreeNode child1;
55
56     /** Reference to the middle child of this node. */
57     public TreeNode child2;
58
59     /** Reference to the right child of this node. */
60     public TreeNode child3;
61
62 }

```