```java
/****
 *
 * TreeNode3 extends TreeNode by adding three child components, which are
 * references other TreeNodes.  Hence, TreeNode3 is used to represent trinary
 * syntactic constructs in a parse tree.
 *
 */
public class TreeNode3 extends TreeNode {

    /**
     * Construct this with the given id and child TreeNode references.
     */
    public TreeNode3(int id, TreeNode child1, TreeNode child2,
            TreeNode child3) {
        super(id);
        this.child1 = child1;
        this.child2 = child2;
        this.child3 = child3;
    }

    /**
     * A la the other constructor, but with line and column numbers.
     */
    public TreeNode3(int id, TreeNode child1, TreeNode child2,
            TreeNode child3, int line, int column) {
        super(id, line, column);
        this.child1 = child1;
        this.child2 = child2;
        this.child3 = child3;
    }

    /**
     * Return the String representation of this subtree, which is the String
     * value of its ID, followed on the next three indented lines by the
     * recursive toString of its three children.  See the documentation for <a
     * href= "TreeNode.html#toString()"> TreeNode.toString() </a> for a general
     * description the way trees are represented as strings.
     */
    public String toString(int level) {
        String indent = "";
        for (int i = 0; i < level; i++) {
            indent += "  ";
        }
        return symPrint(id) + toStringLineAndColumn(" ") + "\n" +
            indent + "  " +
                (child1 == null ? "null" : child1.toString(level+1)) + "\n" +
            indent + "  " +
                (child2 == null ? "null" : child2.toString(level+1)) + "\n" +
            indent + "  " +
                (child3 == null ? "null" : child3.toString(level+1));
    }

    /** Reference to the left child of this node. */
    public TreeNode child1;

    /** Reference to the middle child of this node. */
    public TreeNode child2;

    /** Reference to the right child of this node. */
    public TreeNode child3;

}
```