

```

43 public class TreeBarrier implements Barrier {
44     int radix;
45     Node[] leaf;
46     int leaves;
47     ThreadLocal<Boolean> threadSense;
48
49     public TreeBarrier(int n, int r) {
50         radix = r;
51         leaves = 0;
52         leaf = new Node[n / r];
53         int depth = 0;
54         threadSense = new ThreadLocal<Boolean>() {
55             protected Boolean initialValue() { return true; };
56         };
57         // compute tree depth
58         while (n > 1) {
59             depth++;
60             n = n / r;
61         }
62         Node root = new Node();
63         build(root, depth - 1);
64     }
65     // recursive tree constructor
66     void build(Node parent, int depth) {
67         if (depth == 0) {
68             leaf[leaves++] = parent;
69         } else {
70             for (int i = 0; i < radix; i++) {
71                 Node child = new Node(parent);
72                 build(child, depth - 1);
73             }
74         }
75     }
76     ...
77 }

```

FIGURE 18.7 The TreeBarrier class: initializing a combining tree barrier. The build() method creates r children for each node, and then recursively creates the children's children. At the bottom, it places leaves in an array.