

```
13  private class Node {
14      AtomicInteger count;
15      Node parent;
16      volatile boolean sense;
17
18      public Node() {
19          sense = false;
20          parent = null;
21          count = new AtomicInteger(radix);
22      }
23      public Node(Node myParent) {
24          this();
25          parent = myParent;
26      }
27      public void await() {
28          boolean mySense = threadSense.get();
29          int position = count.getAndDecrement();
30          if (position == 1) { // I'm last
31              if (parent != null) { // Am I root?
32                  parent.await();
33              }
34              count.set(radix);
35              sense = mySense;
36          } else {
37              while (sense != mySense) {};
38          }
39          threadSense.set(!mySense);
40      }
41  }
```

FIGURE 18.6 The TreeBarrier class: internal tree node.