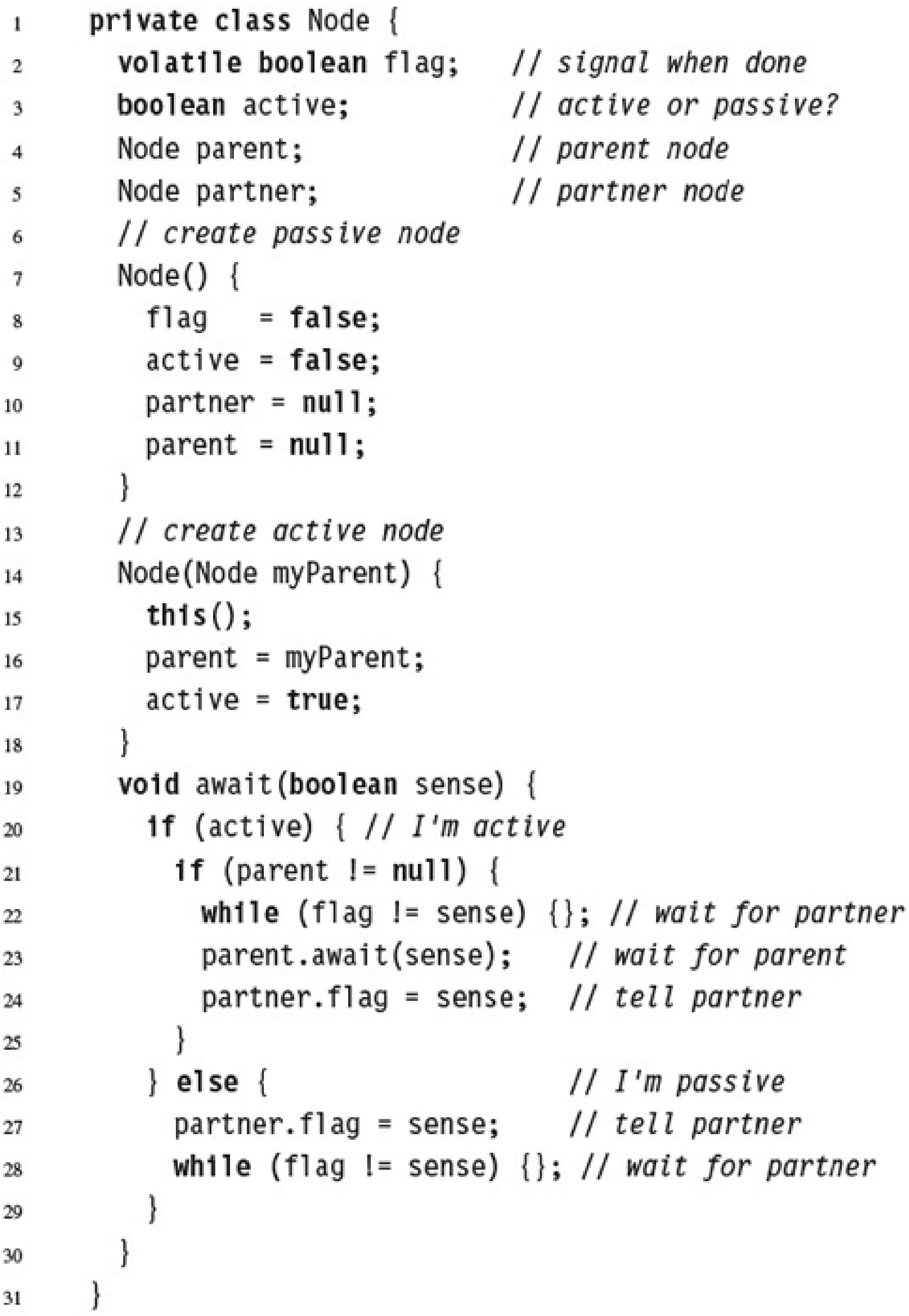

```
private class Node {
  volatile boolean flag;   // signal when done
  boolean active;          // active or passive?
  Node parent;             // parent node
  Node partner;            // partner node
  // create passive node
  Node() {
    flag   = false;
    active = false;
    partner = null;
    parent = null;
  }
  // create active node
  Node(Node myParent) {
    this();
    parent = myParent;
    active = true;
  }
  void await(boolean sense) {
    if (active) { // I'm active
      if (parent != null) {
        while (flag != sense) {}; // wait for partner
        parent.await(sense);    // wait for parent
        partner.flag = sense;   // tell partner
      }
    } else {                    // I'm passive
      partner.flag = sense;     // tell partner
      while (flag != sense) {}; // wait for partner
    }
  }
}
```

FIGURE 18.15 The TourBarrier class.