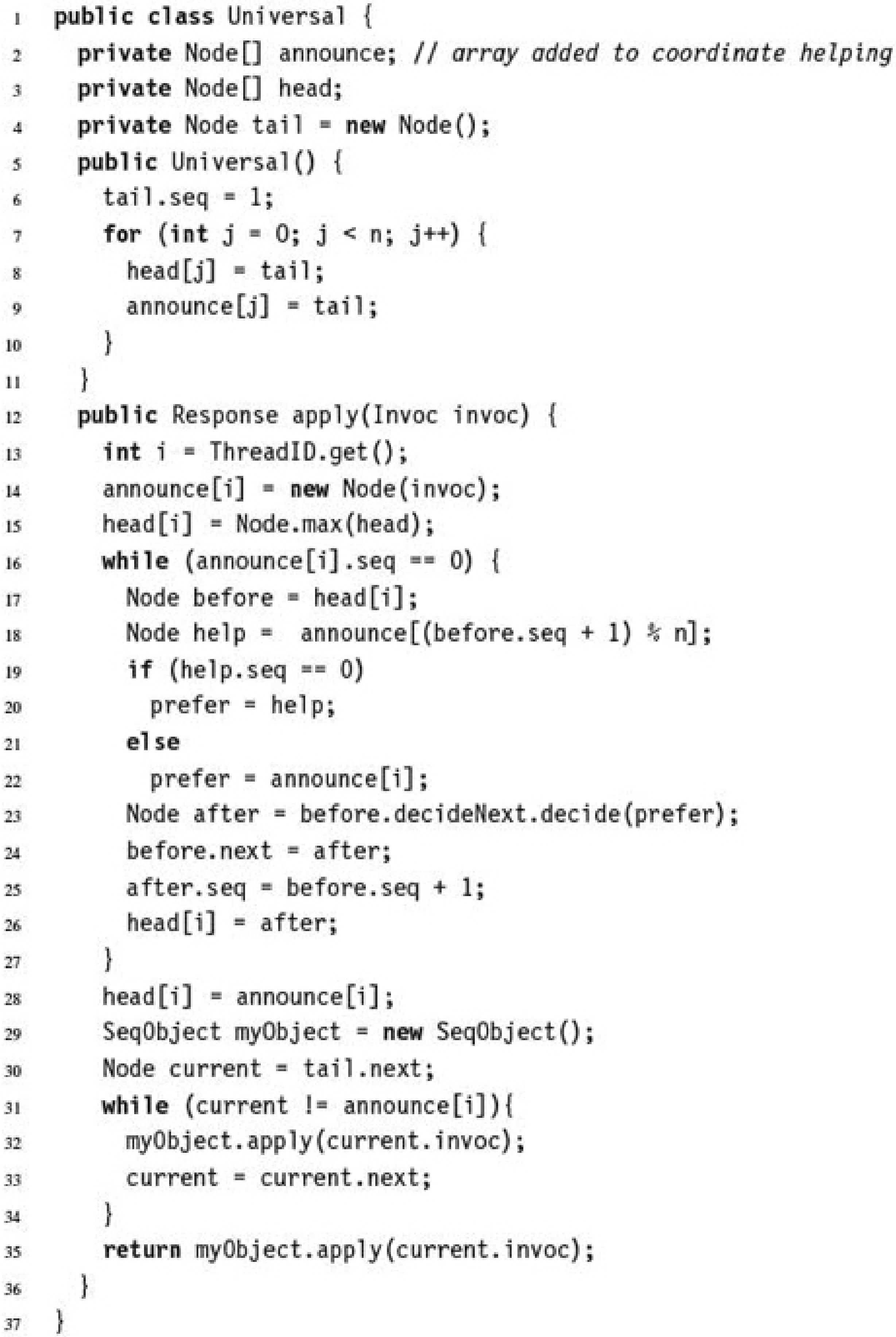

```java
public class Universal {
  private Node[] announce; // array added to coordinate helping
  private Node[] head;
  private Node tail = new Node();
  public Universal() {
    tail.seq = 1;
    for (int j = 0; j < n; j++) {
      head[j] = tail;
      announce[j] = tail;
    }
  }
  public Response apply(Invoc invoc) {
    int i = ThreadID.get();
    announce[i] = new Node(invoc);
    head[i] = Node.max(head);
    while (announce[i].seq == 0) {
      Node before = head[i];
      Node help =  announce[(before.seq + 1) % n];
      if (help.seq == 0)
        prefer = help;
      else
        prefer = announce[i];
      Node after = before.decideNext.decide(prefer);
      before.next = after;
      after.seq = before.seq + 1;
      head[i] = after;
    }
    head[i] = announce[i];
    SeqObject myObject = new SeqObject();
    Node current = tail.next;
    while (current != announce[i]){
      myObject.apply(current.invoc);
      current = current.next;
    }
    return myObject.apply(current.invoc);
  }
}
```

**FIGURE 6.6** The wait-free universal construction.