

```
1 class Driver {
2     void main() {
3         CountDownLatch startSignal = new CountDownLatch(1);
4         CountDownLatch doneSignal = new CountDownLatch(n);
5         for (int i = 0; i < n; ++i) // start threads
6             new Thread(new Worker(startSignal, doneSignal)).start();
7         doSomethingElse();           // get ready for threads
8         startSignal.countDown();   // unleash threads
9         doSomethingElse();           // biding my time ...
10        doneSignal.await();       // wait for threads to finish
11    }
12    class Worker implements Runnable {
13        private final CountDownLatch startSignal, doneSignal;
14        Worker(CountDownLatch myStartSignal, CountDownLatch myDoneSignal) {
15            startSignal = myStartSignal;
16            doneSignal = myDoneSignal;
17        }
18        public void run() {
19            startSignal.await();      // wait for driver's OK to start
20            doWork();
21            doneSignal.countDown(); // notify driver we're done
22        }
23        ...
24    }
25 }
```

FIGURE 8.14 The CountDownLatch class: an example usage.