```
1  class FastPath implements Lock {
2    private Lock lock;
3    private int x, y = -1;
4    public void lock() {
5      int i = ThreadID.get();
6      x = i;                          // I'm here
7      while (y != -1) {}              // is the lock free?
8      y = i;                          // me again?
9      if (x != i)                     // Am I still here?
10       lock.lock();                  // slow path
11   }
12   public void unlock() {
13     y = -1;
14     lock.unlock();
15   }
16 }
```

FIGURE 2.18 Fast-path mutual exclusion algorithm used in Exercise 2.8.