```
1  public class CompositeLock implements Lock{
2    private static final int SIZE = ...;
3    private static final int MIN_BACKOFF = ...;
4    private static final int MAX_BACKOFF = ...;
5    AtomicStampedReference<QNode> tail;
6    QNode[] waiting;
7    ThreadLocal<QNode> myNode = new ThreadLocal<QNode>() {
8      protected QNode initialValue() { return null; };
9    };
10   public CompositeLock() {
11     tail = new AtomicStampedReference<QNode>(null,0);
12     waiting = new QNode[SIZE];
13     for (int i = 0; i < waiting.length; i++) {
14       waiting[i] = new QNode();
15     }
16   }
17   public void unlock() {
18     QNode acqNode = myNode.get();
19     acqNode.state.set(State.RELEASED);
20     myNode.set(null);
21   }
22   ...
23 }
```

FIGURE 7.24 The CompositeLock class: fields, constructor, and unlock() method.