

#	Source		Pattern	
	Original	Modified	Original	Modified
8	<pre> <del>-synchronized</del> void enqueue(final long seqno,     final boolean lastPacketInBlock,     final long offsetInBlock) { <del>- if (running) {</del>     final Packet p = new Packet(...);     LOG.debug(...);      ackQueue.addLast(p);     notifyAll(); } </pre>	<pre> +void enqueue(final long seqno,     final boolean lastPacketInBlock,     final long offsetInBlock) {      final Packet p = new Packet(...);     LOG.debug(...); + synchronized (this) +     if (running) { +         ackQueue.addLast(p); +         notifyAll();     } } </pre>	<pre> synchronized (obj) {     statements1     statements2 } </pre>	<pre> statements1 synchronized (obj) {     statements2 } </pre>
9	<pre> <del>-Membership</del> membership = null; public boolean hasMembers() {     if (membership == null) setupMembership();     return membership.hasMembers(); } synchronized void setupMembership() {     if ( membership == null ) {         membership = new Membership(             super.getLocalMember(true));     } } </pre>	<pre> +volatile Membership membership = null; public boolean hasMembers() {     if (membership == null) setupMembership();     return membership.hasMembers(); } synchronized void setupMembership() {     if ( membership == null ) {         membership = new Membership(             super.getLocalMember(true));     } } </pre>	<pre> T foo; </pre>	<pre> volatile T foo; </pre>
10	<pre> <del>-private volatile</del> int requestCount;  <del>-private volatile</del> int errorCount;  - requestCount++; </pre>	<pre> +private final AtomicInteger requestCount = + new AtomicInteger(0); +private final AtomicInteger errorCount = + new AtomicInteger(0); + requestCount.incrementAndGet(); </pre>	<pre> volatile T foo; </pre>	<pre> TT foo; </pre>