

#	Source		Pattern	
7				
8	<pre> synchronized (buffers) { if (...) { - if (spillWriter != null) { - spillWriter.close(); - } isFinished = true; } } </pre>	<pre> synchronized (buffers) { if (...) { isFinished = true; } +if (spillWriter != null) { + spillWriter.close(); +} </pre>	<pre> synchronized (obj) { statements1 statements2 } </pre>	<pre> synchronized (obj) { statements1 } statements2 </pre>
9				
10	<pre> -protected Membership membership = null; public boolean hasMembers() { if (membership == null) setupMembership(); return membership.hasMembers(); } protected synchronized void setupMembership() { if (membership == null) { membership = new Membership(super.getLocalMember(true)); } } </pre>	<pre> +protected volatile Membership membership = null; public boolean hasMembers() { if (membership == null) setupMembership(); return membership.hasMembers(); } protected synchronized void setupMembership() { if (membership == null) { membership = new Membership(super.getLocalMember(true)); } } </pre>	T foo;	volatile T foo;
11	<pre> -private volatile int requestCount; -private volatile int errorCount; - requestCount++; </pre>	<pre> +private final AtomicInteger requestCount = + new AtomicInteger(0); +private final AtomicInteger errorCount = + new AtomicInteger(0); + requestCount.incrementAndGet(); </pre>	volatile T foo;	TT foo;
12	<pre> -private volatile long genstamp; -public synchronized long nextStamp() { - this.genstamp++; - return this.genstamp; } </pre>	<pre> +private AtomicLong genstamp = + new AtomicLong(); +public long nextStamp() { + return genstamp.incrementAndGet(); } </pre>		