

what
spurious wakeup, 虚假唤醒

where
java.lang.Object#wait() 及其重载
java.util.concurrent.locks.Condition#await() 及其重载
java.util.concurrent.locks.Condition#awaitUntil

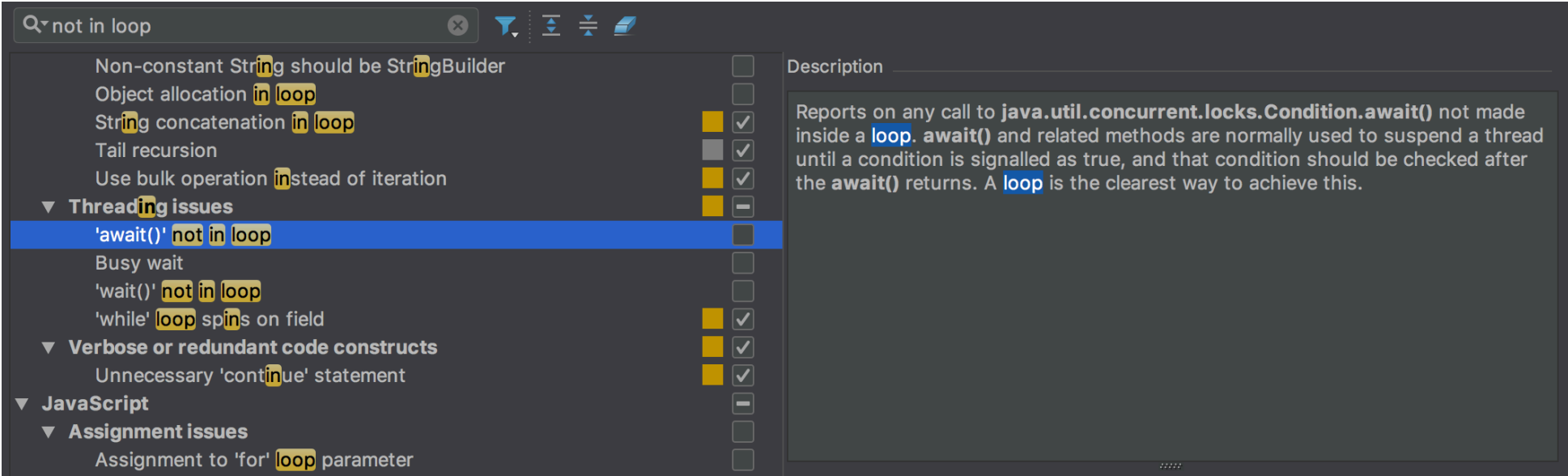
对策：while循环，重新检测条件或者timeout
在各个方法的javadoc中其实就有示例代码了。

Object.wait() is supposed to block until either another thread invokes the Object.notify() or Object.notifyAll() method, or a specified amount of time has elapsed. The various Condition.await() methods have similar behavior. However, it is possible for a thread to wake up without either of those occurring; these are called spurious wakeups. Because of spurious wakeups, Object.wait() and Condition.await() must always be called in a loop. The correct fix for this varies depending on what you are trying to do.

各种正确的用法见
<http://errorprone.info/bugpattern/WaitNotInLoop>

IntelliJ 默认未启用

Reports on any call to wait() not made inside a loop. wait() is normally used to suspend a thread until a condition is true, and that condition should be checked after the wait() returns. A loop is the clearest way to achieve this.



notifyAll、notify、signal、signalAll 执行后，并不会马上唤醒被唤醒线程
java doc 上有写，得等到lock释放后，才会被唤醒。