链滴

【笔记】第 4 章 Java 并发编程基础

作者：Changer0914

原文链接：https://ld246.com/article/1512474264953

来源网站：链滴

<h3 id="等待-通知机制">等待/通知机制</h3>

等待/通知机制，是指一个线程 A 调用了对象 O 的 wait()方法进入等待状态 WAITING，而另一线程 B 调用了对象 O 的 notify()或 notifyAll()方法，线程 A 收到通知后从对象 O 的 wait()方法返回进而执行后续操作。
使用 wait()、notify()和 notifyAll()时需要先对调用对象加锁。
notify()或 notifyAll()方法调用后，等待线程依旧不会从 wait()返回，需要调用 notify()或 notifyAl()方法的线程释放锁之后，等待线程才有机会从 wait()返回。
wait 方法会释放锁，notify 和 notifyAll 方法不会释放锁。
从 wait()方法返回的前提是获得了调用对象的锁。

<h3 id="经典范式">经典范式</h3>

等待方

<pre><code class="language-java highlight-chroma">synchronized(对象){ while(条不满足){   对象.wit(); } 对应的处理逻辑}</code></pre>

通知方

<pre><code class="language-java highlight-chroma">synchronized(对象){ 改变条件 对象.notfyAll();}</code></pre>
<h3 id="管道输入-输出流">管道输入/输出流</h3>

4 个类。PipedOutputStream、PipedInputStream、PipedReader、PipedWriter
使用前需要调用 connect()方法进行绑定。

<h3 id="ThreadLocal">ThreadLocal</h3>
深入剖析 ThreadLocal