目录

[1、 一级标题 2](#_Toc499559732)

[1.1、 二级标题 2](#_Toc499559733)

[1.1.1、 三级标题与正文与标号 2](#_Toc499559734)

# volatile关键字

1. 在当前的Java内存模型下，线程可以把变量保存在本地内存（比如机器的寄存器）中，而不是直接在主存中进行读写。这就可能造成一个线程在主存中修改了一个变量的值，而另外一个线程还继续使用它在寄存器中的变量值的拷贝，造成数据的不一致。
2. Volatile修饰的成员变量在每次被线程访问时，都强迫从共享内存中重读该成员变量的值。而且，当成员变量发生变化时，强迫线程将变化值回写到共享内存。这样在任何时刻，两个不同的线程总是看到某个成员变量的同一个值。

|  |
| --- |
| public class TestWithVolatile {  private static volatile boolean bChanged;  public static void main(String[] args) throws InterruptedException {  new Thread() {  @Override  public void run() {  for (;;) {  if (bChanged == !bChanged) {  System.out.println("!=");  System.exit(0);  } } }  }.start();  Thread.sleep(1);  new Thread() {  @Override  public void run() {  for (;;) {  bChanged = !bChanged;  } }  }.start();  } } |
| 结果：程序输出!=，然后马上退出。 |