

审查Java代码的十一种常见错误（3）PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022__E5_AE_A1_E6_9F_A5Java_c104_145415.htm

七、常见错误7#：用==替代.equals在Java中，有两种方式检查两个数据是否相等：通过使用==操作符，或者使用所有对象都实现的.equals方法。原子类型(int, float, char等)不是对象，因此他们只能使用==操作符，如下所示：
`int x = 4. int y = 5. if (x == y)`

`System.out.println ("Hi"). // This ' if ' test won ' t compile. if (x.equals (y)) System.out.println ("Hi").` 对象更复杂些，==操作符检查两个引用是否指向同一个对象，而equals方法则实现更专门的相等性检查。更显得混乱的是由java.lang.Object所提供的缺省的equals方法的实现使用==来简单的判断被比较的两个对象是否为同一个。许多类覆盖了缺省的equals方法以便更有用些，比如String类，它的equals方法检查两个String对象是否包含同样的字符串，而Integer的equals方法检查所包含的int值是否相等。大部分时候，在检查两个对象是否相等的时候你应该使用equals方法，而对于原子类型的数据，你应该使用==操作符。

八、常见错误8#：混淆原子操作和非原子操作 Java保证读和写32位数或者更小的值是原子操作，也就是说可以在一步完成，因而不可能被打断，因此这样的读和写不需要同步。以下的代码是线程安全(thread safe)的：

```
public class Example{ private int value. // More code here... public void set (int x){ // NOTE: No synchronized keyword this.value = x. } }
```

不过，这个保证仅限于读和写，下面的代码不是线程安全的：

```
public void increment (){ // This is effectively two or three
```

instructions: // 1) Read current setting of ' value ' . // 2) Increment that setting. // 3) Write the new setting back. this.value. } 在测试的时候，你可能不会捕获到这个错误。首先，测试与线程有关的错误是很难的，而且很耗时间。其次，在有些机器上，这些代码可能会被翻译成一条指令，因此工作正常，只有当在其它的虚拟机上测试的时候这个错误才可能显现。因此最好在开始的时候就正确地同步代码：
public synchronized void
increment () { this.value. }
100Test 下载频道开通，各类考试题目
直接下载。详细请访问 www.100test.com