

FindBugs 错误问题详解

作者: [windy1st](#)

原文链接: <https://ld246.com/article/1366080614852>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

1. **equals** **比较不同的对象类型**

Call to equals() comparing different types
This method calls equals(Object) on two references of different class types with no common subclasses. Therefore, the objects being compared are unlikely to be members of the same class at runtime (unless some application classes were not analyzed, or dynamic class loading can occur at runtime). According to the contract of equals(), objects of different classes should always compare as unequal; therefore, according to the contract defined by java.lang.Object.equals(Object), the result of this comparison will always be false at runtime. **说的是equals比较相同的对象类型**

2. **可能产生空指针异常**
A reference value dereferenced here might be null at runtime. This may lead to a NullPointerException when the code is executed.

3. **从未使用的本地变量**
This instruction assigns a value to a local variable, but the value is not read by any subsequent instruction. Often, this indicates an error, because the value computed is never used. Note that Sun's javac compiler often generates dead stores for final local variables. Because FindBugs is a bytecode-based tool, there is no easy way to eliminate these false positives.

4. **应该是一个静态内部类**
This class is an inner class, but does not use its embedded reference to the object which created it. This reference makes the instances of the class larger, and may keep the reference to the creator object alive longer than necessary. If possible, the class should be made static.

5. **方法名称第一个字母小写**
Method names should start with an lower case letter
Methods should be verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized.

6. **用包装类的valueOf代替NEW**
解释: 因为用new Integer(int) 这样的方式会产生一个新的对象 而当译时用valueOf则会被缓存, 并且速度更快。
Method invokes inefficient Number constructor; use static valueOf instead
Using new Integer(int) is guaranteed to always result in a new object whereas Integer.valueOf(int) allows caching of values to be done by the compiler, class library, or JVM. Using of cached values avoids object allocation and the code will be faster.

7. **无效的变量 (这个可能是FindBugs的问题)**
Bug: Dead store to result
Pattern id: DLS_DEAD_LOCAL_STORE, type: DLS, category: STYLE
This instruction assigns a value to a local variable, but the value is not read or used in any subsequent instruction. Often, this indicates an error, because the value computed is never used. Note that Sun's javac compiler often generates dead stores for final local variables. Because FindBugs is a bytecode-based tool, there is no easy way to eliminate these false positives. **该指令指定一个值给一局部变量, 但值不读或后续指令使用。通常, 这表明一个错误, 因为值计算是从未使用过。**
请注意, Sun的javac编译器经常产生局部变量的最终死亡商店。因为FindBugs是一个字节码为基的工具, 没有一种简单的方法来消除这些误报。

8. **可能忽略的异常**
Pattern id: DEIGHT_IGNORE, type: DE, category: BAD_PRACTICE
This method might ignore an exception. In general, exceptions should be handled or reported in some way, or they should be thrown out of the method. **这种方法可能会忽略一个例外。一般来说, 异常应以某种方式处理或报告的, 或者他们应该在方法外被抛出。**

9. **静态常量前增加final关键字**
Pattern id: MS_SHOULD_BE_FINAL, type: MS, category: MALICIOUS_CODE
A mutable static field could be changed by malicious code or by accident from another package. The field could be made final to avoid this vulnerability. **一个可变的静**

变量是可以改变的恶意代码或从另一包事故。使用final可以作出最终以避免这种脆弱性。 </p>