



链滴

关于 Java 中 Integer 和 Long 类型对象对比的问题

作者: [Hassan](#)

原文链接: <https://ld246.com/article/1471924156921>

来源网站: [链滴](#)

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

<p>其实这个问题也是在Java基础面试常常遇到的一个问题，以前没注意过，直到遇到bug，顺便记来吧。</p>

<p>做客户端“忘记密码”功能有一个bug，今天调试时，发现了原因：</p>

<p>功能模块中有一段：</p>

```
<div class="highlight highlight-source-java">
<pre class="brush: java">if(userpo.getId() != Long.valueOf(uid)){
    throw new VerifyException("mobile have been binded for uid=" + uid ,
        AppCode.VERIFY_MOBILE_IS_BIND);
}</pre>
</div>
```

<p>问题就出在两个Long型对象的比较。其实即使是Java初学者，学习过对象比较，也应该知道要用 equals方法，额...这个错误确实比较低级。</p>

<p>疑问：当初代码肯定是通过测试的，那当初是为什么会通过测试？带着疑问，查了一些资料。结果是Integer和Long内部实现是有缓存的。</p>

<p>例如我写了一个测试类测试Integer和Long类型对象的比较：</p>

```
<div class="highlight highlight-source-java">
<pre class="brush: java">public class Test {
    public static void main(String[] args) {
        Integer a = new Integer(10);
        Integer b = new Integer(10);
        System.out.println("a==b:" + (a==b)); //很明显false
        Integer c = 129;
        Integer d = 129;
        System.out.println("c==d:" + (c==d)); //false
        Integer e = 127;
        Integer f = 127;
        System.out.println("e==f:" + (e==f)); //true
        Long h = 128L;
        Long g = 128L;
        System.out.println("h==g:" + (h==g)); //false
        System.out.println("h.equals(g):" + h.equals(g)); //true
        System.out.println("h.compareTo(g):" + h.compareTo(g)); // 0
        Long k = 127L;
        Long m = 127L;
        System.out.println("k==m:" + (k==m)); //true

        System.out.println("k.equals(m):" + k.equals(m)); //true
        System.out.println("k.compareTo(m):" + k.compareTo(m)); // 0
    }
}</pre>
```

</div>

<p>执行结果：</p>

<div class="highlight highlight-source-shell">

```
<pre class="brush: bash">a==b:false
c==d:false
e==f:true
h==g:false
h.equals(g):true
h.compareTo(g):0
```

```
k==m:true  
k.equals(m):true  
k.compareTo(m):0</pre>
```

</div>

<p>查看Integer源代码后发现Integer有个内部类IntegerCache, 它维护了一个Integer数组cache[], 长度为256, 还有一个静态块</p>

<div class="highlight highlight-source-java">

```
<pre class="brush: java">static {  
    for(int i = 0; i &lt; cache.length; i++)  
        cache[i] = new Integer(i - 128);
```

```
</pre>
```

</div>

<p>很明显这个静态块已经默认创建出了 - 128 ~ 127 的 Integer 数据。Integer在创建对象时, 值在(-128到127)范围内, 则直接从缓冲区中取, 若超过该范围则创建新对象, 所以在-128到127范围内Integer对象值相同时, 对象 == 返回true</p>

<p>Long对象同理。</p>