链滴

如何理解 hashCode 的作用

作者：huihui
原文链接：https://ld246.com/article/1480060587100
来源网站：链滴

有许多人学了很长时间的Java，但一直不明白hashCode方法的作用，

我来解释一下吧。首先，想要明白hashCode的作用，你必须要先知道Java中的集合。

总的来说，Java中的集合（Collection）有两类，一类是List，再有一类是Set。

你知道它们的区别吗？前者集合内的元素是有序的，元素可以重复；后者元素无序，但元素不可重复。

那么这里就有一个比较严重的问题了：要想保证元素不重复，可两个元素是否重复应该依据什么来判呢？

这就是Object.equals方法了。但是，如果每增加一个元素就检查一次，那么当元素很多时，后添加集合中的元素比较的次数就非常多了。

也就是说，如果集合中现在已经有1000个元素，那么第1001个元素加入集合时，它就要调用1000次equals方法。这显然会大大降低效率。

于是，Java采用了哈希表的原理。哈希（Hash）实际上是个人名，由于他提出一哈希算法的概念，以就以他的名字命名了。

哈希算法也称为散列算法，是将数据依特定算法直接指定到一个地址上。如果详细讲解哈希算法，那要更多的文章篇幅，我在这里就不介绍了。

初学者可以这样理解，hashCode方法实际上返回的就是对象存储的物理地址（实际可能并不是）。

这样一来，当集合要添加新的元素时，先调用这个元素的hashCode方法，就一下子能定位到它应该置的物理位置上。

如果这个位置上没有元素，它就可以直接存储在这个位置上，不用再进行任何比较了；如果这个位置已经有元素了，

就调用它的equals方法与新元素进行比较，相同的话就不存了，不相同就散列其它的地址。

所以这里存在一个冲突解决的问题。这样一来实际调用equals方法的次数就大大降低了，几乎只需要两次。

所以，Java对于eqauls方法和hashCode方法是这样规定的：

1、如果两个对象相同，那么它们的hashCode值一定要相同；

2、如果两个对象的hashCode相同，它们并不一定相同

上面说的对象相同指的是用eqauls方法比较。

你当然可以不按要求去做了，但你会发现，相同的对象可以出现在Set集合中。同时，增加新元素的率会大大下降。hashcode这个方法是用来鉴定2个对象是否相等的。 那你会说，不是还有equals这方法吗？ 不错，这2个方法都是用来判断2个对象是否相等的。但是他们是有区别的。

一般来讲，equals这个方法是给用户调用的，如果你想判断2个对象是否相等，你可以重写equals方，然后在代码中调用，就可以判断他们是否相等 了。

简单来讲，equals方法主要是用来判断从表面上看或者从内容上看，2个对象是不是相等。举个例子有个学生类，属性只有姓名和性别，那么我们可以 认为只要姓名和性别相等，那么就说这2个对象是等的。 hashcode方法一般用户不会去调用，比如在hashmap中，由于key是不可以重复的，他在判key是不是重复的时候就判断了hashcode 这个方法，而且也用到了equals方法。这里不可以重复是说quals和hashcode只要有一个不等就可以了！

所以简单来讲，hashcode相 当于是一个对象的编码，就好像文件中的md5，他和equals不同就在于返回的是int型的，比较起来不直观。

我们一般在覆盖equals的同时也要 覆盖hashcode，让他们的逻辑一致。举个例子，还是刚刚的例子如果姓名和性别相等就算2个对象相等的话，那么hashcode的方法也要返回姓名 的hashcode值加上别的hashcode值，这样从逻辑上，他们就一致了。