

为什么重写equals方法必须要重写hashCode方法?

网络上解释的很全面但是很枯燥，也有些难懂，其实就是为了保证当该对象作为key时**哈希表**的检索效率。如HashMap的get方法是分两步获取的

- * 第一步通过key的哈希值找到对应的哈希桶
- * 第二步通过equals方法来判断是否为同一个key (因为可能出现哈希冲突)

假设一个Student类有三个属性：学号、姓名、年龄，但是只要学号相同我们就认为是同一个学生。这个时候我们重写equals方法

```
...
public class Student {
    private String idCard;
    private String name;
    private String age;

    @Override
    public boolean equals(Object obj) {
        if (obj instanceof Student) {
            if (this.idCard.equals(((Student) obj).idCard)) {
                return true;
            }
        }
        return false;
    }
}
...
```

如果不重写hashCode方法，当从一个Map里通过Student对象作为key检索时，如果只有学号相同而姓名和年龄不同就会导致hashCode大概率不同，那get方法检索时第一步通过hashCode就找不到这个对象，所以就会get为null，但是我们认为学号相同就是同一个对象，我们是想找到这个对象的，这与我们的初衷相违背。

所以我们约定重写equals方法就必须重写hashCode方法，即“约定一致性”

```
```
public class Student {
 private String idCard;
 private String name;
 private String age;

 @Override
 public boolean equals(Object obj) {
 if (obj instanceof Student) {
 if (this.idCard.equals(((Student) obj).idCard)) {
 return true;
 }
 }
 return false;
 }

 @Override
 public int hashCode() {
 return this.idCard.hashCode();
 }
}
````
```

这样get方法第一步就能通过相同的hashCode找到对象，然后再通过equals方法就能判断是同一个对象了

原文链接: <https://juejin.cn/post/7368823201068695602>