**JAVA   关于Map的key可不可以是自定义对象的学习**

**虽说经常用map，但是我用它的时候，key都是用String或者Integer类型（我相信大多数人都会是这样用的，哈哈），今天偶然想到，可不可以用自己定义的对象作为map的key呢？**

**一、尝试一下**

         首先我建了一个类

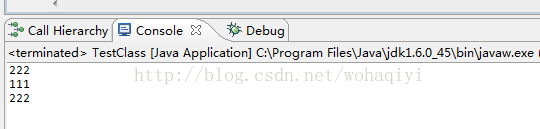
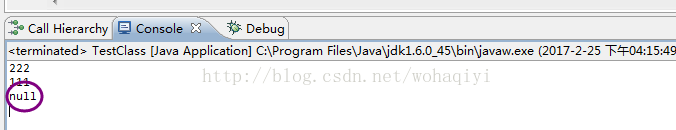
**[plain]** [view plain](https://blog.csdn.net/wohaqiyi/article/details/57083444?locationNum=9&fps=1) [copy](https://blog.csdn.net/wohaqiyi/article/details/57083444?locationNum=9&fps=1)

1. public class Student{
2. private String name;
3. private String sex;
4. public String getName() {
5. return name;
6. }
7. public void setName(String name) {
8. this.name = name;
9. }
10. public String getSex() {
11. return sex;
12. }
13. public void setSex(String sex) {
14. this.sex = sex;
15. }
16. }

           
         又做了如下操作，嘿嘿，一切都感觉好规范，期待结果

**[plain]** [view plain](https://blog.csdn.net/wohaqiyi/article/details/57083444?locationNum=9&fps=1) [copy](https://blog.csdn.net/wohaqiyi/article/details/57083444?locationNum=9&fps=1)

1. public static void main(String[] args) {
2. // TODO 自动生成的方法存根
3. Map map = new HashMap();
4. Student s1 = new Student();
5. s1.setName("一号");
6. s1.setSex("男");
7. Student s2 = new Student();
8. s2.setName("二号");
9. s2.setSex("女");
10. map.put(s1, "111");
11. map.put(s2, "222");
12. Set set = map.keySet(); //试了试遍历key，然后获取
13. for(Student s:set){
14. System.out.println(map.get(s));
15. }
16. System.out.println(map.get(s2)); //试了试单独根据key获取值
17. }

          一运行发现，他妈的好正确。。。  
  
                      
                  
       后来我就新建了一个Student对象，然后再次以这个对象为key去map里取值，然后我懵了，啥也没有了。  
                
                   Student s4 = new Student();  
              s4.setName("二号");  
              s4.setSex("女");  
              System.out.println(map.get(s4));  
  
                

       我再仔细看我写的代码，才发现，太蠢了。

     因为map存储获取都是根据hashcode值和equals方法有联系，前边将map的key拿到再放进去获取value，并没有发生对象的变化，hashcode值也不变，下边我又以新的对象去map中获取，此时的hashcode值已经变了，所以返回为null，因为那是我新new的对象所以，map已经不认识了。

     怎么才能解决这个问题呢。重写Student的hashcode方法和equals方法就可以了,如下：

**[plain]** [view plain](https://blog.csdn.net/wohaqiyi/article/details/57083444?locationNum=9&fps=1) [copy](https://blog.csdn.net/wohaqiyi/article/details/57083444?locationNum=9&fps=1)

1. class Student{
2. private String name;
3. private String sex;
4. public String getName() {
5. return name;
6. }
7. public void setName(String name) {
8. this.name = name;
9. }
10. public String getSex() {
11. return sex;
12. }
13. public void setSex(String sex) {
14. this.sex = sex;
15. }
16. @Override
17. public int hashCode() {
18. final int prime = 31;
19. int result = 1;
20. result = prime \* result + ((name == null) ? 0 : name.hashCode());
21. result = prime \* result + ((sex == null) ? 0 : sex.hashCode());
22. return result;
23. }
24. @Override
25. public boolean equals(Object obj) {
26. if (this == obj)
27. return true;
28. if (obj == null)
29. return false;
30. if (getClass() != obj.getClass())
31. return false;
32. Student other = (Student) obj;
33. if (name == null) {
34. if (other.name != null)
35. return false;
36. } else if (!name.equals(other.name))
37. return false;
38. if (sex == null) {
39. if (other.sex != null)
40. return false;
41. } else if (!sex.equals(other.sex))
42. return false;
43. return true;
44. }
46. }

    通过这种方式，就是new的新对象，只要是内容相同就会有相同的hashcode值，这样的话那map就认识这个新对象了。  
  
**总结：众所周知，map的key以不变的类型，比如String,Integer，作为key是最明智的，当然如果要以可变对象作为key的话，那就必须要重写hashcode和equals方法来达到这个目的，除此之外，别无他法。**