对于数组之间的比较，应该使用Array.equals()方法，如果是原生数组（即数组中的值是几大基本数据类型之一的）之间的比较可以直接用，如果数组中的值不是原生的基本数据类型，那么再使用Array.equals()方法时会主动调用Object.equals()方法，所以如果是自己定义的类，需要重写类的equals方法

# [JAVA中数组Arrays类的常见用法](https://www.cnblogs.com/Downtime/p/7757743.html)

import java.util.Arrays;  
  
int[] array1={7,8,3,2,12,6,5,4};

1.  //克隆clone    
int[] array2=array1.clone();    
System.out.println("arry2克隆的结果是:"+Arrays.toString(array2));

2.  //复制copyof

①方法一：  
int [] array3=Arrays.copyOf(array1, 4); // "4"表示复制的长度  
System.out.println("arry3复制copyof的结果:"+Arrays.toString(array3));    //输出：[7,8,2,12]

②方法二：  
System.arraycopy(src, srcPos, dest, destPos, length);  
src -- 这是源数组.  
srcPos -- 这是源数组中的起始位置。  
dest -- 这是目标数组。  
destPos -- 这是目标数据中的起始位置。  
length -- 这是一个要复制的数组元素的数目。  
  
异常:  
    IndexOutOfBoundsException -- 如果复制会导致访问数组边界之外的数据  
    ArrayStoreException -- 如果src数组中的元素不能被存储到dest数组中，因为类型不匹配  
    NullPointerException -- 如果src或dest为null  
  
例,  
int arr1[] = { 0, 1, 2, 3, 4, 5 };  
int arr2[] = { 11, 10, 20, 30, 40, 50 };  
//从数组arr1中的第0个位置开始复制3个数，复制到arr2中，从b中的1位置开始存放  
System.arraycopy(arr1, 0, arr2, 1, 3);     
System.out.println(Arrays.toString(arr2));  //输出：[11, 0, 1, 2, 40, 50]

3.判断两个数组是否相等：  
        String[] a={"a","b","c"};  
        String[] b={"a","b","c"};  
        System.out.println(a.equals(b));//false  
        System.out.println(a == b);//false  
        System.out.println(Arrays.equals(a, b));//true

注意：

基本数据类型的数组的equals方法没有重载Object的equals方法，所以跟“==”效果一样 ，所以你这里 a.equals(b) 效果和 a == b是一样的，==比较的是内存地址，结果是false。  
如果想要比较内容相同，可以使用Arrays.equals(a,b)。  
   
  
4.  //对数组的第2个到第7个进行排序进行排序  
Arrays.sort(array1,2,8);  
System.out.print("对数组的第2个到第6个元素进行排序进行排序：Arrays.sort(array1,2,7)：");  
System.out.println(Arrays.toString(array1));  
  
5.  //将arry1整体排序 sort    
Arrays.sort(array1);   
//toString 转换为字符串  
System.out.println("排序后的arry1的值:"+Arrays.toString(array1));    
  
6. //排序后arry1与arry2是否还相等 equals    
System.out.println("排序后arry1与arry2是否还相等："+Arrays.equals(array1, array2));    
  
7. //填充 fill    
     Array.fill(arrayname,value)；  
     Array.fill(arrayname ,starting index ,ending index ,value)  
例，  
Arrays.fill(array3, 6);  //将数字6填充到数组array3，输出：{6,6,6,6,6,6,6,6}  
Arrays.fill(array3, 2, 4, 9);  //将数组9从索引值2填充到索引值4  
System.out.println("填充之后arry2的值:"+Arrays.toString(array3));//输出：{7,8,6,6,12,6,5,4}  
    
8. //查询binarySearch   
System.out.println("数字6在arry1中的位置:"+Arrays.binarySearch(array1, 6));   
  
9. //将数组array1转换为List  
        String[] name=new String[]{"a","b","c"};  
        //转换为集合  
        List<String> list=Arrays.asList(name);  
        System.out.println(list);  
  
  
10.数组的删除：  
ArrayList<Integer> a = new ArrayList<Integer>();  
for(int i=0;i<10;i++){  
    a.add(i);  
}  
    a.remove(1);  
           
System.out.println(a);   //输出：[0, 2, 3, 4, 5, 6, 7, 8, 9]

11.数组的查找：  
①第一种方法：（这种方法需要用动态数组ArrayList）  
ArrayList a = new ArrayList();  
for(int i=0;i<10;i++){  
    a.add(i);  
}  
a.remove(0);  
System.out.println(a);   //输出：[1, 2, 3, 4, 5, 6, 7, 8, 9]  
System.out.println(a.contains(1)); //返回boolea值      输出：true  
  
  
②第二种方法：  
int[] aa={1,2,3};  
System.out.println("数字2在数组aa中的位置:"+Arrays.binarySearch(aa, 2));  //返回索引值  输出：1

# 遍历字符串中的字符

char[] ca = new char[26];

        //统计字符串中每个字符串出现的次数

        for (char c : s.toCharArray())

            ca[c - 'a']++;

如果此处写成for(char c : s)

会报错：error: for-each not applicable to expression type

for(char ch:s)

^

required: array or java.lang.Iterable

found: String

迭代的东西应该是数组或者是可迭代的东西