字符串对象是一种特殊的对象.String类是一个不可变的类..也就说,String对象一旦创建就不允许修改

String类有一个对应的String池,也就是 String pool.每一个内容相同的字符串对象都对应于一个pool里的对象.

1 看下面一段代码.

String s = new String("abc");  
  String s1 = "abc";  
  String s2 = new String("abc");  
    
  System.out.println(s == s1);  
  System.out.println(s == s2);  
  System.out.println(s1 == s2);

请问 前面三条语句分别创建了几个对象,分别是什么.后面的输出分别是什么

**(1)**String s = new String("abc"); 这句,创建了两个对象..其内容都是"abc".注意,s不是对象,只是引用.只有new生成的才是对象.

创建的流程是,首先括号里的"abc"先到String pool里看有没"abc"这个对象,没有则在pool里创建这个对象..所以这里就在pool创建了一个"abc"对象.然后 通过new语句又创建了一个"abc"对象..而这个对象是放在内存的堆里. .这里的s指向堆里的对象.

**(2)** String s1 = "abc"; 这条语句,s1当然还是引用.没啥可说的.后面的"abc".其实就是上面括号里的"abc".执行的是相同的操作.即 在pool里查找有没"abc"这个对象.没有则创建一个...很显然,第一条语句在pool里已经创建了一个"abc".所以这条语句没有创建对象,s1指向的是pool中的"abc"

**(3)**String s2 = new String("abc"); 这条语句,其实和第一条是一样的,但是,因为第一条已经在pool中创建了"abc"这个对象,所以,这条语句创建了一个对象.s2指向的是堆里的"abc".注意,虽然内容都是"abc",s与s2表示的是不同的对象

1. 接下来就很好说了.下面的三个==判断.(注意,==永远是判断内存地址是否相等) s与s1,一个指向堆里的对象,一个指向pool里的.很明显是不同的对象.s与s2.上面说了,虽然都是指向堆里的对象,内容也是"abc",但是也不是相同的对象.s1与s2.一个指向pool,一个指向堆.也不是相同的对象.所以三个都返回false.

2 第二个问题

String s = new String("abc");  
  String s1 = "abc";  
  String s2 = new String("abc");  
    
  System.out.println(s == s1.intern());  
  System.out.println(s == s2.intern());  
  System.out.println(s1 == s2.intern());

求最后输出是什么

解答.最后的答案是 false false true

intern()方法.按照jdk的帮助文档来说,是返回字符串对象的规范化表示形式。通俗一点说,就是返回对应这个字符串内容的那个pool里的对象.这样说也许还看不太明白,那可以拿具体例子来说

s1.intern().他的执行流程是,在pool里去查找s1对应的内容(也就是"abc").如果找到,则返回pool里的对象.如果没有(老实说,我没想到有哪种情况是没有的),则在Pool创建这个对象,并返回...

这样就很容易理解了.s1.intern返回的是pool里的"abc"对象.与s这个堆里的对象肯定不同,返回false.同理,s与s2.intern()也肯定不同,返回false.第三个,s1与s2.intern().其中s2.intern()返回的是pool中的"abc"对象,而s1也是指向pool中的"abc"对象.所以返回的是true:

3第三个问题

String hello = "hello";  
  String hel = "hel";  
  String lo = "lo";  
    
  System.out.println(hello == "hel" + "lo");  
  System.out.println(hello == "hel" + lo);

求输出的结果

解答 true false

首先,上面已经说明了,hello hel lo这三个都是指向pool中的对象..

现在我们考虑"hel" + "lo" 按照内容来说,两个相加也就是"hello".这个时候,这个会返回pool中的"hello"对象.所以,hello == "hel" + "lo" 返回的是true .

而"hel" + lo 虽然内容也是"hello",但是它将在堆里面生成一个"hello"对象,并返回这个对象...所以这里的结果是false

总结一下就是,如果加号两边的是字面值(字面值就是直接在""里写的值,比如上面的"hel"与"lo"),那么将在pool里查找有没对应内容的对象(这里的内容就是"hello"),并返回pool里的对象.这和hello是一样的....

如果加号两边不满足上面的条件(比如,两边的值是引用值或者堆里的字符串对象).那么将不会再pool里查找"hello",而是直接在堆里生成一个新的对象...