

# 两道小题目

作者: [eddy](#)

原文链接: <https://ld246.com/article/1477633571211>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## 两道小题目

```
@Test
public void test7() {
    String s1 = "abc";
    String s2 = "ab";
    String s3 = "c";
    String s4 = s2 + s3;

    String st0 = "helloworld";
    String st1 = "helloworld";
    String st2 = "hello" + "word";

    System.out.println(st0 == st2);
    System.out.println(s1 == (s2 + s3));
    System.out.println(s1 == s4);
}
```

```
@Test
public void test8() {
    int i = 0;
    i = i++;
    System.out.println(i);
}
```

- 第一题结果均为 **false**
- 第二题结果为 **0**

## 解析

利用[javap反汇编](#)查看jvm执行指令

### 第一题

```
public void test7();
Code:
  0: ldc      #39          // String abc
  2: astore_1
  3: ldc      #40          // String ab
  5: astore_2
  6: ldc      #41          // String c
  8: astore_3
  9: new      #19          // class java/lang/StringBuilder
 12: dup
 13: invokespecial #20      // Method java/lang/StringBuilder.<init>:()V
 16: aload_2
 17: invokevirtual #22      // Method java/lang/StringBuilder.append:(Ljava/lang/String)
)Ljava/lang/StringBuilder;
 20: aload_3
 21: invokevirtual #22      // Method java/lang/StringBuilder.append:(Ljava/lang/String)
)Ljava/lang/StringBuilder;
```

```

24: invokevirtual #25          // Method java/lang/StringBuilder.toString():Ljava/lang/Stri
g;
27: astore      4
29: ldc        #42          // String helloworld
31: astore      5
33: ldc        #42          // String helloworld
35: astore      6
37: ldc        #43          // String helloworld
39: astore      7
41: getstatic   #18          // Field java/lang/System.out:Ljava/io/PrintStream;
44: aload      5
46: aload      7
48: if_acmpne   55
51: iconst_1
52: goto       56
55: iconst_0
56: invokevirtual #33          // Method java/io/PrintStream.println:(Z)V
59: getstatic   #18          // Field java/lang/System.out:Ljava/io/PrintStream;
62: aload_1
63: new         #19          // class java/lang/StringBuilder
66: dup
67: invokespecial #20          // Method java/lang/StringBuilder.<init>:()V
70: aload_2
71: invokevirtual #22          // Method java/lang/StringBuilder.append:(Ljava/lang/String
)Ljava/lang/StringBuilder;
74: aload_3
75: invokevirtual #22          // Method java/lang/StringBuilder.append:(Ljava/lang/String
)Ljava/lang/StringBuilder;
78: invokevirtual #25          // Method java/lang/StringBuilder.toString():Ljava/lang/Stri
g;
81: if_acmpne   88
84: iconst_1
85: goto       89
88: iconst_0
89: invokevirtual #33          // Method java/io/PrintStream.println:(Z)V
92: getstatic   #18          // Field java/lang/System.out:Ljava/io/PrintStream;
95: aload_1
96: aload      4
98: if_acmpne   105
101: iconst_1
102: goto       106
105: iconst_0
106: invokevirtual #33          // Method java/io/PrintStream.println:(Z)V
109: return

```

可以看出字符串变量相加实际上JVM是优化为java/lang/StringBuilder.append, 最终输出为java/lang/StringBuilder.toString。

<br/>new String 则是直接引用堆中的对象, 而 "" 则是引用常量池中的字符串, 这就是三个输出均为false的原因。

## 第二题

```
public void test8();
```

Code:

```
0: iconst_0 -- 将常量0加载进栈中, 入栈
1: istore_1 -- 将栈顶int赋值给变量1, 出栈
2: iload_1 -- int变量1入栈
3: iinc     1, 1 -- 变量1自增1,不会入栈 i = 1
6: istore_1 -- 将栈顶int赋值给变量1, 出栈 i = 0
7: getstatic #18          // Field java/lang/System.out:Ljava/io/PrintStream;
10: iload_1 -- int变量1入栈
11: invokevirtual #29      // Method java/io/PrintStream.println:(I)V
14: return
```

由于执行完*iinc* 1, 1指令之后并没有将变量*i*入栈, 然后将栈顶的0赋值给了变量*i*  
<br/>而栈顶的0来自于*i++*的0。这就导致了最终的输出结果0。

## 对比

@Test

```
public void test8() {
    int i = 0;
    i++;
    System.out.println(i);
}
```

```
public void test8();
```

Code:

```
0: iconst_0
1: istore_1
2: iinc     1, 1
5: getstatic #18          // Field java/lang/System.out:Ljava/io/PrintStream;
8: iload_1
9: invokevirtual #29      // Method java/io/PrintStream.println:(I)V
12: return
```