**String.intern()原理**

String.intern()是一个Native方法，底层调用C++的 StringTable::intern 方法，源码注释：

**当调用 intern 方法时，如果常量池中已经有该字符串，则返回池中的字符串；否则将此字符串添加到常量池中，并返回字符串的引用。**

package com.ctrip.ttd.whywhy;

class Test {

public static void main(String args[]) {

String s1 = new StringBuilder().append("String").append("Test").toString();

System.out.println(s1.intern() == s1);**//新建并且返回引用**

String s2 = new StringBuilder().append("ja").append("va").toString();

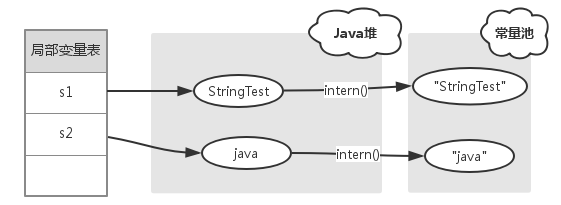
System.out.println(s2.intern() == s2);**//直接返回字符串**

}

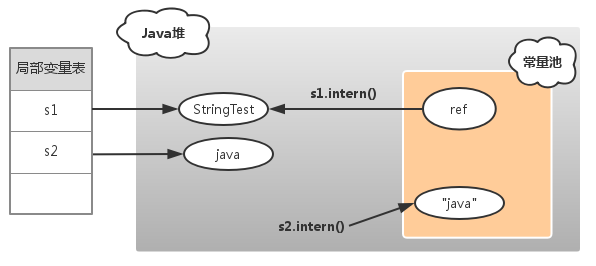
}

在 JDK6 和 JDK7 中结果不一样：

**1、JDK6的执行结果：false false**  
对于这个结果很好理解。在JDK6中，常量池在永久代分配内存，永久代和Java堆的内存是物理隔离的，执行intern方法时，如果常量池不存在该字符串，虚拟机会在常量池中复制该字符串，并返回引用，所以需要谨慎使用intern方法，避免常量池中字符串过多，导致性能变慢，甚至发生PermGen内存溢出。



**2、JDK7的执行结果：true false**  
对于这个结果就有点懵了。在JDK7中，常量池已经在Java堆上分配内存，执行intern方法时，如果常量池已经存在该字符串，则直接返回字符串引用，否则复制该字符串对象的引用到常量池中并返回，所以在JDK7中，可以重新考虑使用intern方法，减少String对象所占的内存空间。



对于变量s1，常量池中没有 "StringTest" 字符串，s1.intern() 和 s1都是指向Java对象上的String对象。  
对于变量s2，常量池中一开始就已经存在 "java" 字符串，所以 s2.intern() 返回常量池中 "java" 字符串的引用。

**String.intern()性能**

常量池底层使用StringTable数据结构保存字符串引用，实现和HashMap类似，根据字符串的hashcode定位到对应的数组，遍历链表查找字符串，当字符串比较多时，会降低查询效率。

在JDK6中，由于常量池在PermGen中，受到内存大小的限制，不建议使用该方法。  
在JDK7、8中，可以通过-XX:StringTableSize参数StringTable大小，下面通过几个测试用例看看intern方法的性能。

public class StringTest {

public static void main(String[] args) {

System.out.println(cost(1000000));

}

public static long cost(int num) {

long start = System.currentTimeMillis();

for (int i = 0; i < num; i++) {

String.valueOf(i).intern();

}

return System.currentTimeMillis() - start;

}

}

**执行一百万次intern()方法，不同StringTableSize的耗时情况如下：**  
1、-XX:StringTableSize=1009， 平均耗时23000ms；  
2、-XX:StringTableSize=10009， 平均耗时2200ms；  
3、-XX:StringTableSize=100009， 平均耗时200ms；  
4、默认情况下，平均耗时400ms；

**在默认StringTableSize下，执行不同次intern()方法的耗时情况如下：**  
1、一万次，平均耗时5ms；  
2、十万次，平均耗时25ms；  
3、五十万次，平均耗时130ms；  
4、一百万次，平均耗时400ms；  
5、五百万次，平均耗时5000ms；  
6、一千万次，平均耗时15000ms；

从这些测试数据可以看出，尽管在Java 7以上对intern()做了细致的优化，但其耗时仍然很显著，如果无限制的使用intern(）方法，将导致系统性能下降，不过可以将有限值的字符串放入常量池，提高内存利用率，所以intern()方法是一把双刃剑。