Java 基础提升篇：理解 String 及 String.intern()在实际中的应用

String 的深入解析

- 首先 String 不属于 8 种基本数据类型，String 是一个对象。

因为对象的默认值是 null，所以 String 的默认值也是 null；但它又是一种特殊的对象，有其它对象没有的一些特性。

- new String()和 new String(“”)都是申明一个新的空字符串，是空串不是 null；

```
String str=”kvill”;
String str=new String (“kvill”);
```

两者的区别：

在这里，我们不谈堆，也不谈栈，只先简单引入常量池这个简单的概念。

常量池(constant pool)指的是在编译期被确定，并被保存在已编译的.class 文件中的一些数据。它包括了关于类、方法、接口等中的常量，也包括字符串常量。

例 1：

```
String s0=”kvill”;
String s1=”kvill”;
String s2=”kv” + “ill”;
System.out.println( s0==s1 );
System.out.println( s0==s2 );
```

结果为：

```
true
true
```

首先，我们要知道 Java 会确保一个字符串常量只有一个拷贝。

因为例子中的 s0 和 s1 中的”kvill”都是字符串常量，它们在编译期就被确定了，所以 s0==s1 为 true；而”kv”和”ill”也都是字符串常量，当一个字符串由多个字符串常量连接而成时，它自己肯定也是字符串常量，所以 s2 也同样在编译期就被解析为一个字符串常量，所以 s2 也是常量池中”kvill”的一个引用。

所以我们得出 s0==s1==s2;

用 new String() 创建的字符串不是常量，不能在编译期就确定，所以 new String() 创建的字符串不放入常量池中，它们有自己的地址空间。

例 2：

```
String s0=”kvill”;
String s1=new String(”kvill”);
String s2=”kv” + new String(“ill”);
System.out.println( s0==s1 );
System.out.println( s0==s2 );
System.out.println( s1==s2 );
```

结果为：

```
false
false
false
```

例 2 中 s0 还是常量池中”kvill”的应用，s1 因为无法在编译期确定，所以是运行时创建的新对象”kvill”的引用，s2 因为有后半部分 new String(“ill”)所以也无法在编译期确定，所以也是一个新创建对象”kvill”的应用;明白了这些也就知道为何得出此结果了。

String.intern()：

再补充介绍一点 :存在于.class 文件中的常量池 ,在运行期被 JVM 装载 ,并且可以扩充。String 的 intern()方法就是扩充常量池的一个方法；当一个 String 实例 str 调用 intern()方法时，Java 查找常量池中是否有相同 Unicode 的字符串常量，如果有，则返回其的引用，如果没有，则在常量池中增加一个 Unicode 等于 str 的字符串并返回它的引用；看例 3 就清楚了。

```
String s0= “kvill”;
String s1=new String(”kvill”);
String s2=new String(“kvill”);
System.out.println( s0==s1 );
System.out.println( “**********” );
s1.intern();
s2=s2.intern(); //把常量池中“kvill”的引用赋给 s2
System.out.println( s0==s1);
System.out.println( s0==s1.intern() );
System.out.println( s0==s2 );
```

结果为：

```
false
** false //虽然执行了 s1.intern(),但它的返回值没有赋给 s1
true //说明 s1.intern()返回的是常量池中”kvill”的引用
true
```

最后我再破除一个错误的理解：

有人说，“使用 String.intern()方法则可以将一个 String 类的保存到一个全局 String 表中，如果具有相同值的 Unicode 字符串已经在这个表中，那么该方法返回表中已有字符串的地址，如果在表中没有相同值的字符串，则将自己的地址注册到表中 “如果我把他说的这个

全局的 String 表理解为常量池的话，他的最后一句话，“如果在表中没有相同值的字符串，则将自己的地址注册到表中”是错的：

```
String s1=new String("kvill");
String s2=s1.intern();
System.out.println( s1==s1.intern() );
System.out.println( s1+" "+s2 );
System.out.println( s2==s1.intern() );
```

结果：

```
false
kvill kvill
true
```

在这个类中我们没有声名一个”kvill”常量，所以常量池中一开始是没有”kvill”的，当我们调用 s1.intern()后就在常量池中新添加了一个”kvill”常量，原来的不在常量池中的”kvill”仍然存在，也就不是“将自己的地址注册到常量池中“了。

```
s1==s1.intern()为 false 说明原来的“kvill”仍然存在；
```

s2 现在为常量池中“kvill”的地址，所以有 s2==s1.intern()为 true。