58从零开始学Java58之String字符串常量池和intern方法

作者：孙玉昌，昵称【**一一哥**】，另外【**壹壹哥**】也是我哦

千锋教育高级教研员、CSDN博客专家、万粉博主、阿里云专家博主、掘金优质作者

前言

在之前的文章中，**壹哥**给大家介绍了String字符串的不可变性及其实现原理，其中给大家提到了字符串常量池的概念。那么什么是常量池？String字符串与常量池有什么关系？常量池中存储的内容有什么特点？要想搞清楚这些问题，**壹哥**还得再利用一篇文章给大家唠唠字符串常量池及String#intern()方法的作用。

----------**前戏已做完，精彩即开始**----------

全文大约【2300】字，不说废话，只讲可以让你学到技术、明白原理的纯干货！本文带有丰富的案例及配图视频，让你更好地理解和运用文中的技术概念，并可以给你带来具有足够启迪的思考.....

配套开源项目资料

Github：

GitHub – SunLtd/LearnJava

Contribute to SunLtd/LearnJava development by creating an account on GitHub.

GitHub

Gitee：

一一哥/从零开始学Java

从零开始学Java系列 稀土掘金专栏地址：https://juejin.cn/column/7175082165548351546 CSDN专...

Gitee

一. 常量池简介

1. 基本概念

常量池是堆中的一块存储区域，用于存储显式的String、float、Integer等数据。这是一个特殊的共享区域，开发时不需要在内存中经常改变的数据，都可以放在这里进行共享。JDK 7及其之前的常量池是在方法区中，从Java8之后，常量池存放到了堆中。

为了让大家更好地理解常量池的作用，**壹哥**给大家分析一下String字符串的内存分配。

2. 实验案例

我们先来编写一行代码，如下所示：

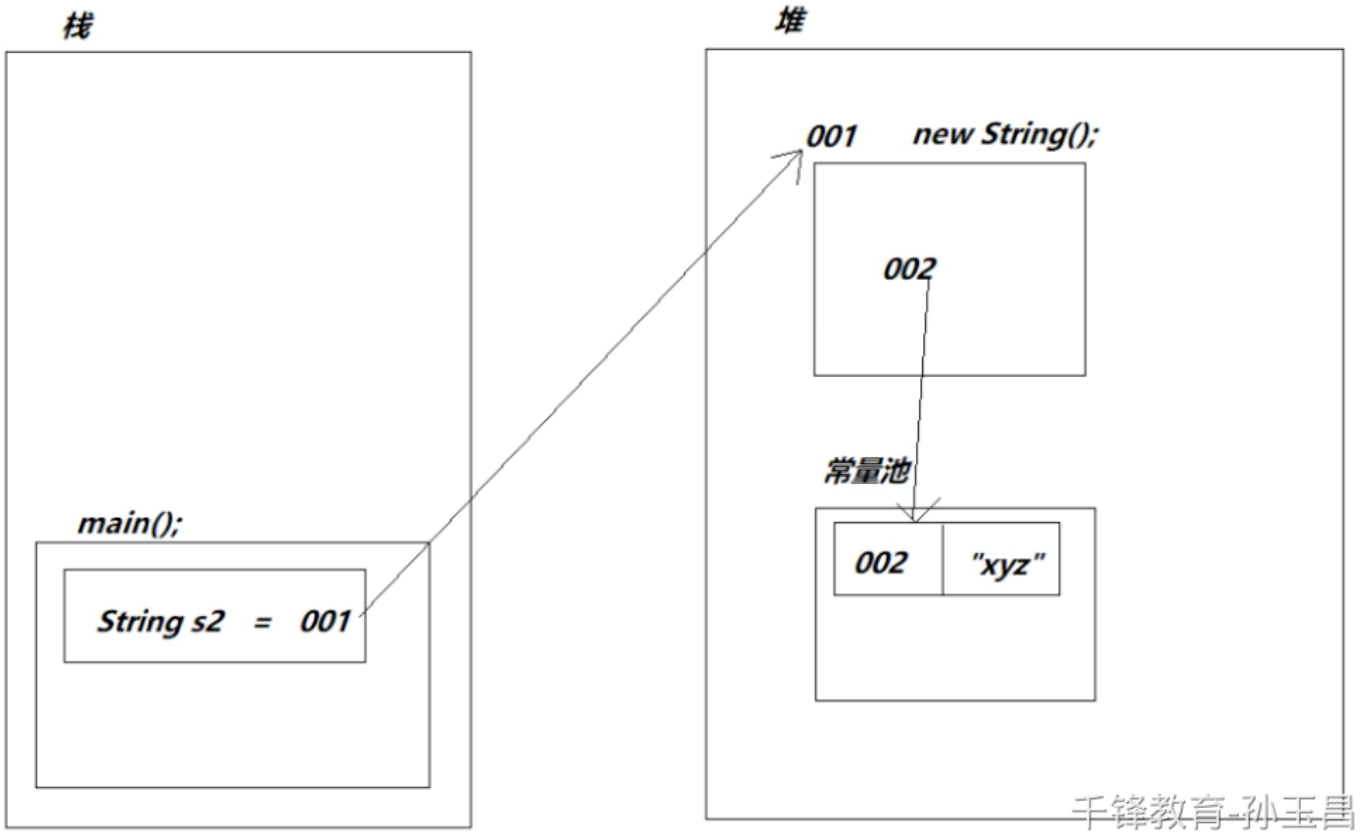
Java | 复制代码

```
1  //String对象创建
2  String s = new String("xyz");
```

这个代码很简单，就一行代码！那么问题来了，这行代码中几个对象的内存分配是如何的？接下来**壹哥**就给大家把这段代码的内存分区绘制一下(本案例开发环境是基于JDK8)。

3. 内存分配(重点)

在 `String s = new String("xyz");` 这行代码中，s是String类型的变量，不是对象！‘xyz’是字符串对象，new String("xyz")也是一个对象，那么它们几个的内存划分在JDK8的环境中，如下图所示：



根据上图，**壹哥**给大家分析一下上述代码的内存分配情况，如下所示：

1. 当JVM在编译阶段加载读取到“xyz”的时候，首先会检查堆中的String常量池，也就是常量缓冲区。检查是否已经有了"xyz"常量对象，如果有，则不会再次创建"xyz"常量对象，并直接返回该字符串的引用地址；如果没有，则创建一个"xyz"常量对象，并为该对象分配一个内存地址002返回；
2. 当JVM在运行阶段加载读取到new关键字的时候，JVM会在堆中为其创建一个对象，即new String()，并为其分配内存地址001，而堆中这个对象的内容是上面"xyz"常量对象的引用地址002，换句话说这个堆中存的就是常量池中"xyz"的引用地址002；
3. 最后，s 是对当前堆中001号对象的一个地址引用，s本身不是一个对象，s只是一个String类型的变量而已！

二. intern()方法(重点)

了解了常量池的内容之后，接下来请大家再跟着**壹哥**来看看String的intern()方法，这个方法很重要，请大家记住哦。

Java | 复制代码

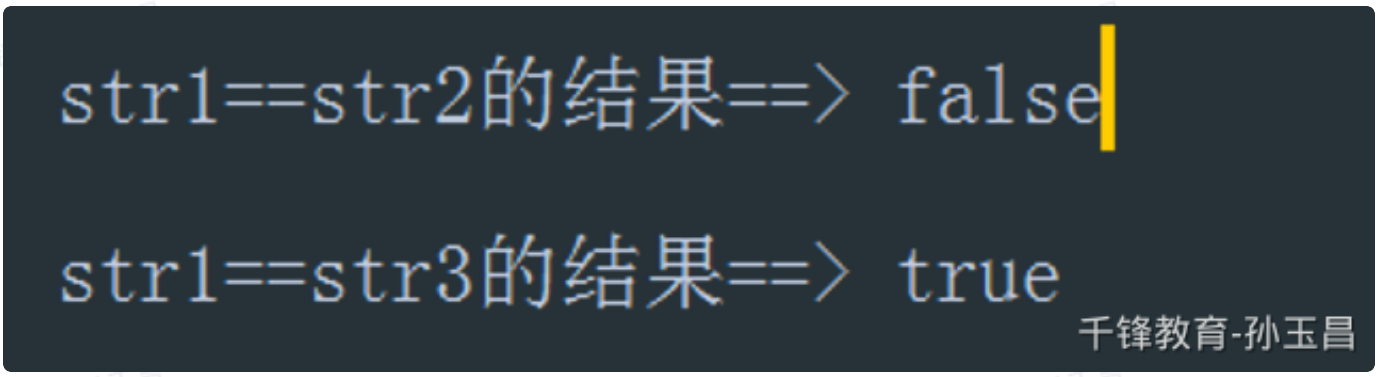
```java
/**
     * Returns a canonical representation for the string object.
     * <p>
     * A pool of strings, initially empty, is maintained privately by the
     * class {@code String}.
     * <p>
     * When the intern method is invoked, if the pool already contains a
     * string equal to this {@code String} object as determined by
     * the {@link #equals(Object)} method, then the string from the pool i
s
     * returned. Otherwise, this {@code String} object is added to the
     * pool and a reference to this {@code String} object is returned.
     * <p>
     * It follows that for any two strings {@code s} and {@code t},
     * {@code s.intern() == t.intern()} is {@code true}
     * if and only if {@code s.equals(t)} is {@code true}.
     * <p>
     * All literal strings and string-valued constant expressions are
     * interned. String literals are defined in section 3.10.5 of the
     * <cite>The Java&trade; Language Specification</cite>.
     *
     * @return  a string that has the same contents as this string, but is
     *          guaranteed to be from a pool of unique strings.
     */
    public native String intern();
```

从上面的源码注释中我们可以知道，intern()是由C语言实现的native底层方法，**用于从String缓存池中获取一个与该字符串内容相同的字符串对象。当这个intern()方法被执行的时候，如果缓存池中已经有这个String内容，则直接从这个缓存池中获取该String内容对象；如果缓存池中没有这个String内容对象，则把这个String内容对象放到缓存池中，并返回这个字符串对象的引用。**现在我们知道了intern()方法的功能，但是该方法的底层原理是什么样的呢？接下来 **壹哥** 给结合一段代码案例，给各位详细说一下：

```java
//常量池与堆的关系
String str1="yiyige";
String str2=new String("yiyige");
System.out.println("str1==str2的结果==> " +(str1==str2));

String str3 = str2.intern();
System.out.println("str1==str3的结果==> " +(str1==str3));
```

执行结果如下图所示：



intern()方法的底层原理如下(重点)：

> **Java专门为String类设计了一个缓存池intern pool，intern pool是在方法区中的一块特殊存储区域，当我们通过 String str="yiyige" 这样的方式来构造一个新的字符串时，String类会优先在缓存池中查找是否已经存在内容相同的String对象。如果有则直接返回该对象的地址引用，如果没有就会构造一个新的String对象，然后放进缓存池，再返回该字符串的地址引用。因此，即使我们构造一万个String str = "yiyige"，但实际上得到的都是同一个地址引用，这样就避免了很多不必要的空间开销。注意：intern池不适用new String("yiyige")的构造形式！**

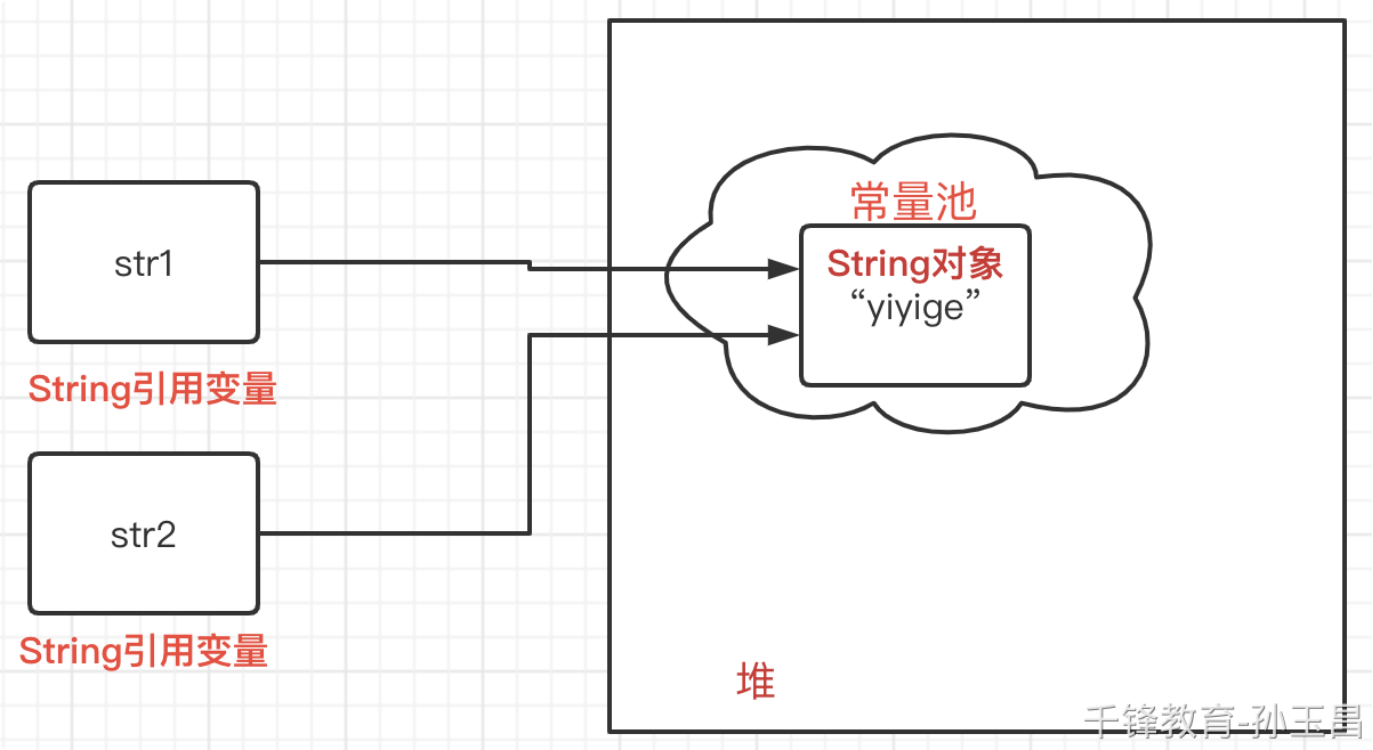
注意：

> 因为字符串常量池存放位置发生了变化，String类对intern()方法也进行了一些修改：
>
> JDK 6 版本中执行intern()方法时，首先会判断字符串常量池中是否存在该字符串字面量，如果不存在则拷贝一份字符串字面量存放到常量池中，最后返回该字符串字面量的唯一引用。如果发现字符串常量池中已经存在，则直接返回该字符串字面量的唯一引用。
>
> JDK 7 以后执行intern()方法时，如果发现字符串常量池中不存在该字符串字面量，则不会再拷贝一份字面量，而是拷贝字面量对应堆中的一个地址引用，然后返回这个引用。

现在我们知道了，原来当一个String对象被创建时，如果发现当前String对象已经存在于String Pool中了，就会返回一个已存在的String对象引用而不会新建一个对象。比如以下代码只会在常量池中创建一个String对象。

```java
String str1 = "yiyige";
String str2 = "yiyige";
```

创建过程如下图所示：



如果一个String是可变的，当改变了A引用指向的String时，可能就会导致其他的B引用得到错误的值，所以Sting就被设计为不可变的。String底层主要是使用intern缓存池将字符串缓存起来，同时允许把一个String字符串的地址赋值给多个String变量来引用，这样就可以保证多个变量安全地共享同一个对象。**如果Java中的String对象可变的话，一个字符串引用的操作改变了对象的值，那么其他的变量就会受到影响。**

另外，关于String源码的解读，及不可变性的相关面试题，**壹哥**在自己的高薪面试题专栏中有过非常详细地讲解，如果你想了解更多，请参考如下链接：

《高薪程序员&面试题精讲系列06之String s=new String(“xyz“)中产生了几个对象？》

三. 结语

至此，**壹哥**就把字符串相关的一些常规原理性知识点，给大家讲解梳理完毕了。当然，关于String原理性的面试题还有很多，受篇幅所限，**壹哥**就不再说这么多了，欢迎大家订阅我的《高薪程序员&面试题精讲系列》。与本文配套的视频链接如下：

https://player.bilibili.com/player.html?bvid=BV1Ja411x7XB&p=124&page=124