对于String的源码剖析

笔记本：	我的第一个笔记本		
创建时间：	2020/8/10 21:35	**更新时间：**	2020/8/10 22:41
作者：	1667567382@qq.com		

对于String的源码剖析

我今天又看到关于字符串常量池和String有关的问题，还夹杂着intern的问题。
根据我的知识，以及String的源码我做出了我的理解。

```
首先
String s1= "abc";
String s2="abc".intern();
String s3=new String("abc");
String s4=new String("abc");
```

还有什么new String("abc")有几个对象这种问题。
两个区域，字符串常量池，堆，还有就是字符串的构造了，对于一个没被加载过的字符串常量，首次遇到肯定会加载到字符串常量池中，而对于new 出来的对象，肯定是在堆上开辟内存空间，创建对象，
我们现在重点看一下创建过程，

```
/**
 * Allocates a new {@code String} so that it represents the sequence of
 * characters currently contained in the character array argument. The
 * contents of the character array are copied; subsequent modification of
 * the character array does not affect the newly created string.
 *
 * @param  value
 *         The initial value of the string
 */
public String( @NotNull() char value[]) { this.value = Arrays.copyOf(value, value.length); }
```

可以看到就是这个构造方法，然后调用了Arrays类的copyOf方法，跟进去看看，

```
@NotNull()
public static <T,U> T[] copyOf( @NotNull() U[] original, int newLength,  @NotNull() Class<? extends T[]> newType) {
    /unchecked/
    T[] copy = ((Object)newType == (Object)Object[].class)
        ? (T[]) new Object[newLength]
        : (T[]) Array.newInstance(newType.getComponentType(), newLength);
    System.arraycopy(original,  srcPos: 0, copy,  destPos: 0,
                     Math.min(original.length, newLength));
    return copy;
}
```

看到它是new了一个Object数组，然后调用arraycopy方法拷贝，可以看出，这就是一个新对象，当然我们还是跟进去看看这个方法。

```
     *
     * @param      src      the source array.
     * @param      srcPos   starting position in the source array.
     * @param      dest     the destination array.
     * @param      destPos  starting position in the destination data.
     * @param      length   the number of array elements to be copied.
     * @exception  IndexOutOfBoundsException  if copying would cause
     *               access of data outside array bounds.
     * @exception  ArrayStoreException  if an element in the <code>src</code>
     *               array could not be stored into the <code>dest</code> array
     *               because of a type mismatch.
     * @exception  NullPointerException if either <code>src</code> or
     *               <code>dest</code> is <code>null</code>.
     */
    public static native void arraycopy( @NotNull() @Flow(...) Object src,  int  srcPos,
                                         @NotNull()
                                         Object dest, int destPos,
                                         int length);
```

这里就是native方法了，当然还是可以理解大概就是对数组赋值的意思。
那么关键可以明白的是这就是一个new出来的一个新数组，当然这里需要提一下String底层是char数组，所以可以确定这个数组跟传入参数的字符串所有的数组完全不是一个。

那么new出来的字符串自然就是一个新对象了。

还有关于intern也解释一下，如果字符串常量池有对象直接返回一个引用就好，如果没有就是堆中有的话，那么堆中的一个引用会被保存到常量池中，然后返回一个引用。