Arrays.*copyOf*(elementData, newCapacity)

****Java 拷贝数组方法 Arrays.copyOf() 是地址传递。****

**Java中System.arraycopy()和Arrays.copyOf()的区别？**

System.arraycopy()的声明：

public static native void arraycopy(Object src,int srcPos, Object dest, int destPos,int length);`

src - 源数组。  
srcPos - 源数组中的起始位置。  
dest - 目标数组。  
destPos - 目标数据中的起始位置。  
length - 要复制的数组元素的数量。

该方法用了native关键字，说明调用的是其他语言写的底层函数。

Arrays.copyOf()的声明：

public static <T,U> T[] copyOf(U[] original, int newLength, Class<? extends T[]> newType) {

@SuppressWarnings("unchecked")

T[] copy = ((Object)newType == (Object)Object[].class)?(T[]) new Object[newLength]:(T[])

Array.newInstance(newType.getComponentType(), newLength);

System.arraycopy(original,0, copy,0, Math.min(original.length, newLength));

return copy;

}

该方法对应不同的数据类型都有各自的重载方法  
original - 要复制的数组  
newLength - 要返回的副本的长度  
newType - 要返回的副本的类型  
仔细观察发现，copyOf()内部调用了System.arraycopy()方法

### 区别在于：

1. arraycopy()需要目标数组，将原数组拷贝到你自己定义的数组里，而且可以选择拷贝的起点和长度以及放入新数组中的位置
2. copyOf()是系统自动在内部新建一个数组，调用arraycopy()将original内容复制到copy中去，并且长度为newLength。返回copy; 即将原数组拷贝到一个长度为newLength的新数组中，并返回该数组。

### 总结

Array.copyOf()可以看作是受限的System.arraycopy(),它主要是用来将原数组全部拷贝到一个新长度的数组，适用于数组扩容。