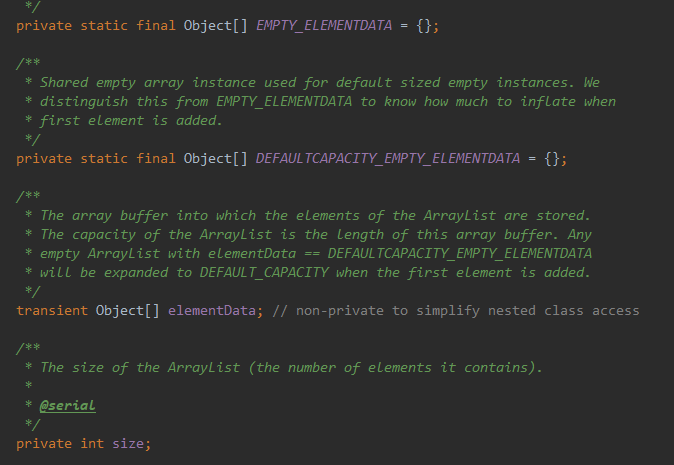
## ArrayList源码解读

### 成员变量



//默认容量是10

private static final int DEFAULT\_CAPACITY = 10;

//当传入ArrayList构造器的容量为0时用这个数组表示：容器的容量为0

private static final Object[] EMPTY\_ELEMENTDATA = {};

//主要作为一个标识位，在扩容时区分：默认大小和容量为0，使用默认容量时采取的是“懒加载”

private static final Object[] DEFAULTCAPACITY\_EMPTY\_ELEMENTDATA = {};

//ArrayList底层使用Object数组保存的元素的

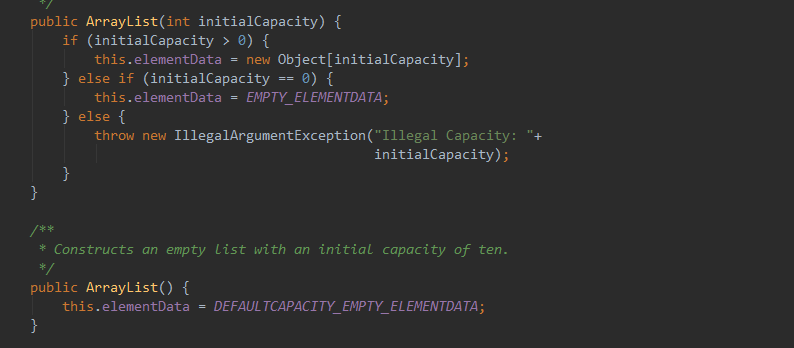
//transient关键词表示这个变量不能被序列化

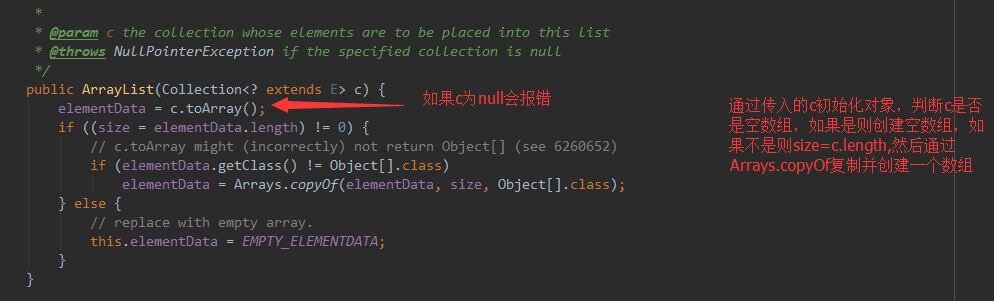
transient Object[] elementData;

//记录当前容器中有多少元素

private int size;

### 构造函数

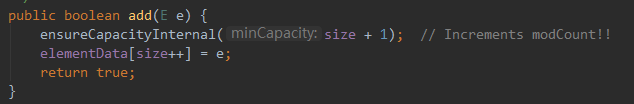




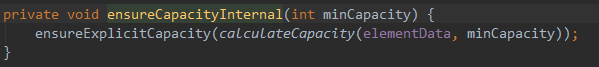
带参数initialCapacity构造函数指定了初始化时创建的内部数组的大小。如果=0则返回默认的空数组，

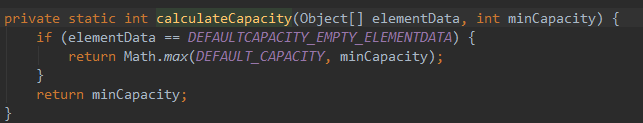
不带参数initialCapacity构造函数默认返回一个空数组

### add方法

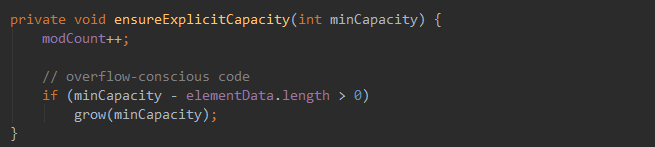


添加元素首先通过方法ensureCapacityInternal进行判断是否需要扩容如果需要则进行扩容，扩容的主要方法是下面的grow

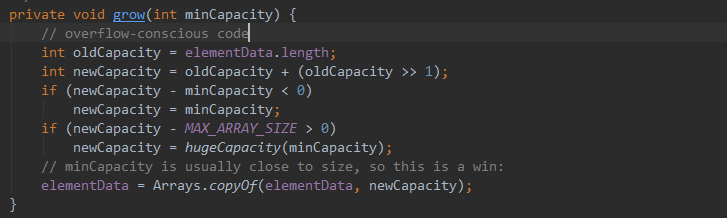




通过判断当前elementData是否是空数组，如果是则扩容大小是DEFAULT\_CAPACITY(默认是10)或者传入的minCapacity



这里会判断是否需要进行扩容,新增时minCcapacity = size + 1 ,通过判断要插入的位置是不是大于数组当前的长度，如果大于了则证明数组不够了需要扩容，如果没有大于则不需要扩容



NewCapacity = oldCapacity + (oldCapacity >> 1) 这个是位运算，主要是扩容1.5倍,例如 oldCapacity = 10 , 则 10 + ( 10 >> 1) = 15，扩容成原来的1.5倍。

与MAX\_ARRAY\_SIZE相比，这里可以看出ArrayList的最大容量是Integer.MAX\_VALUE