1、ArrayList底层有Object数组实现（Object[] elementData）。（泛型：类型擦除）

有两个空Object数组（EMPTY\_ELEMENTDATA和DEFAULT\_EMPTY\_ELEMENTDATA）用来给ArrayList初始化的elementData赋值。

构造函数

**public** ArrayList(**int** initialCapacity) {  
 **if** (initialCapacity > 0) {  
 **this**.**elementData** = **new** Object[initialCapacity];  
 } **else if** (initialCapacity == 0) {  
 **this**.**elementData** = ***EMPTY\_ELEMENTDATA***;  
 } **else** {  
 **throw new** IllegalArgumentException(**"Illegal Capacity: "**+  
 initialCapacity);  
 }  
}

通过Arrays.copyof(原数组，大小，目标数组类型)  
*/\*\*  
 \* Constructs an empty list with an initial capacity of ten.  
 \*/***public** ArrayList() {  
 **this**.**elementData** = ***DEFAULTCAPACITY\_EMPTY\_ELEMENTDATA***;  
}

**public ArrayList(Collection<? extends E> c) {  
 elementData = c.toArray();  
 if ((size = elementData.length) != 0) {  
 // c.toArray might (incorrectly) not return Object[] (see 6260652)  
 if (elementData.getClass() != Object[].class)  
 elementData = Arrays.copyOf(elementData, size, Object[].class);  
 } else {  
 // replace with empty array.  
 this.elementData = EMPTY\_ELEMENTDATA;  
 }  
}**

2、ArrayList扩容：原大小的1.5倍

**int** newCapacity = oldCapacity + (oldCapacity >> 1);

3.add根据grow函数扩容数组（调用**elementData** = Arrays.*copyOf*(**elementData**, newCapacity);）

4.remove函数都借助了System.arrayCopy函数