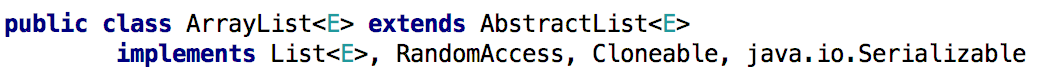
# ArrayList

## 继承关系





AbstractList已经实现了List接口,事实上ArrayList无需实现,这里只是设计错误

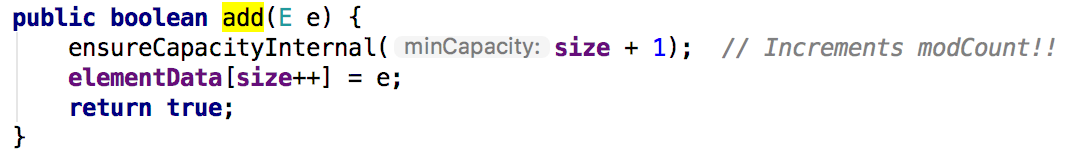
## 构造函数

无参构造函数:会给一个空的数组,添加元素时长度赋值为10

int构造函数:指定长度构造

Collection对象构造:将Collection转成三方数组,并将三方数组地址赋值给底层数组

## add方法

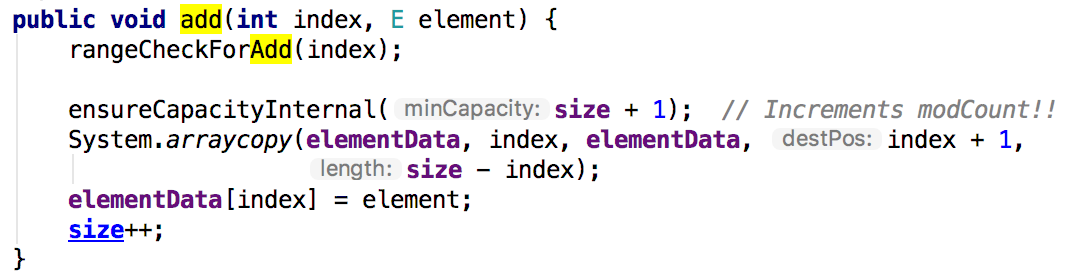


1）确保数组已使用长度（size）加1之后足够存下 下一个数据

2）修改次数modCount 标识自增1，如果当前数组已使用长度（size）加1后的大于当前的数组长度，则调用grow方法，增长数组，grow方法会将当前数组的长度变为原来容量的1.5倍。

3）确保新增的数据有地方存储之后，则将新元素添加到位于size的位置上。

4）返回添加成功布尔值。



1）确保数插入的位置小于等于当前数组长度，并且不小于0，否则抛出异常

2）确保数组已使用长度（size）加1之后足够存下 下一个数据

3）修改次数（modCount）标识自增1，如果当前数组已使用长度（size）加1后的大于当前的数组长度，则调用grow方法，增长数组

4）grow方法会将当前数组的长度变为原来容量的1.5倍。

5）确保有足够的容量之后，使用System.arraycopy 将需要插入的位置（index）后面的元素统统往后移动一位。

6）将新的数据内容存放到数组的指定位置（index）上

## **set方法**

指定index替换元素,方法返回旧元素

## remove**方法**

1）判断索引有没有越界

2）自增修改次数

3）copy不需要删除的元素至临时数组

4）再次回调copy 会覆盖原数组需要删除的元素

5）最后面的一个元素会被置空，等待GC

6）将原来的值oldValue返回

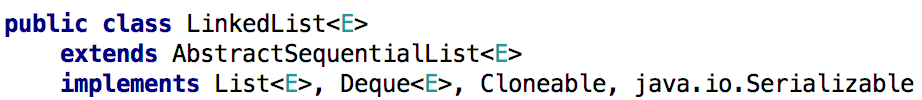
## 总结

1 ) 线程不安全,多线程modCount会有异常

2 ) 删除元素,数组尾部经过处理会置null,等待GC

3 ) 实现了RandomAccess接口,for循环遍历效率高

# LinkedList



## 数据结构

内部是双向链表

