**List接口**ArrayList,LinkedList,Vector

**ArrayList**

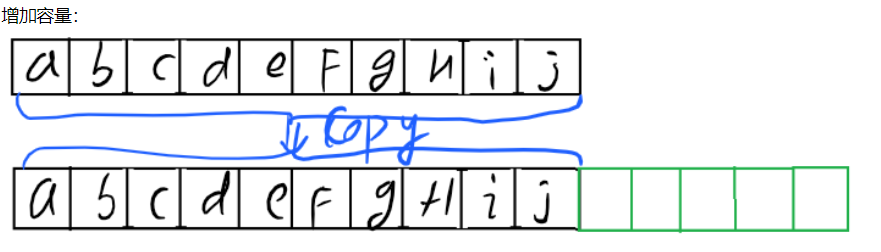
* **ArrayList是基于动态数组实现的，在增删时候，需要数组的拷贝复制**

(navite 方法由C/C++实现)

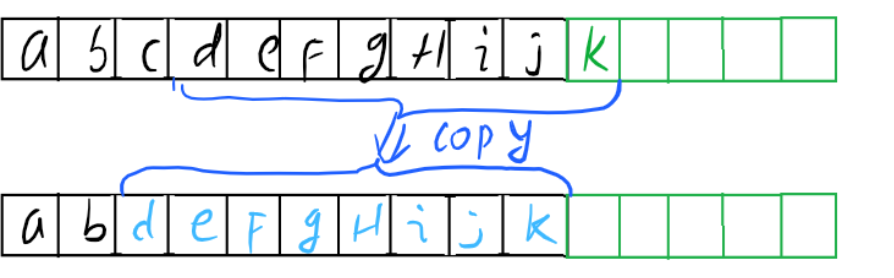
（增指在有数据的地方插入）

* **要同步的ArrayList**

**Collections.synchronizedList(new ArrayList<E>())方法可以将线程不安全的List转成线程安全的List**



**移除index 2的元素，从index 3->最后一元素覆盖源数组的index2到最后一元素**



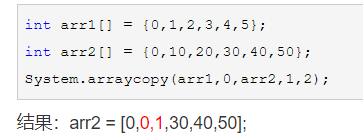
@ src -- 这是源数组

@ srcPos -- 这是源数组中的起始位置

@dest -- 这是目标数组

@ destPos -- 这是目标数据中的起始位置

@ length -- 这是一个要复制的数组元素的数目



**LinkedList(没同步)**

**LinkedList使用双向链表实现存储**(方便实现往前遍历)

**按序号索引数据需要进行前向或后向遍历，但是插入数据时只需要记录本项的前后项即可，所以插入速度较快**

**LinkedList实现了Deque**daiki**接口，可以操作LinkedList像操作队列和栈一样**

**常用方法：**

* **Add()**

**往链表最后插入元素**

* **Remove()**

**从头比较是否有相同的值：若有则把当前节点作为参数传进去删掉**

* **Get()**

**检查下表，若下标小于一半就从头遍历否则从尾遍历**

* **Set()**

**先得到参数index节点，再修改值**