1. ArrayList和Vector的区别?

- ArrayList是List的主要实现类，底层使用Object[ ] 存储，适合频繁查找工作，线程不安全
- Vector是List的古老实现类，底层使用Object[ ]存储，查找、修改速度都比较慢，线程安全

2. ArrayList和LinkedList区别?

- **是否线程安全**：ArrayList和LinkedList都是不同步，线程不安全的
- **底层数据结构**：ArrayList底层是Object[ ]数组；LinkedList底层是双向链表
- **插入和删除是否受元素位置影响**：ArrayList采用数组存储，所以在插入和删除元素的时候时间复杂度受元素位置的影响。LinkedList采用链表存储，所以对于add（E e）方法的插入和删除元素时间复杂度不受元素位置的影响，近似O（1）
- **是否支持快速随机访问**：LinkedList不支持，ArrayList支持。快速随机访问是指通过元素的序号快速获取元素对象（对应get（int index）方法）
- **内存空间占用**：ArrayList的空间浪费主要体现在list列表的结尾会预留一定的容量空间，而LinkedList的空间花费则体现在它每一个元素都需要消耗比ArrayList更多的空间（LinkedList需要存放直接后继和直接前驱以及数据）

> ① ArrayList 执行add(E e)方法的时候， ArrayList 会默认在将指定的元素追加到此列表的末尾，这种情况时间复杂度就是 O(1)。但是如果要在指定位置 i 插入和删除元素的话（add(int index, E element)）时间复杂度就为 O(n-i)。因为在进行上述操作的时候集合中第 i 和第 i 个元素之后的(n-i)个元素都要执行向后位/向前移一位的操作。 ② LinkedList 采用链表存储，所以对于add(E e)方法的插入，删除元素时间复杂度不受元素位置的影响，近似 O(1)，如果是要在指定位置i插入和删除元素的话（(add(int index, E element)）时间复杂度近似为o(n))因为需要先移动到指定位置再插入。

3. ArrayList扩容机制?

以无参构造方法创建ArrayList时，实际上是初始化赋值了一个空数组。只有真正对数组添加元素的时候，才会分配容量，初始容量为10。

ArrayList每次扩容是将容量右移一位扩大为原数组大小的1.5倍，判断扩容后的新容量是否大于最小需要容量，如果是还是小于就将最小需要容量当作数组的新容量。将数组内容复制进新数组。