# 集合树图：



**List**

**特点:**

①有序可以插入元素到指定位置.可以通过下标来访问list里面的元素(类似数组).

②因为有序所以可以通过for循环遍历,迭代器

**LinkedList:**

**特点:**

①实现了List接口,元素可以是null值, 因为有get(), remove(),insert()在链表头尾. 链表可以做成stack, queue.

②没有同步方法,使用多线程时,要手动构造一个同步的List.

③由于基于链表实现,地址随意,在任意点插入,删除元素的时间复杂度都是常数.但是在查询某个元素值时,花费时间复杂度高.

**ArrayList:**

**特点:**

①实现了List接口,元素可以是null值,链表可以做成stack, queue.

②没有同步方法,使用多线程时,要手动构造一个同步的List.

③由于基于数组实现,地址是连续的,在任意点插入,删除元素的时间复杂度都是O(n).但是在查询某个元素值时,花费时间复杂度低(线性)

**总结:访问次数多用ArrayList,增删次数多用LinkedList.**

**Set**

**特点:**

①Set不可以存储相同的元素(equals()方法). 元素在set中的位置其实是由HashCode决定(一般位置也是定死的)

②因为set无序,所以只可以用迭代器.

**TreeSet**

**特点**

是二差树（红黑树的树据结构）实现的,Treeset中的数据是自动排好序的，不允许放入null值