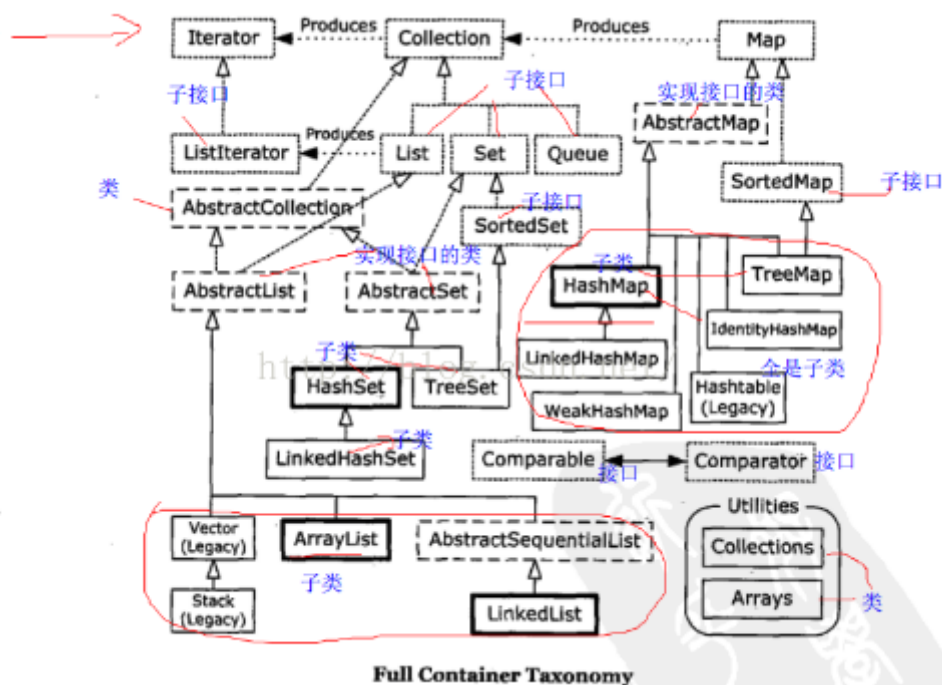


集合：

集合框架：



1.虚线框为抽象类，点框为接口，实现框表示类，空心箭头表示实现了该接口，实心箭头表示可以生成箭头所指向的类的对象

1. LinkedList与ArrayList的区别：

- 1) ArrayList 底层是数组。它可以以 $O(1)$ 时间复杂度对元素进行随机访问。与此对应，LinkedList 是以链表的形式存储它的数据，每一个元素都和它的前一个和后一个元素链接在一起，在这种情况下，查找某个元素的时间复杂度是 $O(n)$ 。
- 2) 相对于 ArrayList，LinkedList的插入，添加，删除操作速度更快，因为当元素被添加到集合任意位置的时候，不需要像数组那样重新计算大小或者是更新索引。
- 3) LinkedList 比 ArrayList 更占内存，因为 LinkedList 为每一个节点存储了两个引用，一个指向前一个元素，一个指向下一个元素。

2.为什么集合类没有实现Cloneable和Serializable接口

在所有的实现中授权克隆和序列化，最终导致更少的灵活性和更多的限制，特定的实现应该决定他是否可以克隆和序列化

3.Comparable和Comparator接口

- 1) Comparator位于包java.util下，而Comparable位于包java.lang下
- 2) Comparable 是在集合内部定义的方法实现的排序,Comparator 是在集合外部实现的排序.Comparable 是一个对象本身就已经支持自比较所需要实现的接口,自定义的类要在加入list容器中后能够排序，可以实现Comparable接口.在用Collections类的sort方法排序时，如果不指定Comparator，那么就以自然顺序排序.而 Comparator 是一个专用的比较器，当这个对象不支持自比较或者自比较函数不能满足你的要求时，你可以写一个比较器来完成两个对象之间大小的比较。用 Comparator 是策略模式（strategy design pattern），就是不改变对象自身，而用一个策略对象（strategy object）来改变它的行为。
- 3) Comparator定义了两个方法，分别是int compare(T o1,T o2)和boolean equals(Object obj).Comparable接口只提供了int compareTo(T o)方法.

4.HashMap和HashTable的区别

- hashtable 是线程安全的， hashMap是线程不安全的
- hashtable不允许null作为key,而hashMap可以
- 在单线程下， hashMap速度快于hashtable
- hashMap用Iterator遍历， HashTable使用Enumeration
- HashMap中hash数组初始容量为16， 扩充翻倍即old*2， 因此必为2的指数。HashTable中hash数组初始容量为11， 扩充容量Old*2+1， 默认的加载因子都是0.75.临界值=容量*加载因子
- hash值使用不同， hashtable直接使用对象的hashCode， Hashtable计算hash是直接使用key的hashCode对table数组的长度直接进行取模。hashMap计算hash值对key的hashCode进行二次hash,以获得更好的散列值， 然后对数组长度取模。

5.HashMap 与TreeMap的区别

- HashMap里面存入的键值对在取出时没有固定的顺序， 是随机的
- TreeMap实现了SortMap接口， 能够把它保存的记录按键排序， 因此， 取出来的是排序后的键值对

6.自定义类作为HashMap的key时需要注意哪些问题

- 如果想根据对象的相关属性来自定义对象是否相等的逻辑，此时就需要重写equals()方法，一旦重写equals()方法，那么就必须重写hashCode()方法
- 当自定义类的多项作为HashMap的key时，最好把这个类设计为不可变类
- 从hashmap的工作原理可以看出，如果两个对象相等，那么这两个对象有着相同的hashCode，反之则不成立。

7.Collection和Collections有什么区别

- Collection是一个集合接口，它提供了对集合对象进行基本操作的通用接口方法，实现该接口的类主要有List和Set。
- Collections是针对集合类的一个包装类，它提供一系列静态方法以实现对各种集合的搜索、排序、线程安全化等操作，Collections类不能实例化，如同一个工具类服务于Collection框架。