今日任务完成 2

**Interface Collection**

方法：boolean add(E e) 添加元素。

boolean addAll(Collection<? extends E> c) 将指定集合中的所有元素添加到此集合。

void clear() 从此集合中删除所有元素。

boolean contains(Object o) 判断集合中是否包含o对象 。

boolean containsAll(Collection<?> c) 判断此集合中是否包含指定集合中的所有元素。

boolean equals(Object o) 将指定的对象与此集合进行比较以获得相等性。

int hashCode() 返回此集合的哈希码值。

boolean isEmpty() 判断集合是否为空 。

Iterator<E> iterator() 返回此集合中的元素的迭代器。

default Stream<E> parallelStream() 返回可能并行的 Stream与此集合作为其来源。

boolean remove(Object o) 从该集合中删除指定元素的单个实例（如果存在）。

boolean removeAll(Collection<?> c) 删除指定集合中包含的所有此集合的元素。

default boolean removeIf(Predicate<? super E> filter) 删除满足给定谓词的此集合的所有元素。

boolean retainAll(Collection<?> c) 仅保留此集合中包含在指定集合中的元素。

int size() 返回此集合中的元素数。

default Spliterator<E> spliterator() 创建一个Spliterator在这个集合中的元素。

default Stream<E> stream() 返回以此集合作为源的顺序 Stream 。

Object[] toArray() 将此集合转换成一个数组。

<T> T[] toArray(T[] a) 返回包含此集合中所有元素的数组; 返回的数组的运行时类型是指定数组的运行时类型。

**Interface List 继承 Interface Collection**

有序，有下标，元素重复

class Arraylist

特点：1.有序，有下标，元素重复；

2.数组结构实现

3.线程不安全（线程异步）；

4.查询快，增删慢；

容量：1、默认容量10个，没有向集合中添加任何元素时,容量是0。

2、加载因子为0.5，扩容为增加原来的0.5倍再+1（容量不够时,每次扩容的后长度是:原长度的1.5倍+1）

3、每次扩容是先创建一个新的长度的数组,再把之前的放进来

**ClassVector**

特点：1、数组结构实现

2、查询快,增删慢

3、线程安全

4、它支持线程的同步，即某一时刻只有一个线程能够写Vector，避免多线程同时写而引起的不一致性，但实现同步需要很高的花费，因此，访问它比访问ArrayList慢。

容量：1、默认容量10个

2、加载因子为1，扩容为增加原来的1倍（容量不够时,每次扩容后的长度是:原长度的2倍）

3、每次扩容是先创建一个新的长度的数组,再把之前的放进来

Class LinkedList

特点：1、双向链表实现

2、多用于存取数据

3、线程不安全

**Class TreeSet 实现 Interface SortedSet 继承 Interface Set**

无序,无下标,元素不可重复

基于排序顺序实现元素不重复

1、底层采用红黑树

2、实现了SortedSet接口,对集合元素自动排序

3、元素对象类型必须实现Comparable接口,指定排序规则

4、通过CompareTo方法确定是否为重复元素

**Class HashSet 实现 Interface Set**

HashMap底层实现(数组+链表+红黑树(Java8以后,如果链表的个数超过8个将采用红黑树，8个以内又会转化为链表))

根据hash凑得计算保存的位置,

如果此位置为空,则直接保存,如果不为空,再执行equals方法,

如果返回ture,则认为是重复,拒绝添加,如果是falst,则形成链表

**Interface Map**

1、用于存储任意键值对

2、键:无序,无下标,不允许重复

3、值:无序,无下标,允许重复

4、键重复将用后面的值覆盖前面的值

方法：

V put(K key,V value) 将对象存入集合中,关联键值,key重复则覆盖原值

V remove(Object key) 删除Key为key值的元素

Object get(Object key) 根据键获取对应的值

void clear() 把map集合中所有的键值删除

Collection<V. values() 返回包含所有值的Collection集合

void putAll(Map<? extends K,? extends V> m) 向map集合中添加指定集合的所有元素

boolean containsKey(Object key) 检出map集合中有没有包含Key为key的元素，如果有则返回true，否则返回false

boolean containsValue(Object value)检出map集合中有没有包含Value为value的元素，如果有则返回true，否则返回false。

Set<Map.Entry<K,V>> entrySet() 返回map到一个Set集合中，以map集合中的Key=Value的形式返回到set中。

boolean equals(Object o) 判断两个Set集合的元素是否相同

V get(Object key) 根据map集合中元素的Key来获取相应元素的Value

int hashCode() 返回map集合的哈希码值

boolean isEmpty() 检出map集合中是否有元素，如果没有则返回true，如果有元素则返回false

Set<K> keySet() 返回map集合中所有Key

int size() 返回map集合中元素个数

Collection<V> values()返回map集合中所有的Value到一个Collection集合

**Class HashMap 实现 Interface Map**

1. 无序,无下标,键一定不能重复,值可以重复

1.HashMap负载因子默认是0.75，可设置，

2.当map填满了75%的bucket时候，将会创建原来HashMap大小两倍的bucket数组，

3.来重新调整map的大小并将原来的对象放入新的bucket数组中,这个过程叫做rehashing，

4.因为它调用hash方法找到新的bucket位置。

**Class TreeMap 实现 nterface SortedMap 继承 Interface Map**

**特点：**

底层实现：桶加上红黑树

TreeMap的底层结构就是一个数组,数组中每一个元素又是一个红黑树.

当添加一个元素(key-value)的时候,根据key的hash值来确定插入到哪一个桶中(确定插入数组中的位置),

当桶中有多个元素时,使用红黑树进行保存;当一个桶中存放的数据过多,那么根据key查找的效率就会降低

容量：

hash数组的默认大小是11,

当hash数组的容量超过初始容量0.75时,增加的方式是old\*2+1