JAVA中集合类的使用

Collection接口

1.是list接口和set接口的父接口

2.通常情况下不被直接使用。

3.定义了一些通用方法。

3.list接口和set接口实现了C接口，所以这些方法对L集合和S结合是通用的。

4。eg;List<String> list = new ArrayList<String>();//实例化一个List。对List使用快捷导入Ctrl+E且导入//的工具包。

list.add("1");

方法摘要	
boolean	**add**(E e) 确保此 collection 包含指定的元素（可选操作）。
boolean	**addAll**(Collection<? extends E> c) 将指定 collection 中的所有元素都添加到此 collection 中（可选操作）。
void	**clear**() 移除此 collection 中的所有元素（可选操作）。
boolean	**contains**(Object o) 如果此 collection 包含指定的元素，则返回 true。
boolean	**containsAll**(Collection<?> c) 如果此 collection 包含指定 collection 中的所有元素，则返回 true。
boolean	**equals**(Object o) 比较此 collection 与指定对象是否相等。
int	**hashCode**() 返回此 collection 的哈希码值。
boolean	**isEmpty**() 如果此 collection 不包含元素，则返回 true。
Iterator<E>	**iterator**() 返回在此 collection 的元素上进行迭代的迭代器。
boolean	**remove**(Object o) 从此 collection 中移除指定元素的单个实例，如果存在的话（可选操作）。
boolean	**removeAll**(Collection<?> c) 移除此 collection 中那些也包含在指定 collection 中的所有元素（可选操作）。
boolean	**retainAll**(Collection<?> c) 仅保留此 collection 中那些也包含在指定 collection 的元素（可选操作）。
int	**size**() 返回此 collection 中的元素数。
Object[]	**toArray**() 返回包含此 collection 中所有元素的数组。
<T> T[]	**toArray**(T[] a) 返回包含此 collection 中所有元素的数组；返回数组的运行时类型与指定数组的运行时类型相同。

当放入数组的空间太小，操作是没有意义的，

List集合

1.List是列表类型，以线性方法存储对象，可以通过对象的索引来操作对象

2.List接口海提供了一些适合与自生

3.常用类为ArrayList和LinkedList通常情况申请为List类型

ArrayList

.采用的是数组结构来保存对象，便于对集合进行快速访问，如果经常需要根据索引位置访问集合中的对象，使用ArrayList类实现的LIst集合的效率较好。缺点：插入和删除对象慢。

LinkList

.采用链表结构，便于集合中插入和删除对象。缺点：随机访问对象速度慢。

Set集合

1.是集类型，集市最简单的一种集合，存放于集中对象不按特定方法排序，知识简单的把对象放入集合中。像在口袋里面放东西，所以不能有重复的一项。

eg:Set<String> s1 = new HashSet<>();//利用HashSet类来实例化Set集合。

Set<String> s2 = new TreeSet<>();//利用TreeSet类来实例化Set集合。

2.不能有重复的元素，不能实现自我添加

HashSet

优点：能够快速定位集合中的元素，要求集合中的对象必须唯一，集合中的对象是无序的。只是不像List集合那样按对象的插入去保存对象

TreeSet

优点：是有序的，从而保证在遍历集合时按照递增的顺序获得对象。因为遍历对象时可能是按照自然顺序递增排列，所以存入由TreeSet类实现的Set集合的对象时必须实现Comparable的接口。

也可能是按照指定比较器对由TreeSet类实现的Set集合中对象进行排序。

Map集合

1.Map集合为映射类型，映射与集合列表由明显的区别，映射中的对象都是成对存在的，都有相应的键对象<key,Value>,就像在字典中查找，键对象必须是唯一的。

eg:Map<Integer,String>m1 = new HashMap<>();//利用HashMap类实例化Map集合；

:Map<Integer,String>m2 = new TreeMap<>();//利用TreeMap类实例化Map集合；

2.主要分为HashMap（遍历集合时无序），TreeMap（遍历集合时有序）

3.clear():删除Map对象中所有的映射。

containkey(Object key):判断Map对象中是否包含指定键的映射；

containsValue（Object value）：判断此Map对象是否包含相应的值；

entrySet():返回Map对象中所包含映射的Set视图；

equals（Objects o）比较指定对象的等价性。

get（Object key）获取指定键的关联的值；

hashCode():返回此Map对象的哈希值

isEmpty()：判断Map对象中是否包含键值映射。

keySet():返回Map对象中包含Set的石头；

put(K key,V value):将指定值与指定键相关联；

putAll(Map<?Ectends k,?extends V>m):将指定Map中所有映射复制到此map对象中；

remove(Object key):从Map对象中删除相应的键和关联的值。

size():返回Map对象中键—值映射的数目；

value（）：返回Map对象中包含值得Collection视图，删除Collection中得元素还将删除Map中相应得映射；