**集合框架与泛型**

**数组长度固定，集合只能存储对象（引用类型），长度可变，可以存储不同类型的对象，两种集合：Collection(单列集合)、Map(双列集合)**

Java.util

**Collection<E>(集合泛型根接口)**

**|--List<E>(接口、可重复、有序)**

**|--ArrayList<E>(数组、查询、效率)**

**|--LinkedList<E>(链表、增删、效率)**

**|--Set<E>(接口、唯一、无序)**

**|--HashSet<E>(唯一)**

**|--TreeSet<E>(排序)**

1. **Collection<E>的功能**

**A:添加功能**

**boolean add(Object obj) 添加一个元素**

boolean addAll(Collection c) 添加一个集合的元素

**B:删除功能**

void clear() 移除所有元素

boolean remove(Object o) 移除一个元素

boolean removeAll(Collection c) 移除一个集合的元素，只要有一个元素被移除了，就返回true

**C:判断功能**

boolean contains(Object o) 判断集合中是否包含指定的元素

boolean containsAll(Colletion c) 判断集合中是否包含指定的集合的所有元素

boolean isEmpty() 判断集合是否为空

**D:获取功能**

**Iterator<E> iterator() 集合用增强for、迭代器遍历**

Collection<String> c=new Collection<String>();

Iterator<String> it=c.iterator();

while(it.hasNext()){

String s=it.next();

}

**E:长度功能**

int size() 元素的个数

**F:集合转数组**

Object[] toArray() 把集合转成数组，可以实现对集合的遍历

1. **List<E>的特有功能**

**A:添加功能**

**void add(int index,Object element) 在指定位置添加元素**

**B:获取功能**

**Object get(int index) 获取指定位置的元素**

**C:删除功能**

**Object remove(int index) 根据索引删除元素，并返回该元素**

**D:修改功能**

**Object set(int index,Object element) 根据索引修改元素，并返回该元素**

1. **LinkedList<E>的特有功能**

**A:添加功能**

**public void addFirst(Object e)**

**public void addLast(Pbject e)**

**B:获取功能**

**public Object getFirst()**

**public Object getLast()**

**C:删除功能**

**public Object removeFirst()**

**public Object removeLast()**

**Map<K,V>(集合泛型根接口)**

**|---HashMap<K,V>(无序、键唯一)**

**|---TreeMap<K,V>(排序)**

1. **Map<K,V>接口的功能**

**A:添加功能**

V put(K key,Vvalue) 添加元素，如果键是第一次存储，就直接存储元素，返回null，如果存在该键，就用值把以前的值替换掉，返回以前的值

**B:删除功能**

void clear() 移除所有的键值对元素

V remove(Object key) 根据键删除键值对元素，并返回值

**C:判断功能**

boolean containsKey(Object key) 判断集合是否包含指定的键

boolean containsValue(Object value)判断集合是否包含指定的值

Boolean isEmpty() 判断集合是否为空

**D:获取功能**

Set<Map.Entry<K,V>> entrySet() 返回的是键值对对象的集合

Map.Entry<K,V>接口：

K getKey() 返回此项对应的键

V getValue() 返回此项对应的值

V get(Object key) 根据键获取值

Set<K> keyset() 获取集合中所有值的集合

Collection<V> values() 获取集合中所有值的集合

**E:长度功能**

int size() 返回集合中的键值对的对数

1. **HashMap<K,V>和HashSet一样依赖hashCode()和equals()方法所以自定义的类如果要唯一一定要重写这两个方法**
2. **TreeMap<K,V>自定义对象要自然排序必须实现Comparable接口，重写compareTo()方法，而且使用的是TreeMap的无参构造。如果要比较器排序必须使用TreeMap的有参构造（参数用匿名内部类对象）new Comarator(){重写compare()}**