

ArrayList和LinkedList的区别：

ArrayList的内存结构是数组，当超出数组大小时需要创建一个新的数组，把原数组的内容拷贝过去。其本质是顺序存储的线性表，插入和删除操作会引发后续元素的移动，附有移动成本的影响，增删效率低，但是随机访问效率高；

LinkedList的内存结构是双向链表存储的，链表存储结构插入和删除效率高，不需要移动，但随机访问效率低，需要从头向后开始依次访问。

Set实现类

**HashSet：**  
基于HashCode计算元素存放位置；  
当存入元素的哈希码相同时，会调用equals进行比较，如果相同则拒绝后者的存入；

**TreeSet：**  
基于排列顺序实现元素不重复；  
实现了SortedSet接口，对集合元素自动排序；  
元素对象的类型必须实现Comparable接口，指定排序规则；  
通过CompareTo方法确定是否为重复元素。

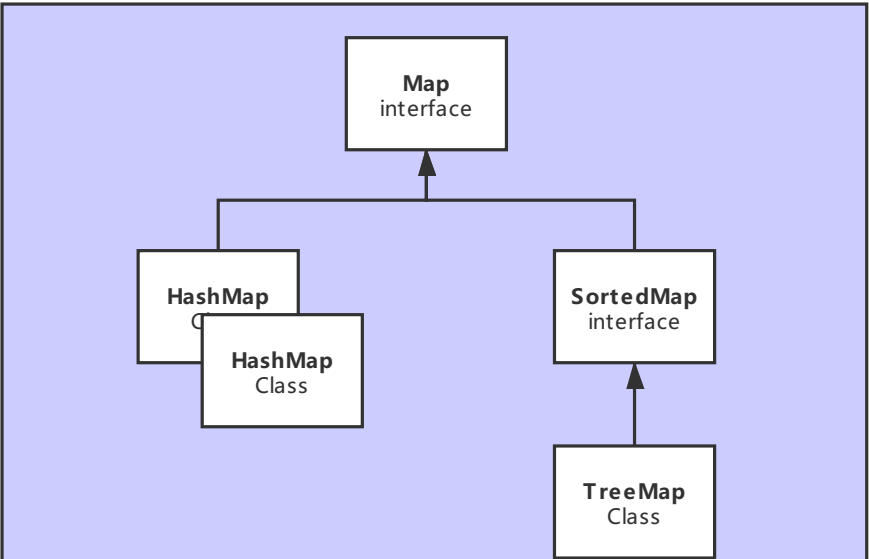
**Collections工具类**

```
public static void reverse (List<?> list) ; //反转集合元素的顺序

public static void shuffle(List<?> list) ; //随机重置集合元素的顺序

public static void sort(List<?> list) ; //升序排序(元素类型必须实现Comparable接口)

...
```



Map接口的特点：

用于存储任意键值对（key-Value）  
键：无序，无下标，不允许重复（唯一）  
值：无序，无下标，允许重复

HashMap：线程不安全的，运行效率高，允许null作为key或者value；

HashTable：线程安全的，运行效率低，不允许null作为key；

Properties：HashTable的子类，要求key和value都是String。通常用于配置文件的读取；

TreeMap：实现SortedMap接口，对key自动排序

HashMap和HashTable的区别？

- 1.HashMap是HashTable的轻量级实现，它们都实现了Map接口，主要区别在于HashMap允许空键值，而HashTable不允许；
- 2.HashMap没有contains方法，而是containsValue和containsKey；
- 3.HashMap不是线程安全的，而HashTable是线程安全的；
- 4.HashMap使用的是Iterator，而HashTable使用的是Enumeration；
- 5.HashMap和HashTable采用的是Hash、ReHash算法；
- ...

