

8. 集合（容器）

CONTENTS

- 集合的引入
- List (ArrayList LinkedList)
- Set (HashSet LinkedHashSet TreeSet)
- Map (HashMap LinkedHashMap TreeMap)
- Iterator
- 使用泛型
- Collections
- 旧集合类和最新线程安全集合类
- 集合的选择



本章技能点列表

技能点名称	难易程度	认知程度	重要程度
集合和数组的联系和区别	易	理解	***
集合框架	易	记忆	***
数据结构知识—线性表	中	理解	***
数据结构知识—栈和队列	中	理解	**
数据结构知识—树和二叉树	中	理解	***
ArrayList使用及其原理	中	应用	***
LinkedList使用及其原理	中	应用	**
HashSet的使用	中	应用	***
哈希表的原理（hashCode和equals）	难	理解	***
TreeSet使用及其原理	中	应用	**
LinkedHashSet	中	应用	**



本章技能点列表

技能点名称	难易程度	认知程度	重要程度
HashMap使用及其原理	中	应用	***
LinkedHashMap	中	应用	**
TreeMap使用及其原理	中	应用	***
Iterator	易	应用	***
泛型的使用	易	了解	*
Collections	易	了解	*
Vector	易	了解	*
Hashtable	易	了解	*
最新线程安全集合类	中	理解	**
集合选择依据	中	理解	***



1. 集合的引入

- 集合的使用场合-Where

就业喜报

更多>>

十年软件开 尚学堂学员就业

更多>>

姓名	
章**	桂林
王**	洁
耿**	
黄**	
何*	广
赵**	吉
刘**	东
袁**	长
林*	吉
李**	吉

星期一(1封)

美国通用电气公司<dwno...

上周(5封)

12306@rails.com.cn<1...

研华(中国)<slcoloeqrifs...

51testing软件测试<zixu...

12306@rails.com.cn<1...

美国国家仪器(NI)<wv...

入职单位: 中加诚信

就业月薪: 8500

毕业院校: 辽宁科技学院

入职时间: 2014-04-15

店铺: 志恋户外

Columbia/哥伦比亚 专柜正品 13秋 冬男式三合一冲锋衣 PM7812

颜色分类: 437宝蓝 尺码: L 180/100A

850.00

店铺: 正品雅戈尔淘宝特

[虫冠价]正品雅戈尔长袖衬衫13年新品 DP免邮(YNA142DP14273-

颜色: 天蓝色 尺码: 42

348.00

店铺: u[1865133596]

2014新款代购正品adidas双肩背包 运动包男女初高中生书包旅行

颜色分类: 蓝银条

569.00 295.00

卖家促销



1. 集合的引入

- 集合的使用场合—Where

- 举例

- 新闻列表
 - 就业喜报
 - 就业明星
 - 邮件列表
 - 购物车

- 当我们需要将一些相同结构的个体整合在一起时，就可以考虑使用集合了。



1. 集合的引入

- 为什么使用集合而不是数组—Why
 - 集合和数组相似点
 - 都可以存储多个对象，对外作为一个整体存在
 - 数组的缺点
 - 长度必须在初始化时指定，且固定不变
 - 数组采用连续存储空间，删除和添加效率低下
 - 数组无法直接保存映射关系
 - 数组缺乏封装，操作繁琐

0	1	2	3	4	5	
aaaa	dddd	cccc	aaaa	eeee	dddd	

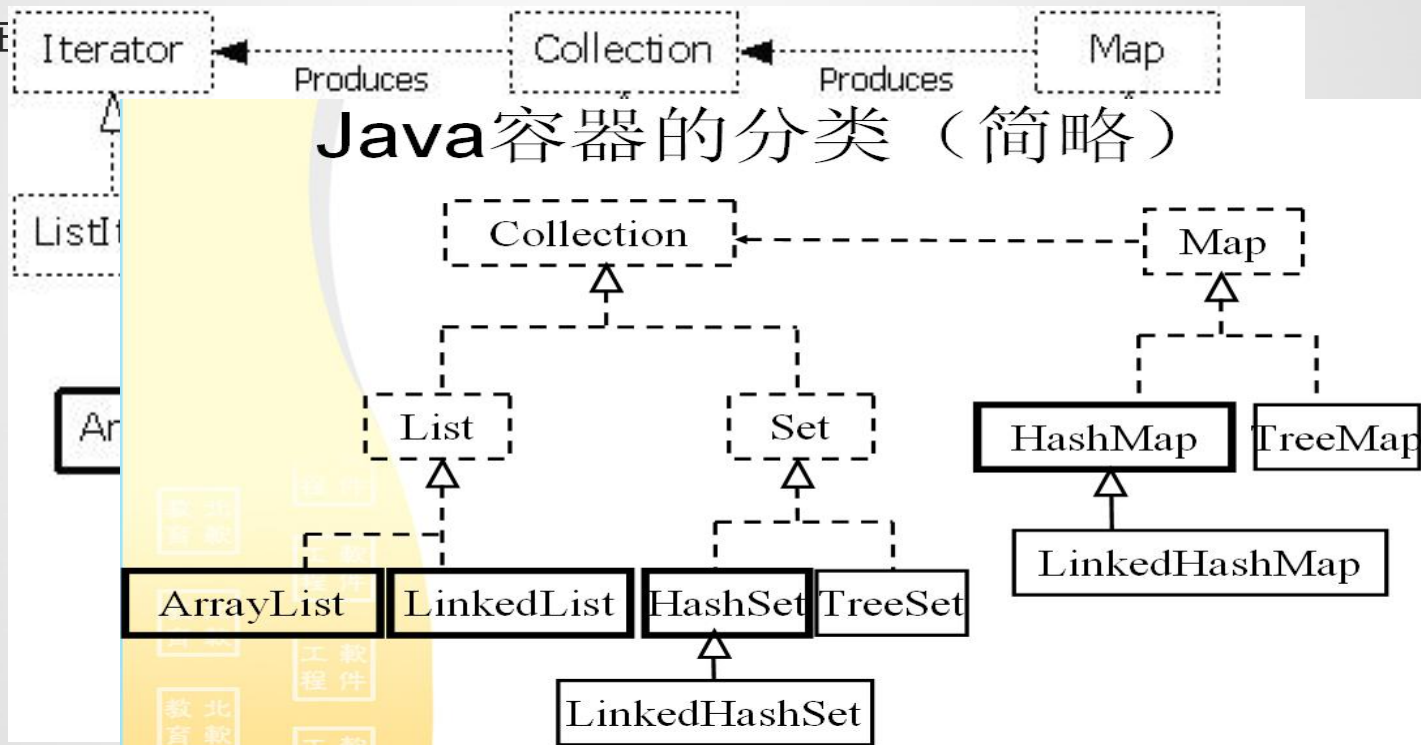


1. 集合的引入

- 集合架构

- Java集合框架提供了一套性能优良、使用方便的接口和类，它们位于java.util包中

- 存放在

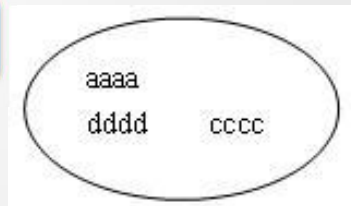
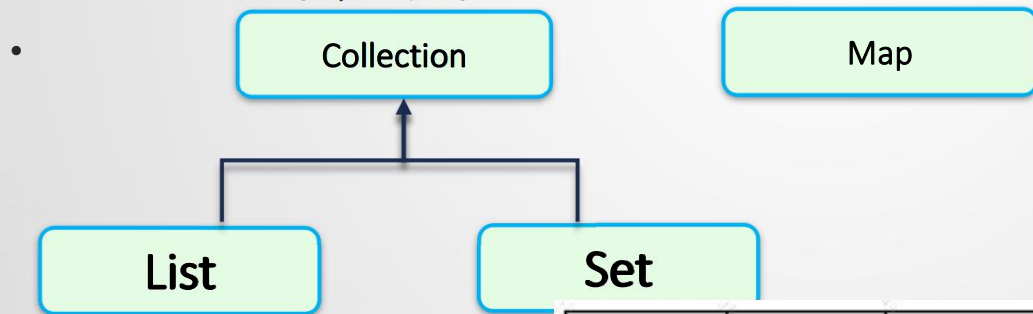
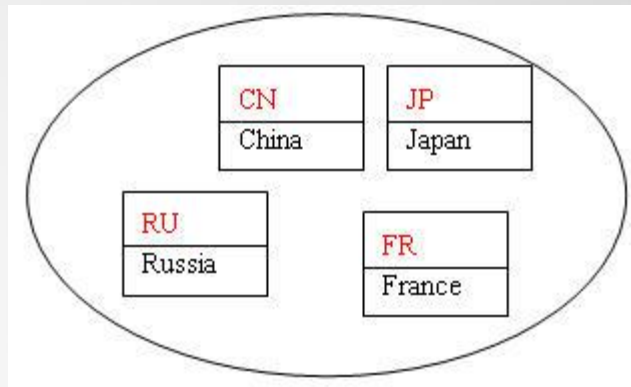




1. 集合的引入

• 集合架构

- Collection 接口存储一组不唯一，无序的对象
- List 接口存储一组不唯一，有序（索引顺序）的对象
- Set 接口存储一组唯一，无序的对象
- Map接口存储一组键值对象，提供key到value的映射
 - Key 唯一 无序
 - value 不唯一 无序



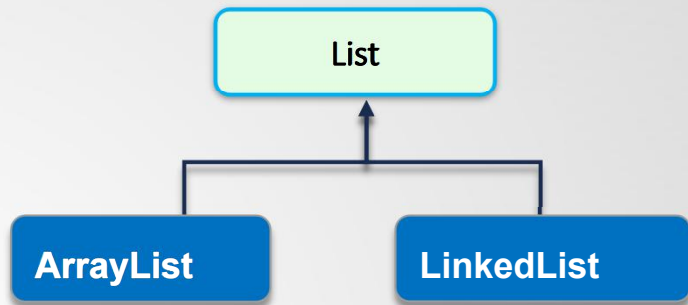
0	1	2	3	4	5	
aaaa	dddd	cccc	aaaa	eeee	dddd	





2. List

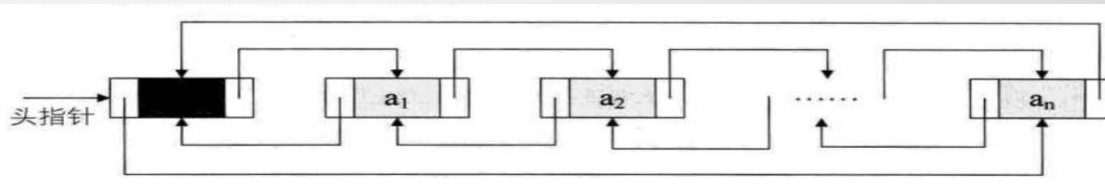
- List
 - 特点：有序 不唯一（可重复）
- ArrayList 线性表中的顺序表
 - 在内存中分配连续的空间，实现了长度可变的数组
 - 优点：遍历元素和随机访问元素的效率比较高
 - 缺点：添加和删除需大量移动元素效率低，按照内容查询效率低，



- LinkedList 线性表中双向链表

0	1	2	3	4	5	
aaaa	dddd	cccc	aaaa	eeee	dddd	

- 采用双向链表存储方式。
- 缺点：遍历和随机访问元素效率低下
- 优点：插入、删除元素效率比较高（但是前提也是必须先低效率查询才可。如果插入删除发生在头尾可以减少查询次数）





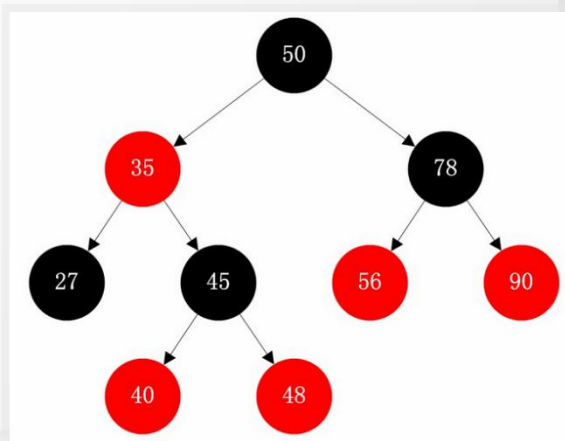
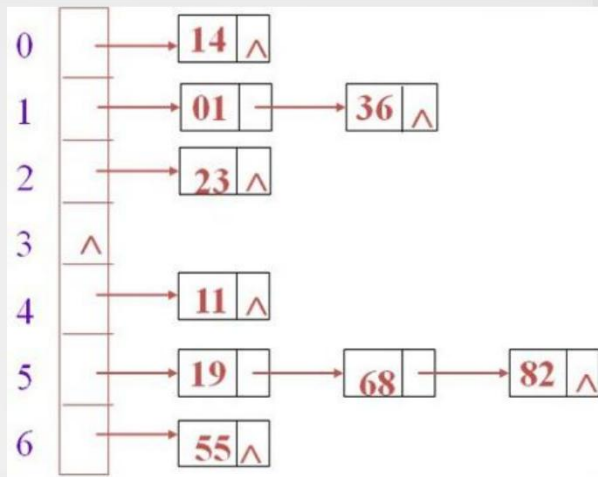
2. List

- List常用方法
 - List相对Collection增加了关于位置操作的方法
- List的遍历方法
 - for
 - for-each
 - Iterator迭代器
- 理解面向接口编程
 - `List list = new ArrayList();`
 - `ArrayList list = new ArrayList();`
- 集合中内容是否相同
 - 通过equals进行内容比较，而是==引用比较



3. Set

- Set
 - 特点：无序 唯一（不重复）
- HashSet
 - 采用Hashtable哈希表存储结构（神奇的结构）
 - 优点：添加速度快 查询速度快 删除速度快
 - 缺点：无序
- LinkedHashSet
 - 采用哈希表存储结构，同时使用链表维护次序
 - 有序（添加顺序）
- TreeSet
 - 采用二叉树（红黑树）的存储结构
 - 优点：有序 查询速度比List快（按照内容查询）
 - 缺点：查询速度没有HashSet快





3. Set

- Set常用方法
 - Set相对Collection没有增加任何方法
- Set的遍历方法
 - for-each
 - Iterator迭代器
 - 无法使用for进行遍历（因为无序，所以没有get(i)）
- HashSet、HashMap或Hashtable中对象唯一性判断
 - 重写其hashCode()和equals()方法
- TreeSet中指明排序依据
 - 实现Comparable接口
 - 创建实现Compator接口的类。



3. Set

- 哈希表存储原理
 - 神奇的表结构，添加和查询数据速度非常快
- 演示添加过程
 - 一次添加成功
 - 数据已经存在
 - 多次添加成功
- 演示查询过程
 - 1次查询获取
 - 多次查询获取
 - 不存在的数据
- hashCode如何计算
 - 整数 实数
 - 字符串 对象



3. Set

23	36	48	77	86	67	76	23	56	78	47
----	----	----	----	----	----	----	----	----	----	----

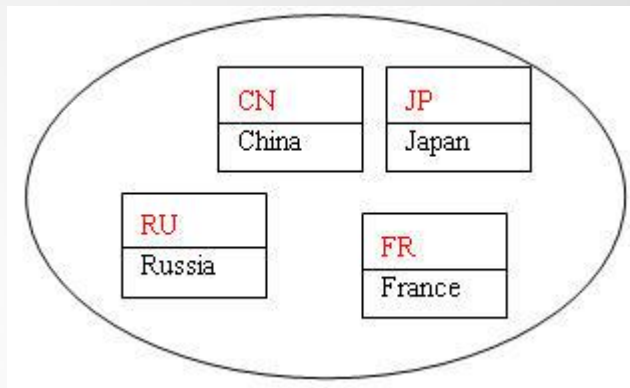
num	23	36	48	77	86	67	76	23	56	78	47
x: 哈希码	23	36	48	77	86	67	76	23	56	78	47
$y=k(x)=x\%11$	1	3	4	0	9	1	10	1	1	1	3

0	1	2	3	4	5	6	7	8	9	10



4. Map

- Map
 - 特点 key-value映射
- HashMap
 - Key无序 唯一 (Set)
 - Value 无序 不唯一 (Collection)
- LinkedHashMap
 - 有序的HashMap 速度快
- TreeMap
 - 有序 速度没有hash快
- 问题：Set和Map有关系吗？
 - 采用了相同的数据结构，只用于map的key存储数据，就是Set





5. Iterator

- 所有集合类均未提供相应的遍历方法，而是把把遍历交给迭代器完成。迭代器为集合而生，专门实现集合遍历
- Iterator是迭代器设计模式的具体实现
- Iterator方法
 - boolean hasNext(): 判断是否存在另一个可访问的元素
 - Object next(): 返回要访问的下一个元素
 - void remove(): 删除上次访问返回的对象。
- 问题：可以使用Iterator遍历的本质是什么
 - 实现Iterable接口



5. Iterator

- For-each循环
 - 增强的for循环，遍历array 或 Collection的时候相当简便
 - 无需获得集合和数组长度，无需使用索引访问元素，无需循环条件
 - 遍历集合时底层调用Iterator完成操作
- For-each缺陷：
 - 数组：
 - 不能方便的访问下标值
 - 不要在for-each中尝试对变量赋值，只是一个临时变量
 - 集合：
 - 与使用Iterator相比，不能方便的删除集合中的内容
- For-each总结：
 - 除了简单遍历并读出其中的内容外，不建议使用增强for



5. Iterator

- ListIterator和Iterator的关系

- public interface ListIterator<E> extends Iterator<E>
- 都可以遍历List

- ListIterator和Iterator的区别

- 使用范围不同
 - Iterator可以应用于更多的集合，Set、List和这些集合的子类型。
 - 而ListIterator只能用于List及其子类型。
- 遍历顺序不同
 - Iterator只能顺序向后遍历; ListIterator还可以逆序向前遍历
 - Iterator可以在遍历的过程中remove();ListIterator可以在遍历的过程中remove()、add()、set()
- ListIterator可以定位当前的索引位置，nextIndex()和previousIndex()可以实现。Iterator没有此功能。



6. 使用泛型

- 起因：
 - JDK1.4以前类型不明确：
 - 装入集合的类型都被当作Object对待，从而失去自己的实际类型。
 - 从集合中取出时往往需要转型，效率低，容易产生错误。
- 解决办法：
 - 泛型，在定义集合的时候同时定义集合中对象的类型
- 好处：
 - 增强程序的可读性和安全性



7. Collections

- 专门用来操作集合的工具类
 - 构造方法私有，禁止创建对象
 - 提供一系列静态方法实现对各种集合的操作
 - 具体操作：搜索、复制、排序、线程安全化等
- 常用方法
 - `Collections.addAll(list, "aaa", "bbb", "ccc", "ccc");`
 - `int index = Collections.binarySearch(list, "ccc");`
 - `Collections.copy(list2, list);`
 - `Collections.fill(list3, "888");`
 - `String max = Collections.max(list4);`
 - `String min = Collections.min(list4);`
 - `Collections.reverse(list4);`
 - `List list5 = Collections.synchronizedList(list4);`



8. 旧的集合类

- Vector
 - 实现原理和ArrayList相同，功能相同，都是长度可变的数组结构，很多情况下可以互用
 - 两者的主要区别如下
 - Vector是早期JDK接口，ArrayList是替代Vector的新接口
 - Vector线程安全，效率低下；ArrayList重速度轻安全，线程非安全
 - 长度需增长时，Vector默认增长一倍，ArrayList增长50%
- Hashtable类
 - 实现原理和HashMap相同，功能相同，底层都是哈希表结构，查询速度快，很多情况下可互用
 - 两者的主要区别如下
 - Hashtable是早期JDK提供的接口，HashMap是新版JDK提供的接口
 - Hashtable继承Dictionary类，HashMap实现Map接口
 - Hashtable线程安全，HashMap线程非安全
 - Hashtable不允许null值，HashMap允许null值



9. 新的线程同步集合类

- 早期集合类Vector、Hashtable：线程安全的
 - 是怎么保证线程安排的，使用synchronized修饰方法
- 为了提高性能，使用ArrayList、HashMap替换，线程不安全，但是性能好。使用ArrayList、HashMap，需要线程安全怎么办呢？
 - 使用Collections.synchronizedList(list);Collections.synchronizedMap(m);解决
 - 底层使用synchronized代码块锁
 - 虽然也是锁住了所有的代码，但是锁在方法里边，并所在方法外边性能可以理解为稍有提高吧。毕竟进方法本身就要分配资源的
- 在大量并发情况下如何提高集合的效率和安全性呢？
 - 提供了新的线程同步集合类，委员java.util.concurrent包下，使用Lock锁
 - ConcurrentHashMap、CopyOnWriteArrayList、CopyOnWriteArraySet：
 - 注意 不是CopyOnWriteHashSet



10. 集合总结

- 集合和数组的比较
- ArrayList和LinkedList 的联系和区别
- Vector和ArrayList的联系和区别
- HashMap和Hashtable的联系和区别
- Collection和Collections的区别
- 各种集合类的特点



10. 集合总结

• 集合和数组的比较

- 数组不是面向对象的，存在明显的缺陷，集合完全弥补了数组的一些缺点，比数组更灵活更实用，可大大提高软件的开发效率而且不同的集合框架类可适用于不同场合。具体如下：
 - 1：数组容量固定且无法动态改变，集合类容量动态改变。
 - 2：数组能存放基本数据类型和引用数据类型的数据，而集合类中只能放引用数据类型的数据。
 - 3：数组无法判断其中实际存有多少元素，length只告诉了array容量；集合可以判断实际存有多少元素，而对总的容量不关心
 - 4：集合有多种数据结构（顺序表、链表、哈希表、树等）、多种特征（是否有序，是否唯一）、不同适用场合（查询快，便于删除、有序），不像数组仅采用顺序表方式
 - 5：集合以类的形式存在，具有封装、继承、多态等类的特性，通过简单的方法和属性调用即可实现各种复杂操作，大大提高软件的开发效率。



10. 集合总结

• ArrayList和LinkedList 的联系和区别

• 联系:

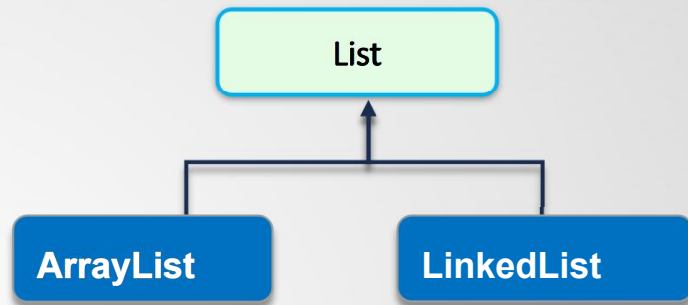
- 都实现了List接口
- 有序 不唯一（可重复）

• ArrayList

- 在内存中分配连续的空间，实现了长度可变的数组
- 优点：遍历元素和随机访问元素的效率比较高
- 缺点：添加和删除需大量移动元素效率低，按照内容查询效率低，

• LinkedList

- 采用链表存储方式。
- 缺点：遍历和随机访问元素效率低下
- 优点：插入、删除元素效率比较高（但是前提也是必须先低效率查询才可。如果插入删除发生在头尾可以减少查询次数）



0	1	2	3	4	5	
aaaa	dddd	cccc	aaaa	eeee	dddd	





10. 集合总结

- Vector和ArrayList的联系和区别
 - 实现原理相同，功能相同，都是长度可变的数组结构，很多情况下可以互用
 - 两者的主要区别如下
 - Vector是早期JDK接口，ArrayList是替代Vector的新接口
 - Vector线程安全，效率低下；ArrayList重速度轻安全，线程非安全
 - 长度需增长时，Vector默认增长一倍，ArrayList增长50%



10. 集合总结

- HashMap和Hashtable的联系和区别
 - 实现原理相同，功能相同，底层都是哈希表结构，查询速度快，在很多情况下可以互用
 - 两者的主要区别如下
 - Hashtable是早期JDK提供的接口，HashMap是新版JDK提供的接口
 - Hashtable继承Dictionary类，HashMap实现Map接口
 - Hashtable线程安全，HashMap线程非安全
 - Hashtable不允许null值，HashMap允许null值



10. 集合总结

- Collection和Collections的区别
 - Collection是Java提供的集合接口，存储一组不唯一，无序的对象。它有两个子接口List和Set。
 - Java中还有一个Collections类，专门用来操作集合类，它提供一系列静态方法实现对各种集合的搜索、排序、线程安全化等操作。



10. 集合总结

	存储结构	顺序	唯一	查询	添加/删除
ArrayList	顺序表	有序（添加）	不唯一	索引查询效率高	效率低
LinkedList	链表	有序（添加）	不唯一	效率低	效率高
HashSet	哈希表	无序	唯一	效率最高	效率最高
HashMap	哈希表	Key无序	key唯一	效率最高	效率最高
LinkedHashSet	哈希表+链表	有序(添加)	唯一	效率最高	效率最高
LinkedHashMap	哈希表+链表	Key有序（添加）	key唯一	效率最高	效率最高
TreeSet	二叉树	有序（自然）	唯一	效率中等	效率中等
TreeMap	二叉树	有序（自然）	key唯一	效率中等	效率中等



10. 集合总结

	Collection		Map	
无序 不唯一	Collection		Map.values()	
有序 不唯一	ArrayList	LinkedList		
无序 唯一	HashSet		HashMap.keySet()	
有序 唯一	LinkedHashSet	TreeSet	LinkedHashMap.keySet()	TreeMap.keySet()