

Java容器2

- What?Why?How?



本章概述

- 容器的概念
- 容器 API
- Collection 接口
- Iterator 接口
- Set 接口
- Comparable 接口
- List 接口
- Map 接口



上机练习

- 创建Cat类, 包含如下属性, 昵称(name), 年龄(age)
- 创建测试类, 使用HashSet泛型集合存储Cat对象,
- 要求: 一、不能添加重复的对象, 提示 (Cat类重写equals, 和hashCode方法)
- 二、实现comparable接口定义内部比较器按昵称比较
- 三、实现comparator接口定义外部比较器按昵称的长度比较, 长度相同, 再按照昵称的内容比较

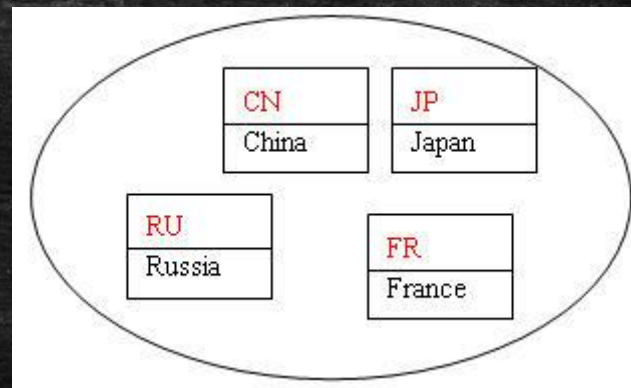
```
hs.add(new Cat("美美", 3));  
hs.add(new Cat("丽丽", 2));  
hs.add(new Cat("亚亚", 4));  
hs.add(new Cat("美美", 3));
```

昵称	年龄
亚亚	4
丽丽	2
美美	3

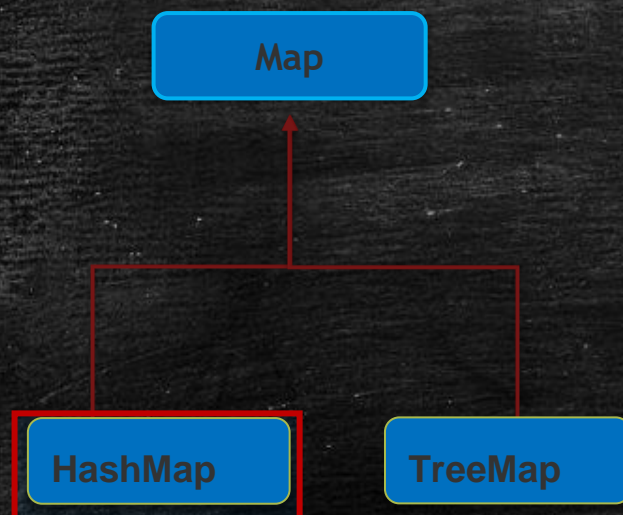


Map

- 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与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, 以上是Set



Map接口



序号	方法	作用
1(添加)	put(key,value)	添加元素
2(删除)	clear() remove(key)	清除所有 根据key去移除
3(判断)	containsKey(key) containsValue(value) isEmpty()	是否包含指定的key 是否包含指定的值 判断集合中元素是否为空
4(遍历)	get(key) size() values() entrySet() keySet()	



Collections工具类

Collections和Collection不同，前者是集合的操作类，后者是集合接口

Collections提供的静态方法

addAll():批量添加

sort():排序

binarySearch(): 二分查找

fill(): 替换

shuffle():随机排序

reverse():逆序



上机练习

- 1、使用List, map容器存放如下数据, 并从map中取出“李四”
姓名: 张三 年龄: 18 体重: 90 地址: 北京
姓名: 李四 年龄: 28 体重: 50 地址: 上海
- 2、假如有以下email数据
“aa@sohu.com, bb@163.com, cc@sina.com, ...” 现需要把
email中的用户部分和邮件地址部分分离, 分离后以键值对应的方式放入HashMap?



上机练习

- 3、定义一个Worker类, 属性:
- name:String, age:int, salary:double
- a). 把若干Worker对象放在List中, 排序并遍历输出, 按照age升序排列
- b). 把若干Worker对象放在Set中并遍历, 要求没有重复元素
- c). 把若干Worker对象放在Map中并按照三种方式分别遍历, 要求以Worker的姓名作为key。



集合总结

名称	存储结构	顺序	唯一性	查询效率	添加/删除效率
ArrayList	顺序表	有序(添加)	不唯一	索引查询效率高	低
LinkedList	链表	有序(添加)	不唯一	低	最高
HashSet	哈希表	无序	唯一	最高	最高
HashMap	哈希表	Key无序	Key唯一	最高	最高
LinkedHashSet	哈+链	有序(添加)	唯一	最高	最高
LinkedHashMap	哈+链	Key有序(添加)	Key唯一	最高	最高
TreeSet	二叉树	有序(升序)	唯一	中等	中等
TreeMap	二叉树	有序(升序)	Key唯一	中等	中等



集合总结

特性	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



面试题

- 集合与数组的比较
- Collection和Collections的区别
- 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



集合与数组的比较

▪ 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不同场合。具体如下：
- 1：数组能存放基本数据类型和对象，而集合类中只能存放对象。
- 2：数组容易固定无法动态改变，集合类容量动态改变。
- 3：数组无法判断其中实际存有多少元素，length只告诉了数组的容量，而集合的size()可以确切知道元素的个数
- 4：集合有多种实现方式和不同适用场合，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即可实现各种复杂操作，大大提高了软件的开发效率



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还有一个Collections类，专门用来操作集合类，它提供了一系列的静态方法实现对各种集合的搜索、排序、线程安全化等操作。

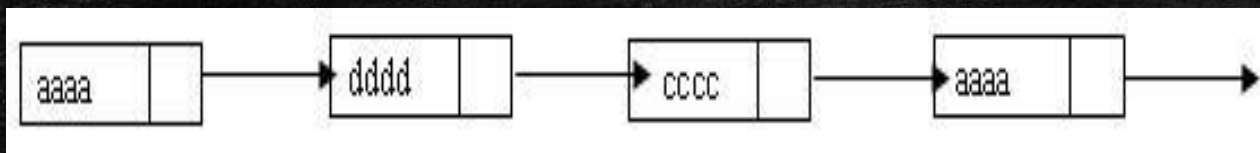


ArrayList和LinkedList的联系和区别

- ArrayList实现了长度可变的数组，在内存中分配连续空间。遍历元素和随机访问元素效率比较高。

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

- LinkedList采用链表存储方式。插入、删除元素效率比较高



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，很多时候可以互用
- 两者的主要区别如下
 - Vector是早期的JDK接口，ArrayList是替代Vector的新接口
 - Vector线程安全，ArrayList重速度轻安全，线程非安全
 - 长度需要增长时，Vector默认增长一倍，ArrayList增长50% (1.5+1)



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是早期的JDK提供的接口，HashMap是新版的JDK提供的接口
 - Hashtable继承Dictionary类，HashMap实现Map接口
 - Hashtable是线程安全，HashMap线程非安全
 - Hashtable不允许null值，HashMap允许null值

