

Java Day 12

`TreeSet`: 有序, 不可重复。可以指定一个顺序, 对象存入之后会按照指定的顺序排列

自然顺序(*Comparable*)

`TreeSet` 类的 `add()` 方法中会把存入的对象提升为 `Comparable` 类型

调用对象的 `compareTo()` 方法和集合中的对象比较

根据 `compareTo()` 方法返回的结果进行存储

`TreeSet` 中存放自定义类对象-->`TreeSetTest2.java`

比较器排序(`Comparator`)-->`TreeSetTest3.java`

创建 `TreeSet` 的时候可以指定一个 `Comparator`

如果传入了 `Comparator` 的子类对象, 那么 `TreeSet` 就会按照

比较器中的顺序排序

`add()` 方法内部会自动调用 `Comparator` 接口中 `compare()` 方法排序

调用的对象是 `compare` 方法的第一个参数, 集合中的对象是

compare 方法的第二个参数

`public int compare(String o1, String o2)` : 比较其两个参数的顺序。

两个对象比较的结果有三种：大于，等于，小于。

如果要按照升序排序， 则 o1 小于 o2，返回（负数），相等返回 0，
o1 大于 o2 返回（正数） 如果要按照降序排序 则 o1 小于 o2，返回（正数），相等返回 0， o1 大于 o2 返回（负数）

两种方式的区别

TreeSet 构造函数什么都不传，默认按照类中 Comparable 的顺序(没有就报错 ClassCastException); TreeSet 如果传入 Comparator，就优先按照 Comparator

Map

特点

将键映射到值的对象

一个映射不能包含重复的键

每个键最多只能映射到一个值

Map 和 Collection 的不同

Map 是双列的,Collection 是单列的

Map 的键唯一,Collection 的子体系 Set 是唯一的

Map 的数据结构只针对键有效,跟值无关;Collection 的数据结构是针对元素有效

常用方法

`public V put(K key, V value)` : 把指定的键与指定的值添加到 Map 集合中。

`public V remove(Object key)` : 把指定的键所对应的键值对元素从 Map 集合中删除, 返回被删除元素的值。

`public V get(Object key)` 根据指定的键, 在 Map 集合中获取对应的值。

`public Set<K> keySet()` : 获取 Map 集合中所有的键, 存储到 Set 集合中。

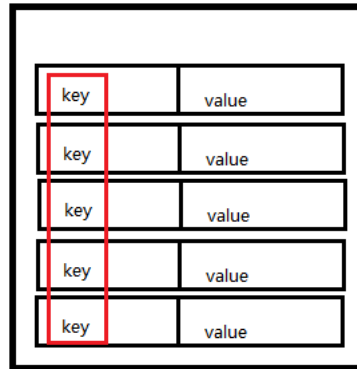
`public Set<Map.Entry<K,V>> entrySet()` : 获取到 Map 集合中所有的键值对对象的集合(Set 集合)。

HashMap

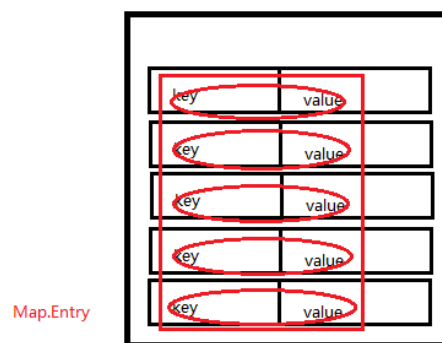
存储数据采用的哈希表结构, 元素的存取顺序不能保证一致。由于要保证键的唯一、不重复, 需要重写键的 `hashCode()` 方法、`equals()`

方法。

集合遍历 1: `keySet()`



集合遍历 2: `entrySet()`



LinkedHashMap

HashMap 下有个子类 LinkedHashMap, 存储数据采用哈希表结构+链表结构。通过链表结构可以保证元素的存取顺序一致; 通过哈希表结构可以保证的键的唯一、不重复, 需要重写键的 `hashCode()` 方法、`equals()` 方法。

HashMap 和 Hashtable 的区别

Hashtable 是 JDK1.0 版本出现的, 是线程安全的, 效率低; HashMap 是 JDK1.2 版本出现的, 是线程不安全的, 效率高

Hashtable 不可以存储 null 键和 null 值;HashMap 可以存储 null 键和 null 值

TreeMap

可以根据键进行排序，排序规则的指定同 TreeSet 相同

java.util.Properties

通过 load() 操作，从文件中加载键值对

控制台传参：

1) 应用程序参数: java com.briup.Test 1 2

通过 main 方法中的 args 获取

2) 虚拟机参数 : java -Dname=briup -Dage=20
com.briup.Test

通过 System.getProperties();
getProperty("key");

java.util.Stack

集合辅助类 *java.util.Collections*

Collection 和 Collections 区别？

java.util.Collections 是集合工具类，用来对集合进行操作。

部分方法如下：

public static <T> boolean addAll(Collection<T> c,
T... elements) :往集合中添加一些元素。

public static void shuffle(List<?> list) :打乱集合
顺序。

public static <T> void sort(List<T> list) :将集合中
元素按默认规则排序。

public static <T> void sort(List<T> list, Comparator<?
super T>) :将集合中元素按指定规则排序。

增强 for 循环：简化数组和集合的遍历

```
for(元素的数据类型 变量 : Collection 集合 or 数组){  
  
    //操作代码  
}
```