b+树，分布式的cap和base理论。如何保证异常的事务回滚，幂等机制

命名法:

驼峰命名法,混合大小字母来构成变量和函数的名称.(分为大驼峰式命名法和小驼峰式命名法)

匈牙利命名法:(windows常用)标识符的名字以一个或者多个小写字母开头作为前缀,前缀之后是一个首字母大写的一个单词或者多个单词组合,该单词要指明变量的用处,

规则:属性+类型+描述

属性:一般是小写字母+\_

g\_:全局变量

m\_:类成员变量

s\_:静态变量

c\_:常量

类型:

b:bool

sz:以零结束的字符串

p:指针

n:整整

dw:双字

l:长整型

无符号:u

函数:fn

匈牙利命名法分为系统和应用两种,系统着眼于类型,应用着眼于语义.

帕斯卡命名法:类似于驼峰命名法,只不过首字母是大写的

下划线命名法:每个逻辑断点都会以下划线分开

hashmap和hashtable的区别

1.map比table作者上多了并发大神doug lea,

2.table产生早于map,线程安全的,效率低,.不符合驼峰命名规则

3.父类不同hashmap继承AbstractMap hashtable继承dictionary类

都实现了map,Cloneable(可复刻),serializable(可序列化)三个接口

4.接口table多了Elements和contains方法,返回table 中value的枚举,contains方法是否包含value,containsvalue只是调用了一下contains方法.

5.对空key和空value的支持,

table不支持空key和空value(key为对象调方法,value为空时,抛出异常)

map中,null可以作为key,可以有一个或者多个键对应的值为null,get不能作为判断有没有key的依据,应该用containsKey判断.

6.线程安全性不同日常使用大部分是单线程,,需要多线程操作时,使用线程安全的concurrenthashmap.是线程安全的,并且效率比table高好多倍,concurrent使用分段锁,不对整个数据进行锁定.

7.遍历的内部实现上不同

都是用了iterator,table还使用了enumeration的.

map是fail-fast迭代器,

fast-fail机制:

8.初始容量和每次扩充容量不同

table的初始容量时11,每次扩容,容量变为2N+1

如果给定了初始容量值,table会直接使用,而map会扩充为2 的幂次方大小,也就是,table会直接使用素数,奇数,hashmap作为hash表的大小.table关注于hash的结果更为简单,map则关注于为运算的结果更为简单,导致hash分布不均匀,,计算hash值得方式不同

hashset和treeset的区别

1.数据结构 hashmap表结构存储 Treemap树存储结构

2.set复写hashcode和equals保证唯一 Compareable接口的compareTo保证

3.set无序 treeSet有序

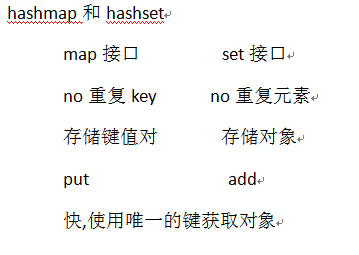
4.存储原理

treeset数据结构是二叉树,进行排序,提高排序性能,,返回0表示元素重复.

hashmap和treemap

treemap:基于红黑二叉树的navigableMap实现,线程非安全,不允许null,存入TreeMap的元素应当实现,Comparable接口,Comparator接口,会按照排序后的顺序迭代元素,两个相比较的key不得抛出classcast异常,主要用于存储元素的时候对元素进行自动排序,得带输出的时候按排序顺序输出.

hashmap和hashset



---------------------------------------------------------------------------------------