java集合

1.java集合框架

java最初的的集合类:vector,stack,hashtable,Array

在线程安全的情况下,使用泛型和并发集合类,阻塞接口及实现.

集合框架的优点:使用核心集合类,降低开发成本,

集合框架,经过严格的测试,能提高代码质量

降低代码维护成本,

复用性和可操作性

2.泛型 优点:

出现类型的问题,会在编译期报错,避免了classcastException,不需要显示转换和instanceof,不会产生类型检查的字节码指令.

3.基础接口

list更像是长度动态变化的数组.其他的接口Queue,DeQueue,SortedSet,sortedMap,ListIterator

迭代器允许在迭代过程中移除元素

Enumeration速度是iterator的两倍,也是用了更少的内存,iterator 的优点在于安全,在遍历过程中,他会阻止其他线程修改集合,不能保证迭代的顺序,所以在迭代过程中不能添加元素.

ListIterator和Iterator

Iterator遍历set和list,只能向前遍历

ListIterator遍历list,增加额外功能,添加,替换,获取前面元素和后面元素的索引.

迭代器fail-fast 属性,每次尝试获取下一个元素时,fail-fast属性检查,有任何改动,就会跑出concurrentModificationException,connection中的所有Iterator的实现都是按fail-fast设计的,并发类集合除外.

fail-fast ,fail-fast(并发集合包的设计)从不会跑出那个异常.

使用并发类结合遍历集合避免跑出那个异常.

Iterator接口没有具体的实现,,他的实现是集合实现类的责任,允许集合去选择遍历的迭代器是fail-fast还是fail-safe

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

hashmap和hashtable,map允许NULL;table同步,线程安全;hashmap可以实现遍历顺序(子类,linkedhashmap),,map是fail-fast的,在迭代时要修改map,使用concurrentHashMap.

TreeMap会对有序的key集合进行遍历.

ArrayList和Vector

两者基于索引,内部有数组支持,

维护插入的顺序,获取元素有序,

迭代器都是Fail-fast,

两者都允许NULL值,使用索引值对元素进行随机访问.

不同:

vector同步,如果在迭代过程中,想要修改使用copyonWriteArrayList,

ArrayList比vector快,

数组和list

数组放置基本类型和对象,List只能容纳对象.

数组指定大小,list大小固定,

使用Array的情况,列表的大小指定,,基本数据类型,集合 的自动装箱减轻编码工作,多维数组更容易.

ArrayList和LinkedList

都实现了list,Array数组支持的基于索引的数据结 构,linkedlist从节点开始遍历,遍历到索引的节点,然后返回 元素.

linkedList在插入元素时,不会改变数组的大小,或者更新 索引.

linkedlist消耗更多的内存,需要存储前后节点的引用.

随机访问:

ArrayList,HashMap,TreeMap,HashTable都有随机访问的 方法.

线程安全的:

vector,hashtable,Properties,Stack

并发集合类:

copyOnWriteArrayList,conCurrentHashMap,CopyOnWriteArraySet

队列和栈

队列,实现类在java的并发包中,队列允许先出检索元素,

Deque,允许从两端检索元素,栈先进后出,stack是一个扩展 子vector的类,queue是一个接口.

collections折半搜索,排序,逆序.

comparable和comparator接口,被用来对对象集合 或者数组进行排序,able提供自然的排序,tor提供不同的排 序算法

函数不能修改集合,使用 collections.unmodifiableCollection(c)来定义集合,创建一个 只读集合,任何改变集合的操作都会抛出不支持操作异常.