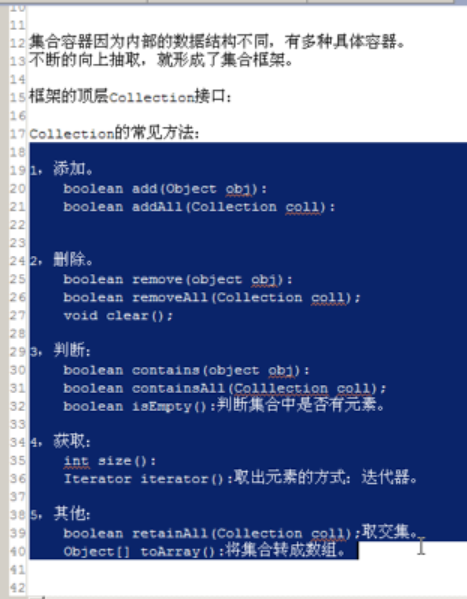
# Collection



集合中的System.***out***.println(l);输出某个集合时，若基本类型输出看到的是真值，若是复杂类型若是内部实现了tostring（）方法，则也可以看出值，否则输出的就是hashcode指

Hashcode

默认的hashCode()返回的值就是对象在内存中的地址。

Tostring

而默认的toString()方法就是打印出对象的地址。

容器之所以能区分就是因为内部的数据结构不一样

1：retainAll()求交集

2：removeAll求差集

3：iterator it=;

While(it.hasnext()){

}

it在while之后还能用（在内存）课改成for循环

4：For（iterator it=;it.hasnext()；）{

It.next（）

}

5：迭代器原理：在容器内部实现的内部类，

## 一：list：



### 特点

有序。怎么存进去怎么取出来（存入取出顺序一致比如存进去时123，取出来的也是123）

元素都有索引有角标

元素可以重复

### 1：常见方法：

特点：都可以操作角标index

添加：add

删除：remove

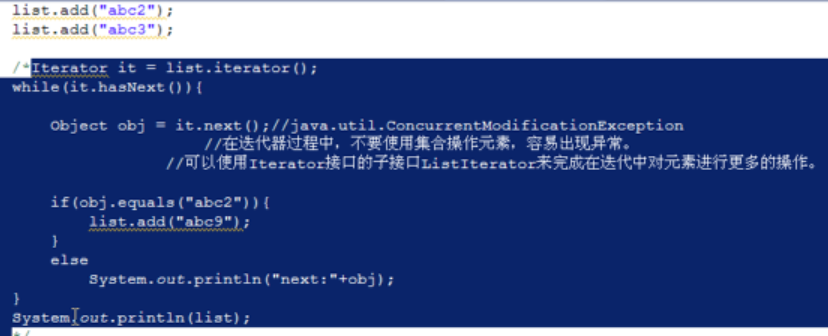
修改：set

获取：get

### 2：取全部元素：

iterator,get()循环（特有）

### 3：list的迭代器

:

迭代器操作元素时，集合操作元素（add操作）。（ 因为集合操作时，迭代器不知道。你加了元素迭代器是显示还是不显示）两个同时对元素修改异常（快速失败原则）

解决：迭代过程中不集合操作。别两个同时操作。

使用iterator,接口的子接口ListIterator（只有list才有）来完成在迭代过程中对元素进行跟多的操作（增删改查）

### 4：常见子类：

#### vector

1:可增长变数组数据结构，同步的100%延长

2特有的取全部元素：

Enumeration en=v.elements();

en.hasMoreElement()

En.nextElement()

功能和迭代器的一样

#### Arraylist

arrayList可以存放null。

1:特点：查询快。大小可变数组数据结构，不同步，替代了vector。50%延长

2:大小可变实现原理：创建一个新数组，将原来数组的内容复制到新数组里去。

3:查询原理：空间连续？？？？

4：默认初始容量10个

5：判断元素相等仅用equals（）--contains---remove用equals判断

6：实现原理：http://zhangshixi.iteye.com/blog/674856

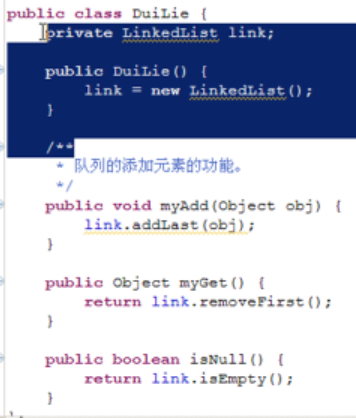
#### LinkedList：

1：链表数据结构，不同步，增删快

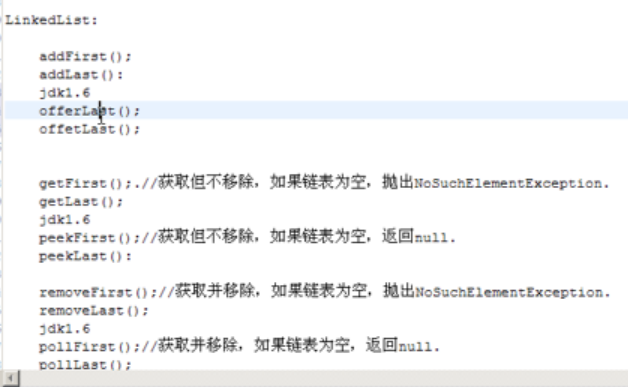
2：多了操作头和尾的first，last（add，remove，get）返回操作的元素

3：面试使用LinkedList来模拟一个堆栈或者队列，即做一个容器，自己分装

队列：



4: 1.6JDK升级：getfirst---peekfirst（如果链表没有元素，peek不跑出异常）



## 二：set

1：元素不能重复，无序

2：方法和collection 一样

3：判断两个元素相等：equals

### Hashset

实际上是个hashmap

0：无序原因。按hash存取

1:如何存值：利用对象的hashcode

2：hash表判断元素是否相同

（1）判断两个元素的哈希值是否相同

（2）hash值相同（hashcode相同）判断内容是否相同用的值equals方法

3：所以一般是要覆盖equals方法和hashcode方法-

4：原理：http://zhangshixi.iteye.com/blog/673143

### LinkedHashSet

1：hashset的子类

2：hash和链表数据结构。

3：有序的。维护了元素的插入顺序

### TreeSet

数据结构： 二叉树：相等的进不来（现在的可能是红黑树排序）

一：

:A:对集合元素进行自然排序，需要对对象类实现comparable接口覆盖compareTO方法

实现了该comparable接口的有：bigdecimal,biginteger,character,boolean,以及所有数值类型包装类，string，date，time

1;判断元素唯一性的方式：compareTO方法看这个方法返回是否为0

2： 排序：降序，a>b 改变两个参数顺序

3：string本身就实现了comparable接口

4：排序原理：二叉树比较原理，当添加一个元素时从根开始比根大走右边，比根小邹左边

二：

1：因为在compareTO方法中强制转换了类型若是不知道类型:？？？？？？。若是比较的方式改变，不能再回去修改类的compareTO

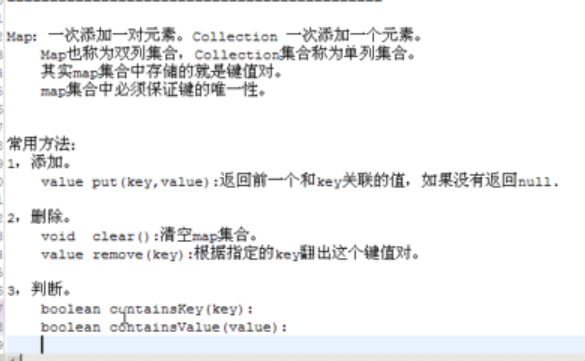
2：让集合自身具备比较功能 得给集合的构造函数传一个comparator对象

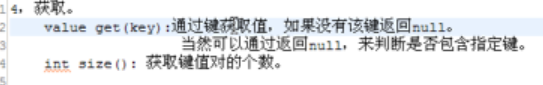
3：comparator接口 方法compare（object 1，object 1）

三：添加和取元素的原理

## Map

### 基本

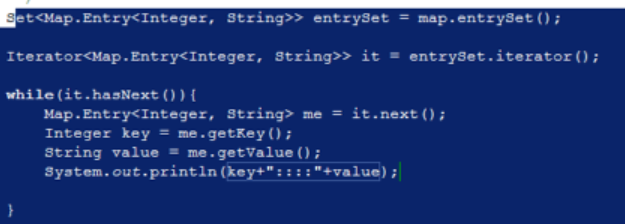


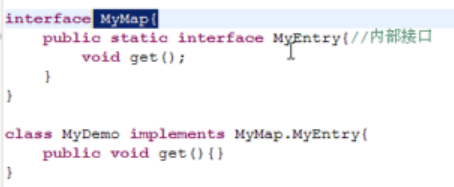


1：取元素：（1）keyset遍历，让map返回键的集合set

（2）entryset让map返回有迭代器的集合（返回的是映射关系的集合包含key-value对的）（每个集合元素都是map）

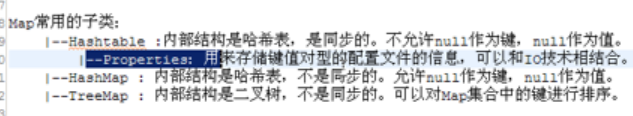
为什么返回set而不返回list，键的唯一性





Entry静态的内部接口

### 子类



#### Properties

1：是hashtable的子类，只是key和value都是string的

2：常用方法：getproperty（string key，string defalutvalue）

Setproperty（string key，string value）

Void load（inputstream instream）从属性文件（以输入流表示）中加载属性名=属性值。把加载到的键值对最佳到properties中

Void store（outputstream，stringcomments）将properties中的键值对写入指定属性文件

#### Linkedhashmap

有序的，存入和取出顺序一样

3：练习题：用map记录字母的次数

用字母做键，次数做value

4：原理：http://zhangshixi.iteye.com/blog/673789

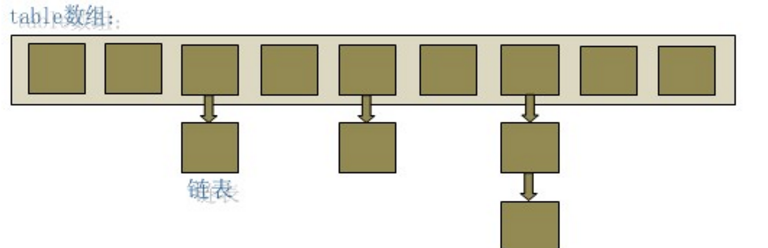
#### Hashmap

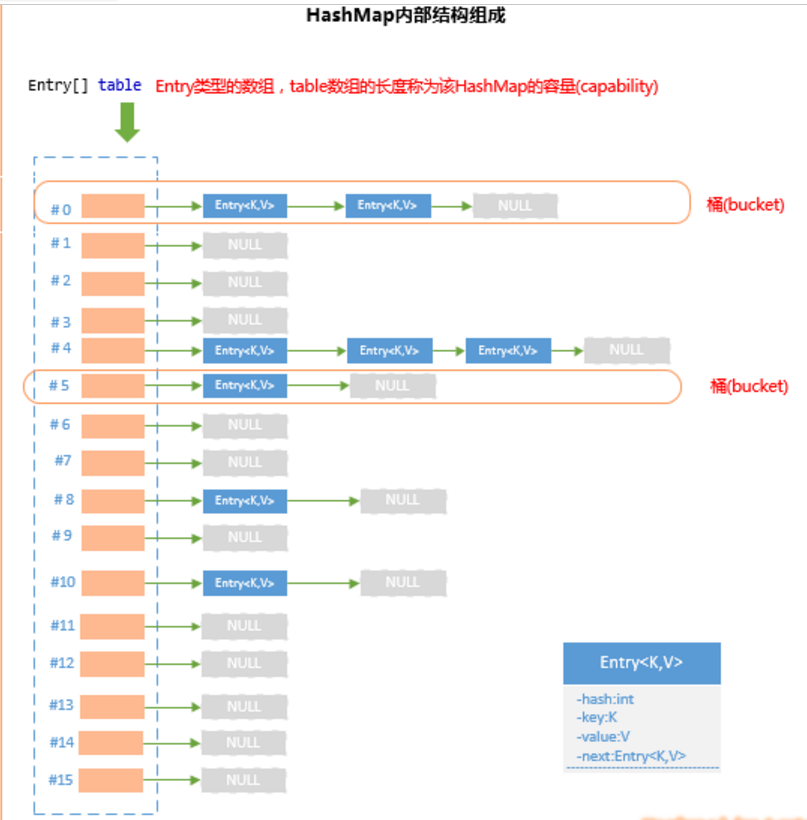
1：允许使用null作为key，value，由于hashmap的key不能重复，所以其最多只有一项key-value里key为null

2：判断两个值是否相等，hashcode和equals

3：结构：

http://zhangshixi.iteye.com/blog/672697





从上图中可以看出，HashMap底层就是一个数组结构，数组中的每一项又是一个链表。当新建一个HashMap的时候，就会初始化一个数组。

1. **transient** Entry[] table;
3. **static** **class** Entry<K,V> **implements** Map.Entry<K,V> {
4. **final** K key;
5. V value;
6. Entry<K,V> next;
7. **final** **int** hash;
8. ……
9. }

#### Hashtable

1：不允许使用null作为key，value

2：判断两个值是否相等，hashcode和equals

3：线程同步的

# 集合工具类collectionS

1：实现排序，反转，查找，替换，同步控制，设置不可变类

2：提供了多个synchronizedxxx返回指定集合对象的同步对象

3：emptyxxx返回一个空的，不可变的集合对象。

4：singletonxxx

5：Arrays类