复习：

1. Collection：集合的根接口。存储同一类型的多个元素的容器，大小不定。---泛型用于规定元素类型

2. List：列表。元素有序可以重复。---所以元素存在下标，能够精确操作指定的位置

ArrayList：顺序表。基于数组的。内存空间是连续的。默认初始容量是10，每次扩容之后增加一半。是一个线程不安全的列表。查询相对容易而增删相对复杂

LinkedList：链表。基于节点的。内存空间是不连续的。是一个线程不安全的列表。增删相对容易而查询相对复杂。

Vector：向量。基于数组的。内存空间是连续的。默认初始容量是10，每次默认增加1倍。是一个线程安全的列表。

Stack：栈。FILO。继承了Vector。

### Queue---队列

遵循先进先出(FIFO)的原则--但是可以使用集合的一些方法

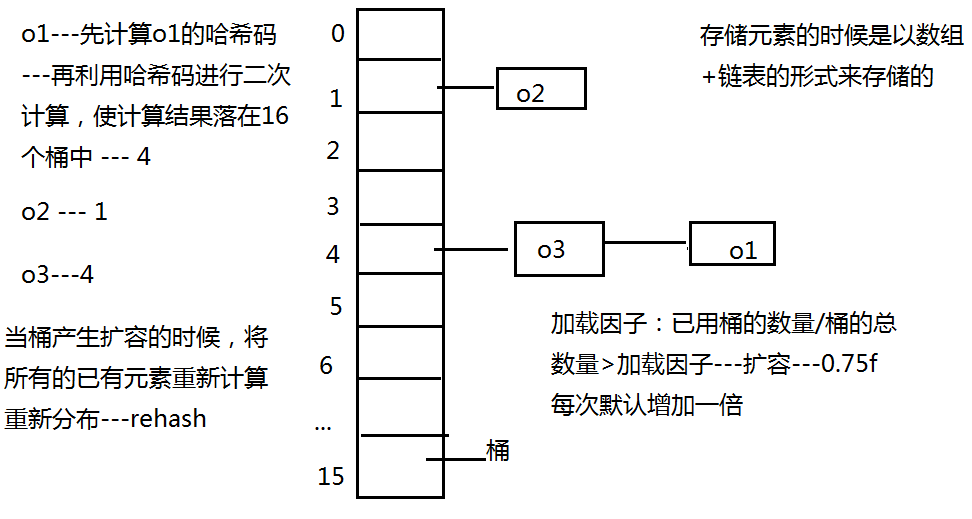
Deque--双向队列

Java中主要的线性集合：主要是List，Queue

### Set---散列集合

set中的元素是唯一的。

HashSet---不保证元素的存入顺序---是一个线程不安全的集合



#### TreeSet

要求存入的元素对应的类必须实现Comparable接口，在没有指定特殊规则的情况下会根据重写compareTo方法进行排序---TreeSet本身是一个线程不安全的集合

如果需要给某一个TreeSet来指定比较规则，这时候需要创建的是Comparator(比较器)对象，重写的compare方法

### 迭代器

Iterator---本质上就是通过指针的挪动来依次获取元素---在迭代过程中不允许增删原集合中的元素

Collection中的iterator()方法是从Itarable继承来的

如果一个对象想要被增强for循环遍历，这个对象对应的类必须实现Itarable接口。---增强for循环本质上就是在进行迭代遍历---增强for循环是JDK1.5的特性之一

### Collections

操作集合的工具类。

练习：最大疯狂值

输入一个数字，输入对应个数的整数

5

15 12 17 48 32

疯狂值：相邻两个数字之差的绝对值之和

|15 - 12| + |12 - 17| + |17 - 48| + |48 - 32| = 55

## 泛型

参数化类型---ParameterizedType---JDK1.5的特性之一

将泛型替换为具体类型的过程---泛型的擦除---发生在编译期、

? extends 类/接口---上限

? super 类/接口---下限

? 表示一个通配符/占位符

## Map<K,V>---映射

由键和值两组元素来组成的。一个键对应一个值。每一个键和对应的值称之为是一个键值对。--一个映射实际上由多个键值对来构成的。---键是唯一的。

如果将每一个键值对看做一个对象，那么就可以抽取出一个代表键值对的接口---Entry

所以实际上一个Map对象是由多个Entry对象来构成的。

### 遍历映射

1. 先获取所有的键，根据键获取值

2. 将所有的entry对象放到一个集合或者数组中

练习：输入一个字符串，统计每一个元素出现的次数

映射是集合吗？---不是，映射是集合框架的一员

集合框架不是集合，包含了集合、映射、数组以及操作它们对应的一些工具类

HashMap---底层数组+链表结构存储数据。默认大小是16，默认加载因子是0.75f，每次默认增加一倍。允许键或者值为null。是一个异步式线程不安全的映射。

Hashtable---默认大小是11，默认加载因子是0.75f。---不允许键或者值为null。是一个同步式线程安全的映射。---基本上只存于面试中

ConcurrentHashMap---异步式线程安全的映射