

在这里帮大家总结一下HashMap和hashtable方面的知识点吧：

1. 关于HashMap的一些说法：

a) HashMap实际上是一个“链表散列”的数据结构，即数组和链表的结合体。

HashMap的底层结构是一个数组，数组中的每一项是一条链表。

b) HashMap的实例有两个参数影响其性能：“初始容量”和装填因子。

c) HashMap实现不同步，线程不安全。HashTable线程安全。

d) HashMap中的key-value都是存储在Entry中的。

e) HashMap可以存null键和null值，不保证元素的顺序恒久不变，它的底层使用的是数组和链表，通过hashCode()方法和equals方法保证键的唯一性。

f) 解决冲突主要有三种方法：定址法，拉链法，再散列法。

HashMap是采用拉链法解决哈希冲突的。注：链表法是将相同hash值的对象组成一个链表放在hash值对应的槽位；

用开放定址法解决冲突的做法是：当冲突发生时，使用某种探查(亦称探测)技术在散列表中形成一个探查(测)序列。沿此序列逐个单元地查找，直到找到给定的关键字，或者碰到一个开放的地址(即该地址单元为空)为止（若要插入，在探查到开放的地址，则可将待插入的新结点存入该地址单元）。

拉链法解决冲突的做法是：将所有关键字为同义词的结点链接在同一个单链表中。若选定的散列表长度为m，则可将散列表定义为一个由m个头指针组成的指针数组T[0..m-1]。凡是散列地址为i的结点，均插入到以T[i]为头指针的单链表中。T中各分量的初值均应为空指针。在拉链法中，装填因子 α 可以大于

1，但一般均取 $\alpha \leq 1$ 。拉链法适合未规定元素的大小。

2. Hashtable和HashMap的区别：

a) 继承不同。 public class Hashtable extends Dictionary implements Map

public class HashMap extends AbstractMap implements Map

b) Hashtable中的方法是同步的，而HashMap中的方法在缺省情况下是非同步的。在多线程并发的环境下，可以直接使用Hashtable，但是要使用HashMap的话就要自己增加同步处理了。

c) Hashtable 中，key 和 value 都不允许出现 null 值。在 HashMap 中，null 可以作为键，这样的键只有一个；可以有一个或多个键所对应的值为 null。当 get() 方法返回 null 值时，即可以表示 HashMap 中没有该键，也可以表示该键所对应的值为 null。因此，在 HashMap 中不能由 get() 方法来判断 HashMap 中是否存在某个键，而应该用 containsKey() 方法来判断。

d) 两个遍历方式的内部实现上不同。Hashtable、HashMap都使用了Iterator。而由于历史原因，Hashtable还使用了Enumeration的方式。

e) 哈希值的使用不同，HashTable直接使用对象的hashCode。而HashMap重新计算hash值。

f) Hashtable和HashMap它们两个内部实现方式的数组的初始大小和扩容的方式。

HashTable中hash数组默认大小是11，增加的方式是 $old * 2 + 1$ 。HashMap中hash数组的默认大小是16，而且一定是2的指数。

注： HashSet子类依靠hashCode()和equal()方法来区分重复元素。

HashSet内部使用Map保存数据，即将HashSet的数据作为Map的key值保存，这也是HashSet中元素不能重复的原因。而Map中保存key值的,会去判断当前Map中是否含有该Key对象，内部是先通过key的hashCode,确定有相同的hashCode之后，再通过equals方法判断是否相同。