

HashSet

一、基本介绍

1、特点：

无序性：无序性指的是元素在底层存储的位置是无序的，不等于随机性；

不可重复性：当向Set中添加相同的元素时，只能添加进第一个，后面的不能被添加进去

2、常用构造方法

HashSet(Collection<? extends E> c)

构造一个包含指定集合中的元素的新集合。

HashSet(int initialCapacity)

构造一个新的空集合；背景HashMap实例具有指定的初始容量和默认负载因子（0.75）。

HashSet(int initialCapacity, float loadFactor)

构造一个新的空集合；背景HashMap实例具有指定的初始容量和指定的负载因子。

3、什么是初始化大小与装载因子：

初始化尺寸就是当创建哈希表（HashSet内部用哈希表的数据结构）的时候桶（buckets）的数量。如果当前的尺寸已经满了，那么桶的数量会自动增长。

装载因子衡量的是在HashSet自动增长之前允许有多满。当哈希表中实体的数量已经超出装载因子与当前容量的积，那么哈希表就会再次进行哈希（也就是内部数据结构重建），这样哈希表大致有两倍桶的数量。

表中已经存储的元素的数量

装载因子 = $\frac{\text{表中已经存储的元素的数量}}{\text{哈希表的大小}}$

例如：如果内部容量为16，装载因子为0.75，那么当表中有12个元素的时候，桶的数量就会自动增长

4、HashSet中的一些重要方法：

boolean add(E e)：如果不存在则添加，存在则返回false。

void clear()：移除Set中所有的元素

boolean contains(Object o)：如果这个元素在set中存在，那么返回true。

boolean remove(Object o)：如果这个元素在set中存在，那么从set中删除。

Iterator iterator()：返回set中这个元素的迭代器。

二、HashSet中元素的存储原理（哈希算法）：

当向Set中添加对象时，首先调用此对象所在类的hashCode()方法，计算该对象的哈希值，此哈希值决定了此对象在Set中存放的位置；若此位置没有被存储对象则直接存储，若已有对象则通过对象所在类的equals()比较两个对象是否相同，相同则不能被添加。

HashSet存放的是**哈希值**，Hashset存储元素的顺序并不是按照**存入时的顺序**(和List显然不同)，是按照哈希值来存的，所以取数据也是按照哈希值取的。

HashSet不存入重复元素的规则：使用hashCode和equals。那么HashSet是如何检查重复？其实原理：HashSet会通过元素的hashCode()和equals()方法进行判断，当试图将元素加入到Set集合中，HashSet首先会使用对象的hashCode来判断对象加入的位置。

同时也会与其他已经加入的对象的hashCode进行比较，如果没有相等的hashCode，HashSet就认为这个对象之前不存在，如果之前存在同样的hashCode值，就会进一步的比较equals()方法，如果equals()比较返回结果是true，那么认为该对象在集合中的对象是一模一样的，不会将其加入；

如果比较返回的是false,那么HashSet认为新加入的对象没有重复，可以正确加入。

如图所示：当两个对象的hashCode不一样时，说明两个对象是一定不相等的，在存储时如左图所示，当两个对象的hashCode相等，但是equals()不相等，在实际中，会在同一个位置，用**链式结构来**保存多个对象，而HashSet访问集合元素时也是根据元素的hashCode值快速定位，如果HashSet中两个以上的元素具有相同的hashCode值，将会导致性能下降。

