2019.09.20

20 September 2019 10:47

Java Day 11

1、集合

a) 概念：存储其它对象的对象

b) 集合和数组的区别：

数组的长度是固定的，集合的长度是可变的

数组中存储的是同一类型的元素，可以存储基本数据类型值，

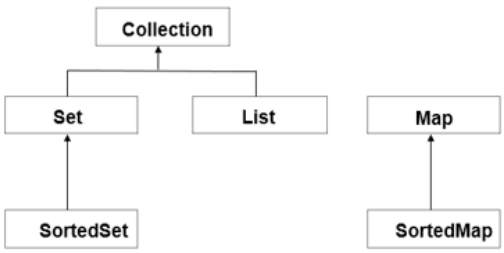
集合存储的都是对象，而且对象的类型可以不一致

如果元素个数是固定的推荐用数组；如果元素个数不是固定的，推荐用集合。

c) 集合框架

JavaSE 提供了满足各种需求的 API，在使用这些 API 前，先了解其继承与接口操作架构，才能了解何时采用哪个类，以及类之间如何彼此合作，从而达到灵活应用。

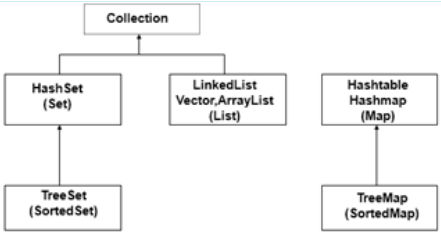
集合按照其存储结构可以分为两大类，分别是单列集合 java.util.Collection 和双列集合 java.util.Map



List：有序、可重复

Set：无序，不重复

SortedSet：有序、不重复



d) 集合操作

Collection 是所有单列集合的父接口，因此在 Collection 中定义了单列集合(List 和 Set)通用的一些方法，这些方法可用于操作所有的单列集合。方法如下：

* public boolean add(E e)
* public void clear();
* public boolean remove(Ee)
* public boolean contains(E e)
* public boolean isEmpty()
* public int size()
* public Object[] toArray();

2、迭代器

Iterator 接口的常用方法如下：

public E next(); 返回迭代的下一个元素

public boolean hasNext();

迭代器是对集合进行遍历，而每个集合内部的存储结构都是不同的，所以每一个集合存和取都是不一样的，那么就需要在每一个类中定义 hasNext() 和 next() 方法，这样做是可以的，但是会让整个集合体系过于臃肿，迭代器是将这样的方法向上抽取出接口，然后在每个类的内部，定义自己的迭代方式。

好处：

第一规定了在个集合体系的遍历方法都是 hasNext() 和 next() 方法。第二，代码有底层内部实现，使用者不用管是怎么实现的，会用即可。

原理：

在调用 Iterator 的 next 方法之前，迭代器的索引位于第一个元素之前，不指向任何元素，当第一次调用迭代器的 next 方法后，迭代器的索引会向后移动一位，指向第一个元素并将该元素返回，当再次调用 next 方法时，迭代器的索引会指向第二个元素并将该元素返回，依此类推，直到 hasNext() 方法返回 false，表示到达了集合的末尾，终止对元素的遍历。

3. List

a) 特点：

它是一个元素存取有序的集合，即元素的存入顺序和取出顺序一致。

它是一个带有索引的集合，通过索引就可以精确地操作集合中的元素。

集合中可以有重复的元素，通过元素的 equals 方法，来比较是否为重复的元素。

b) 常用操作

继承了 Collection 接口中的全部方法，而且还增加了一些根据元素索引来操作集合的特有方法：

public void add(int index, E element) : 将指定的元素，添加到该集合中的指定位置上
public E get(int index) :返回集合中指定位置的元素。
public E remove(int index) : 移除列表中指定位置的元素, 返回的是被移除的元素。
public E set(int index, E element) :用指定元素替换集合中指定位置的元素,返回值的更新前的元素。

c) ArrayList →

ArrayList 集合数据存储的结构是数组结构。

元素增删慢，查找快，由于日常开发中使用最多的功能为查询数据。

遍历数据，所以 ArrayList 是最常用的集合。

ArrayList 中放自定义类对象 ☆

注意：

1）ArrayList.remove(int)按照下标删除，不会自动装箱
2）迭代器遍历的同时，添加元素，并发修改异常，通过ListIterator.add(...)解决 java.util.ConcurrentModificationException
3）遍历集合删除元素，使用Iterator.remove()操作，不要直接使用List.remove()
4）集合数组互转Arrays.asList()/ArrayList.toArray()

d) LinkedList

LinkedList 集合数据存储的结构是双向链表结构。方便 增添加、删除的集合。

public void addFirst(E e) :将指定元素插入此列表的开头。
public void addLast(E e) :将指定元素添加到此列表的结尾。
public E getFirst() :返回此列表的第一个元素。
public E getLast() :返回此列表的最后一个元素。
public E removeFirst() :移除并返回此列表的第一个元素。
public E removeLast() :移除并返回此列表的最后一个元素。

笔试面试题：

笔试面试题

List的三个子类（ArrayList、LinkedList、Vector）的特点？
ArrayList:底层数据结构是数组，查询快，增删慢。线程不安全，效率高。
LinkedList:底层数据结构是链表，查询慢，增删快。线程不安全，效率高。
Vector:底层数据结构是数组，查询快，增删慢。线程安全，效率低。
Vector相对ArrayList查询慢(线程安全的) -->VectorTest.java
Vector相对LinkedList增删慢(数组结构)
Vector和ArrayList的区别?
    Vector是线程安全的,效率低
    ArrayList是线程不安全的,效率高
    共同点:都是数组实现的
ArrayList和LinkedList的区别？
    ArrayList底层是数组,查询和修改快
    LinkedList底层是链表结构的,增和删比较快,查询和修改比较慢
共同点:都是线程不安全的

4. Set

a) 特点：元素无序、不重复

b) HashSet

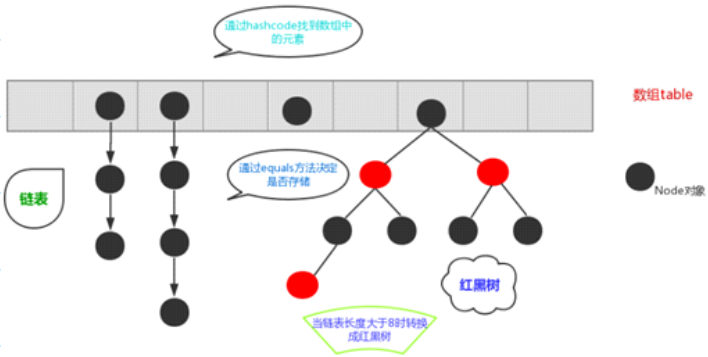
我们使用Set集合都是需要去掉重复元素的。如果在存储的时候逐个 equals()比较，效率很低，哈希算法提高了去重复的效率，降低了使用 equals()方法的次数

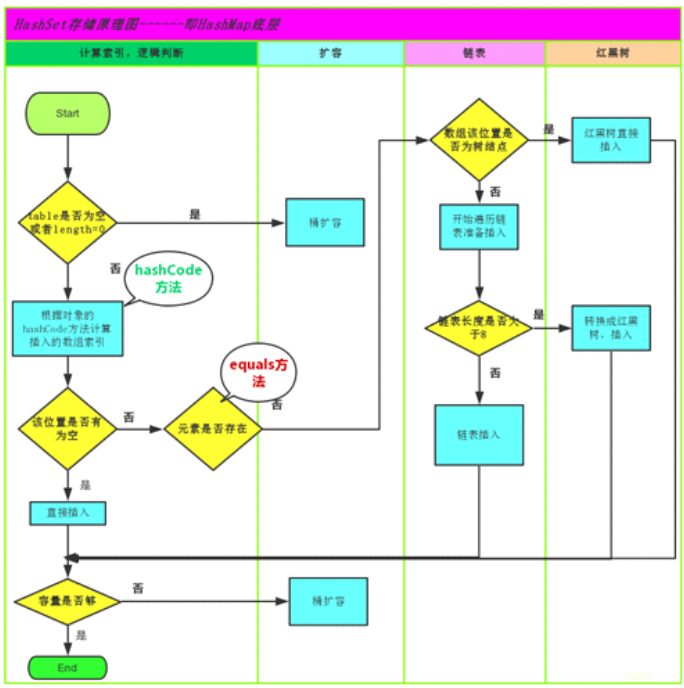
根据对象的哈希值来确定元素在集合中的存储位置，具有良好的存取和查找性能。

保证元素唯一性的方式依赖于：hashCode 与 equals 方法

HashSet 集合存储数据的结构(哈希表)：

数组 + 链表 + 红黑树 (Jdk1.8 中加了红黑树部分)





当HashSet调用add()方法存储对象的时候, 先调用对象的hashCode()方法得到一个哈希值, 然后在集合中查找是否有哈希值相同的对象

如果没有哈希值相同的对象就直接存入集合

如果有哈希值相同的对象, 就和哈希值相同的对象逐个进行equals()比较,比较结果为false就存入, true则不存

将自定义类的对象存入HashSet去重复

类中必须重写hashCode()和equals()方法

hashCode(): 属性相同的对象返回值必须相同, 属性不同的返回值尽量不同(提高效率)

equals(): 属性相同返回true, 属性不同返回false,返回false的时候存储

扩展了解：

1）散列表,它是基于高速存取的角度设计的，也是一种典型的“空间换时间”的做法。顾名思义，该数据结构能够理解为一个线性表，可是当中的元素不是紧密排列的，而是可能存在空隙。

散列表（Hash table，也叫哈希表），是依据关键码值(Key value)而直接进行訪问的数据结构。也就是说，它通过把关键码值映射到表中一个位置来訪问记录，以加快查找的速度。这个映射函数叫做散列函数，存放记录的数组叫做散列表。

比方我们存储70个元素，但我们可能为这70个元素申请了100个元素的空间。70/100=0.7，这个数字称为负载因子。我们之所以这样做，也是为了“高速存取”的目的。我们基于一种结果尽可能随机平均分布的固定函数H为每一个元素安排存储位置，这样就能够避免遍历性质的线性搜索，以达到高速存取。可是因为此随机性，也必定导致一个问题就是冲突。所谓冲突，即两个元素通过散列函数H得到的地址同样，那么这两个元素称为“同义词”。这类似于70个人去一个有100个椅子的饭店吃饭。散列函数的计算结果是一个存储单位地址，每一个存储单位称为“桶”。设一个散列表有m个桶，则散列函数的值域应为[0,m-1]。

加载(负载)因子是表示Hsah表中元素的填满的程度.若:加载因子越大,填满的元素越多,好处是,空间利用率高了,但:冲突的机会加大了.反之,加载因子越小,填满的元素越少,好处是:冲突的机会减小了,但:空间浪费多了.

2）红黑树

是二叉树的一种，红黑树本身就是一颗二叉查找树，将节点插入后，该树仍然是一颗二叉查找树。也就意味着，树的键值仍然是有序的。

红黑树的约束:

1. 节点可以是红色的或者黑色的
2. 根节点是黑色的
3. 叶子节点(特指空节点)是黑色的
4. 每个红色节点的子节点都是黑色的
5. 任何一个节点到其每一个叶子节点的所有路径上黑色节点数相同

红黑树的特点:

速度特别快,趋近平衡树,查找叶子元素最少和最多次数不多于二倍

a) **LinkedHashSet**

它是链表和哈希表组合的一个数据存储结构。元素的存入顺序和取出顺序一致

EOF

2016005851