集合框架

1. ArrayList 和 Vector 的区别。

这两个类都实现了 List 接口（List 接口继承了 Collection 接口），他们都是有序集合，即存储在这两个集合中的元素的位置都是有顺序的，相当于一种动态的数组，我们以后可以按位置索引号取出某个元素，并且其中的数据是允许重复的，这是 HashSet 之类的集合的最大不同处，HashSet 之类的集合不可以按索引号去检索其中的元素，也不允许有重复的元素（本来题目问的与 hashset 没有任何关系，但为了说清楚 ArrayList 与 Vector 的功能，我们使用对比方式，更有利于说明问题）。接着才说 ArrayList 与 Vector 的区别，这主要包括两个方面。

- 同步性：

Vector 是线程安全的，也就是说是它的方法之间是线程同步的，而 ArrayList 是线程序不安全的，它的方法之间是线程不同步的。如果只有一个线程会访问到集合，那最好是使用 ArrayList，因为它不考虑线程安全，效率会高些；如果有多个线程会访问到集合，那最好是使用 Vector，因为不需要我们自己再去考虑和编写线程安全的代码。

备注：对于 Vector&ArrayList、Hashtable&HashMap，要记住线程安全的问题，记住 Vector 与 Hashtable 是旧的，是 java 一诞生就提供了的，它们是线程安全的，ArrayList 与 HashMap 是 java2 时才提供的，它们是线程不安全的。所以，我们讲课时先讲老的。

- 数据增长：

ArrayList 与 Vector 都有一个初始的容量大小，当存储进它们里面的元素的个数超过了容量时，就需要增加 ArrayList 与 Vector 的存储空间，每次要增加存储空间时，不是只增加一个存储单元，而是增加多个存储单元，每次增加的存储单元的个数在内存空间利用与程序效率之间要取得一定的平衡。Vector 默认增长为原来两倍，而 ArrayList 的增长策略在文档中没有明确规定（从源代码看到的是增长为原来的 1.5 倍）。ArrayList 与 Vector 都可以设置初始的空间大小，Vector 还可以设置增长的空间大小，而 ArrayList 没有提供设置增长空间的方法。

总结：即 Vector 增长原来的一倍，ArrayList 增加原来的 0.5 倍。

2. 说说 ArrayList,Vector, LinkedList 的存储性能和特性。

ArrayList 和 Vector 都是使用数组方式存储数据，此数组元素数大于实际存储的数据以便增加和插入元素，它们都允许直接按序号索引元素，但是插入元素要涉及数组元素移动等内存操作，所以索引数据快而插入数据慢，Vector 由于使用了 synchronized 方法（线程安全）。

通常性能上较 ArrayList 差，而 LinkedList 使用双向链表实现存储，按序号索引数据需要进行前向或后向遍历，但是插入数据时只需要记录本项的前后项即可，所以插入速度较快。

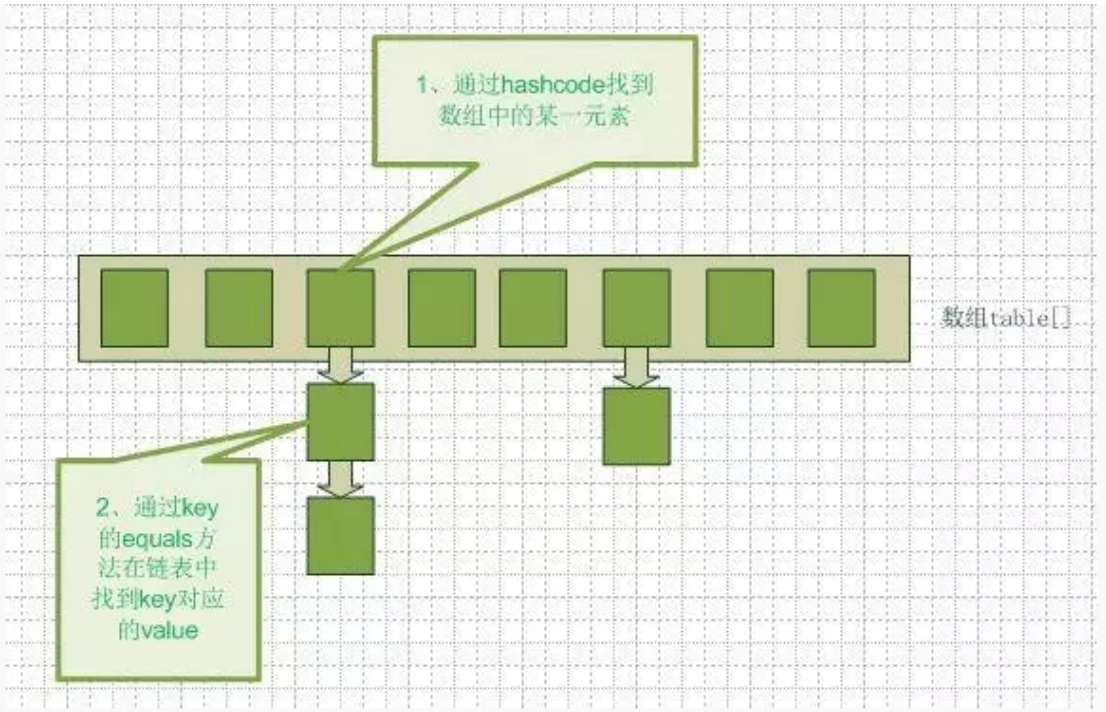
ArrayList 在查找时速度快，LinkedList 在插入与删除时更具优势。

3. 快速失败（fail-fast）和安全失败（fail-safe）的区别是什么？

Iterator 的安全失败是基于对底层集合做拷贝，因此，它不受源集合上修改的影响。java.util 包下面的所有的集合类都是快速失败的，而 java.util.concurrent 包下面的所有的类都是安全失败的。快速失败的迭代器会抛出 ConcurrentModificationException 异常，而安全失败的迭代器永远不会抛出这样的异常。

4. hashmap 的数据结构。

在 java 编程语言中，最基本的结构就是两种，一个是数组，另外一个是模拟指针（引用），所有的数据结构都可以用这两个基本结构来构造的，hashmap 也不例外。Hashmap 实际上是一个数组和链表的结合体（在数据结构中，一般称之为 “链表散列 “）



enter image description here

5. HashMap 的工作原理是什么?

Java 中的 HashMap 是以键值对 (key-value) 的形式存储元素的。HashMap 需要一个 hash 函数，它使用 hashCode()和 equals()方法来向集合 / 从集合添加和检索元素。当调用 put() 方法的时候，HashMap 会计算 key 的 hash 值，然后把键值对存储在集合中合适的索引上。 如果 key 已经存在了，value 会被更新成新值。HashMap 的一些重要的特性是它的容量 (capacity)，负载因子 (load factor) 和扩容极限(threshold resizing)。

6. Hashmap 什么时候进行扩容呢？

当 hashmap 中的元素个数超过数组大小 loadFactor 时，就会进行数组扩容，loadFactor 的默认值为 0.75，也就是说，默认情况下，数组大小为 16，那么当 hashmap 中元素个数超过 16*0.75=12 的时候，就把数组的大小扩展为 2*16=32，即扩大一倍，然后重新计算每个元素在数组中的位置，而这是一个非常消耗性能的操作，所以如果我们已经预知 hashmap 中元素的个数，那么预设元素的个数能够有效的提高 hashmap 的性能。比如说，我们有 1000 个元素 new HashMap(1000), 但是理论上来讲 new HashMap(1024) 更合适，不过上面 annegu 已经说过，即使是 1000，hashmap 也自动会将其设置为 1024。 但是 new HashMap(1024) 还不是更合适的，因为 0.75*1000 < 1000, 也就是说为了让 0.75 * size > 1000, 我们必须这样 new HashMap(2048) 才最合适，既考虑了 & 的问题，也避免了 resize 的问题。

7. List、Map、Set 三个接口，存取元素时，各有什么特点？

这样的题属于随意发挥题：这样的题比较考水平，两个方面的水平：一是要真正明白这些内容，二是要有较强的总结和表述能力。如果你明白，但表述不清楚，在别人那里则等同于不明白。

首先，List 与 Set 具有相似性，它们都是单列元素的集合，所以，它们有一个功共同的父接口，叫 Collection。Set 里面不允许有重复的元素，所谓重复，即不能有两个相等（注意，不是仅仅是相同）的对象 ，即假设 Set 集合中有了一个 A 对象，现在我要向 Set 集合再存入一个 B 对象，但 B 对象与 A 对象 equals 相等，则 B 对象存储不进去，所以，Set 集合的 add 方法有一个 boolean 的返回值，当集合中没有某个元素，此时 add 方法可成功加入该元素时，则返回 true，当集合含有与某个元素 equals 相等的元素时，此时 add 方法无法加入该元素，返回结果为 false。Set 取元素时，没法说取第几个，只能以 Iterator 接口取得所有的元素，再逐一遍历各个元素。

List 表示有先后顺序的集合， 注意，不是那种按年龄、按大小、按价格之类的排序。当我们多次调用 add(Obj e) 方法时，每次加入的对象就像火车站买票有排队顺序一样，按先来后到的顺序排序。有时候，也可以插队，即调用 add(int index,Obj e) 方法，就可以指定当前对象在集合中的存放位置。一个对象可以被反复存储进 List 中，每调用一次 add 方法，这个对象就被插入进集合中一次，其实，并不是把这个对象本身存储进了集合中，而是在集合中用一个索引变量指向这个对象，当这个对象被 add 多次时，即相当于集合中有多个索引指向了这个对象，如图 x 所示。List 除了可以以 Iterator 接口取得所有的元素，再逐一遍历各个元素之外，还可以调用 get(index i) 来明确说明取第几个。

Map 与 List 和 Set 不同，它是双列的集合，其中有 put 方法，定义如下：put(obj key,obj value)，每次存储时，要存储一对 key/value，不能存储重复的 key，这个重复的规则也是按 equals 比较相等。取则可以根据 key 获得相应的 value，即 get(Object key) 返回值为 key 所对应的 value。另外，也可以获得所有的 key 的结合，还可以获得所有的 value 的结合，还可以获得 key 和 value 组合成的 Map.Entry 对象的集合。

List 以特定次序来持有元素，可有重复元素。Set 无法拥有重复元素，内部排序。Map 保存 key-value 值，value 可多值。

HashSet 按照 hashcode 值的某种运算方式进行存储，而不是直接按 hashCode 值的大小进行存储。例如，"abc" ---> 78，"def" ---> 62，"xyz" ---> 65 在 hashSet 中的存储顺序不是 62,65,78，这些问题感谢以前一个叫崔健的学员提出，最后通过查看源代码给他解释清楚，看本次培训学员当中有多少能看懂源码。LinkedHashSet 按插入的顺序存储，那被存储对象的 hashcode 方法还有什么作用呢？学员想想! hashset 集合比较两个对象是否相等，首先看 hashcode 方法是否相等，然后看 equals 方法是否相等。new 两个 Student 插入到 HashSet 中，看 HashSet 的 size，实现 hashcode 和 equals 方法后再看 size。

同一个对象可以在 Vector 中加入多次。往集合里面加元素，相当于集合里用一根绳子连接到了目标对象。往 HashSet 中却加不了多次的。

8. Set 里的元素是不能重复的，那么用什么方法来区分重复与否呢? 是用 == 还是 equals()? 它们有何区别?

Set 里的元素是不能重复的，元素重复与否是使用 equals() 方法进行判断的。

equals() 和 == 方法决定引用值是否指向同一对象 equals() 在类中被覆盖，为的是当两个分离的对象的内容和类型相配的话，返回真值。

9. 两个对象值相同 (x.equals(y) == true)，但却可有不同的 hash code，这句话对不对?

对。如果对象要保存在 HashSet 或 HashMap 中，它们的 equals 相等，那么，它们的 hashcode 值就必须相等。

如果不是要保存在 HashSet 或 HashMap，则与 hashcode 没有什么关系了，这时候 hashcode 不等是可以的，例如 arrayList 存储的对象就不用实现 hashcode，当然，我们没有理由不实现，通常都会去实现的。

10. heap 和 stack 有什么区别。

Java 的内存分为两类，一类是**栈内存**，一类是**堆内存**。栈内存是指程序进入一个方法时，会为这个方法单独分配一块私属存储空间，用于存储这个方法内部的局部变量，当这个方法结束时，分配给这个方法的栈会释放，这个栈中的变量也将随之释放。

堆是与栈作用不同的内存，一般用于存放不放在当前方法栈中的那些数据，例如，使用 new 创建的对象都放在堆里，所以，它不会随方法的结束而消失。方法中的局部变量使用 final 修饰后，放在堆中，而不是栈中。

11. Java 集合类框架的基本接口有哪些？

集合类接口指定了一组叫做元素的对象。集合类接口的每一种具体的实现类都可以选择以它 自己的方式对元素进行保存和排序。有的集合类允许重复的键，有些不允许。

Java 集合类提供了一套设计良好的支持对一组对象进行操作的接口和类。Java 集合类里面 最基本的接口有：

Collection：代表一组对象，每一个对象都是它的子元素。

Set：不包含重复元素的 Collection。

List：有顺序的 collection，并且可以包含重复元素。

Map：可以把键（key）映射到值（value）的对象，键不能重复。

12. HashSet 和 TreeSet 有什么区别？

HashSet 是由一个 hash 表来实现的，因此，它的元素是无序的。add()，remove()，contains()

TreeSet 是由一个树形的结构来实现的，它里面的元素是有序的。因此，add()，remove()，contains() 方法的时间复杂度是 O(logn)。

13. HashSet 的底层实现是什么?

通过看源码知道 HashSet 的实现是依赖于 HashMap 的，HashSet 的值都是存储在 HashMap 中的。在 HashSet 的构造法中会初始化一个 HashMap 对象，HashSet 不允许值重复，因此，HashSet 的值是作为 HashMap 的 key 存储在 HashMap 中的，当存储的值已经存在时返回 false。

14. LinkedHashMap 的实现原理?

LinkedHashMap 也是基于 HashMap 实现的，不同的是它定义了一个 Entry header，这个 header 不是放在 Table 里，它是额外独立出来的。

LinkedHashMap 通过继承 hashMap 中的 Entry，并添加两个属性 Entry before,after，和 header 结合起来组成一个双向链表，来实现按插入顺序或访问顺序排序。LinkedHashMap 定义了排序模式 accessOrder，该属性为 boolean 型变量，对于访问顺序，为 true；对于插入顺序，则为 false。一般情况下，不必指定排序模式，其迭代顺序即为默认为插入顺序。

15. 为什么集合类没有实现 Cloneable 和 Serializable 接口？

克隆（cloning）或者是序列化（serialization）的语义和含义是跟具体的实现相关的。因此，应该 由集合类的具体实现来决定如何被克隆或者是序列化。

16. 什么是迭代器（Iterator）？

Iterator 接口提供了很多对集合元素进行迭代的方法。每一个集合类都包含了可以返回迭代 器实例的迭代方法。迭代器可以在迭代的过程中删除底层集合的元素，但是不可以直接调用集合的 remove(Object Obj) 删除，可以通过迭代器的 remove() 方法删除。

17. Iterator 和 ListIterator 的区别是什么？

下面列出了他们的区别：

Iterator 可用来遍历 Set 和 List 集合，但是 ListIterator 只能用来遍历 List。

Iterator 对集合只能是前向遍历，ListIterator 既可以前向也可以后向。

ListIterator 实现了 Iterator 接口，并包含其他的功能，比如：增加元素，替换元素，获取前一个和后一个元素的索引，等等。

18. 数组（Array）和列表（ArrayList）有什么区别？什么时候应该使用 Array 而不是 ArrayList？

Array 可以包含基本类型和对象类型，ArrayList 只能包含对象类型。

Array 大小是固定的，ArrayList 的大小是动态变化的。

ArrayList 处理固定大小的基本数据类型的时候，这种方式相对比较慢。

19. Java 集合类框架的最佳实践有哪些？

- 假如元素的大小是固 定的，而且能事先知道，我们就应该用 Array 而不是 ArrayList。
- 有些集合类允许指定初始容量。因此，如果我们能估计出存储的元素的数目，我们可以设置 初始容量来避免重新计算 hash 值或者是扩容。
- 为了类型安全，可读性和健壮性的原因总是要使用泛型。同时，使用泛型还可以避免运行时的 ClassCastException。
- 使用 JDK 提供的不变类（immutable class）作为 Map 的键可以避免为我们自己的类实现 hashCode()和 equals()方法。
- 编程的时候接口优于实现。
- 底层的集合实际上是空的情况下，返回长度是 0 的集合或者是数组，不要返回 null。

20. Set 里的元素是不能重复的，那么用什么方法来区分重复与否呢？是用 == 还是 equals()？它们有何区别？

Set 里的元素是不能重复的，那么用 iterator() 方法来区分重复与否。equals() 是判读两个 Set 是否相等

equals() 和 == 方法决定引用值是否指向同一对象 equals() 在类中被覆盖，为的是当两个分离的对象的内容和类型相配的话，返回真值

21. Comparable 和 Comparator 接口是干什么的？列出它们的区别。

Java 提供了只包含一个 compareTo() 方法的 Comparable 接口。这个方法可以个给两个对象排序。具体来说，它返回负数，0，正数来表明输入对象小于，等于，大于已经存在的对象。

Java 提供了包含 compare() 和 equals() 两个方法的 Comparator 接口。compare() 方法用来给两个输入参数排序，返回负数，0，正数表明第一个参数是小于，等于，大于第二个参数。equals() 方法需要一个对象作为参数，它用来决定输入参数是否和 comparator 相等。只有当输入参数也是一个 comparator 并且输入参数和当前 comparator 的排序结果是相同的时 候，这个方法才返回 true。

22. Collection 和 Collections 的区别。

collection 是集合类的上级接口，继承与它的接口主要是 set 和 list。

collections 类是针对集合类的一个帮助类. 它提供一系列的静态方法对各种集合的搜索，排序，线程安全化等操作。