1、集合与数组

数组是定长的，给定一个数组后，就只能自改数组中的值，而不能修改数组的长度。如需修改长度，则只能重新建立数组对象。【并且数组可以存储基本数据类型，而集合不行】

集合是动态数组，底层还是数组。所谓的动态，是指当数组长度不够的时候，能自动扩容。该自动扩容实际就是新建数组，后将原数组中的数据copy进新数组中。

ArrayList底层：

Arrays.copyOf(elementData, newCapacity);

2、集合的层次关系

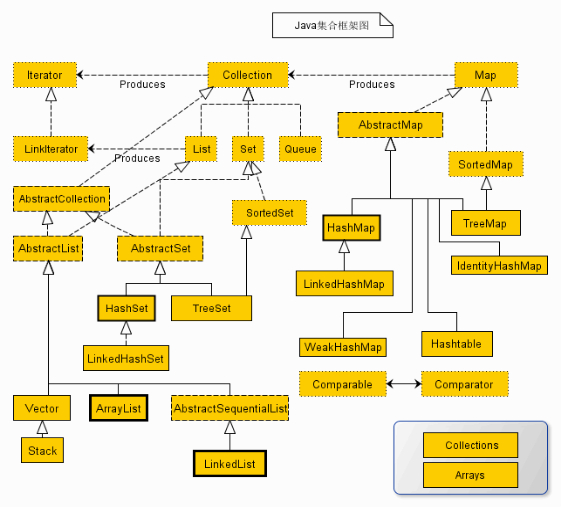
Iterator->Collection->{List, Set }->{AbstractList, AbstractSet}

Map->{HashMap, HashTable,TreeMap,LinkedHashMap}

AbstractList->{ArrayList, LinkedList, Vector}

AbstractSet->{HashSet,LinkedHashSet, TreeSet}

如图所示：图中，实线边框的是实现类，折线边框的是抽象类，而点线边框的是接口



3、集合中几个重要的接口和简介

Iterator：实现了该接口的集合都可以使用迭代器迭代集合中的数据。

Collection：List和Set的父类

List：其内元素：有序并可重复（有序：记录元素的添加顺序）

Set：其内元素：无序但是不可重复

Map：其内元素：无序旦不可重复(隶属于：java.util)

4、集合的遍历

List集合的遍历：

1、for循环 (因为)

2、Iterator迭代器

3、foreach：增强for循环（实际就是迭代器）

4、ListIterator：是Iterator的子接口，专门用于输出List的内容

而在List的实现类中，由于各个类 的底层实现不一致，所以在选取遍历的遍历方法上也有差别。

1. ArrayList：因实现了快速访问的接口RandomAccess,能通过索引进行获取数据。所以在使用for循环会比迭代器更快。
2. LinkedList：因为其是链表，所以不存在索引， 故而使用迭代器获取数据会更快捷。
3. Vector：与ArrayList一样。

Set集合的遍历：

因不存在索引，所以使用迭代器进行遍历。

Map集合的遍历：

1、通过keySet()获取key的值，在通过get(key)方法，获取value

2、通过entrySet()获取Map中的保存key-value的entry集合，在通过

getKey(),getValue(),获取key和value

5、ArrayList和LinkedList

ArrayList：底层是动态数组，在元素个数>数组长度时，会进行扩容操作。相应的会保存元素的索引值。因为存在索引，所以查询和修改快。

LinkedList：底层是双向链表，每个节点都会保存上一个和下一个的节点，内存消耗处。 因为是链表，所以新增和删除快，只需要修改节点的指向。

6、Map集合（链地址法）

HashMap：（key-value键值对） key可以为null， value都可以

底层数据结构：数组 + 链表 + 红黑树

【初始容量：16， 2倍扩容， 负载因子：0.75， 并且扩容的空间是2的幂次方，这样使得在存储数据时，能尽可能的降低hash冲突，使得数据尽可能的均匀的分布在数组上。】

LinkedHashMap：是HashMap的子类，与父类属性差不多

但是，LinkedHashMap底层多维护了一条双向链表，记录了数据存储的顺序，从而做到有序。

HashTable：初始容量 11， 2倍再 +1 扩容， 线程安全

Key 和 value 都不能为null

TreeMap：所添加的对象必须实现了Comparable接口或comparator接口

Key 不能为空

【concurrentHashMap】:jdk 1.7 使用的是segment锁（分段锁）

Jdk 1.8 以后：

使用的是cas + synchronized

7、主要实现类的区别小结

Vector和ArrayList  
1、ctor是线程同步的，所以它也是线程安全的，而arraylist是线程异步的，是不安全的。如果不考虑到线程的安全因素，一般用arraylist效率比较高。  
2、果集合中的元素的数目大于目前集合数组的长度时，vector增长率为目前数组长度的100%，而arraylist增长率为目前数组长度的50%。如果在集合中使用数据量比较大的数据，用vector有一定的优势。  
3、果查找一个指定位置的数据，vector和arraylist使用的时间是相同的，如果频繁的访问数据，这个时候使用vector和arraylist都可以。而如果移动一个指定位置会导致后面的元素都发生移动，这个时候就应该考虑到使用linklist,因为它移动一个指定位置的数据时其它元素不移动。  
ArrayList 和Vector是采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，都允许直接序号索引元素，但是插入数据要涉及到数组元素移动等内存操作，所以索引数据快，插入数据慢，Vector由于使用了synchronized方法（线程安全）所以性能上比ArrayList要差，LinkedList使用双向链表实现存储，按序号索引数据需要进行向前或向后遍历，但是插入数据时只需要记录本项的前后项即可，所以插入数度较快。

arraylist和linkedlist  
1、rrayList是实现了基于动态数组的数据结构，LinkedList基于链表的数据结构。  
2、于随机访问get和set，ArrayList觉得优于LinkedList，因为LinkedList要移动指针。  
3、于新增和删除操作add和remove，LinedList比较占优势，因为ArrayList要移动数据。 这一点要看实际情况的。若只对单条数据插入或删除，ArrayList的速度反而优于LinkedList。但若是批量随机的插入删除数据，LinkedList的速度大大优于ArrayList. 因为ArrayList每插入一条数据，要移动插入点及之后的所有数据。

HashMap与TreeMap  
1、HashMap通过hashcode对其内容进行快速查找，而TreeMap中所有的元素都保持着某种固定的顺序，如果你需要得到一个有序的结果你就应该使用TreeMap（HashMap中元素的排列顺序是不固定的）。  
2、在Map 中插入、删除和定位元素，HashMap是最好的选择。但如果您要按自然顺序或自定义顺序遍历键，那么TreeMap会更好。使用HashMap要求添加的键类明确定义了hashCode()和 equals()的实现。  
两个map中的元素一样，但顺序不一样，导致hashCode()不一样。  
同样做测试：  
在HashMap中，同样的值的map,顺序不同，equals时，false;  
而在treeMap中，同样的值的map,顺序不同,equals时，true，说明，treeMap在equals()时是整理了顺序了的。

HashTable与HashMap  
1、同步性:Hashtable是线程安全的，也就是说是同步的，而HashMap是线程序不安全的，不是同步的。

2、HashMap允许存在一个为null的key，多个为null的value 。

3、hashtable的key和value都不允许为null。

8、集合框架是什么？集合框架的有点有哪些？

每种编程语言中都有集合，最初的Java版本包含几种集合类：Vector、Stack、HashTable和Array。随着集合的广泛使用，Java1.2提出了囊括所有集合接口、实现和算法的集合框架。在保证线程安全的情况下使用泛型和并发集合类，Java已经经历了很久。它还包括在Java并发包中，阻塞接口以及它们的实现。集合框架的部分优点如下：

（1）使用核心集合类降低开发成本，而非实现我们自己的集合类。

（2）随着使用经过严格测试的集合框架类，代码质量会得到提高。

（3）通过使用JDK附带的集合类，可以降低代码维护成本。

（4）复用性和可操作性。

9、集合框架中的泛型有什么优点？

Java1.5引入了泛型，所有的集合接口和实现都大量地使用它。泛型允许我们为集合提供一个可以容纳的对象类型，因此，如果你添加其它类型的任何元素，它会在编译时报错。这避免了在运行时出现ClassCastException，因为你将会在编译时得到报错信息。泛型也使得代码整洁，我们不需要使用显式转换和instanceOf操作符。它也给运行时带来好处，因为不会产生类型检查的字节码指令。

10、集合框架中的基础接口有哪些？

Collection为集合层级的根接口。一个集合代表一组对象，这些对象即为它的元素。Java平台不提供这个接口任何直接的实现。

    Set是一个不能包含重复元素的集合。这个接口对数学集合抽象进行建模，被用来代表集合，就如一副牌。

    List是一个有序集合，可以包含重复元素。你可以通过它的索引来访问任何元素。List更像长度动态变换的数组。

    Map是一个将key映射到value的对象.一个Map不能包含重复的key：每个key最多只能映射一个value。

   一些其它的接口有Queue、Dequeue、SortedSet、SortedMap和ListIterator。

11、为何Collection不从Cloneable和Serializable接口继承？

Collection接口指定一组对象，对象即为它的元素。如何维护这些元素由Collection的具体实现决定。例如，一些如List的Collection实现允许重复的元素，而其它的如Set就不允许。很多Collection实现有一个公有的clone方法。然而，把它放到集合的所有实现中也是没有意义的。这是因为Collection是一个抽象表现。重要的是实现。

    当与具体实现打交道的时候，克隆或序列化的语义和含义才发挥作用。所以，具体实现应该决定如何对它进行克隆或序列化，或它是否可以被克隆或序列化。

在所有的实现中授权克隆和序列化，最终导致更少的灵活性和更多的限制。特定的实现应该决定它是否可以被克隆和序列化。

12、为何Map接口不继承Collection接口？

尽管Map接口和它的实现也是集合框架的一部分，但Map不是集合，集合也不是Map。因此，Map继承Collection毫无意义，反之亦然。

如果Map继承Collection接口，那么元素去哪儿？Map包含key-value对，它提供抽取key或value列表集合的方法，但是它不适合“一组对象”规范。

Collection提供了很多的方法，但是这些方法都是基于单个对象（key）的，不适用于一组（key-value）对象。

13、Iterator是什么？

Iterator接口提供遍历任何Collection的接口。我们可以从一个Collection中使用迭代器方法来获取迭代器实例。迭代器取代了Java集合框架中的Enumeration。迭代器允许调用者在迭代过程中移除元素。

14、Enumeration和Iterator接口的区别？

Enumeration的速度是Iterator的两倍，也使用更少的内存。Enumeration是非常基础的，也满足了基础的需要。但是，与Enumeration相比，Iterator更加安全，因为当一个集合正在被遍历的时候，它会阻止其它线程去修改集合。

迭代器取代了Java集合框架中的Enumeration。迭代器允许调用者从集合中移除元素，而Enumeration不能做到。为了使它的功能更加清晰，迭代器方法名已经经过改善。

15、为何没有像Iterator.add()这样的方法，向集合中中添加元素？

  语义不明，已知的是，Iterator的协议不能确保迭代的次序。然而要注意，ListIterator没有提供一个add操作，它要确保迭代的顺序。

16、为何迭代器没有一个方法可以直接获取下一个元素，而不需要移动游标？

它可以在当前Iterator的顶层实现，但是它用得很少，如果将它加到接口中，每个继承都要去实现它，这没有意义。

17、Iterator和ListIterator之间的区别？

（1）我们可以使用Iterator来遍历Set和List集合，而ListIterator只能遍历List。

（2）Iterator只可以向前遍历，而LIstIterator可以双向遍历。

（3）ListIterator从Iterator接口继承，然后添加了一些额外的功能，比如添加一个元素、替换一个元素、获取前面或后面元素的索引位置。

18、遍历一个List有哪些不同的方式？

19、通过迭代器fail-fast属性，你明白了什么？

20、fail-fast与fail-safe有什么区别？

21、在迭代一个集合的时候，如何避免ConcurrentModificationException?

22、为何Iterator接口没有具体实现？

23、UnsupportedOperationException是什么？

24、在Java中，HashMap是如何工作的？

25、hashCode()和equals()方法有何重要性？

26、我们能否使用任何类作为Map的key?

27、Map接口提供了哪些不同的集合视图？

28、HashMap和HashTable有何不同？

29、如何决定选用HashMap还是TreeMap？

30、ArrayList和Vector有何异同？

31、Array和ArrayList有何区别？什么时候更合适用Array?

32、ArrayList和LinkedArrayList有何区别？

33、哪些集合类提供对元素的随机访问？

34、EnumSet是什么？

35、哪些集合类是线程安全的？

36、并发集合类是什么？

37、BlockingQueue是什么？

38、队列和栈是什么，列出他们的区别？

39、Collections类是什么？

40、Comparable 和 Comparator接口是什么？

41、Comparable 和Comparator接口有何区别？

42、我们如何对一组对象进行排序？

43、当一个集合被作为参数传递给一个函数时，如何 才可以确保函数不能修改它？

44、我们如何从给定集合那里创建一个synchronized的集合？

45、集合框架里实现的通用算法有哪些？

46、大写的0是什么？举几个例子？

47、与java接框架相关的有哪些最好的实践？