# Java常见面试题

集合与数组？

数组：（可以存储基本数据类型）是用来存储对象的一种容器，但是数组的长度固定，不适合在对象数量未知的情况下使用

集合：（只能存储对象，对象类型可以不一样）集合的长度可变，可在多数情况下使用

Collection接口是集合类的根接口，Java中没有提供这个接口的直接的实现类，但是却让其被继承产生了两个接口

Set：不包含重复的元素

List：有序的集合，可以包含重复的元素，提供了按索引访问的方式

Map是Java.util包中的另一个接口，它和Collection接口没有关系，是相互独立的，但是都属于集合的一部分，Map包含了Key – Value对，Map不能包含重复的Key，但是可以包含相同的Value。

Iterator，所有的集合类都实现了Iterator接口，这是一个用于遍历集合中元素的接口，主要包含以下三种方法：

hasnext()：是否还有下一个元素

next()：返回下一个元素

remove()：删除当前元素

几种重要的接口和类简介

List（有序，可重复）

List里存放的对象是有序的，同时也是可以重复的，List关注的是索引，拥有一系列和索引相关的方法，查询速度快，因为往List集合里插入或删除数据时，会伴随着后面数据的移动，所以插入数据速度很慢

Set（无序，不能重复）

Set里存放的对象是无序的，不能重复的，集合中的对象不按特定的方式排序，只是简单的把对象放入集合中。

Map（键值对，键唯一，值不唯一）

Map集合中存储的时键值对，键不能重复，值可以重复，根据键得到值，对map集合遍历时先得到键的Set集合，对Set集合遍历，得到相对应的值。

遍历

在集合中提供了以下四种的常见的输出方式

Iterator：迭代输出，是使用最多的输出方式

ListIterator：是Iterator的子接口，用于专门输出List中的内容‘

Foreach输出：JDK1.5之后提出的新功能。可以输出数组或集合

For循环

ArrayList和LinkedList

ArrayList和LinkedList在用法上没有区别，但是在功能上还是有区别的

LinkedList经常用在增删操作较多而查询操作很少的情况

Arraylist经常用在查询较多，增删操作较少的情况下。

Map集合

实现类：HashMap，HashTable，LinkedHashMap和TreeMap

HashMap

HashMap是最常用的Map它根据键的HashCode来存储元素，根据键可以直接获取它的值，具有很快的访问速度。

遍历时，取得数据的顺序是完全随机的，因为键对象不可以重复，所以HashMap最多只允许一条记录的键为Null，允许多条记录的值为Null，是非同步的

HashTable

HashTable与HashMap类似，是HashMap的线程安全版，它支持线程的同步，任一时刻只有一个线程能写HashTable

因此也导致了HashTable在写入时会比较慢，它集成子Dictionary类，不同的时它不允许记录的键或值为null，同时效率较低

LinkedHashMap

LinkedHashMap保存来记录的插入顺序，在用Iteraor遍历LinkedHashMap时，先得到的记录肯定是先插入的，在遍历的时候会比HashMap慢，有HashMap的全部特性

TreeMap

TreeMap实现SortMap接口，能够把它保存的记录根据键顺序，默认是按键值的升序排序（自然顺序），可以指定排序的比较器，当用Iterator遍历ThreeMap时，得到的记录是排过序的，不允许Key值为空，非同步

Map的遍历

KeySet（）；

将Map中所有的键存入到set集合中因为set几倍迭代器，所以可以迭代取出所有的键，再根据get方法获取每一个键对应的值，KeySet（）迭代后只能通过get（）取Key。

ebtrySet（）

Set<Map.Entry<K,V>>entrySet（）返回此映射值中包含的映射关系的Set视图（一个关系就是一个兼职对）就是把（Key-value）作为一个整体一对一的存放到Set集合中，Map.Entry表示映射关系

EntrySet（）迭代后可以e.getKey() ,e.getValue() 两种方法来去Key和Value，返回的值是Entry接口

推荐使用entrySet（）方法，效率较高

对于KeySet其实是遍历了2次，一次转为iterator，一次就是从HashMap中取出Key所对应的Value

Entryset（）只是遍历了第一次，它把Key和Value都放到了entry中，所以快了。两种遍历的遍历时间相差还是很明显的。

ArrayList和LinkedList的区别

ArrayList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

对于随机访问get和set，ArrayList性能优于LinkedList，因为LinkedList要移动指针

对于新增和删除操作，add和remove，linkedList比较占优势，因为ArrayList要移动数据，这一点要看实际情况的，若值对单条数据插入或删除，ArrayList的速度反而要优于LinkedList，但是要是批量随机的插入删除数据，LinkedList的书读大大优于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

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许多结果你就应该去使用TreeMap（HashMap中元素排列顺序是不固定的）

在Map中插入，删除和定位元素，HashMap是最好的选择，但如果要按照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

连个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

在TreeMap中，同样的值的map，顺序不同，equals时，true，说明TreeMap在equals（）时是整理了顺序的。

HashTable与HashMap的区别

同步性：HashTable是线程安全的，也就是说是同步的，而HashMap是线程不安全的，不是同步的

HashMap允许存在一个为null 的key，多个为null的元素

HashTable的Key和Value都不允许为null

Java集合框架是什么，说出一些集合框架的优点

每种编程语言中都有集合，最初的Java版本中包含击中集合类：Vector，Stack，HashTable和Array，随着集合的广泛使用，Java1.2提出囊括所有集合接口，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，阻塞接口以及他们的实现。集合框架的部分优点如下

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

通过使用经过严格测试的结合框架，代码质量会得到提高

通过使用JDK附带的集合类，可以降低代码维护成本

复用性和可操作性

集合框架中的泛型有什么优点

Java1.5中引入了泛型，所有的集合接口和实现都大量的使用它，泛型允许我们为集合提供一个可以容纳的对象类型，因此，如果你想添加其他类型的任何元素，它都会在编译时报错，避免了再运行时出现错误，因此你将会在编译时得到报错信息，泛型也是的代码整洁，我们不需要使用显式转换和instanceOf操作符。

Java集合框架的基础接口有哪些

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不提供这个接口的任何直接实现。

Set是一个包含不重复元素的集合，这个接口对数字集合抽象进行建模，被用来代表集合，就如一副牌一样。

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

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

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中使用迭代器的方法来获取迭代器实例，迭代器允许调用者在迭代过程中移出元素

HashMap是如何工作的

HashMap使用哈希算法，在put和get方法中，他是用hashCode()和equals()方法，当我们通过传递Key-value对，调用put方法的时候，HashMap使用key hashCode()和哈希算法，来找出存储Key-value对的索引Entry存储在LinkedList中，如果存在Entry，它使用equals()方法来检查传递的Key是否已经存在如果存在，它会覆盖Value，如果不存在，它会创建一个新的entry然后保存，让我们通过传递key调用get（）方法是，它再次使用hashCode()来找到数组中的索引，然后使用equals()方法找出正确的Entry，然后返回它的值。

HashCode()和equals()方法有和重要性

HashMap使用Key对象的HashCode()和equals()方法去决定Key-Value对的索引，当我们试着从HashMap中获取值的时候，这些方法也会被用到，如果这些方法没有正确的实现，这种情况下，两个不同Key也会产生相同的HashCode()和equals()输出

HashMap将会认为他们是相同的，然后覆盖他们，而非把他们存储到不同的地方，同样的所有不允许存储重复数据的集合类都使用HashCode()和equals()去查找重复，所以他们非常重要

Equals()和hashCode（）的实现应该遵循以下规则：

如果o1.equals(o2)，那么o1,hashCode()==o2.hashCode()总是为true 的

如果o1.hashCode()==02.hashCode()，并不意味着o1.equals(o2)会为true。

简单来讲：

HashCode相同，不一定对象相同

对象相同，HashCode一定相同。

能否使用任何类作为Map的Key

可以！

使用之前要考虑几点

如果类重写了equals()方法，它也应该重写HashCode()方法

类的所有实例要遵循与equals()和HashCode()相关的规则

如果一个类没有使用equals()，你不应该在HashCode()中使用它

HashMap和HashTable有何不同

HashMap允许key和value为空，而HashTable不允许

HashTable是同步的是安全的，而HashMap不是，所以HashMap适合单线程环境，HashTable适合多线程环境

如何决定选用HashMap还是ThreeMap

对于在Map中插入，删除和定位元素这类的操作，HashMap是孔的选择，然而，假如你需要对一个有序的Key集合进行遍历，ThreeMap是更高的选择，基于你的Collection的大小，也许想HashMap中添加元素会更快，将Map换位TheeMap进行有序Key的遍历

ArrayList 和 Vertor有何异同

ArrayList和Vector在很多时候都很类似

两者都是基于索引的，内部又一个数组支持

两者维护插入的顺序，我们可以根据插入顺序来获取元素

ArrayList和Vector两者允许null值，也可以使用索引值对元素进行随机访问

不同点

Vector是同步的是安全的，而ArrayList不是

ArrayList比Vector快

ArrayList更加的通用

Array和ArrayList有何区别？什么时候更适合用Array?

Array可以容纳基本类型和对象，而ArrayList只能容纳对象

Array是指定大小的，而ArrayList大小是固定的

Array没有ArrayList那么多的功能，比如addAll，rumoveAll和Iterator等

如果列表的大小已经指定，大部分的情啊狂还是存储和遍历它们

如果要使用多维数组，使用Array[][]要比List<List<>>更容易

ArrayList和LinkedList有何区别？

ArrayList和Linked两者都实现了List接口，但是它们之间有些不同

ArrrayList是时间了基于动态数组的数据结构，ListedList是基于链表的数据结构。

对于随机访问get（）和set（）ArrayList速度优于LinkedList，因为LinkedList要移动指针。

对于新增和删除操作add和remove，LinkedList比较占优势，因为ArrayList要移动数据。

两者都是现成不安全的，允许重复的。

哪些集合提供对元素的随机访问？

ArrayList，HashMap，TreeMap和HashTable都提供对元素的随机访问。