## 一面题目（电话面试）

1. java8函数式编程

使用函数式代码的好处：

* 减少了可变量(Immutable Variable)的声明
* 能够更好的利用并行(Parallelism)
* 代码更加简洁和可读





1. 回调函数，函数式编程，面向对象之间区别
2. 面向对象编程的优点：面向对象程序设计要比以往的做法更加便于学习，因为它能够让人们更简单地设计并维护程序，使得程序更加便于分析、设计、理解。同时它也是易拓展的，由于继承、封装、多态的特性，自然设计出高内聚、低耦合的系统结构，使得系统更灵活、更容易扩展，而且成本较低。缺点：面向对象虽然开发效率高但是代码的运行效率比起面向过程要低很多，这也限制了面向对象的使用场景不能包括那些对性能要求很苛刻的地方。
3. 回调函数：
4. 函数式编程优点：在函数式编程中，由于数据全部都是不可变的，所以没有并发编程的问题，是多线程安全的。可以有效降低程序运行中所产生的副作用，对于快速迭代的项目来说，函数式编程可以实现函数与函数之间的热切换而不用担心数据的问题，因为它是以函数作为最小单位的，只要函数与函数之间的关系正确即可保证结果的正确性。缺点：由于所有的数据都是不可变的，所以所有的变量在程序运行期间都是一直存在的，非常占用运行资源。

**3.面向对象的3大特点是什么？接口和抽象类的区别是什么？**

1.继承：一个类可以继承现有类的属性和方法，同时我们可以修改继承来的方法让其更加的符合自己的实际需求。新类被称作为现有类的子类，现有类被称作为新类的父类。

2.封装：封装是面向对象的特征之一，面向对象所要讲的一切事物皆为对象，对象一般都会包括属性和方法，但是有些属性是私有的不可见的，有些是可见的，所以我们想要访问到那些私有的属性的时候，必须通过对象对外提供的方法去访问那些不可见的属性。

3.多态：指一个类实例的相同方法在不同情形有不同表现形式。多态机制使具有不同内部结构的对象可以共享相同的外部接口。这意味着，虽然针对不同对象的具体操作不同，但通过一个公共的类，它们（那些操作）可以通过相同的方式予以调用。

4.接口和抽象类的区别：

1）。接口中的成员变量都是public static final类型的，而抽象类中的成员属性没有做限制。

2）。接口中不能有对方法的具体实现，而抽象类中可以有。

3）。抽象类和接口都是不能实例化的。

4）。一个类一次只能继承一个抽象类，但是一个类可以实现多个接口。

4.说说对集合类的了解？

１. **HashSet**  
HashSet有以下特点  
¬ 不能保证元素的排列顺序，顺序有可能发生变化  
¬ 不是同步的  
¬ 集合元素可以是null,但只能放入一个null  
当向HashSet集合中存入一个元素时，HashSet会调用该对象的hashCode()方法来得到该对象的hashCode值，然后根据 hashCode值来决定该对象在HashSet中存储位置。  
简单的说，HashSet集合判断两个元素相等的标准是两个对象通过equals方法比较相等，并且两个对象的hashCode()方法返回值相 等  
注意，如果要把一个对象放入HashSet中，重写该对象对应类的equals方法，也应该重写其hashCode()方法。其规则是如果两个对 象通过equals方法比较返回true时，其hashCode也应该相同。另外，对象中用作equals比较标准的属性，都应该用来计算 hashCode的值。

**TreeSet类**  
TreeSet是SortedSet接口的唯一实现类，TreeSet可以确保集合元素处于排序状态。TreeSet支持两种排序方式，自然排序 和定制排序，其中自然排序为默认的排序方式。向TreeSet中加入的应该是同一个类的对象。  
TreeSet判断两个对象不相等的方式是两个对象通过equals方法返回false，或者通过CompareTo方法比较没有返回0  
自然排序  
自然排序使用要排序元素的CompareTo（Object obj）方法来比较元素之间大小关系，然后将元素按照升序排列。  
Java提供了一个Comparable接口，该接口里定义了一个compareTo(Object obj)方法，该方法返回一个整数值，实现了该接口的对象就可以比较大小。  
obj1.compareTo(obj2)方法如果返回0，则说明被比较的两个对象相等，如果返回一个正数，则表明obj1大于obj2，如果是 负数，则表明obj1小于obj2。  
如果我们将两个对象的equals方法总是返回true，则这两个对象的compareTo方法返回应该返回0  
定制排序  
自然排序是根据集合元素的大小，以升序排列，如果要定制排序，应该使用Comparator接口，实现 int compare(T o1,T o2)方法。

２. “集合框架”提供两种常规的 Map 实现：HashMap 和TreeMap。和所有的具体实现一样，使用哪种实现取决于您的特定需要。在Map 中插入、删除和定位元素，HashMap 是最好的选择。但如果您要按顺序遍历键，那么TreeMap 会更好。根据集合大小，先把元素添加到 HashMap，再把这种映射转换成一个用于有序键遍历的 TreeMap 可能更快

３.  1.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。   
     2.对于随机访问get和set，ArrayList觉得优于LinkedList，因为LinkedList要移动指针。   
     3.对于新增和删除操作add和remove，LinedList比较占优势，因为ArrayList要移动数据。

５.ｈａｓｈＭａｐ，当ｋｅｙ为自定义的类的时候，需要实现重新ｅｑｕａｌｓ和ｈａｓｈＣｏｄｅ方法

１. 用自定义类作为key，必须重写equals()和hashCode()方法。

自定义类中的equals() 和 hashCode()都继承自Object类。

Object类的hashCode()方法返回这个对象存储的内存地址的编号。

而equals()比较的是内存地址是否相等。

在Map，set等容器中，在存取时都需要大量的equals操作，会影响效率。所以引入了hashCode()。

通过hashCode()能够计算出一个hash值，通过hash值来判断两个对象的值是否相等,如果hash值不相等则说明这两个对象不相等，如果hash值相等则继续用equals()去判断两个对象是否相等。

６.常见的集合的扩容。

.１.HashMap默认容量为16的Entry数组，默认加载因子为0.75

扩容：使用一个容量更大的数组来代替已有的容量小的数组，transfer()方法将原有Entry数组的元素拷贝到新的Entry数组里。put过程是先计算hash然后通过hash与table.length取摸计算index值，然后将key放到table[index]位置，当table[index]已存在其它元素时，会在table[index]位置形成一个链表，将新添加的元素放在table[index]，原来的元素通过Entry的next进行链接，这样以链表形式解决hash冲突问题，当元素数量达到临界值(capactiy\*factor)时，则进行扩容，是table数组长度变为table.length\*2

Hashtable：初始容量为11，扩容时是容量翻倍+1即:capacity\*2+1。

HashSet：底层实现和hashmap一样，不是key value结构，仅仅是存储不重复的元素，HashSet内部就是使用HashMap实现，只不过HashSet里面的HashMap所有的value都是同一个Object而已，因此HashSet也是非线程安全的

 ArrayList是实现List接口的动态数组，初始容量10，扩容(原始容量x3)/2 + 1

７．ｈａｓｈＭａｐ和ｃｏｎｃｕｒｒｅｎｔＨａｓｈＭａｐ的区别吗？

ｈａｓｈＭａｐ是线程不安全的而ｈａｓｈｔａｂｌｅ是线程安全的，ｈａｓｈｔａｂｌｅ是使用ｓｙｎｃｈｒｏｎｉｚｅｄ来保证线程安全的，但是ｃｏｎｃｕｒｒｅｎｔＨａｓｈＭａｐ是使用分段锁，来保证线程安全。

ConcurrentHashMap允许多个修改操作并发进行，其关键在于使用了锁分离技术。它使用了多个锁来控制对hash表的不同部分进行的修改。ConcurrentHashMap内部使用段(Segment)来表示这些不同的部分，每个段其实就是一个小的hash table，它们有自己的锁。只要多个修改操作发生在不同的段上，它们就可以并发进行。

８.ｖｏｌａｔｉｌｅ关键字作用？

用volatile修饰的变量，线程在每次使用变量的时候，都会读取变量修改后的最的值。volatile很容易被误用，用来进行原子性操作。

对于volatile修饰的变量，jvm虚拟机只是保证从主内存加载到线程工作内存的值是最新的，但是不能保证ｊｖｍ的内存区域中加载的变量是最新的。