Android最全BAT面试题

1. **java面试题**

熟练掌握java是很关键的，大公司不仅仅要求你会使用几个api，更多的是要你熟悉源码实现原理，甚至要你知道有哪些不足，怎么改进，还有一些java有关的一些算法，设计模式等等。

##### （一） java基础面试知识点

* java中==和equals和hashCode的区别
* ==比较的是地址,equals比较的对象引用地址,hashcode是生成一个hash码,用来比较hash码是否相同的.
* int、char、long各占多少字节数
* 4,2,8
* int与integer的区别
* 探探对java多态的理解
* String、StringBuffer、StringBuilder区别
* String是不能改变的字符串,stringbuffer是线程安全的,stringbuilder是线程不安全的,stringbuffer和stringbuilder是可以改变的字符串
* 什么是内部类？内部类的作用
* 抽象类和接口区别
* 抽象类的意义
* 抽象类与接口的应用场景
* 抽象类是否可以没有方法和属性？
* 接口的意义
* 泛型中extends和super的区别
* 作为一个限定域,extends指的是子类可以,super
* 父类的静态方法能否被子类重写
* 进程和线程的区别
* final，finally，finalize的区别
* Final是关键字，不能作为父类被继承不能和abstract配合使用。Finally是try
* Catch无论是否发生异常都会执行的语句，finalize是垃圾回收中的一个方法，确定这个对象没有被引用的时候调用。
* 序列化的方式
* Serializable 和Parcelable 的区别
* Serializable是写在硬盘上的，用到了反射，parcelable用的是内存
* 静态属性和静态方法是否可以被继承？是否可以被重写？以及原因？
* 静态内部类的设计意图
* 成员内部类、静态内部类、局部内部类和匿名内部类的理解，以及项目中的应用
* 谈谈对kotlin的理解
* 闭包和局部内部类的区别
* string 转换成 integer的方式及原理

##### （二） java深入源码级的面试题（有难度）

* 哪些情况下的对象会被垃圾回收机制处理掉？

通过引用计数法和可达性分析法来判断对象是否还活着，如果否，就会被垃圾回收机制回收掉

* 讲一下常见编码方式？

GBK，UTF-8，UTF-16，ASCLL码

* utf-8编码中的中文占几个字节；int型几个字节？

少数3，大多数4，int是4个

* 静态代理和动态代理的区别，什么场景使用？

静态代理，需要手动的生成代理类源码再编译成代理类，代理类和委托类在运行前就已经确定关系了，动态代理类，程序运行期间由JVM根据反射机制动态生成的，代理类和委托类的关系在运行时确定

* Java的异常体系

异常往上抛

* 谈谈你对解析与分派的认识。
* 修改对象A的equals方法的签名，那么使用HashMap存放这个对象实例的时候，会调用哪个equals方法？
* Java中实现多态的机制是什么？
* 如何将一个Java对象序列化到文件里？
* 说说你对Java反射的理解
* 说说你对Java注解的理解
* 说说你对依赖注入的理解
* 说一下泛型原理，并举例说明
* Java中String的了解

1.String类是final类是不可以被继承的，并且他的成员方法默认是final方法。java中final方法是不可以被继承的  
2.String对象一旦被创建就是固定不变的了，对String对象的任何改变都不影响到原对象，相关的任何change操作都会生成新的对象  
3.每当我们创建字符串常量时，JVM会首先检查字符串常量池，如果该字符串已经存在常量池中，那么就直接返回常量池中的实例引用。如果字符串不存在常量池中，就会实例化该字符串并且将其放到常量池中。由于String字符串的不可变性我们可以十分肯定常量池中一定不存在两个相同的字符

* String为什么要设计成不可变的？

**1字符串常量池的需要，**假如此字符串值已经存在于常量池中,则不会创建一个新的对象,而是引用已经存在的对象 2允许String对象缓存hashcode码，字符串不变，可以保证hashcode的唯一性，不用每次都去计算 3安全性考虑，string被很多方法当做参数使用，如果可变，则不安全

* Object类的equal和hashCode方法重写，为什么？

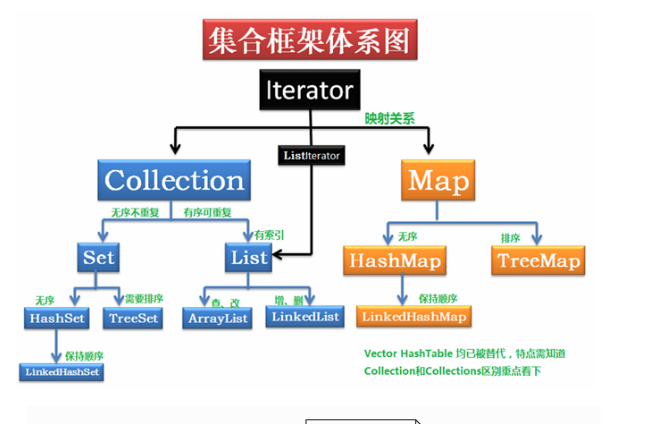
如果两个对象相同（使用equal），则hashcode也相同

如果两个对象的hashcode相同，那么他们不一定equlas=true；

##### （三） 数据结构

* 常用数据结构简介
* 并发集合了解哪些？

阻塞式集合和非阻塞式集合

* 列举java的集合以及集合之间的继承关系
* 集合类以及集合框架
* 容器类介绍以及之间的区别（容器类估计很多人没听这个词，Java容器主要可以划分为4个部分：List列表、Set集合、Map映射、工具类（Iterator迭代器、Enumeration枚举类、Arrays和Collections），具体的可以看看这篇博文 [Java容器类](https://link.jianshu.com?t=http://alexyyek.github.io/2015/04/06/Collection/" \t "_blank)）
* List,Set,Map的区别
* List和Map的实现方式以及存储方式
* HashMap的实现原理

hash值得到这个元素在数组中的位置（即下标），如果数组该位置上已经存放有其他元素了，那么在这个位置上的元素将以链表的形式存放，新加入的放在链头，最先加入的放在链尾。如果数组该位置上没有元素，就直接将该元素放到此数组中的该位置上。

* HashMap数据结构？

hashMap实际上是一个链表散列的数据结构，数组和链表的结合体。

* HashMap源码理解
* HashMap如何put数据（从HashMap源码角度讲解）？
* HashMap怎么手写实现？
* ConcurrentHashMap的实现原理
* ArrayMap和HashMap的对比
* HashTable实现原理
* TreeMap具体实现
* HashMap和HashTable的区别
* HashMap与HashSet的区别
* HashSet与HashMap怎么判断集合元素重复？

Hashset不能添加重复数据，当添加数据的时候，会调用object的hashcode来判断是否存在，不存在才添加。而实现的原理是利用hashmap的key唯一性来判断是否重合。

Hashmap中会根据key来判断元素是否重合，如果重合那么就会覆盖当前的元素，因为他是数组链表形式，所以他会将最新添加的数据放在链头部，之前的数据放在链尾部。

* 集合Set实现Hash怎么防止碰撞

底层封装一个hashmap利用key的唯一性来判断是否重合

* ArrayList和LinkedList的区别，以及应用场景

Arraylist是基于动态数组的数据结构，linkedlist是基于链表的数据结构

ArrayList对于随机访问例如get或者set，性能优于LinkedList

ArrayList对于新增和删除，他的性能低于LinkedList。

* 数组和链表的区别

数组在堆，链表在栈

数组静态分配内存，链表动态分布内存

数组利用下标定位，时间复杂度为O（1），链表时间复杂度为O（n）所以get（）set（）方法会快

数组增加删除元素时间复杂度为O（n），链表时间复杂度为O（1），所以增删操作比较耗费时间。

* 二叉树的深度优先遍历和广度优先遍历的具体实现

深度优先搜索，沿着树的深度遍历节点，尽可能的搜索树的分支，实现：先访问根节点，之后左节点，再右节点，所以可以根据堆栈的特性，将右节点压入栈中再将左节点压入，这样就可以保证左子树先于右子树被遍历。

广度优先遍历是沿着树结构横向遍历，先根节点然后根据宽度进行遍历，可以利用队列实现广度搜索

* 堆的结构

是一堆数组，逻辑存储是完全二叉树结构

* 堆和树的区别

堆是一种特殊的树，根节点的值小于子节点，下面的子节点又是堆结构，就是由大到小或者由小到大堆起来的一个结构

* 堆和栈在内存中的区别是什么(解答提示：可以从数据结构方面以及实际实现方面两个方面去回答)？

堆存对象，栈存对象的地址和基本数据类型的变量。

* 什么是深拷贝和浅拷贝
* 手写链表逆序代码
* 讲一下对树，B+树的理解
* 讲一下对图的理解
* 判断单链表成环与否？
* 链表翻转（即：翻转一个单项链表）
* 合并多个单有序链表（假设都是递增的）

##### （四） 线程、多线程和线程池

* 开启线程的三种方式？

1继承Thread，2实现Runanable，3使用callable和future创建

* 线程和进程的区别？

线程是系统进行资源分配和调度的一个独立单元

线程是实体，是资源分配和调度的最小单位

* 为什么要有线程，而不是仅仅用进程？
* run()和start()方法区别

Run（）方法可以调用多次，而start（）只能调用一次

* 如何控制某个方法允许并发访问线程的个数？

构造函数中创建5个信号，所以当需要访问时最多只有5个线程并发访问，超过五个就要排队等候，当消息运行完成，就会释放，才能有下一个消息进来。

* 在Java中wait和seelp方法的不同；

Seelp是Thread类中的，wait是object类中的，seelp只是暂时出让给cpu其他的线程，但是他的状态还是存在，到指定时间就会重新恢复，而且线程锁不会被释放。

Wait线程会放弃线程锁，此线程被挂起，直到其他线程调用同一对象的notify（）方法才会重新激活调用者。

* 谈谈wait/notify关键字的理解
* 什么导致线程阻塞？

该线程放弃cpu的使用，暂时挂起，等待导致阻塞的原因消除才会被恢复。或者被其他线程中断，也会导致线程阻塞

原因：1执行seelp操作，2执行了一个对象的wait（）,3有同步线程锁，但是锁暂时还未获得，所以进入线程阻塞状态，4IO操作时等待一些资源，而进入线程阻塞状态。

* 线程如何关闭？

使用退出标志来退出，比如用wehile循环来做，当标识为true的时候就一直运转，如果标识为false的时候就取消，使用stop来做，使用interrupt来操作，interrupt是不能中断正在运行的线程的，他只能改变中断状态。

* 讲一下java中的同步的方法

使用synchronized关键字，对象都会有一个内置锁，内置锁会保护整个方法，调用此方法时需要获取到内置锁，否则则会引起阻塞

使用synchronized修饰同步代码块

使用volatile关键字

使用重入锁来实现线程同步

* 数据一致性如何保证？

1使用推，2使用拉，3使用冥等性，多次重复操作，为消息加一个标识，并且保存这个消息的状态，相同的消息只处理一次，并做状态判断

* 如何保证线程安全？

原子性，同一时刻只能有一个线程对数据操作，可见性，当一个线程对内存的修改其他的线程可以看到，有序性，在jvm编译器和处理器对指令重新排序，它不会影响到单线程，但是会影响到多线程的并发执行正确性，需要用到synchronized，volitile，lock保证有序性。

* 如何实现线程同步？

Synchronized关键字，同步代码块儿，使用sysnnchronized修饰的代码块，使用volile特殊域变量实现线程同步。

* 两个进程同时要求写或者读，能不能实现？如何防止进程的同步？

可以，三种类型，1读者优先，如果读者源源不断，写者得不到资源，造成写者饥饿。2写者优先，如果在之前已经进入队列的读者还是可以优先获取资源的，只要写者源源不断，读者得不到资源，读者饥饿。3读写公平，只要之前已经排队的读者，写者就算获取到资源，也要等待所有等待中的读者结束。

* 线程间操作List
* Java中对象的生命周期
* Synchronized用法
* synchronize的原理
* 谈谈对Synchronized关键字，类锁，方法锁，重入锁的理解
* static synchronized 方法的多线程访问和作用
* 同一个类里面两个synchronized方法，两个线程同时访问的问题
* volatile的原理
* 谈谈volatile关键字的用法
* 谈谈volatile关键字的作用
* 谈谈NIO的理解
* synchronized 和volatile 关键字的区别
* synchronized与Lock的区别
* ReentrantLock 、synchronized和volatile比较
* ReentrantLock的内部实现
* lock原理
* 死锁的四个必要条件？
* 怎么避免死锁？
* 对象锁和类锁是否会互相影响？
* 什么是线程池，如何使用?
* Java的并发、多线程、线程模型
* 谈谈对多线程的理解
* 多线程有什么要注意的问题？
* 谈谈你对并发编程的理解并举例说明
* 谈谈你对多线程同步机制的理解？
* 如何保证多线程读写文件的安全？
* 多线程断点续传原理
* 断点续传的实现

##### （五）并发编程有关知识点（这个是一般Android开发用的少的，所以建议多去看看）：

平时Android开发中对并发编程可以做得比较少，Thread这个类经常会用到，但是我们想提升自己的话，一定不能停留在表面，,我们也应该去了解一下java的关于线程相关的源码级别的东西。

1. **Android面试部分**

Android面试题包括Android基础，还有一些源码级别的、原理这些等。所以想去大公司面试，一定要多看看源码和实现方式，常用框架可以试试自己能不能手写实现一下，锻炼一下自己。

##### （一）Android基础知识点

* 四大组件是什么
* Activity，service，broadcasts，content provider
* 四大组件的生命周期和简单用法
* Activity，oncreate，onstart，onresume，onpuse，onstop，ondestroy
* Service oncreate onstartcommand，ondestroy或者oncreate，onbind，onunbind，ondestory
* content provider 内容提供者，实现跨进程通信。
* Broadcasts分为有序广播和无序广播，无序广播效率高，没有顺序，无法被截断，有序广播只有这个收到广播逻辑执行完毕，才会继续传递，可以截断，有顺序的广播
* Activity之间的通信方式
* Intent，静态变量，全局变量，application，借助外在工具sp存储 ，文件存储，数据库存储，使用service来通信。
* Activity各种情况下的生命周期
* 横竖屏切换的时候，Activity 各种情况下的生命周期
* 如果没有设置configchange那么会销毁重新创建，并且会调用onsaveinstancestatus方法，设置了configchange则不会重新创建
* Activity与Fragment之间生命周期比较
* Fragment在oncreat之前多了一个onttach方法，在oncreat之后多了oncreatview，onactivitycreat，在ondestroy之前加了一个ondestoryview，在ondestroy之后加了一个ondetach
* Activity上有Dialog的时候按Home键时的生命周期
* 正常的生命周期，当按了home键，activity退出到后台，调用onstop挂起到后台两个Activity 之间跳转时必然会执行的是哪几个方法？
* 前台切换到后台，然后再回到前台，Activity生命周期回调方法。弹出Dialog，生命值周期回调方法。
* A oncreat onstart，onresume，onpuse，Boncreat onstart，onresume，Aonstop，如果A长时间在后台会被销毁，再回到前台的时候，Bonpause，Aoncreat onstart，onresume，Bonstop，如果没有被销毁，B onpause， Aonrestart onstart onresume，B onstop
* Activity的四种启动模式对比
* Standard模式： 每次启动在栈顶创建一个新的实例，即使处于栈顶位置也会重新创建
* singleTop模式： 判断是否处于栈顶，如果处于栈顶就直接复用，如果不处于，则会创建
* Singleintask模式： 判断栈中是否有当前activity实例，如果存在将其他的全部出栈，并且复用
* Singleinstance： 会启动过一个新的任务栈来管理activity实例，并且该实例在整个系统中只存在一个，无论从哪启动都会将当前的任务栈转移到前台来，而使activity显示。
* Activity状态保存于恢复
* 使用Onsaveinstancestate方法保存状态，使用onrestoreinstancestate方法恢复
* fragment各种情况下的生命周期
* Fragment状态保存startActivityForResult是哪个类的方法，在什么情况下使用？
* 如何实现Fragment的滑动？
* 和viewpager绑定，就可以滑动了，viewpager来处理touch事件。
* fragment之间传递数据的方式？
* 采用接口回调方式传递，采用evenbus传递数据，在fragment中写一个setdata方法，调用方法传递数据。
* Activity 怎么和Service 绑定？
* 通过bindservice方法进行绑定，绑定成功之后，service会将代理类用回调的方式传递给serviceconnection，这样就可以获取到service提供的对象了。
* 怎么在Activity 中启动自己对应的Service？
* 使用startservice,生命周期为oncreat，onstartcommand，关闭的时候，ondestory
* 使用onbind启动，oncreat，onbind，关闭的时候，onunbind ondestory
* service和activity怎么进行数据交互？
* Service的开启方式
* 请描述一下Service 的生命周期
* 谈谈你对ContentProvider的理解
* 说说ContentProvider、ContentResolver、ContentObserver 之间的关系
* 请描述一下广播BroadcastReceiver的理解
* 广播的分类
* 广播使用的方式和场景
* 在manifest 和代码中如何注册和使用BroadcastReceiver?
* 本地广播和全局广播有什么差别？
* BroadcastReceiver，LocalBroadcastReceiver 区别
* AlertDialog,popupWindow,Activity区别
* Application 和 Activity 的 Context 对象的区别
* Android属性动画特性
* 如何导入外部数据库?
* LinearLayout、RelativeLayout、FrameLayout的特性及对比，并介绍使用场景。
* 谈谈对接口与回调的理解
* 回调的原理
* 写一个回调demo
* 介绍下SurfView
* 不会阻碍主线程，并且在底层实现了双缓存机制，他能够支持openGL开发，执行的效率比较高，允许其他线程更新视图对象。
* RecycleView的使用
* 序列化的作用，以及Android两种序列化的区别
* 差值器
* 估值器
* Android中数据存储方式
* Sp存储
* File存储
* Sqlite存储
* 网络存储
* Content provider存储

##### （二）Android源码相关分析

* Android动画框架实现原理

通过parentView不断的调整childView的位置来实现的，比如说在parentview的dispatchdraw方法中发现了childView有一个动画，通过调整画布的traslate方法，告诉起始位置，之后不断的调用invalid来画帧，导致不断的调用dispatch函数，产生了后续的帧，直到动画播放完成。

* Android各个版本API的区别
* Requestlayout，onlayout，onDraw，DrawChild区别与联系

requestLayout会导致调用measure和layout过程，onlayout是对子视图重新布局，ondraw是绘制视图本身，drawchild重新回调每个子视图的draw方法。

* invalidate和postInvalidate的区别及使用

Invalidate必须是在主线程UI线程中进行刷新页面的，postInvalidate是在工作者线程中被调用的。

* Activity-Window-View三者的差别

Activity调用attach创建一个window，子类是phoneWindow，调用setcontentView实际上是调用getwindow（）.setcontentview（）

Window是存放view的容器

* 谈谈对Volley的理解
* 如何优化自定义View

删除重复代码，避免gc，初始化或者动画间隙分配内存，不要在动画过程中分配内存，容易导致gc，通过GPU加速

* 低版本SDK如何实现高版本api？

可以去support包中找相应的方法。

* 描述一次网络请求的流程

域名解析然后发起TCP三次握手，建立TCP之后发起http请求，服务器响应http请求，浏览器得到html代码，浏览器解析html代码，呈现给用户。

* HttpUrlConnection 和 okhttp关系

在android 4.4之后httpUrlConnection的实现是通过调用okhttp来完成的。通过httphandler这个桥梁，以及okhttpcilent，httpengine中增加一些方法来实现的。

* Bitmap对象的理解

在API10之前，像素数据存储在native堆上的，而bitmap对象存储在dvm上，所以需要销毁的时候得调用bitmap.recyle（）方法，在API10以后，都存储在dvm上，所以只需要保证在dvm回收之前，不被任何gc强引用回收掉这部分内存就行了。

* looper架构

Handler中有一个成员就是looper，looper内部存在一个messageQueue，looper.looper方法是一个死循环，会不断的从消息队列中取消息，然后调用hanler实例化中的方法，来执行操作。一般有post（runnable），sendmessage（）等方法来实现。

* ActivityThread，AMS，WMS的工作原理
* 自定义View如何考虑机型适配
* 自定义View的事件
* AstncTask+HttpClient 与 AsyncHttpClient有什么区别？
* LaunchMode应用场景
* AsyncTask 如何使用?
* SpareArray原理
* 请介绍下ContentProvider 是如何实现数据共享的？
* AndroidService与Activity之间通信的几种方式
* IntentService原理及作用是什么？
* 说说Activity、Intent、Service 是什么关系
* ApplicationContext和ActivityContext的区别
* SP是进程同步的吗?有什么方法做到同步？
* 谈谈多线程在Android中的使用
* 进程和 Application 的生命周期
* 封装View的时候怎么知道view的大小
* RecycleView原理
* AndroidManifest的作用与理解

##### （三）常见的一些原理性问题

* Handler机制和底层实现
* Handler、Thread和HandlerThread的差别
* handler发消息给子线程，looper怎么启动？
* 关于Handler，在任何地方new Handler 都是什么线程下?
* ThreadLocal原理，实现及如何保证Local属性？
* 请解释下在单线程模型中Message、Handler、Message Queue、Looper之间的关系
* 请描述一下View事件传递分发机制
* Touch事件传递流程
* 事件分发中的onTouch 和onTouchEvent 有什么区别，又该如何使用？
* View和ViewGroup分别有哪些事件分发相关的回调方法
* View刷新机制
* View绘制流程
* 自定义控件原理
* 自定义View如何提供获取View属性的接口？
* Android代码中实现WAP方式联网
* AsyncTask机制
* AsyncTask原理及不足
* 如何取消AsyncTask？
* 为什么不能在子线程更新UI？
* ANR产生的原因是什么？

主线程出现了耗时操作，大量的数据读写，数据库操作结束未关闭，硬件操作， service忙导致未响应。

* ANR定位和修正
* oom是什么？

内存溢出

* 什么情况导致oom？

Activity没有对栈进行管理，开启过多就会导致内存溢出，加载大图片，数据库操作结束未关闭，单例引起的内存泄露。Handler造成的内存泄露，无限循环的属性动画导致的内存泄露。遍历过度。

* 有什么解决方法可以避免OOM？

使用缓存技术，避免占用太多内存，对于不适用的资源要尽快进行释放。

* Oom 是否可以try catch？为什么？

不行，因为如果try 中的对象不是引起oom的对象，在catch中还是会出现oom。

* 内存泄漏是什么？

程序中分配的内存一直无法得到释放，导致占用太多，引起程序运行速度变慢，甚至崩溃。

* 什么情况导致内存泄漏？

Activity没有对栈进行管理，开启过多就会导致内存溢出，加载大图片，数据库操作结束未关闭，单例引起的内存泄露。Handler造成的内存泄露，无限循环的属性动画导致的内存泄露。遍历过度。

* 如何防止线程的内存泄漏？
* 内存泄露场的解决方法
* 内存泄漏和内存溢出区别？
* LruCache默认缓存大小

ActivityManager am = (ActivityManager)

getSystemService(Context.ACTIVITY\_SERVICE);

int availMenInBytes = am.getMemoryClass() \* 1024 \*1024;

LruCache bitmapCache =

new LruCache<String,Bitmap>(availMenInBytes /8);

* ContentProvider的权限管理(解答：读写分离，权限控制-精确到表级，URL控制)
* 如何通过广播拦截和abort一条短信？
* 广播是否可以请求网络？

可以开启一个新的线程来操作联网请求，但是一般用广播吊起service进行网络操作。

* 广播引起anr的时间限制是多少？

前台是10秒，后台是60秒，service前台20秒，后台200秒，输入是5秒，contentprovider是10秒。

* 计算一个view的嵌套层级

遍历，看当前的view的getparent方法得到的是不是null，为null跳出遍历。

* Activity栈
* Android线程有没有上限？
* 线程池有没有上限？
* ListView重用的是什么？
* Android为什么引入Parcelable？
* 有没有尝试简化Parcelable的使用？

##### （四）开发中常见的一些问题

* ListView 中图片错位的问题是如何产生的?
* 混合开发有了解吗？
* 知道哪些混合开发的方式？说出它们的优缺点和各自使用场景？（解答：比如:RN，weex，H5，小程序，WPA等。做Android的了解一些前端js等还是很有好处的)；
* 屏幕适配的处理技巧都有哪些?
* 服务器只提供数据接收接口，在多线程或多进程条件下，如何保证数据的有序到达？
* 动态布局的理解
* 怎么去除重复代码？
* 画出 Android 的大体架构图
* Recycleview和ListView的区别
* ListView图片加载错乱的原理和解决方案
* 动态权限适配方案，权限组的概念
* Android系统为什么会设计ContentProvider？
* 下拉状态栏是不是影响activity的生命周期
* 如果在onStop的时候做了网络请求，onResume的时候怎么恢复？
* Bitmap 使用时候注意什么？
* Bitmap的recycler()
* Android中开启摄像头的主要步骤
* ViewPager使用细节，如何设置成每次只初始化当前的Fragment，其他的不初始化？
* 点击事件被拦截，但是想传到下面的View，如何操作？
* 微信主页面的实现方式
* 微信上消息小红点的原理
* CAS介绍（这是阿里巴巴的面试题，我不是很了解，可以参考博客: [CAS简介](https://link.jianshu.com?t=http://blog.csdn.net/jly4758/article/details/46673835" \t "_blank)）

1. **高端技术面试部分**

**这里讲的是大公司需要用到的一些高端Android技术，这里专门整理了一个文档，希望大家都可以看看。这些题目有点技术含量，需要好点时间去研究一下的。**

##### （一）图片

* 图片库对比
* 图片库的源码分析
* 图片框架缓存实现
* LRUCache原理
* 图片加载原理
* 自己去实现图片库，怎么做？
* Glide源码解析
* Glide使用什么缓存？
* Glide内存缓存如何控制大小？

##### （二）网络和安全机制

* 网络框架对比和源码分析
* 自己去设计网络请求框架，怎么做？
* okhttp源码
* 网络请求缓存处理，okhttp如何处理网络缓存的
* 从网络加载一个10M的图片，说下注意事项
* TCP的3次握手和四次挥手
* TCP与UDP的区别
* TCP与UDP的应用
* HTTP协议
* HTTP1.0与2.0的区别
* HTTP报文结构
* HTTP与HTTPS的区别以及如何实现安全性
* 如何验证证书的合法性?
* https中哪里用了对称加密，哪里用了非对称加密，对加密算法（如RSA）等是否有了解?
* client如何确定自己发送的消息被server收到?
* 谈谈你对WebSocket的理解
* WebSocket与socket的区别
* 谈谈你对安卓签名的理解。
* 请解释安卓为啥要加签名机制?
* 视频加密传输
* App 是如何沙箱化，为什么要这么做？
* 权限管理系统（底层的权限是如何进行 grant 的）？

##### （三）数据库

* sqlite升级，增加字段的语句
* 数据库框架对比和源码分析
* 数据库的优化
* 数据库数据迁移问题

##### （四）算法

* 排序算法有哪些？
* 最快的排序算法是哪个？
* 手写一个冒泡排序
* 手写快速排序代码
* 快速排序的过程、时间复杂度、空间复杂度
* 手写堆排序
* 堆排序过程、时间复杂度及空间复杂度
* 写出你所知道的排序算法及时空复杂度，稳定性
* 二叉树给出根节点和目标节点，找出从根节点到目标节点的路径
* 给阿里2万多名员工按年龄排序应该选择哪个算法？
* GC算法(各种算法的优缺点以及应用场景)
* 蚁群算法与蒙特卡洛算法
* 子串包含问题(KMP 算法)写代码实现
* 一个无序，不重复数组，输出N个元素，使得N个元素的和相加为M，给出时间复杂度、空间复杂度。手写算法
* 万亿级别的两个URL文件A和B，如何求出A和B的差集C(提示：Bit映射->hash分组->多文件读写效率->磁盘寻址以及应用层面对寻址的优化)
* 百度POI中如何试下查找最近的商家功能(提示：坐标镜像+R树)。
* 两个不重复的数组集合中，求共同的元素。
* 两个不重复的数组集合中，这两个集合都是海量数据，内存中放不下，怎么求共同的元素？
* 一个文件中有100万个整数，由空格分开，在程序中判断用户输入的整数是否在此文件中。说出最优的方法
* 一张Bitmap所占内存以及内存占用的计算
* 2000万个整数，找出第五十大的数字？
* 烧一根不均匀的绳，从头烧到尾总共需要1个小时。现在有若干条材质相同的绳子，问如何用烧绳的方法来计时一个小时十五分钟呢？
* 求1000以内的水仙花数以及40亿以内的水仙花数
* 5枚硬币，2正3反如何划分为两堆然后通过翻转让两堆中正面向上的硬8币和反面向上的硬币个数相同
* 时针走一圈，时针分针重合几次

11次

* N\*N的方格纸,里面有多少个正方形

1+2平方+3平方.....n平方个正方形

* x个苹果，一天只能吃一个、两个、或者三个，问多少天可以吃完？

##### （五）插件化、模块化、组件化、热修复、增量更新、Gradle

* 对热修复和插件化的理解
* 插件化原理分析
* 模块化实现（好处，原因）
* 热修复,插件化
* 项目组件化的理解
* 描述清点击 Android Studio 的 build 按钮后发生了什么

##### （六）架构设计和设计模式

* 谈谈你对Android设计模式的理解
* MVC MVP MVVM原理和区别
* 你所知道的设计模式有哪些？
* 项目中常用的设计模式
* 手写生产者/消费者模式
* 写出观察者模式的代码
* 适配器模式，装饰者模式，外观模式的异同？
* 用到的一些开源框架，介绍一个看过源码的，内部实现过程。
* 谈谈对RxJava的理解
* RxJava的功能与原理实现
* RxJava的作用，与平时使用的异步操作来比的优缺点
* 说说EventBus作用，实现方式，代替EventBus的方式

Eventbus采用的是观察者模式，是一款发布者订阅者消息总线，将事件放到队列中，用于管理和分发，保证各部分高效的通信，模块解耦。

* 从0设计一款App整体架构，如何去做？
* 说一款你认为当前比较火的应用并设计(比如：直播APP，P2P金融，小视频等)
* 谈谈对java状态机理解
* Fragment如果在Adapter中使用应该如何解耦？
* Binder机制及底层实现
* 对于应用更新这块是如何做的？(解答：灰度，强制更新，分区域更新)？
* 实现一个Json解析器(可以通过正则提高速度)
* 统计启动时长,标准

##### （七）性能优化

* 如何对Android 应用进行性能分析以及优化?
* ddms 和 traceView
* 性能优化如何分析systrace？
* 用IDE如何分析内存泄漏？
* Java多线程引发的性能问题，怎么解决？
* 启动页白屏及黑屏解决？
* 启动太慢怎么解决？
* 怎么保证应用启动不卡顿？
* App启动崩溃异常捕捉
* 自定义View注意事项
* 现在下载速度很慢,试从网络协议的角度分析原因,并优化(提示：网络的5层都可以涉及)。
* Https请求慢的解决办法（提示：DNS，携带数据，直接访问IP）
* 如何保持应用的稳定性
* RecyclerView和ListView的性能对比
* ListView的优化
* RecycleView优化
* View渲染
* Bitmap如何处理大图，如一张30M的大图，如何预防OOM
* java中的四种引用的区别以及使用场景
* 强引用置为null，会不会被回收？

##### （八）NDK、jni、Binder、AIDL、进程通信有关

* 请介绍一下NDK
* 什么是NDK库?
* jni用过吗？
* 如何在jni中注册native函数，有几种注册方式?
* Java如何调用c、c++语言？
* jni如何调用java层代码？
* 进程间通信的方式？
* Binder机制
* 简述IPC？
* 什么是AIDL？
* AIDL解决了什么问题？
* AIDL如何使用？
* Android 上的 Inter-Process-Communication 跨进程通信时如何工作的？
* 多进程场景遇见过么？
* Android进程分类？
* 进程和 Application 的生命周期？
* 进程调度
* 谈谈对进程共享和线程安全的认识
* 谈谈对多进程开发的理解以及多进程应用场景
* 什么是协程？

##### （九）framework层、ROM定制、Ubuntu、Linux之类的问题

* java虚拟机的特性
* 谈谈对jvm的理解
* JVM内存区域，开线程影响哪块内存
* 对Dalvik、ART虚拟机有什么了解？
* Art和Dalvik对比
* 虚拟机原理，如何自己设计一个虚拟机(内存管理，类加载，双亲委派)
* 谈谈你对双亲委派模型理解
* JVM内存模型，内存区域
* 类加载机制
* 谈谈对ClassLoader(类加载器)的理解
* 谈谈对动态加载（OSGI）的理解
* 内存对象的循环引用及避免
* 内存回收机制、GC回收策略、GC原理时机以及GC对象
* 垃圾回收机制与调用System.gc()区别
* Ubuntu编译安卓系统
* 系统启动流程是什么？（提示：Zygote进程 –> SystemServer进程 –> 各种系统服务 –> 应用进程）
* 大体说清一个应用程序安装到手机上时发生了什么
* 简述Activity启动全部过程
* App启动流程，从点击桌面开始
* 逻辑地址与物理地址，为什么使用逻辑地址？
* Android为每个应用程序分配的内存大小是多少？
* Android中进程内存的分配，能不能自己分配定额内存？
* 进程保活的方式
* 如何保证一个后台服务不被杀死？（相同问题：如何保证service在后台不被kill？）比较省电的方式是什么？
* App中唤醒其他进程的实现方式