#### 主要分为以下几部分：

（1）java面试题  
（2）Android面试题  
（3）高端技术面试题  
（4）非技术性问题&HR问题汇总

## 一、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

详解：<https://www.cnblogs.com/kexianting/p/8508207.html>

个人归纳：1.“==” 可用于基本数据类型及引用类型（类、接口、数组）之间的比较。 a.用于基本数据类型(byte,short,char,int,long,float,double,boolean)之间比较时，比较的是它们的值是否相等； b.用于引用类型之间比较时，比较的是它们在内存中的存放地址是否一样。所以，除非是指向同一个new出来的对象，否则比较后的结果都是false。对象是放在堆中的，栈中存放的是对象的引用（地址）。由此可见'=='是对栈中的值进行比较的。如果要比较堆中对象的内容是否相同，那么就要重写equals方法了。

1. “equals”:a.默认情况（没有覆盖equals方法）下equals方法都是调用Object类的equals方法，而Object的equals方法主要用于判断对象的内存地址引用是不是同一个地址（是不是同一个对象）；b.要是类中重写了equals方法，那就要具体问题具体分析了，一般都是通过对象的内容是否相等来判断对象是否相等，如String类中equals的重写。
2. “hashCode”:哈希算法也称为散列算法，是将数据依特定算法直接指定到一个地址上。初学者可以这样理解，hashCode方法实际上返回的就是对象存储的物理地址（实际可能并不是）。所有散列函数都有如下一个基本特性：  
   1：如果a=b，则h(a) = h(b)。  
   2：如果a!=b，则h(a)与h(b)可能得到相同的散列值。
3. 在每个覆盖了equals方法的类中，也必须覆盖hashCode方法。如果不这样做的话，就会违反Object.hashCode的通用约定，从而导致该类无法结合所有基于散列的集合一起正常运作，这样的集合包括HashMap、HashSet和Hashtable。

* int、char、long各占多少字节数

Int占32位为4字节；

char占16位为2字节；

Long占64位为8字节；

* int与integer的区别

Int是基本数据类型，integer是引用类型

详解：<http://perfy315.iteye.com/blog/1480517>

* 谈谈对java多态的理解

* String、StringBuffer、StringBuilder区别

String是一个final class，即不可变类，

* 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
* 父类的静态方法能否被子类重写

父类的静态方法可以被继承，但不能被重写

* 进程和线程的区别
* final，finally，finalize的区别
* 序列化的方式
* Serializable 和Parcelable 的区别
* 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？
* 讲一下常见编码方式？
* utf-8编码中的中文占几个字节；int型几个字节？
* 静态代理和动态代理的区别，什么场景使用？
* Java的异常体系
* 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
* String为什么要设计成不可变的？
* Object类的equal和hashCode方法重写，为什么？

##### （三） 数据结构

* 常用数据结构简介
* 并发集合了解哪些？
* 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/)）
* 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
* HashMap数据结构？
* 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？
* 集合Set实现Hash怎么防止碰撞
* ArrayList和LinkedList的区别，以及应用场景
* 数组和链表的区别
* 二叉树的深度优先遍历和广度优先遍历的具体实现
* 堆的结构
* 堆和树的区别
* 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)？
* 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？
* 线程和进程的区别？
* 为什么要有线程，而不是仅仅用进程？
* run()和start()方法区别
* 如何控制某个方法允许并发访问线程的个数？
* 在Java中wait和seelp方法的不同；
* 谈谈wait/notify关键字的理解
* 什么导致线程阻塞？
* 线程如何关闭？
* 讲一下java中的同步的方法
* 数据一致性如何保证？
* 如何保证线程安全？
* 如何实现线程同步？
* 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？
* 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。

**学习的参考资料如下：**

Java 内存模型

* [java线程安全总结](https://link.jianshu.com?t=http://www.iteye.com/topic/806990)
* [深入理解java内存模型系列文章](https://link.jianshu.com?t=http://ifeve.com/java-memory-model-0/)

线程状态：

* [一张图让你看懂JAVA线程间的状态转换](https://link.jianshu.com?t=https://my.oschina.net/mingdongcheng/blog/139263)

锁：

* [锁机制：synchronized、Lock、Condition](https://link.jianshu.com?t=http://blog.csdn.net/vking_wang/article/details/9952063)
* [Java 中的锁](https://link.jianshu.com?t=http://wiki.jikexueyuan.com/project/java-concurrent/locks-in-java.html)

并发编程：

* [Java并发编程：Thread类的使用](https://link.jianshu.com?t=http://www.cnblogs.com/dolphin0520/p/3920357.html)
* [Java多线程编程总结](https://link.jianshu.com?t=http://blog.51cto.com/lavasoft/27069)
* [Java并发编程的总结与思考](https://www.jianshu.com/p/053943a425c3)
* [Java并发编程实战-----synchronized](https://link.jianshu.com?t=http://www.cnblogs.com/chenssy/p/4701027.html)
* [深入分析ConcurrentHashMap](https://link.jianshu.com?t=http://www.infoq.com/cn/articles/ConcurrentHashMap%23)

## 二、Android面试题

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
* 四大组件的生命周期和简单用法
* Activity之间的通信方式
* 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
* Activity与Fragment之间生命周期比较
* Activity上有Dialog的时候按Home键时的生命周期
* 两个Activity 之间跳转时必然会执行的是哪几个方法？

**一般情况下比如说有两个activity,分别叫A,B。**

**当在A 里面激活B 组件的时候, A会调用onPause()方法,然后B调用onCreate() ,onStart(), onResume()。**

**这个时候B覆盖了A的窗体, A会调用onStop()方法。**

**如果B是个透明的窗口,或者是对话框的样式, 就不会调用A的onStop()方法。**

**如果B已经存在于Activity栈中，B就不会调用onCreate()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。

前->后->前：onPause->onStop->onRestart->onResume；弹出Dialog，生命周期没变化

* Activity的四种启动模式对比

Standard,singleTop,singleInstance, singleTask；对比略

* Activity状态保存于恢复
* 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？

startActivityForResult是Activity的方法，当你希望得到目标跳转界面的回调值时使用。

* 如何实现Fragment的滑动？

ViewPager控件

* fragment之间传递数据的方式？

setArguments(bundle)；接口回调；EventBus等等

* Activity 怎么和Service 绑定？

通过bindService方法

* 怎么在Activity 中启动自己对应的Service？
* service和activity怎么进行数据交互？

先绑定service，然后通过binder进行交互

* 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

Content Provider主要用于在不同的应用程序之间（因而是完成IPC的一种）实现数据共享的功能，它提供了一套完整的机制，允许一个程序访问另一个程序中的数据，同时还能保证被访数据的安全性。目前，使用ContentProvider是 Android 实现跨程序共享数据的标准方式,因为它提供了统一的数据访问方式。

* 说说ContentProvider、ContentResolver、ContentObserver 之间的关系

第一个是内容提供者,可以对系统的数据库文件提供增删改查的方法,第二个是内容解析者,内容解析者可以通过特定的uri对内容提供者进行访问,调用里面的增删改查的方法,最后一个是内容监听者,可以监听数据库内容的改变,当数据库内容发生改变的时候,内容监听者会受到消息,做相应的处理.

* 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

AlertDialog是Dialog的子类，用于显示一些按钮、文字等等，也可以添加自定义的View；PopupWindow可以显示任意的View，它是一个悬浮的容器出现在当前activity上面；Activity为android四大组件之一，主要用于与用户交互。

* Application 和 Activity 的 Context 对象的区别

Application的Context是一个全局静态变量，SDK的说明是只有当你引用的这个context的生命周期超过了当前activity的生命周期，而和整个应用的生命周期挂钩时，才去使用这个application的context。

一种推荐的用法就是应用程序注册广播broadcast时，使用这个context。

引用application context并不会导致内存泄漏。引用activity的context才会导致内存泄漏，当你的引用对象的生命周期超过了当前activity的生命周期时，系统回收你的activity时发现还有一个context的强引用，而不去销毁这个activity，导致内存泄漏，在开发过程中，需要非常重视这点。

* Android属性动画特性

作用对象进行了扩展：不只是View对象，甚至没对象也可以

动画效果：不只是4种基本变换，还有其他动画效果

作用领域：API11后引入的

不止是视觉效果改变，事件响应效果也会跟着改变

* 如何导入外部数据库?

先把外部数据库文件拷贝到工程的assets目录下，然后通过编写代码读该目录下的数据库文件，然后写到data/data/包名/databases/目录下。

* LinearLayout、RelativeLayout、FrameLayout的特性及对比，并介绍使用场景。

[三种布局性能分析](https://www.cnblogs.com/chenlong-50954265/p/5942182.html)。尽量减少层级嵌套，用ViewStub、include、Merge标签等等

* 谈谈对接口与回调的理解

接口是一个抽象类型，是抽象方法的集合。主要做为一个规范，实现接口的类都需要实现接口里面的方法。回调的作用通常是不想或者不能一直等待一个结果的返回而采取的让结果出来后，主动通知调用者的行为。

* 回调的原理
* 写一个回调demo
* 介绍下SurfView

总的归纳起来SurfaceView和View不同之处有：

1. SurfaceView允许其他线程更新视图对象(执行绘制方法)而View不允许这么做，它只允许UI线程更新视图对象。

2. SurfaceView是放在其他最底层的视图层次中，所有其他视图层都在它上面，所以在它之上可以添加一些层，而且它不能是透明的。

3. 它执行动画的效率比View高，而且你可以控制帧数。

4. SurfaceView在绘图时使用l了双缓冲机制，而View没有。

* RecycleView的使用
* 序列化的作用，以及Android两种序列化的区别

[比较Serializable和Parcelable](https://blog.csdn.net/wangchunlei123/article/details/51345130)

* 插值器

Interpolatort通常在动画中使用，主要来改变动画变化率。例如，匀速、加速等等

* 估值器

动画过程中TypeEvaluator（估值器）的作用：当我们ValueAnimator.ofObject()函数来做动画效果的时候就会用到估值器了，估值器说白了就是用来确定在动画过程中每时每刻动画的具体值。换句话说就是确定ValueAnimator.getAnimatedValue()返回的具体对象类型

* Android中数据存储方式

[五种存储方式](https://blog.csdn.net/amazing7/article/details/51437435)

##### （二）Android源码相关分析

* Android动画框架实现原理

[android三种动画实现原理及使用](https://blog.csdn.net/zoky_ze/article/details/54934361)

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

requestLayout：要求父布局对其进行重新测量、布局、绘制。所以只要调用了requestlayout, 那么measure和onMeasure，以及layout，onlayout，draw onDraw都会被调用。

invalidate：View本身调用迫使view重画。 刷新视图，相当于View.OnDraw()方法。没有请求view树重新测量（measure）、布局（layout）。

invalidate只回调onDraw，一般用于刷新当前view，使当前view进行重绘，不会进行测量及布局;  
requestLayout则顺序回调onMeasure->onLayout->onDraw。需要父布局对其进行重新测量、布局、绘制时使用;  
requestLayout可以替代invalidate，反过来则不行。

View中，onLayout及onDraw都是空方法，需要子类来具体实现业务逻辑。例如，LinearLayout等

ViewGroup中有drawChild方法，调用的是里面子View的draw方法。即让子View自己绘制自己。

* invalidate和postInvalidate的区别及使用

前者在主线程调用。后者在子线程中调用。都是用于刷新当前view。

* Activity-Window-View三者的差别

关系：Activity是Android四大组件之一，主要用来与用户交互。Window则是由Activity创建的窗口，它负责管理View的增、删、改操作。三者结合就组成了与用户交互的界面。三者关系可总结为：Activity创建了Window，Window对View即视图控件进行管理，最终展现出能与用户交互的界面。

原理分析：Window是一个抽象类，具体实现类为PhoneWindow。它的实现在Activity的attach方法中，该方法里面还实例化了WindowManager。WM是一个接口，它继承自ViewManager，它的实现类为WindowManagerImpl。而VM只定义了三个方法，分别是addView、updateViewLayout及removeView。由此可以，WM实际是上对View进行管理操作。我们平时写的布局，是通过Activity的setContentView来展示的，而setContentView实际上是通过PhoneWindow的setContentView来实现的。PhoneWindow通过setContentView创建了顶级View，即DecorView，然后通过inflate方法把我们的布局加入到DecorView中。如此一来，我们就能看到我们所写的界面的。

Activity的很多操作实际上都是交给Window来处理的。例如，setContentView，findViewById，getLayoutInflater等等。

* 谈谈对Volley的理解

volley：适合网络通信频繁操作，并能同时实现多个网络通信、扩展性强、通过接口配置之类的优点。

[Volley](https://blog.csdn.net/qq_26358311/article/details/79611461)

* 如何优化自定义View

[自定义View调优](https://blog.csdn.net/whb20081815/article/details/74474736)

不要在onDraw方法内创建对象，不要滥用invalidate或者requestLayout，硬件加速等等。

* 低版本SDK如何实现高版本api？

导入support包，运用@TargetApi标签，根据不同api用不同方法实现业务逻辑。

* 描述一次网络请求的流程

[Android网络请求心路历程](https://www.jianshu.com/p/3141d4e46240)

* HttpUrlConnection 和 okhttp关系

从Android4.4开始HttpURLConnection的底层实现(http/https)采用的是okHttp。

* Bitmap对象的理解

[Bitmap的深入理解](http://www.mamicode.com/info-detail-1440626.html)

* [looper架构](https://www.aliyun.com/jiaocheng/1560.html)
* ActivityThread，AMS，WMS的工作原理

[Launcher启动过程详解](https://blog.csdn.net/u012481172/article/details/51305247)

[WMS与AMS](https://www.jianshu.com/p/47eca41428d6)

[Android中APP、AMS、WMS的Binder IPC](https://blog.csdn.net/qq526459753/article/details/51034625)

* 自定义View如何考虑机型适配

这里要考虑的是屏幕的问题：

合理使用warp\_content，match\_parent.

尽可能的是使用RelativeLayout

针对不同的机型，使用不同的布局文件放在对应的目录下，android会自动匹配。

尽量使用点9图片。

使用与密度无关的像素单位dp，sp

引入android的百分比布局。

切图的时候切大分辨率的图，应用到布局当中。在小分辨率的手机上也会有很好的显示效果。

* 自定义View的事件
* AstncTask+HttpClient 与 AsyncHttpClient有什么区别？
* [LaunchMode应用场景](https://blog.csdn.net/xiaodongrush/article/details/28597855)
* [AsyncTask 如何使用](https://blog.csdn.net/iispring/article/details/50639090)?
* [SparseArray原理](https://www.cnblogs.com/RGogoing/p/5095168.html)
* [请介绍下ContentProvider 是如何实现数据共享的](https://www.jianshu.com/p/b8c2a1cee0a6)？
* [AndroidService与Activity之间通信的几种方式](https://blog.csdn.net/xiaanming/article/details/9750689)
* [IntentService原理及作用是什么](https://www.cnblogs.com/SoulCode/p/5468049.html)？
* 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](https://www.jianshu.com/p/bdebf741221e)？
* 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](https://www.cnblogs.com/-OYK/archive/2011/05/31/2064818.html)

##### （三）常见的一些原理性问题

* [Handler机制和底层实现](https://blog.csdn.net/itheimach/article/details/51170857)
* Handler、Thread和HandlerThread的差别

Handler：负责消息发送及处理；Thread：线程，负责处理耗时任务；HandlerThread：线程的子类，也是线程。不过，它无法并发执行，只能一个任务一个任务地执行，没执行到的任务只能在后面排队。

* 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](https://blog.csdn.net/fenganit/article/details/53750265)？

onTouch是View的OnTouchListener接口里面的方法，也是它唯一的方法。当一个控件通过setOnTouchListener(new OnTouchListener)来注册触摸事件监听时，当控件被触摸后，便会触发onTouch方法。如果onTouch返回false，则事件继续传递到onTouchEvent方法；如果onTouch返回true，则事件不再传递，onTouchEvent不会被触发。

* View和ViewGroup分别有哪些事件分发相关的回调方法

View：onTouchEvent，dispatchTouchEvent；

ViewGroup：onTouchEvent，dispatchTouchEvent，onInterceptTouchEvent

* View刷新机制

Invalidate，requestLayout

* View绘制流程

measure过程：对于View，measure实现其宽高的测量，对于ViewGroup，measure不仅要完成自身的测量还要完成子View的测量。

layout过程：用于确定子元素的位置。

draw过程：将View绘制到屏幕上。

* 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？

安卓系统的UI控件并非线程安全的，如果多线程并发访问同一个UI控件时会引发UI控件状态的混乱。

* [ANR产生的原因是什么？](https://www.colabug.com/2452481.html)
* [ANR定位和修正](https://www.colabug.com/2452481.html)
* [oom是什么？](https://www.colabug.com/2452481.html)
* [什么情况导致oom？](https://www.colabug.com/2452481.html)
* [有什么解决方法可以避免OOM？](https://www.colabug.com/2452481.html)
* [Oom 是否可以try catch？为什么？](https://www.colabug.com/2452481.html)
* [内存泄漏是什么？](https://www.colabug.com/2452481.html)
* [什么情况导致内存泄漏？](https://www.colabug.com/2452481.html)
* [如何防止线程的内存泄漏？](https://www.colabug.com/2452481.html)
* [内存泄露场的解决方法](https://www.colabug.com/2452481.html)
* [内存泄漏和内存溢出区别？](https://www.colabug.com/2452481.html)
* [LruCache默认缓存大小](https://www.colabug.com/2452481.html)
* [ContentProvider的权限管理(解答：读写分离，权限控制-精确到表级，URL控制)](https://www.colabug.com/2452481.html)
* [如何通过广播拦截和abort一条短信？](https://www.colabug.com/2452481.html)
* [广播是否可以请求网络？](https://www.colabug.com/2452481.html)
* [广播引起anr的时间限制是多少？](https://www.colabug.com/2452481.html)
* [计算一个view的嵌套层级](https://www.colabug.com/2452481.html)
* [Activity栈](https://www.colabug.com/2452481.html)
* [Android线程有没有上限？](https://www.colabug.com/2452481.html)
* [线程池有没有上限？](https://www.colabug.com/2452481.html)
* [ListView重用的是什么？](https://www.colabug.com/2452481.html)
* [Android为什么引入Parcelable？](https://www.colabug.com/2452481.html)
* [有没有尝试简化Parcelable的使用？](https://www.colabug.com/2452481.html)

##### （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?

由于ListView的缓存机制导致的。缓存机制缓存了视图的同时，也会把数据一同缓存起来。所以，当该视图被复用的时候，如果不对数据重新赋值，就会出现错位的问题。

* 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](https://blog.csdn.net/aptentity/article/details/68942962)?
* 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](https://blog.csdn.net/mottled233/article/details/52171904)
* [怎么去除重复代码](https://blog.csdn.net/u011494050/article/details/40593605)？
* [画出 Android 的大体架构图](https://juejin.im/post/5a93a1f1f265da4e8b300059)
* Recycleview和ListView的区别

两者都使用了适配器（Adapter）模式和观察者模式。

区别：1.ListView需要自己创建一个ViewHolder、手动判断是否有缓存View等，而RecycleView则不需要开发者去干这些事，这些都由RecycleView内部实现。它定义了ViewHolder类型标准，封装了View缓存判断逻辑。

2.ListView只能在竖直方向滑动，而Recycleview可以通过LayoutManager将RecycleView显示为不同的样式，例如ListView、GridView形式、瀑布流形式。

* ListView图片加载错乱的原理和解决方案

列表缓存机制：obtainView函数定义了列表控件的Item View的复用逻辑，首先会从RecyclerBin中获取一个缓存的View，如果有缓存则将这个缓存的View传递到Adapter的getView第二个参数中，这也是我们对Adapter的最常见的优化方式，即判断getView的convertView是否为空，如果为空则从xml中创建视图，否则使用缓存的View。这样避免了每次都从xml加载布局的消耗，能够显著提升ListView等列表控件的效率。通过上面的缓存机制，ListView实际上只有填满一屏所需的Item View存在内存中。第一次加载整屏数据项时，getView的第二个参数convertView都为空，此时每次都需要从xml中加载布局文件，填充数据之后返回给ListView。当整屏的数据项加载完成之后，用户向下滚动屏幕，此时item1滚出屏幕，并且一个新的项目从屏幕底端上来时，ListView再请求一个视图，此时item1被缓存起来，在下一项数据加载时传递给getView的第二个参数convertView，因此，convertView此时不是空值，它的值是item1。此时只需设定新的数据返回给convertView，这样就避免了每次都从xml加载、初始化视图，减少了时间、性能上的消耗。

图片加载错乱的原理：由于视图被缓存了，视图中的数据也会被缓存。

解决：因此，你需要在每次获取到了Item View时对每个数据项重新赋值，否则会出现数据错误的现象。

* 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

不影响。

* 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](https://www.cnblogs.com/sxzheng/p/5632977.html)？
* [Bitmap的recycler()](https://blog.csdn.net/zhoumushui/article/details/78712649)

这是一个高级调用，通常不应该调用，因为通常情况下如果此bitmap没有被引用，垃圾回收进程会释放占用的内存空间。如果一定要用，请确保该bitmap已经不会再用到了才用。

* 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？](https://www.2cto.com/kf/201606/520983.html)

懒加载

* 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](https://blog.csdn.net/linxi7/article/details/51849667)
* 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)）

## 三、高端技术面试题

**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](https://www.jianshu.com/p/c6a2628a1063)
* 图片加载原理
* 自己去实现图片库，怎么做？
* [Glide源码解析](http://www.jcodecraeer.com/a/anzhuokaifa/androidkaifa/2018/0403/9555.html)
* [Glide使用什么缓存？](https://www.jianshu.com/p/325bd2f56ca7)

磁盘缓存+内存缓存。

磁盘缓存：日志文件（1份）+ 图片缓存二进制文件（n份）

* Glide内存缓存如何控制大小？

通过自定义一个类继承自AppGlideModule，然后重写applyOptions方法，通过builder.setMemoryCache(new LruResourceCache(CACHE\_DEFAULTSIZE));设置内存缓存大小。

##### （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](https://www.jianshu.com/p/6b2e35fdaf2c?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)
* [TCP与UDP的区别](https://zhidao.baidu.com/question/21979549.html)
* [TCP与UDP的应用](https://blog.csdn.net/u013777351/article/details/49226101)
* [HTTP协议](https://baike.baidu.com/item/http/243074?fr=aladdin&fromid=1276942&fromtitle=HTTP协议)
* [HTTP1.0与2.0的区别](https://blog.csdn.net/u012657197/article/details/77877840)
* [HTTP报文结构](https://blog.csdn.net/a19881029/article/details/14002273)
* [HTTP与HTTPS的区别以及如何实现安全性](https://blog.csdn.net/aspenstars/article/details/78043535)
* [如何验证证书的合法性?](https://www.cnblogs.com/wqhwe/p/5407468.html)

<https://www.cnblogs.com/funny11/p/6978908.html>

* [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?](https://blog.csdn.net/a1232345/article/details/44594867)
* client如何确定自己发送的消息被server收到?
* [谈谈你对WebSocket的理解](https://www.cnblogs.com/fuqiang88/p/5956363.html)
* [WebSocket与socket的区别](https://blog.csdn.net/wwd0501/article/details/54582912)
* [谈谈你对安卓签名的理解](https://www.cnblogs.com/wanqieddy/p/3556060.html)。
* 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](https://blog.csdn.net/ljheee/article/details/53191397)？
* 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
* N\*N的方格纸,里面有多少个正方形
* 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
* 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

## 四、非技术性问题&HR问题汇总

**这里整理的是一些与技术没有直接关系的面试题，但是能够考察你的综合水平，所以不要以为不是技术问题，就不看，往往有时候就是这样一些细节的题目被忽视，而错过了一次次面试机会。**

##### （一）非技术问题

* 介绍你做过的哪些项目
* 都使用过哪些框架、平台？
* 都使用过哪些自定义控件？
* 研究比较深入的领域有哪些？
* 对业内信息的关注渠道有哪些？
* 最近都读哪些书？
* 有没有什么开源项目？
* 自己最擅长的技术点，最感兴趣的技术领域和技术点
* 项目中用了哪些开源库，如何避免因为引入开源库而导致的安全性和稳定性问题
* 实习过程中做了什么，有什么产出？

##### （二）HR提出的面试问题

* 您在前一家公司的离职原因是什么？
* 讲一件你印象最深的一件事情
* 介绍一个你影响最深的项目
* 介绍你最热爱最擅长的专业领域
* 公司实习最大的收获是什么？
* 与上级意见不一致时，你将怎么办？
* 自己的优点和缺点是什么？并举例说明？
* 你的学习方法是什么样的？实习过程中如何学习？实习项目中遇到的最大困难是什么以及如何解决的？
* 说一件最能证明你能力的事情
* 针对你你申请的这个职位，你认为你还欠缺什么
* 如果通过这次面试我们单位录用了你，但工作一段时间却发现你根本不适合这个职位，你怎么办？
* 项目中遇到最大的困难是什么？如何解决的？
* 你的职业规划以及个人目标、未来发展路线及求职定位
* 如果你在这次面试中没有被录用，你怎么打算？
* 评价下自己，评价下自己的技术水平，个人代码量如何？
* 通过哪些渠道了解的招聘信息，其他同学都投了哪些公司？
* 业余都有哪些爱好？
* 你做过的哪件事最令自己感到骄傲？
* 假如你晚上要去送一个出国的同学去机场，可单位临时有事非你办不可，你怎么办？
* 就你申请的这个职位，你认为你还欠缺什么？
* 当前的offer状况；如果BATH都给了offer该如何选？
* 你对一份工作更看重哪些方面？平台，技术，氛围，城市，还是money？
* 理想薪资范围；杭州岗和北京岗选哪个？
* 理想中的工作环境是什么？
* 谈谈你对跳槽的看法
* 说说你对行业、技术发展趋势的看法
* 实习过程中周围同事/同学有哪些值得学习的地方？
* 家人对你的工作期望及自己的工作期望
* 如果你的工作出现失误，给本公司造成经济损失，你认为该怎么办？
* 若上司在公开会议上误会你了，该如何解决？
* 是否可以实习，可以实习多久？
* 在五年的时间内，你的职业规划
* 你看中公司的什么？或者公司的那些方面最吸引你？