1. 线程、多线程、线程同步、进程保活、threadlocal、多线程断点续传

1.1：Java多线程干货系列

<http://blog.csdn.net/u014695188/article/details/71424550>

1.2 Android线程总结

<https://www.cnblogs.com/whoislcj/p/5603277.html>

1.3 线程状态

<http://blog.csdn.net/lifei128/article/details/20363257>

1. 动态权限
2. 网络请求缓存处理，okhttp如何处理缓存，retrifit2

okhttp缓存：

<http://blog.csdn.net/briblue/article/details/52920531>

1. 图片加载、大图加载
2. Listview recyleview区别，图片加载错乱的原理和解决方案
3. http https的区别、https验证证书的合法性，即其中包含的对称加密和非对称加密、请求满，DNS,携带数据，直接访问IP
4. 加密(对称加密、非对称加密)
5. Lrucahe 缓存原理

<https://www.jianshu.com/p/b49a111147ee>

1. 视频加密传输
2. Classloader
3. 动态布局
4. 热修复，插件化
5. Hashmap、concurrenthashmap、sparearray、arraylist、linklist、list、set等集合类
6. Sp是否是进程同步，如何做到同步
7. Surfview
8. Broadcastreceive、LocalBroadcastreceive区别

LocalBroadcastreceive本地广播，用于应用内组件通信。使用handler机制更加高效，避免接收第三方应用发送的广播，synchronize线程安全

1. Bundle机制
2. Handler(消息机制)机制
3. 事件传递机制

结论1：事件一定是先到达父控件上。  
结论2：事件简单来说可以分为三种：Down事件、Move事件、Up事件，结合结论1可以得到，Down事件、Move事件Up事件均是最先到达父控件上。  
结论3：父控件和父类不是一回事，这两个概念初学者很容易混淆。

1. 多线程操作list崩溃
2. APP启动流程，源码着手
3. GC回收机制
4. Dalvik art jvm
5. 生命周期（activity、service、broadcatreciver、contentpovider、进程、线程、）
6. 4大组件
7. 数据库迁移
8. 设计模式(android中观察者模式和单例模式等)
9. 网络通信编程(TCP、UDP)WEBSOCKET socket

<http://blog.csdn.net/yuanyuanispeak/article/details/51729191>

1. 进程间通信
2. Java注解
3. Android相关优化（内存、网络、布局、电量、业务）
4. 常用插件原理eventbus retirfit2 rxjava2 okhttp
5. 内部类，静态内部类，匿名内部类等
6. 泛型
7. Activity栈
8. 自定义view过程、layoutmannager流程、viewgrounp流程
9. Service、activity数据交互
10. 状态栏下拉是否会影响activity，onstop的网络请求，onsume怎么恢复
11. 框架（MVC MVP MVVM）
12. 动画

**1.Activity的启动过程（不要回答生命周期）**   
<http://blog.csdn.net/luoshengyang/article/details/6689748>

**2.Activity的启动模式以及使用场景**   
（1）manifest设置，（2）startActivity flag   
<http://blog.csdn.net/CodeEmperor/article/details/50481726>   
此处延伸：栈(First In Last Out)与队列(First In First Out)的区别

**3.Service的两种启动方式**   
（1）startService()，（2）bindService()   
<http://www.jianshu.com/p/2fb6eb14fdec>

**4.Broadcast注册方式与区别**   
（1）静态注册(minifest)，（2）动态注册   
<http://www.jianshu.com/p/ea5e233d9f43>   
此处延伸：什么情况下用动态注册

**5.HttpClient与HttpUrlConnection的区别**   
<http://blog.csdn.net/guolin_blog/article/details/12452307>   
此处延伸：Volley里用的哪种请求方式（2.3前HttpClient，2.3后HttpUrlConnection）

**6.http与https的区别**   
<http://blog.csdn.net/whatday/article/details/38147103>   
此处延伸：https的实现原理

**7.http与https的区别**   
<http://blog.csdn.net/whatday/article/details/38147103>   
此处延伸：https的实现原理

**8.手写算法（选择冒泡必须要会）**   
<http://www.jianshu.com/p/ae97c3ceea8d>

**9.进程保活（不死进程）**   
<http://www.jianshu.com/p/63aafe3c12af>   
此处延伸：进程的优先级是什么（下面这篇文章，都有说）   
<https://segmentfault.com/a/1190000006251859>

**10.进程间通信的方式**   
（1）AIDL，（2）广播，（3）Messenger   
AIDL : <http://www.jianshu.com/p/ae97c3ceea8d>   
Messenger : <http://blog.csdn.net/lmj623565791/article/details/47017485>   
此处延伸：简述Binder ， <http://blog.csdn.net/luoshengyang/article/details/6618363/>

**11.加载大图** 。   
<http://blog.csdn.net/lmj623565791/article/details/49300989>

**12.三级缓存（各大图片框架都可以扯到这上面来）**   
（1）内存缓存，（2）本地缓存，（3）网络   
内存：<http://blog.csdn.net/guolin_blog/article/details/9526203>   
本地：<http://blog.csdn.net/guolin_blog/article/details/28863651>

**13.MVP框架**  
<http://blog.csdn.net/lmj623565791/article/details/46596109>   
此处延伸：手写mvp例子，与mvc之间的区别，mvp的优势

**14.讲解一下Context**   
<http://blog.csdn.net/lmj623565791/article/details/40481055>

**15.JNI**   
<http://www.jianshu.com/p/aba734d5b5cd>   
此处延伸：项目中使用JNI的地方，如：核心逻辑，密钥，加密逻辑

**16.java虚拟机和Dalvik虚拟机的区别**   
<http://www.jianshu.com/p/923aebd31b65>

**17.线程sleep和wait有什么区别**   
<http://blog.csdn.net/liuzhenwen/article/details/4202967>

**18.View，ViewGroup事件分发**   
<http://blog.csdn.net/guolin_blog/article/details/9097463>   
<http://blog.csdn.net/guolin_blog/article/details/9153747>

**19.保存Activity状态**   
onSaveInstanceState()   
<http://blog.csdn.net/yuzhiboyi/article/details/7677026>

**20.WebView与js交互（调用哪些API）**   
<http://blog.csdn.net/cappuccinolau/article/details/8262821/>

**21.内存泄露检测，内存性能优化**   
<http://blog.csdn.net/guolin_blog/article/details/42238627>   
这篇文章有四篇，很详细。   
此处延伸：   
（1）内存溢出（OOM）和内存泄露（对象无法被回收）的区别。   
（2）引起内存泄露的原因

**22.布局优化**   
<http://blog.csdn.net/guolin_blog/article/details/43376527>

**23.自定义view和动画**   
以下两个讲解都讲得很透彻，这部分面试官多数不会问很深，要么就给你一个效果让你讲原理。   
（1）<http://www.gcssloop.com/customview/CustomViewIndex>   
（2）<http://blog.csdn.net/yanbober/article/details/50577855>

**24.设计模式（单例，工厂，观察者。作用，使用场景）**   
一般说自己会的就ok，不要只记得名字就一轮嘴说出来，不然有你好受。   
<http://blog.csdn.net/jason0539/article/details/23297037/>   
此处延伸：Double Check的写法被要求写出来。

**25.String，Stringbuffer，Stringbuilder 区别**   
<http://blog.csdn.net/kingzone_2008/article/details/9220691>

**26.开源框架，为什么使用，与别的有什么区别**   
这个问题基本必问。在自己简历上写什么框架，他就会问什么。   
如：Volley，面试官会问我Volley的实现原理，与okhttp和retrofit的区别。   
开源框架很多，我就选几个多数公司都会用的出来（框架都是针对业务和性能，所以不一定出名的框架就有人用）   
网络请求：Volley，okhttp，retrofit   
异步：RxJava，AsyncTask   
图片处理：Picasso，Glide   
消息传递：EventBus   
以上框架请自行查找，太多了就不贴出来了。

**27.RecyclerView**   
这个挺搞笑的。有另外一个同事也在找工作，面试官嫌他没用过RecyclerView直接pass掉。   
<http://blog.csdn.net/lmj623565791/article/details/45059587>