

1 java基础

1.1数据结构

- 1.1.1 数组 (★★★★)
- 1.1.2 集合 (★★★★)
- 1.1.3 链表 (★★★★)
- 1.1.4 Map (★★★★)
- 1.1.5 队列 (★★★★)

1.2.常用算法 (★★★★)

- 1.2.1 冒泡排序
- 1.2.2 简单选择排序
- 1.2.3 插入排序
- 1.2.4 快速排序
- 1.2.5 二分查找

1.3 Java反射机制, Java代理模式 (★★★★)

1.4 Java泛型 (★★★★)

1.5 java 线程相关 (★★★★)

- 1.5.1 Synchronized原理 (★★★★★)
- 1.5.2 Volatile实现原理 (★★★★★)
- 1.5.3 方法锁、对象锁、类锁的意义和区别 (★★★★★)
- 1.5.4 线程同步的方法: Synchronized、lock、reentrantLock分析 (★★★★)
- 1.5.5 java锁的种类: 公平锁、乐观锁、互斥锁、分段锁、偏向锁、自旋锁等 (★★)
- 1.5.6 ThreadLocal的原理和用法 (★★★★)
- 1.5.7 ThreadPool的用法和示例 (★★★★★)
- 1.5.8 wait()和sleep()的区别 (★★★★)
- 1.5.9 Thread的中断机制(interrupt) (★★★★★)
- 1.5.10 线程池的实现原理 (★★★★★)

2 java 高阶

- 2.1 Java虚拟机, Java运行, Java GC机制(可达性分析法, 引用计数法) (★★★★★)
- 2.2 Java对象的完整生命周期 (★★★★)
- 2.3 JVM内存模型 (★★★★)
- 2.4 进程间通信, 线程间通信 (★★)
- 2.5 JVM类加载机制 (★★★★)
- 2.6 Java引用类型 (★★★★★)
- 2.7 设计模式: 除常用设计模式之外, 特别的, 反射机制, 代理模式 (★★★★★)
- 2.8 HTTP协议和HTTPS协议 (★★★★★)
- 2.9 Socket协议, Socket实现长连接 (★★★★)
- 2.10 TCP和UDP协议 (★★★★)
- 2.11 HTTP协议中GET和POST的具体实现 (★★★★★)
- 2.12 序列化和反序列化 (★★★★)
- 2.13 数据库基础知识: 多表查询、索引、数据库事务 (★★★★)

3 Android 初级

- 3.1 生命周期 (★★★★★)
- 3.2 Activity 几种启动模式? 每一种模式分别使用什么场景? (★★★★★)
- 3.3 Service ,IntentService用法与区别? 他们分别如何与Activity 进行通信? (★★★★★)
- 3.4 Fragment 系列 (★★★★★)
- 3.5 ContentProvider实例详解 (★★★)

- 3.6 BroadcastReceiver使用总结 (★★★★★)
- 3.7 Android Handler 原理机制讲解 (★★★★★)
- 3.8 Binder机制, 共享内存实现原理 (★★)
- 3.9 Android 事件分发机制 (★★★★★)
- 3.10 Android 多线程的实现: Thread、HandlerThread、AsyncTask、IntentService、RxJava (★★★★★)
- 3.11 Activity启动流程, ActivityThread解析 (★★)
- 3.12 嵌套滑动原理 (★★★★☆)
- 3.13 RecyclerView与ListView(缓存原理, 区别联系, 优缺点) (★★★★☆)
- 3.14 View的绘制原理, 自定义View, 自定义ViewGroup (★★★★★)
- 3.15 View、SurfaceView 与 TextureView (★★★)
- 3.16 ViewPager的缓存实现 (★★★★★)
- 3.17 requestLayout, invalidate, postInvalidate区别与联系 (★★★★★)
- 3.18 AndroidP新特性 (★★★★★)
- 3.19 Android两种虚拟机 有那些区别 (★★★★★)
- 3.20 ADB常用命令 (★★★★★)
- 3.21 Asset目录与res目录的区别 (★★★★★)
- 3.22 主线程Looper.loop为什么不会造成死循环 (★★★★★)
- 3.23 Android 进程 和 线程 Process and Threads (★★★★★)
- 3.24 Android页面恢复 (★★★★★)
- 3.25 Anroid mipmap 和 drawable 的区别 (★★★★★)

4 Android 高级

- 4.1 AIDL 是什么? 如何用AIDL 实现进程间通信? (★★★)
- 4.2 自定义LayoutManager, RecyclerView中如何自定义LayoutManager (★★☆)
- 4.3 VLayout实现原理 (★★★★☆)
- 4.4 Glide (★★★★★)
- 4.5 Retrofit的实现与原理 (★★★★★)
- 4.6 OKHttp3 (★★★★★)
- 4.7 EventBus (★★★★★)
- 4.8 ButterKnife (★★★★★)
- 4.9 RxJava (★★★★★)
- 4.10 Dagger依赖注入 (★★★★★)
- 4.11 热修复 (★★★★★)
- 4.12 插件化开发 (★★★★★)
- 4.12 组件化原理和解决方案 (★★★★★)
- 4.13 第三方SDK 的使用 (★★★★★)
- 4.14 Android WebView应用 (★★★★★)
- 4.15 Gradle, 自动化构建 (★★★★★)
- 4.16Android APK 编译过程 (★★★★★)

5 Android 架构

- 5.1 android 架构的集中模式:
- 5.2 Android Architecture Components(★★★★★)

6 android 性能优化(★★★★★)

7 开发各种调试工具(★★★★★)

8.部分高频问题

- 1.Handler
 - 1.1 Handler 原理或者机制
 - 1.2 如何在子线程中创建Handler(IntentService)
 - 1.3 handler 使用不当为什么会造成内存泄漏, 如何避免

- 1.4 项目中那些地方用到Handler,具体如何使用的
- 1.5 AsyncTask
- 2.事件分发
 - 2.1 Android事件分发的机制或者流程
 - 2.2 Activity 的分发机制（如何传递到ViewGroup的）
 - 2.3 ViewGroup的分发机制（如何传递子view的）
 - 2.4 View 的事件分发机制
- 3.屏幕适配
 - 3.1 为什么要屏幕适配？
 - 3.2 屏幕适配有哪几种方案？
 - 3.2.1 android 原生屏幕如何适配
 - 3.2.2 百分比适配原理
 - 3.2.3 今日头条适配原理
 - 3.3 刘海屏，水滴屏 如何屏幕适配？
- 4. Activity
 - 4.1 Activity的四种启动模式，每种模式的使用场景？
 - 4.2 Activity中内存泄漏及解决办法
 - 4.3 Activity 的生命周期
- 5. Android 四大组件
 - 5.1 Android 四大组件分别是什么？
 - 5.2 组件之间通信有那些方式？
 - 5.3 四大组件的应用场景
 - 5.4 本地广播（了解一下）
- 6.Service
 - 6.1 Service是什么，一般用来干什么？
 - 6.2 Service与IntentService的区别？
 - 6.3 两种启动Service 的区别？
 - 6.4 通过不同的方式启动servie，如何销毁service？
 - 6.5 servive 中如何执行耗时操作？
 - 6.6 Service 如何保活？
 - 6.7 Activity 如何和服务进行通信？，有那些方式？
 - 6.8 Android 8.0 对Service 有什么限制？
- 7. Broadcast 广播
 - 7.1 广播有哪两种注册方式?有什么区别？
 - 7.3 什么是粘性广播？
 - 7.4 广播的优先级是什么？可以设置优先级吗？
 - 7.5 广播可以拦截吗？
 - 7.6 广播的生命周期？
 - 7.7 android 7.0 和 8.0 对广播有什么限制？
- 8. Android 性能优化
 - 8.1 UI 优化（UI 优化主要是指xml 布局优化）
 - 8.1.1 xml 布局有那些优化方案？
 - 8.1.2 如何发现层级嵌套过多？
 - 8.2 内存优化
 - 8.2.1 为什么要内存优化？
 - 8.2.2 如何发现app 存在内存问题（包括内存频繁分配，回收 和内存泄漏问题）？
 - 8.2.3 使用什么工具去检查内存泄漏（mat,leakcanary）？
 - 8.2.4 内存优化包括代码优化，和图片优化（大图，多图），如何优化

- 8.2.5 java 几种引用的 (强, 软, 弱, 虚)
 - 8.3 网络优化
 - 9.内存相关问题
 - 9.1 内存泄漏和内存溢出有什么区别?
 - 9.2 gc垃圾回收机制(新生代, 老年代, 永久代)
 - 9.3 堆和栈的区别
 - 10. 进程和线程相关问题
 - 10.1 进程和线程的区别?
 - 10.2 进程之间如何通信? 有那些方式?
 - 10.3 线程之间如何通信? 有那些方式?
 - 10.4 多线程如何保证安全?
 - 10.5 多线程断点下载?
 - 10.6 类锁和对象锁的区别?
 - 10.7 什么是线程池? 有哪几种线程池? 线程池有什么优势?
 - 11.ListView 和 RecyclerView 的区别
 - 12. Fragment 如何实现懒加载
 - 13. 自定义view
 - 13.1 自定义View 分为自定义View 和 自定义ViewGroup,分别如何实现?
 - 13.2 自定义View 涉及到那几个方法? 分别是干甚的?
 - 13.3 View的绘制流程
 - 14. MVC,MVP 的区别
-

1 java基础

1.1数据结构

1.1.1 数组 (★★★★)

1. 数组的内存结构式什么样的?
2. 数组大小可以改变吗?

1.1.2 集合 (★★★★)

1. 集合的超类是什么?
2. 集合的继承关系是什么?
3. ArrayList 的内存结构是什么样的?
4. ArrayList 为什么线程不安全?
5. ArrayList 是查找速度快还是增删快?
6. ArrayList 使用迭代器遍历的时候如何删除一个元素?

7. 如何获取线程安全的List

1.1.3 链表 (★★★★)

1. 链表的的数据结构是什么样的?
2. 链表是如何进行添加, 删除元素的?
3. 链表相对于ArrayList 有什么区别
4. 链表是查询快还是增删快?

1.1.4 Map (★★★★)

1. `HashMap` 的数据结构是什么样的?
2. `HashMap` 添加一个元素时是如何进行添加的 (源码步骤是什么样的)?
3. `HashMap` 的 `loadFactory` 和 `threshold` 是什么? 是如何进行扩容的?
4. 为什么 `HashMap` 是线程不安全的?
5. 如何获取一个线程安全的HashMap?
6. `HashMap` 使用迭代器遍历的时候如何删除一个元素?
7. Android 中 `SparseArray` VS `HashMap`
8. `ArrayMap` VS `HashMap`
9. `SparseBooleanArray`, `SparseIntArray`, `SparseLongArray`
10. `HashMap` 和 `HashTable` 区别

1.1.5 队列 (★★★★)

1. 队列的数据结构是什么样的?
 2. 如何往队列里面添加数据和取出数据?
-

1.2.常用算法 (★★★★)

把几种排序算法拍个照放手机里面

1.2.1 冒泡排序

1.2.2 简单选择排序

1.2.3 插入排序

1.2.4 快速排序

1.2.5 二分查找

1.3 Java反射机制，Java代理模式 (★★★★★)

1.4 Java泛型 (★★★★★)

1.5 java 线程相关 (★★★★★)

1.5.1 Synchronized原理 (★★★★★)

1.5.2 Volatile实现原理 (★★★★★)

1.5.3 方法锁、对象锁、类锁的意义和区别 (★★★★★)

1.5.4 线程同步的方法：Synchronized、lock、reentrantLock 分析 (★★★★★)

1.5.5 java锁的种类: 公平锁、乐观锁、互斥锁、分段锁、偏向锁、自旋锁等 (★★)

1.5.6 ThreadLocal的原理和用法 (★★★)

1.5.7 ThreadPool的用法和示例 (★★★★★)

1.5.8 wait()和sleep()的区别 (★★★)

1.5.9 Thread的中断机制(interrupt) (★★★★★)

1.5.10 线程池的实现原理 (★★★★★)

2 java 高阶

2.1 Java虚拟机，Java运行，Java GC机制（可达性分析法，引用计数法） (★★★★★)

2.2 Java对象的完整生命周期 (★★★)

2.3 JVM内存模型 (★★★)

2.4 进程间通信，线程间通信 (★★)

2.5 JVM类加载机制 (★★★)

2.6 Java引用类型 (★★★★★)

2.7 设计模式：除常用设计模式之外，特别的，反射机制，代理模式 (★★★★★)

2.8 HTTP协议和HTTPS协议 (★★★★★)

2.9 Socket协议，Socket实现长连接 (★★★)

2.10 TCP和UDP协议 (★★★★)

2.11 HTTP协议中GET和POST的具体实现 (★★★★★)

2.12 序列化和反序列化 (★★★★)

2.13 数据库基础知识：多表查询、索引、数据库事务 (★★★★)

3 Android 初级

3.1 生命周期 (★★★★★)

- `Application` 生命周期
- `Activity` 生命周期（横竖屏切换时生命周期变化）
- `Fragment` 生命周期（横竖屏切换时生命周期变化）
- `Activity` 的 `onNewIntent` 什么情况下回调用？

注意：以上生命周期讲解必须结合demo 输入log 日志讲解

3.2 Activity 几种启动模式？每一种模式分别使用什么场景？ (★★★★★)

注意：理论+demo的方式讲解

3.3 Service ,IntentService用法与区别？他们分别如何与Activity 进行通信？（★★★★★）

注意：理论+demo的方式讲解

3.4 Fragment 系列（★★★★★）

- 添加 `Fragment` 的几种方式？
- `add`, `replace`, `delete`, `attach`, `detach`, `show`, `hide` 方法有什么却别？调用不同方法时都会执行那些生命周期方法
- `Fragment` 回退栈机制？回退栈里面是保存的 `Fragment` 吗？如果不是那是保存的什？
- 如果一个 `Activity` 里面添加了5个 `Fragment` ,并且这个5个 `Fragment` 都调用 `addToBackStack` 方法加入到了回退栈里面，这5个 `Fragment` 的顺序是 `F1` 在最下面，`F5` 在最上面，如何在 `Fragment5` 里面按返回键的时候回退到 `Fragment2` 而不是回退到 `Fragment4` ？
- `Fragment` 如何添加动画？动画方法里面每一个参数表示什么意思？
- `Fragment` 之间如何传值？
- `FragmentPagerAdapter` 和 `FragmentStatePagerAdapter`

注意：理论+demo的方式讲解s

3.5 ContentProvider实例详解（★★★）

注意：理论+demo的方式讲解

3.6 BroadcastReceiver使用总结（★★★★★）

- `BroadcastReceiver` 的声明周期什么时候结束？
- 能否在 `BroadcastReceiver` 的 `onReceive` 方法里面开启一个线程去执行耗时操作？如果要执行耗时操应该怎么办？

注意：理论+demo的方式讲解

3.7 Android Handler 原理机制讲解 (★★★★★)

3.8 Binder机制，共享内存实现原理 (★★)

3.9 Android 事件分发机制 (★★★★★)

- 为什么要有事件分发？
- 事件分发设计到那几个方法？ `Activity` , `ViewGroup` , `View` 分别有那几个方法？
- 事件分发时传递的顺序 (`Activity` , `ViewGroup` , `View` 之间) ？
- 事件分发是通过那个事件 (`Down` , `Move` , `Up`) 去寻找要消耗事件的控件？
- 每一个涉及事件分发的方法里面 调用 `super` , 返回 `true` , 返回 `false` 的区别？

注意：理论+demo的方式讲解

3.10 Android 多线程的实现：Thread、HandlerThread、AsyncTask、IntentService、RxJava (★★★★★)

- 如果再想买里面要执行一个耗时操作你如何处理？是直接 `new` 一个 `Thread` 吗？
- 如何实现一个运行在子线程里面的 `Handler` ？
- `AsyncTask` 的内部机制？它有什么限制？

注意：理论+demo的方式讲解

3.11 Activity启动流程，ActivityThread解析 (★★)

3.12 嵌套滑动原理 (★★★★☆)

这是一个能提现技术和实力的问题，如果能回答得比较好能够在给面试加分。对于想要20k（包含）以上的同学能好好去研究一下这个问题，即使你不要20K，如果有空最好也去研究一下，再次强调这是一个能提现技术实力的题

3.13 RecyclerView与ListView(缓存原理，区别联系，优缺点) (★★★★☆)

3.14 View的绘制原理，自定义View，自定义ViewGroup (★★★★)

3.15 View、SurfaceView 与 TextureView (★★★)

3.16 ViewPager的缓存实现 (★★★★)

3.17 requestLayout, invalidate, postInvalidate区别与联系 (★★★★)

3.18 AndroidP新特性 (★★★★)

3.19 Android两种虚拟机 有那些区别 (★★★★)

这个问题简述一下有那两种虚拟机，有什么区别就行

3.20 ADB常用命令 (★★★★)

3.21 Asset目录与res目录的区别 (★★★★)

虽然这个问题被问到的几率不大，但是这个问题比较简单，所以必须去掌握一下

3.22 主线程Looper.loop为什么不会造成死循环 (★★★★)

3.23 Android 进程 和 线程 Process and Threads (★★★★★)

- [Android 进程 和 线程](#) (★★★★★)
- Android 多进程和Application关系? (★★★★★)
- [多进程通信解决方案: Andromeda](#) (★★★)

3.24 Android页面恢复 (★★★★★)

- Android的页面恢复采用以下两个方法，这两个方法都在什么时候调用：
 - onSaveInstanceState(Bundle outState)
 - onRestoreInstanceState(Bundle savedInstanceState)

在想想之前讲的ViewPager+FragmentManager 时是如何在Fragment 销毁是保存数据的？

3.25 Android mipmap 和 drawable 的区别 (★★★★★)

4 Android 高级

引子：Android高级工程师招聘要求：

1. 熟悉Android SDK，熟悉Android UI，熟悉Android各种调试工具；
2. 有丰富的Android应用架构能力，能够独立主导并架构App；
3. Mobile Web 开发经验；具备各种复合技能：熟悉iOS、H5、Python、.NET等多种开发语言的优先考虑；
4. 对Android性能优化，安全，软件加固，自动化测试有深刻认识；
5. 博客，开源项目

Android技术难点

AIDL、Binder、多进程、View的绘制流程、事件分发、消息队列等。

这类知识对于定位自己为高级Android工程师的人来说是必须掌握的，同时他也是能鉴别高级和初级工程师的一块试金石，其中binder是Android系统进程间通信最重要的手段之一，现阶段app的发展离不开多进程的运用，经常会启动例如定位、推送等需要在后台启动的进程来保证主进程的内存运行；所以合理的使用多进程也是十分必要的；

view的绘制是我们自定义控件的理论基础，只有掌握了view是如何绘制的才能个性化的自定义控件；

事件分发一直是Android开发的难点之一，也是必须掌握的；

关于handler机制也是android的一块难点，因为包括AsyncTask、系统启动、IntentService等底层都是通过handler来实现的，所以掌握后handler机制不仅能提高你的实战开发能力，更能让你系统的了解整个android系统运作的情况。

Android框架层源码掌握

Android框架层有很多东西，以下几个是高级程序员必须要掌握的：

1. Android包管理机制，核心PackageManagerService (★★)
2. Window管理，核心WindowManagerService (★★)
3. Android Activity启动和管理，核心ActivityManagerService (★★)
4. 根Activity工作流程 (★★)
5. Context关联类 (★★)

4.1 AIDL 是什么？如何用AIDL 实现进程间通信？ (★★★★)

这个问题先不用去管原理，知道AIDL是啥？用来干什么？如何定义AIDL 文件实现进程间通信？

4.2 自定义LayoutManager，RecyclerView中如何自定义LayoutManager (★★★☆☆)

4.3 VLayout实现原理 (★★★★☆)

VLayout 是什么？可以用它来实现什么效果？原理搞不懂的可以先不管，但是要会用

4.4 Glide (★★★★★★)

4.5 Retrofit的实现与原理 (★★★★★★)

4.6 OKHttp3 (★★★★★★)

4.7 EventBus (★★★★★★)

4.8 ButterKnife (★★★★★★)

4.9 RxJava (★★★★★★)

4.10 Dagger依赖注入 (★★★★★★)

4.11 热修复 (★★★★★★)

4.12 插件化开发 (★★★★★)

4.12 组件化原理和解决方案 (★★★★★)

4.13 第三方SDK 的使用 (★★★★★)

1. 统计SDK (★★★★★)

- 统计用户行为
 - 统计错误日子
2. 支付sdk (★★★★)
 3. 即时通讯 (★★★)
 4. 直播 (★★★)
 5. 视屏 (★★★)

4.14 Android WebView应用 (★★★★★)

- APP调用WebView加载url (★★★★★)
- 掌握WebView的封装，了解所有的WebSettings配置，掌握WebViewClient、WebChromeClient (★★★★★)
- JS 交互 (★★★★★)
- weiview 内存泄漏已经释放

4.15 Gradle，自动化构建 (★★★★★)

1. 如何进行多渠道打包？如果不同渠道包代码都完全一样，只有渠道信息不一样，如何快速的打出上百个不同渠道包？
2. 如何实现不同渠道包包含不同代码或者资源文件？

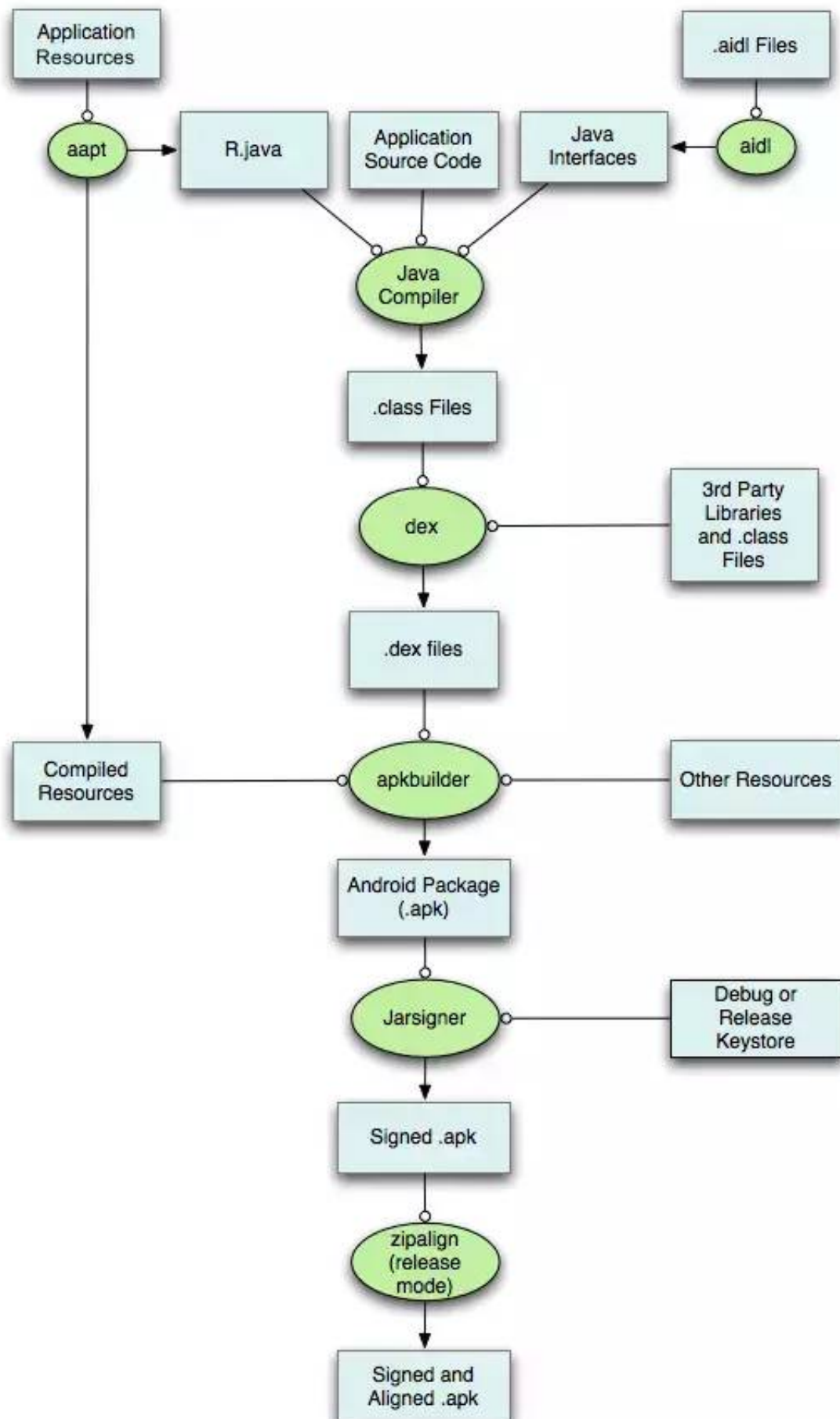
4.16 Android APK 编译过程 (★★★★★)

其中使用到的编译工具：

aapt、aidl、Java Compiler、dex、zipalign

主要步骤描述：

1. 通过aapt打包res资源文件，生成R.java、resources.arsc和res文件（二进制 & 非二进制如res/raw和pic保持原样）
2. 处理.aidl文件，生成对应的Java接口文件
3. 通过Java Compiler编译R.java、Java接口文件、Java源文件，生成.class文件
4. 通过dex命令，将.class文件和第三方库中的.class文件处理生成classes.dex
5. 通过apkbuilder工具，将aapt生成的resources.arsc和res文件、assets文件和classes.dex一起打包生成apk
6. 通过Jarsigner工具，对上面的apk进行debug或release签名
7. 通过zipalign工具，将签名后的apk进行对齐处理。



5 Android 架构

在Android源码中最重要的三个类：ActivityManagerService / PackageManagerService / View，推荐大家周末的时候可以去阅读下这部分的源码，阅读源码能提高我们今后设计架构自己代码的能力，同时也能从底层了解整个android系统的运行原理，其他一些比如主线程的消息循环、主线程如何和AMS如何跨进程交互、SystemServer进程中的各种Service的工作方式、AsyncTask的工作原理等。这些知识也是作为一个Android高级开发工程师必须掌握的，不能整天沉溺于ui和四大组件的交互，要站在更高的角度去考虑Android的有些问题。

[移动端架构的思考](#)

5.1 android 架构的集中模式：

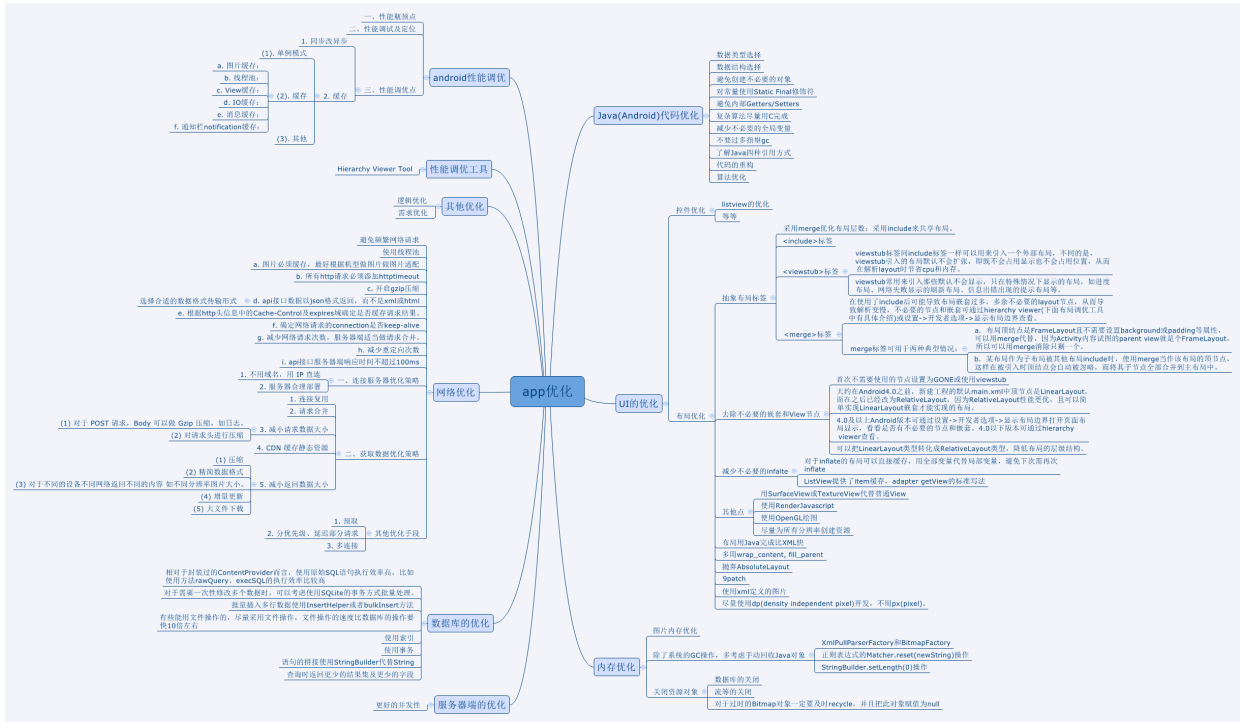
- MVC模式(★★★★★)
- MVP模式(★★★★★)
- MVVM模式(★★★★★)
- CLEAN模式(★★★)
- 跨平台开发：Flutter、ReactNative(★★★)

5.2 Android Architecture Components(★★★★★)

- Lifecycles
- LiveData
- ViewModel
- Data Binding
- Room

以上这些东西都要了解他是什么？能用来干什么？如何使用？

6 android 性能优化(★★★★★)



7 开发各种调试工具(★★★★)

- 性能分析工具：Memory Monitor
- 内存泄漏检测工具 MAT, Leakcanory
- 性能追踪及方法执行分析： TraceView
- 视图分析： Hierarchy Viewer
- ApkTool- 用于反向工程Android Apk文件的工具
- Dex2Jar- 使用android .dex和java .class文件的工具
- 反混淆工具

8.部分高频问题

1.Handler

1.1 Handler 原理或者机制

1.2 如何在子线程中创建Handler(IntentService)

1.3 handler 使用不当为什么会造成内存泄漏，如何避免

1.4 项目中那些地方用到Handler,具体如何使用的

1.5 AsyncTask

2.事件分发

2.1 Android事件分发的机制或者流程

2.2 Activity 的分发机制（如何传递到ViewGroup的）

2.3 ViewGroup的分发机制（如何传递子view的）

2.4 View 的事件分发机制

3.屏幕适配

3.1 为什么要屏幕适配？

3.2 屏幕适配有哪几种方案？

3.2.1 android 原生屏幕如何适配

3.2.2 百分比适配原理

3.2.3 今日头条适配原理

3.3 刘海屏，水滴屏 如何屏幕适配？

4. Activity

4.1 Activity的四种启动模式，每种模式的使用场景？

4.2 Activity中内存泄漏及解决办法

4.3 Activity 的生命周期

5. Android 四大组件

5.1 Android 四大组件分别是什么？

5.2 组件之间通信有那些方式？

5.3 四大组件的应用场景

5.4 本地广播（了解一下）

6.Service

6.1 Service是什么，一般用来干什么？

6.2 Service与IntentService的区别？

6.3 两种启动Service 的区别？

6.4 通过不同的方式启动service，如何销毁service？

6.5 service 中如何执行耗时操作？

6.6 Service 如何保活？

6.7 Activity 如何和Service 进行通信？，有那些方式？

6.8 Android 8.0 对Service 有什么限制？

7. Broadcast 广播

7.1 广播有哪两种注册方式？有什么区别？

7.3 什么是粘性广播？

7.4 广播的优先级是什么？可以设置优先级吗？

7.5 广播可以拦截吗？

7.6 广播的生命周期？

7.7 android 7.0 和 8.0 对广播有什么限制？

8. Android 性能优化

8.1 UI 优化（UI 优化主要是指xml 布局优化）

8.1.1 xml 布局有那些优化方案？

1. 根布局使用 ConstraintLayout ，减少层级嵌套
2. 采用 `merge` 标签（去了解一下merge标签是干啥的）
3. 使用 `ViewStub` 标签减少可能不需要加载的布局，节约系统资源
4. 控件背景图片尽量用 `.9` 图片或者通过xml 定义的 `shape` 图片，矢量图
5. 去掉不必要的背景（比如Activity 的背景，根布局的背景）

8.1.2 如何发现层级嵌套过多？

8.2 内存优化

8.2.1 为什么要内存优化？

8.2.2 如何发现app 存在内存问题（包括内存频繁分配，回收 和内存泄漏问题）？

8.2.3 使用什么工具去检查内存泄漏（mat,leakcanary）？

8.2.4 内存优化包括代码优化，和图片优化（大图，多图），如何优化

8.2.5 java 几种引用的（强，软，弱，虚）

8.3 网络优化

9.内存相关问题

9.1 内存泄漏和内存溢出有什么区别？

9.2 gc垃圾回收机制(新生代，老年代，永久代)

9.3 堆和栈的区别

10. 进程和线程相关问题

10.1 进程和线程的区别？

10.2 进程之间如何通信？有那些方式？

10.3 线程之间如何通信？有那些方式？

10.4 多线程如何保证安全？

10.5 多线程断点下载？

10.6 类锁和对象锁的区别？

10.7 什么是线程池？有哪几种线程池？线程池有什么优势？

11.ListView 和 RecyclerView 的区别

12. Fragment 如何实现懒加载

13. 自定义view

13.1 自定义View 分为自定义View 和 自定义ViewGroup,分别如何实现？

13.2 自定义View 涉及到那几个方法？ 分别是干甚的？

13.3 View的绘制流程

14. MVC,MVP 的区别
