1. 开场自我介绍

自我介绍优化版本：

面试官，你好，我叫肖志，我是2017年从华南理工大学硕士毕业，专业是电子通信工程，

我从事Android开发有5年多了，到目前为止有三家工作经历；

第一家是华为，在公司的消费者智能穿戴产品线，负责运动健康App项目

运动相关业务开发及App架构的开发工作

第二家是Shopee公司，在公司的供应链业务线，负责东南亚物流订单揽收及派送相关业务开发工作，并对物流App引入一些动态化技术

第三家是在区块链公司深圳星辰大海做Torrent App主力开发，该App定位是提供种子资源下载及音视频资源播放，我在里面负责P2P传输协议技术的开发与维护，App广告竞价与投放的方案调研与输出

我擅长的技术点有：Android的性能优化、组件化、热修复等技术

深耕Android技术开发，成为领域专家或者领域项目管理者

面试官，你好；我叫肖志，我是2017年从华南理工大学硕士毕业，专业是电子通信工程，毕业后先后在华为、Shopee工作

(1)首先是校招加入华为，在华为消费者BG部门下从事运动健康App开发，App是超级App，面向全球市场，累积用户过2亿，日活超千万

整个App前期是采用模块化开发，主要以Native和H5开发为主，少量快应用项目

我主要工作内容是负责运动健康相关业务开发，参与对 APP 瘦身、主进

程和守护进程内存优化、启动速度等性能优化，此外负责参与App12.0服务解耦框架设计与开发，对App做组件化改造，并参与部分模块插件化工作；在这段经历中熟练负责App日常开发与维护、App性能优化并有着App组件化与插件化等相关实践经验；

专利：一种运动参数获取方法及装置、电子设备、存储介质(CN2020106709445)

(2)2021年6月底加入Shopee，在供应链业务线，从事针对东南亚8个以上市场快递业务司机端作业App开发，这是Shopee公司除Shopee App之外第二大App，日派送订单达千万级

App整体采用组件化开发方式，使用MVP、MVVM架构，用到RxJava、OKHttp三方框架；并紧跟Google平台，比如使用谷歌公司的Firebase监控平台、Room数据库等

我这边工作负责快递从卖家发货到买家收货大部分环节的业务核心开发，并针对东南亚弱网痛点引入离线模式方案，骑手离线作业比例占比到2%以上，尤其提高了骑手在揽收和派送两大重要流程的作业效率；此外，针对App引入了热修复技术，支持多渠道补丁包自动化编译、构建及部署完整流程，让App在线上运行时能动态、快速修复问题；在这段经历中，接触了海外App的开发所涉及到一些平台及一些三方框架，并学习到项目从立项、评审、开发迭代到上线不一样的开发流程

(3)2023年5月加入了个搞区块链公司，做Torrent种子下载App，面向海外市场，日活百万级，主要涉及P2P传输协议技术、广告竞价及投放等功能，为了提高广告收入，通过调研 Github 开源 OpenMediation SDK，自研一套聚合网络，建立从流量客户端到广告产生端整个链路，并结合用户画像精准投放对应类型广告

好，我的介绍完毕

英文自我介绍

Interviewer, hello; My name is Xiao Zhi. I graduated with a master's degree in Electronic Communication Engineering from South China University of Technology in 2017. After graduation, I worked at Huawei and Shopee successively

(1) Firstly, the school recruited Huawei and worked on the development of sports and health apps under Huawei's consumer BG department. The app is a super app targeting the global market, with a cumulative user base of over 200 million and daily activity exceeding 10 million

The entire app was developed using modularity in the early stages, mainly focusing on Native and H5 development, with a small number of fast application projects

My main job responsibilities are responsible for the development of sports and health related businesses, and participating in the slimming and promotion of the app

Program and daemon memory optimization, startup speed and other performance optimization. In addition, he is responsible for participating in the design and development of App12.0 service decoupling framework, making componentized transformation of App, and participating in the plug-in work of some modules; Proficient in the daily development and maintenance of apps, optimizing app performance, and having practical experience in app componentization and plugination during this experience;

(2) Joined Shopee at the end of June 2021, working in the supply chain business line as a driver side express delivery app targeting more than 8 markets in Southeast Asia. This is the second largest app of Shopee company besides Shopee App, with daily delivery orders reaching tens of millions

The overall development of the app adopts a component-based approach, using MVP and MVVM architectures, and using a tripartite framework of RxJava and OKHttp; And closely follow the Google platform, such as using Google's Firebase monitoring platform, Room database, etc

I am responsible for the core business development of most aspects of express delivery from sellers to buyers, and introducing offline mode solutions for weak network pain points in Southeast Asia. The proportion of riders working offline accounts for over 2%, especially improving the efficiency of riders in the two important processes of collection and delivery; In addition, hot fix technology has been introduced for the app, supporting the complete process of automated compilation, construction, and deployment of multi-channel patch packages, allowing the app to dynamically and quickly fix problems when running online; During this experience, I was exposed to some platforms and third-party frameworks involved in the development of overseas apps, and learned about the different development processes of projects from project initiation, review, development iteration to online launch

Okay, that's all for my introduction

英文简洁版本

Interviewer, hello; My name is Xiao Zhi. I graduated with a master's degree in Electronic Communication Engineering from South China University of Technology in 2017. After graduation, I worked at Huawei and Shopee successively

(1) Firstly, the school recruited Huawei and worked on the development of sports and health apps under Huawei's consumer BG department. The app is a super app targeting the global market, with a cumulative user base of over 200 million and daily activity exceeding 10 million

In the early stage of the entire app, modular development was adopted, mainly focusing on Native and H5 development, with a small number of fast application projects. During this experience, I am proficient in daily app development and maintenance, app performance optimization, and have practical experience in app componentization and plugination;

(2) Joined Shopee at the end of June 2021, working in the supply chain business line as a driver side express delivery app targeting more than 8 markets in Southeast Asia. This is the second largest app of Shopee company besides Shopee App, with daily delivery orders reaching tens of millions

The overall development of the app adopts a component-based approach, using MVP and MVVM architectures, and using a tripartite framework of RxJava and OKHttp; And closely follow the Google platform, such as using Google's Firebase monitoring platform, Room database, etc

During this experience, I was exposed to some platforms and third-party frameworks involved in the development of overseas apps, and learned about the different development processes of projects from project initiation, review, development iteration to online launch

Okay, that's all for my introduction

1. 简历项目总结

1.1热修复

1.2离线

1. 业务侧：用力过猛，揽收订单全流程都要求做离线；实际作业过程涉及order和task多个状态扭转，在App本地操作存在与服务端对不齐的高风险

在需求评审阶段，Client、BE、QA我们私下都一致认为这个离线功能完成风险较大，收益很小，做的必要性不大；而鉴于此需求是产品老大定的OKR重点交付需求，再去讨论裁减这个需求可能性较小，当然如果选择小范围做离线是可以做的

1. 技术测

BE：原有揽收订单存在2个维度：order和task，里面又涉及到批量扫描和实时扫描场景

客户端需要承载原先服务端大量查询多个数据库、数据校验等逻辑，这很大程度增加了客户端的复杂程度，而且这些操作如果用户在离线下操作，监控上也很难及时去发现问题

APP：实际考虑用户有网到无网，无网到有网过程，并且还需要对多个场景页面处理客户端数据和服务端数据不一致问题，冲突数据如何展示，离线数据如何保证顺序同步到服务端

1. 数据预加载，缓存：全量/增量， 定时同步
2. APP本地建立room数据库缓存并更新离线数据
3. 在线/离线切换时，大体采用在线优先策略，请求失败后构造离线数据；本地数据校验通过后默认成功
4. 建立一套离线数据同步机制，保证时序以及支持数据重试、限制同步频率等

1.3运动健康APP组件化

1.4 DriverApp项目架构

1.5 国家差异化框架

1.6 手机定位实现

网络定位数据如何获取的？

1. Android技术知识点

2.1 内存泄漏

<https://zhuanlan.zhihu.com/p/567342744>

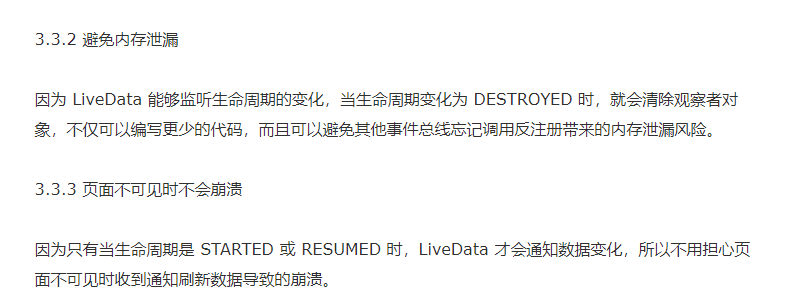
<https://www.51cto.com/article/715124.html>

2.2 Android架构模式(MVC、MVP、MVVM)

<https://blog.csdn.net/zg0601/article/details/123587933>

<https://blog.csdn.net/weixin_37577039/article/details/78504357?utm_medium=distribute.pc_relevant.none-task-blog-2~default~baidujs_baidulandingword~default-1-78504357-blog-118092341.pc_relevant_multi_platform_whitelistv3&spm=1001.2101.3001.4242.2&utm_releva>

2.3 LiveData原理



<https://www.infoq.cn/article/D7kp6cP3XBLXgTCAp8KI/>

2.4 Kotlin/Kotlin协程

2.5 性能优化

<https://zhuanlan.zhihu.com/p/348646268>

【性能优化】大厂OOM优化和监控方案

<https://blog.csdn.net/ajsliu1233/article/details/124370065>

2.5.1 启动优化

2.5.2 内存优化

2.5.3 帧优化

2.5.4 APK大小优化

2.5.5 稳定性优化

2.5.6 绘制优化

2.5.7 网络优化

2.5.8 存储优化

2.5.9 日志优化

2.6 RxJava

<https://blog.csdn.net/sinat_35622297/article/details/89844471?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-89844471-blog-121902338.pc_relevant_default&depth_1-utm_source=distribute.pc_relevant.no>

2.7 Retrofit

<https://www.jianshu.com/p/89d74db4c824>

Retrofit 原理篇

<https://blog.csdn.net/augfun/article/details/123888096?ops_request_misc=&request_id=&biz_id=102&utm_term=Retrofit&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~sobaiduweb~default-4-123888096.article_score_rank_blog&spm=1018.2226.3001.4450>

2.8 Okhttp

<https://www.jianshu.com/p/16a7e83eeb79>

<https://mp.weixin.qq.com/s?__biz=MzA5MzI3NjE2MA==&mid=2650267613&idx=1&sn=a4f1ae77700723e548cf9209bbab61b5&chksm=88632eb2bf14a7a44d752ef76ebdb55297c9478ca5bd485d13167f0ee985c5e7b172fd984996&scene=27>

Android高工：okhttp原理详解

<https://zhuanlan.zhihu.com/p/391928581>

面试常问题目：

<https://blog.csdn.net/weixin_39069034/article/details/114687185?spm=1001.2101.3001.6661.1&utm_medium=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-114687185-blog-115124578.235%5Ev27%5Epc_relevant_landingrelevant&depth_1-utm_source>

2.9 RN/Flutter

2.10 APP组件化

2.11 APP插件化

<https://zhuanlan.zhihu.com/p/467882549>

<https://blog.csdn.net/qq_29966203/article/details/123792407>

插件化介绍和原理解析

<https://www.jianshu.com/p/04aeb7b7ee56>

爱奇艺开源 Qigsaw，基于 Android App Bundle 的动态化框架

<http://www.see-source.com/news/1893.html>

<https://blog.csdn.net/u014294681/article/details/107605305>

2.12 APP编译打包

<https://mp.weixin.qq.com/s?__biz=MzA5MzI3NjE2MA==&mid=2650265470&idx=1&sn=b0a8a269958339a9e33b6777572c97bb&chksm=88632611bf14af079ce3ee1df90dc4e830aa3c106a12def378a5c957ef948f0a1802cd51d404&scene=27>

2.13 JetPack（DataBinding、ViewModel、LiveData）

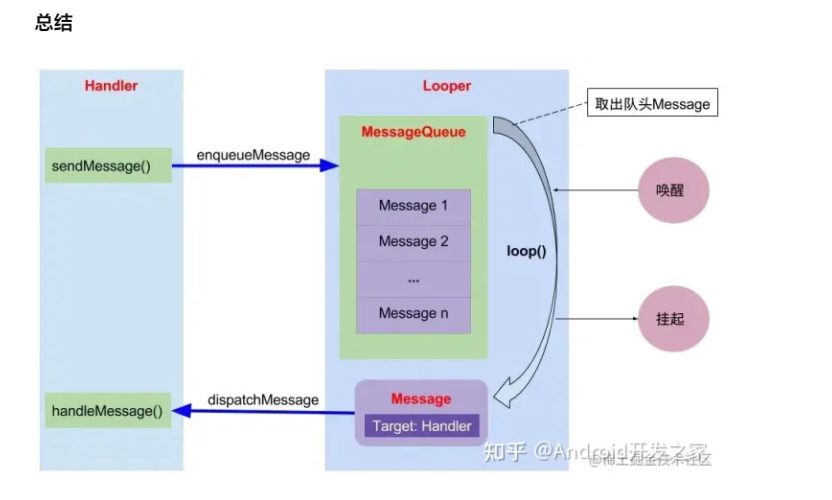
<https://www.infoq.cn/article/D7kp6cP3XBLXgTCAp8KI/>

2.14 View绘制原理

<https://blog.csdn.net/qq_21864095/article/details/127440709>

2.15 Handler消息机制

<https://zhuanlan.zhihu.com/p/462204108>



handler发送延迟处理消息的原理

<https://blog.csdn.net/thh159/article/details/103644489>

Android开发Handler中加数据，内部是如何确保线程安全的?【Handler的消息机制学习】

<https://www.jianshu.com/p/bf88974b0ca6>

2.16 recyclerView原理

<https://blog.csdn.net/qjyws/article/details/123237071>

<https://zhuanlan.zhihu.com/p/80475040>

<https://www.jianshu.com/p/e62537875e92>

2.17 Glide缓存原理

<https://blog.csdn.net/wenzhi20102321/article/details/119337059>

2.18 事件分发机制

<https://blog.csdn.net/qq_41095045/article/details/122266940>

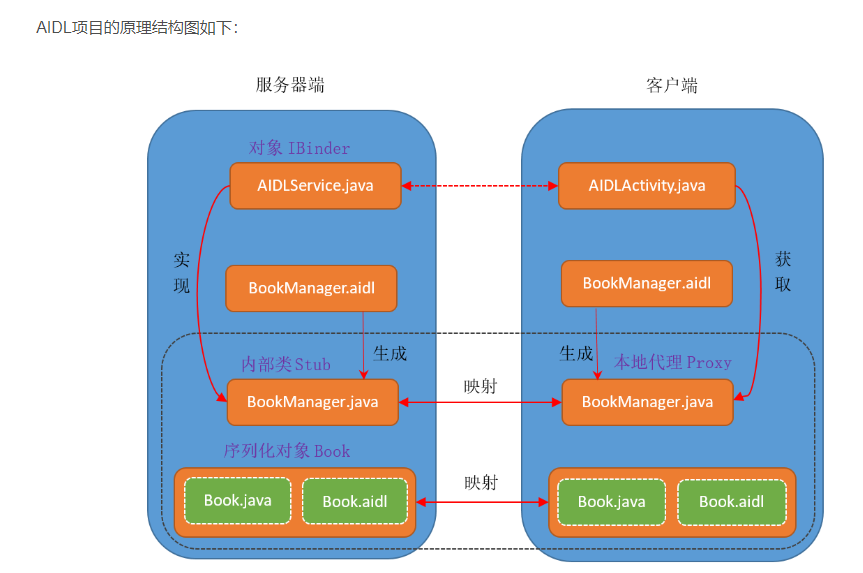
<https://www.jianshu.com/p/9b6f6bd6d541>

2.19 Android虚拟机

2.20 AIDL怎么通信

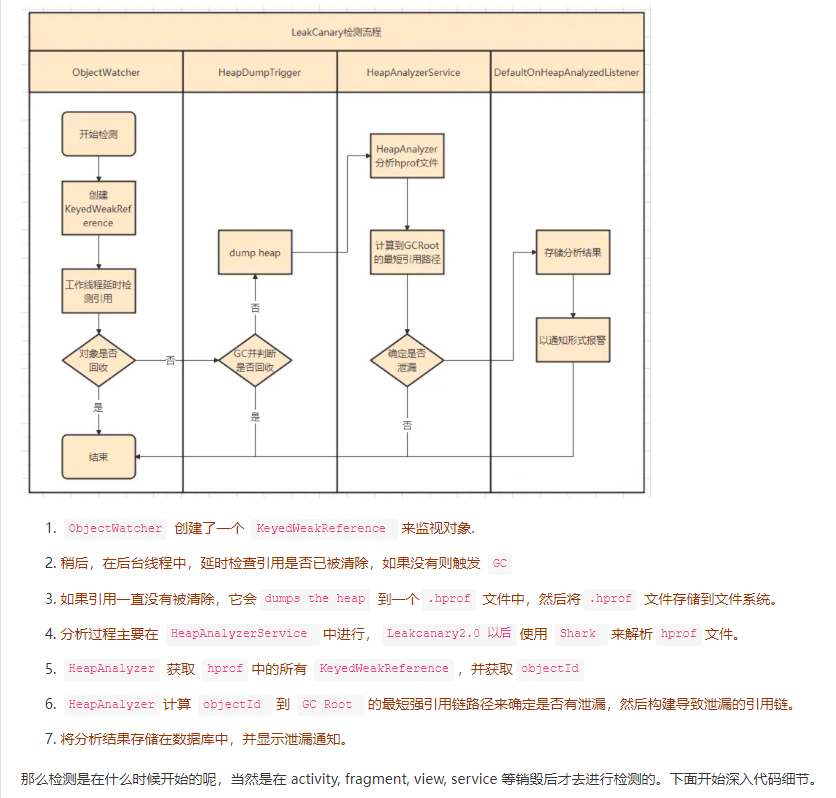
<https://blog.csdn.net/qq_41739313/article/details/123322808>





2.21 LeakCanary检测原理

<https://www.cnblogs.com/huansky/p/15489365.html>



2.22 市场差异化编译原理

2.23 Android动画

2.24 约束布局

2.25 View滑动冲突解决

2.26 四种引用

java对象的引用包括  
  强引用，软引用，弱引用，虚引用

<https://zhuanlan.zhihu.com/p/552644187>

2.27 WebView

<https://www.cnblogs.com/JQstronger/p/WebView.html>

2.28 Gradle常用配置参数

<https://blog.csdn.net/HakuMaster/article/details/127805670>

2.29 JSBridge

JSBridge是横跨原生运行环境和JavaScript运行环境的一道桥梁。这个桥梁，是双端进行通信的媒介。

<https://mp.weixin.qq.com/s/-pH2CN8QvbRqYcGQWUxkRA>

2.30 APP启动流程

<https://baijiahao.baidu.com/s?id=1728062966946873234&wfr=spider&for=pc>

ContentProvider初始化流程分析

<https://juejin.cn/post/6996512754899091487>

2.31 Intent传输大小限制

<https://blog.csdn.net/hequnwang10/article/details/125006697>

2.32 classloader

2.33 Binder通信

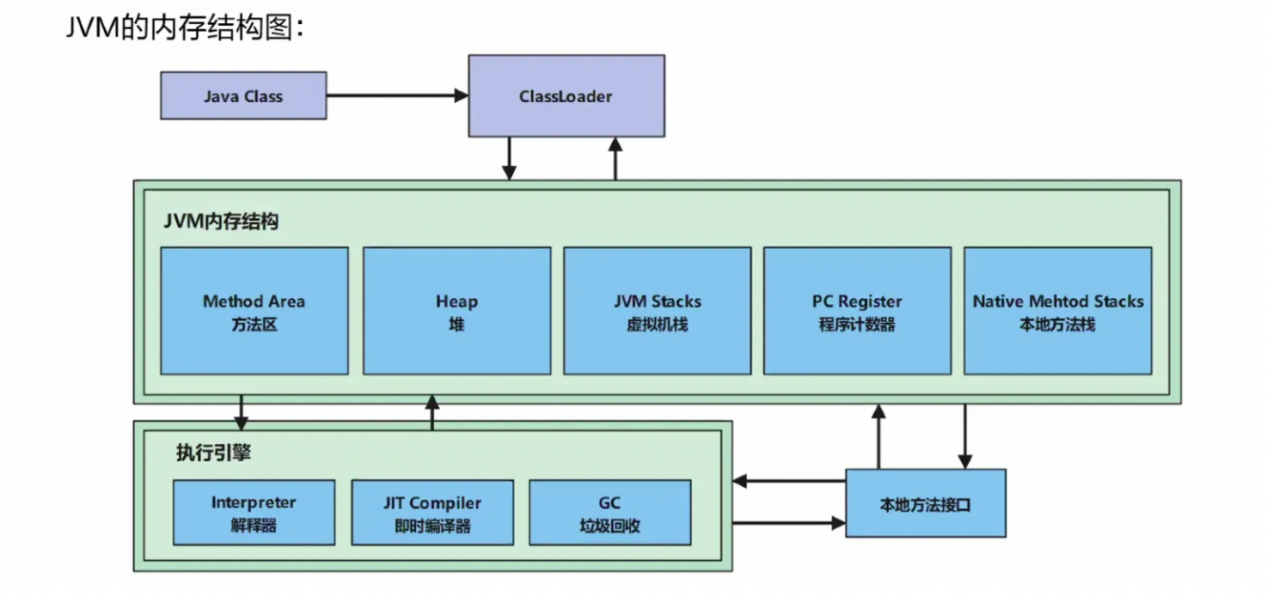
<https://juejin.cn/post/7208488647535214649>

1. Java技术知识点

3.1 JVM

<https://www.cnblogs.com/chihirotan/p/6486436.html>

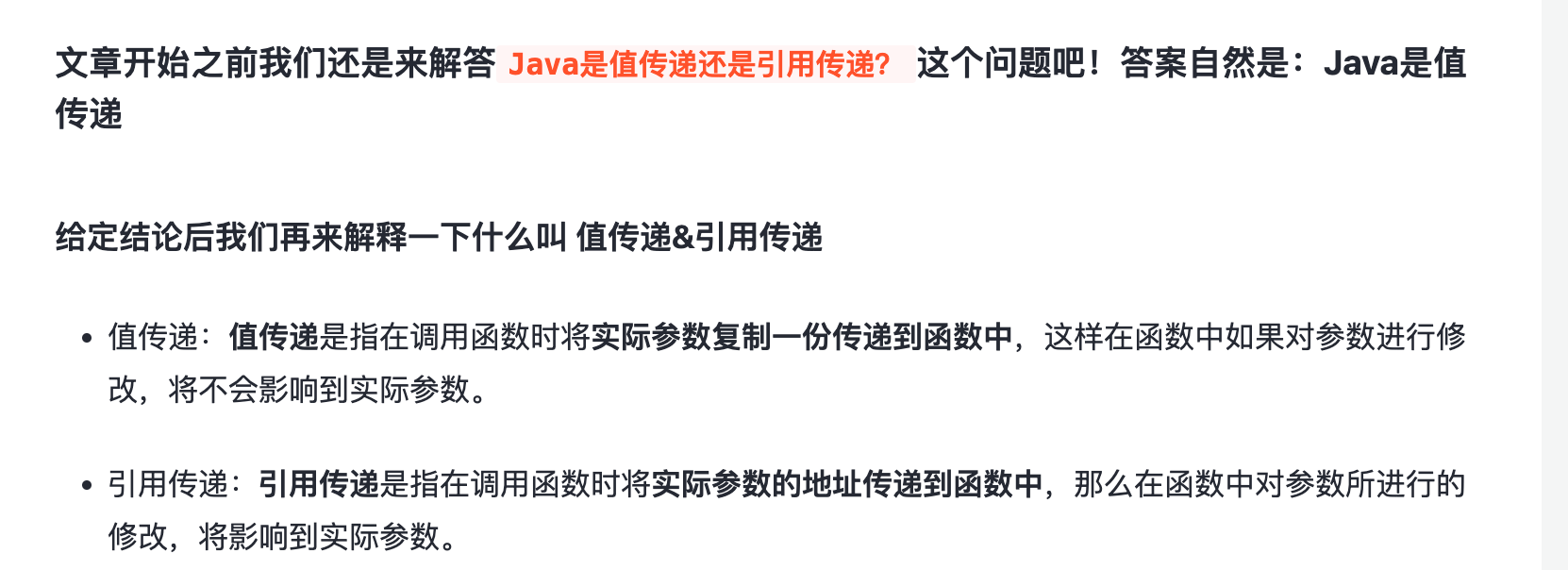
<https://zhuanlan.zhihu.com/p/411021919>



3.2 Java垃圾回收

<https://blog.csdn.net/u012672300/article/details/103600925>

3.3 Java值传递和引用传递



<https://segmentfault.com/a/1190000016773324>

3.4 锁的用法及原理

3.4.1 悲观锁和乐观锁

<https://blog.csdn.net/qq_34337272/article/details/81072874>

[https://muxiaonong.blog.csdn.net/article/details/106351873?spm=1001.2101.3001.6661.1&utm\_medium=distribute.pc\_relevant\_t0.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-106351873-blog-81072874.pc\_relevant\_aa2&depth\_1-utm\_source=distribute.pc\_relevant\_t0.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-106351873-blog-81072874.pc\_relevant\_aa2&utm\_relevant\_index=1](https://muxiaonong.blog.csdn.net/article/details/106351873?spm=1001.2101.3001.6661.1&utm_medium=distribute.pc_relevant_t0.none-task-blog-2~default~CTRLIST~Rate-1-106351873-blog-81072874.pc_relevant_aa2&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-2~default~CTRLIST~Rate-1-106351873-blog-81072874.pc_relevant_aa2&utm_relevant_index=1)

3.4.2 公平锁和非公平锁

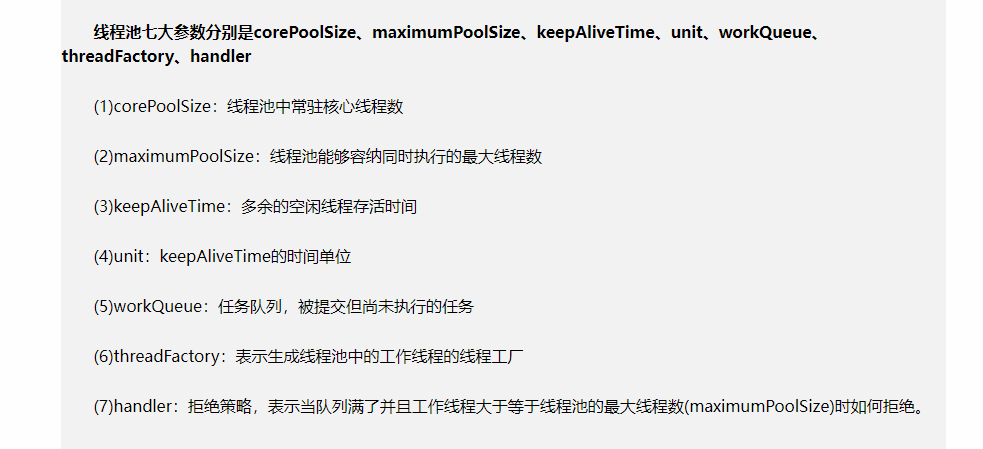
<https://blog.csdn.net/weixin_42480780/article/details/116646365>

3.5 voliate关键字、原子变量、lock与synchronize区别

<https://segmentfault.com/a/1190000042671275?utm_source=sf-similar-article>

3.6 线程池

[https://blog.csdn.net/weixin\_40271838/article/details/79998327?spm=1001.2101.3001.6661.1&utm\_medium=distribute.pc\_relevant\_t0.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-79998327-blog-106911277.pc\_relevant\_multi\_platform\_whitelistv3&depth\_1-utm\_source=distribute.pc\_relevant\_t0.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-79998327-blog-106911277.pc\_relevant\_multi\_platform\_whitelistv3&utm\_relevant\_index=1](https://blog.csdn.net/weixin_40271838/article/details/79998327?spm=1001.2101.3001.6661.1&utm_medium=distribute.pc_relevant_t0.none-task-blog-2~default~CTRLIST~Rate-1-79998327-blog-106911277.pc_relevant_multi_platform_whitelistv3&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-2~default~CTRLIST~Rate-1-79998327-blog-106911277.pc_relevant_multi_platform_whitelistv3&utm_relevant_index=1)



线程池的核心参数怎么确定

<http://m.studyofnet.com/736755380.html>

3.7 如何保证线程安全

3.8 一个Java类加载过程

<https://www.cnblogs.com/TheGCC/p/14738123.html>

3.9 Java中的堆和栈

<https://zhuanlan.zhihu.com/p/529280783>

3.10 线程同步方式

3.11 线程的几种状态

3.12 HashMap实现原理

<https://blog.csdn.net/weixin_45804960/article/details/108572434>

简单来说，HashMap由数组+链表组成的，数组是HashMap的主体，链表则是主要为了解决哈希冲突而存在的，如果定位到的数组位置不含链表（当前entry的next指向null）,那么查找，添加等操作很快，仅需一次寻址即可；如果定位到的数组包含链表，对于添加操作，其时间复杂度为O(n)，首先遍历链表，存在即覆盖，否则新增；对于查找操作来讲，仍需遍历链表，然后通过key对象的equals方法逐一比对查找。所以，性能考虑，HashMap中的**链表出现越少，性能才会越好**。

<https://mikechen.cc/974.html>

<http://www.52xingchen.cn/detail/49>

3.13 双亲委派机制

<https://blog.csdn.net/weixin_45962068/article/details/124087845>

3.14 ThreadLocal的原理

<https://blog.csdn.net/zhiyikeji/article/details/125473692>

3.15 两个应用间如何传输数据

3.16 线程通信方式

<https://www.cnblogs.com/linyufeng/p/9671844.html>

3.17 进程通信方式

<https://blog.csdn.net/GMLGDJ/article/details/124627224>

3.18 多线程面试总结

<https://blog.csdn.net/guorui_java/article/details/113872241>

3.19 hashcode和equals详解

<https://blog.csdn.net/wangqsse/article/details/107198059?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-107198059-blog-108983642.pc_relevant_default_base2&depth_1-utm_source=distribute.pc_relevant.>

1. 设计模式

<https://www.runoob.com/design-pattern/design-pattern-tutorial.html>

根据设计模式的参考书 Design Patterns - Elements of Reusable Object-Oriented Software（中文译名：设计模式 - 可复用的面向对象软件元素） 中所提到的，总共有 23 种设计模式。这些模式可以分为三大类：创建型模式（Creational Patterns）、结构型模式（Structural Patterns）、行为型模式（Behavioral Patterns）。当然，我们还会讨论另一类设计模式：J2EE 设计模式。

1、开闭原则（Open Close Principle）

开闭原则的意思是：对扩展开放，对修改关闭。在程序需要进行拓展的时候，不能去修改原有的代码，实现一个热插拔的效果。简言之，是为了使程序的扩展性好，易于维护和升级。想要达到这样的效果，我们需要使用接口和抽象类，后面的具体设计中我们会提到这点。

2、里氏代换原则（Liskov Substitution Principle）

里氏代换原则是面向对象设计的基本原则之一。 里氏代换原则中说，任何基类可以出现的地方，子类一定可以出现。LSP 是继承复用的基石，只有当派生类可以替换掉基类，且软件单位的功能不受到影响时，基类才能真正被复用，而派生类也能够在基类的基础上增加新的行为。里氏代换原则是对开闭原则的补充。实现开闭原则的关键步骤就是抽象化，而基类与子类的继承关系就是抽象化的具体实现，所以里氏代换原则是对实现抽象化的具体步骤的规范。

3、依赖倒转原则（Dependence Inversion Principle）

这个原则是开闭原则的基础，具体内容：针对接口编程，依赖于抽象而不依赖于具体。

4、接口隔离原则（Interface Segregation Principle）

这个原则的意思是：使用多个隔离的接口，比使用单个接口要好。它还有另外一个意思是：降低类之间的耦合度。由此可见，其实设计模式就是从大型软件架构出发、便于升级和维护的软件设计思想，它强调降低依赖，降低耦合。

5、迪米特法则，又称最少知道原则（Demeter Principle）

最少知道原则是指：一个实体应当尽量少地与其他实体之间发生相互作用，使得系统功能模块相对独立。

6、合成复用原则（Composite Reuse Principle）

合成复用原则是指：尽量使用合成/聚合的方式，而不是使用继承。

5.1 单例模式

5.2 策略模式

5.3 工厂模式

5.4 责任链模式

5.5 观察者模式

1. 数据结构与算法

(1)无重复字符子串

(2)字符相加以及相乘

(3)快排

1. 带团队经验

团队高效管理的“五大核心要素”！

https://baijiahao.baidu.com/s?id=1624075930131860402&wfr=spider&for=pc

1. 离职原因、职业规划、优缺点

**星晨大海科技一个月离职原因：**

和自己预期不太符合

1、团队只有2个Android，而且都是刚来不久

2、所从事业务为美国那边交接过来，由于时差及权限控制，存在较多约束问题

3、业务方向比较朦胧和局限

由于个人成长受限，存在诸多不确定性以及团队存在一定解散风险，个人选择了离职

**Shopee离职原因：**

1. 我现在已经有一定经验和技能，~~由于现在所在团队是做To B产品，很多方面不太关注，~~我希望能够拓宽自己的知识面，~~比如跨平台等技术~~，进而有更深入的学习和实践；
2. 企业发展状况不佳，~~然后在2月底公司出了个政策，给员工三天时间考虑，可自愿申请离职，然后拿N+2补偿；~~
3. 目前行业发展进入低谷，想看比较稳定点好的平台

**华为离职原因：**

1、当时部门组织架构调整，从领导到下面员工换了很多人，自己所从事工作与个人职业发展不太匹配

2、猎头推荐外面机会挺吸引人，尝试跳槽过去了Shopee

3、所在团队一起并肩作战工作的多位同事都走了

4、当时住在宝安那里，每天通勤来回2小时较远

**个人职业规划：**

**随着公司的发展一起成长！**

感谢面试官提出这么深刻的问题。我的兴趣是XX，优势是XX，因此我选择了XX行业，这是一个可以将我的兴趣和工作结合起来的行业，是我非常喜欢的，所以我会跟用心对待XX这个岗位。说到职业规划，近期三到五年，我打算在XX行业做到XX，希望可以不断提升自己的技术，持续学到更多的知识，后续可以在XX岗位独当一面，独立负责XX事物，解决XX问题。谈到远期的规划，我会根据环境的变化，工作内容的变化，以及我自身能力的变化，不断进行调整。对于职业规划，我暂时的考虑是这个样子的。

我是这么想的， 我对XX方面很有兴趣，贵公司正好在XX方面很有优势，我计划在半年（一年）专攻XX技术。不过我对XX方面有点想法，我想在三年（五年）之后，能对XX进行构架方面的改进。这只是我目前的计划，当然我知道计划赶不上变化，如果业务和团队有变化，我也会做出相应的调整。

**短期规划：**

1. 进一步深入钻研Android框架技术、比如插件化、热修复、跨平台等技术
2. 结合岗位要求及工作内容，提升自己技术，拓宽技术视野，把握前沿技术动态，持续涉猎更多相关信息

**长期规划：**

根据环境变化、工作内容变化以及自身能力不断调整，技术/管理方向个人倾向于管理方向

**优缺点：**

优点：做事有计划、抗压能力强、天生乐观

缺点：缺乏管理经验、做事顾虑太多可能错过机会、有时候缺乏全局观容易陷入细节

**性格如何：**

1. 乐观
2. 好学上进
3. 为人真诚、乐于助人
4. 其他

9.1 TCP和UDP区别

TCP 和 UDP 是负责提供端到端通信的传输层协议。TCP 是面向连接的协议，而 UDP 是无连接协议。

TCP是一种面向连接的协议，也就是说，在收发数据前，必须和对方确认已经建立了可靠的连接。

建立连接的方式又称为三次握手：

（1）第一次握手，主机A向主机B发出请求数据包：“我想给你发数据，可以吗？”这是第一次对话。

（2）第二次握手，主机B向主机A发送同意连接，并要求同步的数据包（同步就是两台主机协调工作，一台在发送，一台在接收）：“可以，你什么时候发？”这是第二次对话。

（3）第三次握手，主机A再发出一个数据包确认主机B的要求同步：“我现在就发，你接收吧！”这是第三次对话。

这三次对话的目的是使数据包的发送和接收同步，TCP协议是一定要经过这三次对话之后，主机A才可以向主机B正式发送数据。

TCP建立连接需要进行三次握手，但是断开连接却要进行四次挥手：

（1）第一次挥手，在主机A完成数据传输后，将控制位FIN置1，提出停止TCP连接的请求。

（2）第二次挥手，主机B收到FIN位置上的1信息后对其做出响应，确认这一方向上的TCP连 接将关闭，将ACK置1。

（3）第三次挥手，B主机再提出反方向的关闭请求，并将控制位FIN置1，发送给A主机，并关闭连接。

（4）第四次挥手，主机A对主机B的请求进行确认，将ACK置1，并关闭连接，至此双方关闭连接。

UDP主要有以下几个特点：

（1）UDP是一个非连接的协议，传输数据之前，源端和终端不建立连接，当它想传送时，就简单地去抓取来自应用程序的数据，并尽可能快地把它扔到网络上。

在发送端，UDP传送数据的速度仅受应用程序生成数据的速度、计算机的能力和传输带宽的限制；在接收端，UDP把每个消息段放在队列中，应用程序每次从队列中读一个消息段。

（2）由于传输数据不建立连接，因此不需要维护连接状态，包括收发状态等。一台服务机可同时向多个客户机传输相同的消息。

（3）UDP信息包的包头很短，只有8字节，相对于TCP的20字节包头信息，UDP的包头开销很小。

（4）吞吐量不受拥挤控制算法的调节，只受应用软件生成数据的速率、传输带宽、源端和终端主机性能的限制。

（5）UDP会尽最大努力去传输和接受数据且没有限制，但并不保证可靠的数据交付，主机也不需要维持复杂的链接状态表（里面有许多参数）。

（6）UDP是面向报文的。发送方的UDP对应用程序传过来的报文，在添加包头后就向下交付给IP层。既不拆分，也不合并，而只是保留这些报文的边界，因此，应用程序需要自己限制合适的报文大小，以免报文太大导致丢失率高。

总结一下就是，TCP安全、可靠、面向连接，但是传输速度慢。UDP不安全、不可靠、面向非连接，但是传输速度快。

9.2 蓝牙传输协议

9.3 CPU调度时间分片

9.4 NDK调用实现方式

9.5 一些重要得计算机网络协议

<https://www.cnblogs.com/fzz9/p/8964513.html>

9.6 Http协议

<https://blog.csdn.net/le_17_4_6/article/details/124017572>

彻底搞懂HTTPS的加密原理

<https://zhuanlan.zhihu.com/p/43789231>

# 附：Android面试复习整理

<https://blog.csdn.net/cpcpcp123/category_9148883.html>

Flutter实战

<https://book.flutterchina.club/>

10、HR面候选人了解的问题

(1)公司的人才发展和培养体系是怎么样的？

(2)公司的薪酬福利能介绍下吗？有补充医疗保险吗？

(3)社保和公积金的缴纳基数和比例是多少？全额吗？

~~(4)公司工资发放形式？ 选择USDT和现金发放有啥区别？如果选择USDT发放，员工~~

~~提现风控怎么保证？避税是怎么做？员工的社保公司是怎么缴纳？~~

(5)公司的上下班时间是怎么样的？加班情况如何？周末需要上班吗？

(6)年假和病假这块是什么额度？

(7)餐补和房补是怎么样的？

(8)公司具体地址是在哪里？

(9)深圳这边有多少员工？

(10)合同是签几年？

(11)期望薪酬：希望和现在至少能持平

(12)公司绩效考核是怎样的？ 年终奖是怎么分配的？

(13)试用期几个月？试用期工资和转正工资一样吗？

(14)公司每年有调薪窗口吗？一般调薪幅度是多少？

11、HR面问题总结

(1)工作中遇到的最大挑战是什么？

在Shopee这边做离线化需求是我工作中遇到最大的一个挑战

**挑战难点：**App离线化对于我来说是第一次尝试做，并且由于涉及业务场景加多，夹杂历史包袱，离线具体流程交互产品和设计未明确给出，需要客户端和后台一起出方案敲定，评审时开发、测试、产品各方已经评估上线后出现风险较大，属于一个投入回报比低的需求

**如何解决：**通过调研业内离线化方案，并结合我们项目业务特点，前期通过多次和产品、测试以及后端讨论对齐细节，设计了一个匹配当前App离线化各方都能接受的方案，并在提测前列出了可能出现风险的checklist知会产品；此外，针对App端本地离线操作的各个数据流做了监控与埋点，尽可能保证上限后能第一时间发现问题并快速解决问题。

**效果：**上线后运行至今未出舆情问题，离线签收率达到了2%以上，部分市场能达到10%；

提高了骑手的作业效率