

问题一

1. 部门内安卓的技术栈的重点能大致描述一下吗？希望能再明确一下后续重点学习的切入点（路线、重点、时间已经大致规划好了，但是可能有偏差，需要验证一下）
2. 学习过程中，能否有量化的标准（或指标），比如以某个开源项目为目标，限定多久复刻或读懂核心源码合适？

目前的想法以及准备的辅助学习的资源大概是下面这样(是否可行,如何改进?)

思路：自顶向下（自底向上时间方面代价太大，目前的目标是尽快投入生产，但是不能太功利）

1. 语言方面：Kotlin 以使用为主、Java 以底层原理为主
2. 基础方面：快速过掉，搭建持续学习扩展的框架，不断积累（靠书籍/公众号/博客/视频/开源项目/常见考点）
3. 熟悉开发框架、第三方依赖，从开源项目下手，在实践中学习，总结出现的核心技术点
4. 后续发展/持续学习：
 1. 纵向：着重 Framework 及更下层（内核）、性能优化
 2. 横向：关注大前端（Flutter、H5、小程序、ReactNative、鸿蒙等）

每日需要积累的：算法、设计模式、Kotlin、Java 底层

短期计划（一个月，重在搭建持续学习的环境，构建好整个知识体系的框架）：

已经掌握了大部分通用的技术，重新过一遍会比较快

1. 基础 + 项目管理 (7)
2. 数据持久层 (2)
3. 权限管理 (2)
4. 自定义控件、项目实战进阶、发布分发 (4)
5. 常用开发库、开发框架 (7)
6. 底层原理、开源项目源码 (3)
7. Framework、性能优化、内核 (3)

学习资源

1. 书籍：疯狂Android讲义、深入理解Android、Android源码设计模式、Android框架揭秘、深入理解Android 系列
2. 公众号（相关10+，其他领域40+）
3. 博客（相关8，其他领域20+）、视频博主（1）、线上课程（1）
4. 开发框架、开源项目（5）
5. 交流群（3）+ 常见问题（PDF 总结）

问题二

1. 技术学习方面，关于**发展趋势、收集信息与持续学习**等有较为系统一点的方法论吗？
2. 有推荐的「书籍/博客/公众号/开源项目/视频博主」或者其他的建议吗？

目前我的方式大概就是下面这样：

收集的主要分为：

1. 技术发展趋势
2. 碎片化知识
3. 优质技术文章

途径：

1. Github Trending (关注开源项目)
2. 技术相关的公众号 (关注了 50+，基本每天都能学习新知识) + 微信标签整理
3. 网页收藏 + Onetab 整理分类网页 (便于分类，目前收藏大几百)
4. Rsshub + RSS 客户端主动订阅 + 技术博客
5. 技术交流群
6. 视频博主 (Bilibili&Youtube，快速实践，目前关注 260+)
7. 书籍 + Calibre

其他：

1. 碎片化知识的整合主要是靠：Typora + 滴答清单
2. 提升效率方面，我是采用 UTools + AutoHotKey + Groupy 快速分类、搜索
3. 后续考虑 NAS + PT 自动化收集

问题三

平时可以使用哪些辅助开发的工具来提高效率？我目前常用的大概有：

1. ChatGPT + New Bing
2. Claude
3. 文心一言
4. CodeGeeX
5. Llama 2