## [扫清Android面试障碍](http://chaojimiaomiao.github.io/2016/05/08/%E6%89%AB%E6%B8%85Android%E9%9D%A2%E8%AF%95%E9%9A%9C%E7%A2%8D.html)

怎样快速突破初级瓶颈，变身高级开发？怎样在短时间内提高自我身价，月薪提高50%？你是否是个代码高手，面试中却发挥不出来，想进阶却摸不着头脑。博主在互联网行业摸爬滚打，百面成钢。特来总结与分享自己面试的心路历程和经验。

本系列将分为四大部分切入，包括**第一部分面试前的准备**：从长期来看要做点什么来提高自己的身价，短期内怎样迅速进入面试状态，以及如何准备简历才能吸引HR的眼球。**第二部分是安卓初级中级高级面试题**，大概会有几十个问题，基本覆盖了小白到大牛进阶的所有会被问到的问题。**第三部分是除了安卓经验**，你还会被问到哪些问题。**第四部分是关于 offer的选择**，以及一起探讨未来的职业规划。从这四部分看来，已经涵盖了一个程序员的方方面面。我一直觉得面试也是一种考试，只要是考试就有套路，有套路就能被总结，能被总结就能被练习，就好比高考模拟卷或者培训班，能教你的也就是反复练习。

先交代一下背景：博主本科毕业于国内某985大学，10年大二开始接触Android开发，参加了两届谷歌举办的全国大学生安卓应用竞赛并获奖。实习和工作都在上海的互联网公司做app开发。曾误入 iOS深处，沉醉不知归路。后来惊醒还是转投回安卓的世界中去。目前工作三年。本文的目的在于总结回顾上一波的面试，给未来的自己看看当时的努力与心情，更重要的是希望对业内还在迷茫中想要提高自己的开发们指点迷津。

前两个月，我面试了上海多家互联网公司，按规模上至阿里携程新美大，中至爱奇艺安居客B站平安科技，下至格瓦拉触宝科技蜻蜓fm，偏至招行信用卡微鲸科技等。并非有意要面那么多，只是一开始行动力太旺盛不小心简历投多了。秉承“自己投的简历，跪着也要面完”；以及想一窥上海互联网公司发展情况等多种心理下，横扫了各大Android面试题。经验积累了不少，从简历准备到技术准备再到扯淡环节，甚至于怎样针对不同的公司不同的面试官提出不一样的问题等等。

这整个过程不是一帆风顺的，就像给程序做优化一样，一点点完善、一点点让自己的表现趋近完美。我当时甚至会在家对着镜子针对可能被问到的问题，练习说话的语速和表情。请理解一个心机 girl 总会想太多，比如有的问题面试官或许会觉得这道题是他的杀手锏，那你就不要表现出一副做了充分准备、分分钟秒杀他的气势。毕竟咱面的是普通开发，最后还得在人手底下做事呢～ 这种情况下最好的办法是面露一点点思考与回忆交加的表情，一点点说出对方希望你说出来的答案；而不是一股脑全说完了。当然了，不同的面试官有不同的套路，比如我最欣赏携程的一个初面面试官，能感觉到那是一个智商超群的人。和聪明人讲话，你就不用再拐弯抹角了。Anyway，假如这个面试官最后露出了满意的笑容，那就恭喜你成功了

面试真的是个体力活，有两次我都想放弃了懒得再面了，让我继续坚持下去的除了有始有终的信念以外，还有照云兄一句“ 所谓自由就是随时拥有说’不’的权利。” 我理解就是给自己多一种选择。这一波面试下来，最遗憾的还是阿里几个想让我去的部门都在杭州，但我是上海人还要在交大在职读研。另外，上海的BAT Android岗位实在是太少了，腾讯基本就不在上海招。曾有冲动为了工作收起行囊就那么离家去杭州，但理智最终制止了我。阿里现在有点像软件界的华为，基本打来的电话都在晚上八点，有种血汗工厂的感觉。总体来说，上海的互联网公司规模都没成气候，还是中小企业居多。看了好几年上海互联网的起起落落，唯有”恨其不争“能形容这种感受了。PS：我最后选择了一份”钱多事少离家近“的工作：请不必问我现在何处，因为我一定会对公司和薪资守口如瓶。

话不多扯，上文的各种体会是为了下文抛砖引玉。这个系列的文章除了全面，更重要的补全除技术以外的短板。比如，当一个面试官问你：**你的职业规划是什么**。相信每个面试者都被问过。

大部分人的内心OS想必是： what，职业规划是什么？ what，这个问题究竟有什么意义？ what，我该怎么回答他？

如果技术面过了，栽在这种问题上岂不吃亏？相反假如你答得好，却会为你的薪资增加筹码。其实这个问题究竟该怎么回答，要具体情况具体分析。首先要看这家公司处于它这个行业的什么位置，它未来的公司战略规划是什么，它想招进来一个怎样的人，它希望你发挥的是什么作用，坐在你对面的是app leader还是CTO 抑或是HR？都会有不同的最佳回答。因为每个人想听到的是不一样的“实话”，实话打引号表示经过包装以后的答案。

关键就是你得理解这话背后的意思，面试官为什么要问这个问题呢，他想知道的无非是：

1. 你会在这家公司呆多久？
2. 你未来可能处于公司什么样的角色中。
3. 你能否适应公司现有文化？
4. 你能为公司发挥多少价值。
5. 面试官还有自己的小算盘。

如果坐在你对面的是app leader，你说你以后想做到这里的技术leader，那你一定是撞枪口上了，除非他就是要离职前招个leader进来。请注意这种情况是很有可能发生的。大多数面试官进来不会介绍他的职位，面试level 也未必和面试顺序正相关。这就好像在一个暗箱中博弈，通过察言观色和判断得到的信息是有限的，知己知彼你才能不出错，然后才是怎样最大化说出对方想听到的接近答案。

## 面试前的准备

* [长期准备：](http://www.bingjie.me/2016/05/12/%E6%89%AB%E6%B8%85%E9%9D%A2%E8%AF%95%E5%89%8D%E7%9A%84%E5%87%86%E5%A4%87.html#section)
* [1. 订阅几个高质量的公众号](http://www.bingjie.me/2016/05/12/%E6%89%AB%E6%B8%85%E9%9D%A2%E8%AF%95%E5%89%8D%E7%9A%84%E5%87%86%E5%A4%87.html#section-1)
* [2. 加入一个本地android组织](http://www.bingjie.me/2016/05/12/%E6%89%AB%E6%B8%85%E9%9D%A2%E8%AF%95%E5%89%8D%E7%9A%84%E5%87%86%E5%A4%87.html#android)
* [3. 看几本必看的进阶书](http://www.bingjie.me/2016/05/12/%E6%89%AB%E6%B8%85%E9%9D%A2%E8%AF%95%E5%89%8D%E7%9A%84%E5%87%86%E5%A4%87.html#section-2)
* [4. 收藏几个博客，紧跟几个专家](http://www.bingjie.me/2016/05/12/%E6%89%AB%E6%B8%85%E9%9D%A2%E8%AF%95%E5%89%8D%E7%9A%84%E5%87%86%E5%A4%87.html#section-3)
* [5. 写自己的独立技术博客](http://www.bingjie.me/2016/05/12/%E6%89%AB%E6%B8%85%E9%9D%A2%E8%AF%95%E5%89%8D%E7%9A%84%E5%87%86%E5%A4%87.html#section-4)
* [6. 看源码](http://www.bingjie.me/2016/05/12/%E6%89%AB%E6%B8%85%E9%9D%A2%E8%AF%95%E5%89%8D%E7%9A%84%E5%87%86%E5%A4%87.html#section-5)
* [7. 提交自己的开源代码](http://www.bingjie.me/2016/05/12/%E6%89%AB%E6%B8%85%E9%9D%A2%E8%AF%95%E5%89%8D%E7%9A%84%E5%87%86%E5%A4%87.html#section-6)
* [短期准备：](http://www.bingjie.me/2016/05/12/%E6%89%AB%E6%B8%85%E9%9D%A2%E8%AF%95%E5%89%8D%E7%9A%84%E5%87%86%E5%A4%87.html#section-7)
* [1. 去NewCoder刷题](http://www.bingjie.me/2016/05/12/%E6%89%AB%E6%B8%85%E9%9D%A2%E8%AF%95%E5%89%8D%E7%9A%84%E5%87%86%E5%A4%87.html#newcoder)
* [2. 去极客学院阅读几本微书](http://www.bingjie.me/2016/05/12/%E6%89%AB%E6%B8%85%E9%9D%A2%E8%AF%95%E5%89%8D%E7%9A%84%E5%87%86%E5%A4%87.html#section-8)
* [3. 看几篇分析源码的文章](http://www.bingjie.me/2016/05/12/%E6%89%AB%E6%B8%85%E9%9D%A2%E8%AF%95%E5%89%8D%E7%9A%84%E5%87%86%E5%A4%87.html#section-9)
* [4. 梳理一下知识体系，找出薄弱环节](http://www.bingjie.me/2016/05/12/%E6%89%AB%E6%B8%85%E9%9D%A2%E8%AF%95%E5%89%8D%E7%9A%84%E5%87%86%E5%A4%87.html#section-10)
* [简历的准备：](http://www.bingjie.me/2016/05/12/%E6%89%AB%E6%B8%85%E9%9D%A2%E8%AF%95%E5%89%8D%E7%9A%84%E5%87%86%E5%A4%87.html#section-11)[有几点](http://www.bingjie.me/2016/05/12/%E6%89%AB%E6%B8%85%E9%9D%A2%E8%AF%95%E5%89%8D%E7%9A%84%E5%87%86%E5%A4%87.html#section-12)[千万别…](http://www.bingjie.me/2016/05/12/%E6%89%AB%E6%B8%85%E9%9D%A2%E8%AF%95%E5%89%8D%E7%9A%84%E5%87%86%E5%A4%87.html#section-13)

有的时候我们在职场上会碰到各种各样的坑…… 不管是创业公司倒闭啦，新同事合不来啦，氛围太差受不了啦。如果说女人的安全感来自于独立，那么程序员的安全感就是手中有技术，心中有源码。想飞得更高，有哪些是你平时可以做的呢？

本篇接上一篇 [扫清面试障碍](http://www.bingjie.me/2016/05/08/%E6%89%AB%E6%B8%85Android%E9%9D%A2%E8%AF%95%E9%9A%9C%E7%A2%8D.html)。上篇主要抛砖引玉，提到了许多技术和非技术相关的面试技巧等。本篇讲述面试前的准备，这不仅是技术实力的积累，还有相当一部分软实力的积累。即使是暂时没有跳槽想法的技术人，也应当生于忧患之中。行业的发展非常迅速，逆水行舟不进则退。危机意识很重要，厚积薄发才能更上一层楼。

友情刺激大家： [《高二Android大牛是这样炼成的》](http://chuansong.me/n/329510551337)

面试前的准备可分为长期准备，和短期准备。首先声明，文中提到的书籍以及公众号，博客等，都是我自己觉得特别优秀的，绝没收任何广告费

### 长期准备：

#### 1. 订阅几个高质量的公众号

很多大厂都有自己的微信订阅号，很多人的阅读时间非常琐碎，那么你就可以在地铁上、在等公交车的间隙去刷刷最新技术的发展。订阅号贵精不贵多，某个订阅号里的文章一天也最多一篇，不然就容易粗制滥造。这里推荐几个高质量的android公众号：

腾讯Bugly: weixinBugly。



image

是[腾讯bugly](http://www.bingjie.me/2016/05/12/bugly.qq.com)，专门做性能监控的平台，里面的文章也都是总结bugly bbs上企鹅工程师文章的精华。发送频率很低，质量都不错。后期面试中很多关于内存泄漏的答案都来自于此。

移动开发前线：bornmobile。info主编徐川个人的微信公众号，里面的文章有原创也有很多精华集锦。关键是都比较严肃，不像某些个人公众号发展到后期变个人的独白段子手，干货也多，由于大都是搬很多个人博客和采访，所以质量还不错。

Android程序员：androidtrending。公众号所有者应该是汤涛。虽然Android技术公众号不少，个人感觉这个公众号是比较能紧跟潮流的。也没有什么废话而是纯技术。

程序媛：programgirl。一个程序媛发声的平台，更新不定期，欢迎所有女程序员都来投稿，投稿地址：chaojimiaomiao@qq.com. 比如本期android面试就会连载在里面。

#### 2. 加入一个本地android组织

大学时期我开始沉迷于android技术，但当时不善社交自命清高，往往是孤军奋战。回过头看，一个人的力量是有限的，只有交流才能进步，闭关锁国是没前途的。结识一批有同样追求的同行，甚至有很多比自己厉害许多的专家人才，你才能看到你未来努力的方向。人外有人，天外有天，千万别做井底之蛙。信息时代信息的交换是有价值的，小伙伴们一起努力也会将你带入一个好的氛围。

但是要谨慎地选择伙伴。我知道男生聚在一起最容易犯的错误是互相影响，比如大家一起去打一盘游戏啦，各种吹牛或恭维啦，，这些都是不可取的。低调而务实的小伙伴们聚在一起才有创造力，你们共同完成一个梦想一件事。

这里举个例子，https://github.com/LittleFriendsGroup/ 里的AndroidSdkSourceAnalysis。 他们发起了一个开源项目：18天认领android源码解析。这就是共同进步共同成长的典型呐~ 朋友应当是志同道合的，而非没目标的乌合之众。

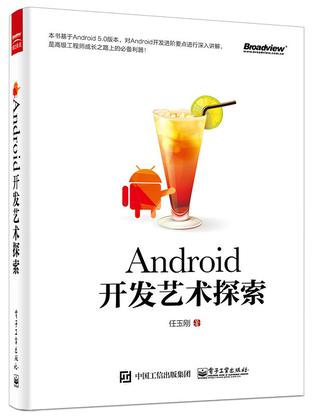
#### 3. 看几本必看的进阶书

市面上入门书籍泛滥，而适合中高级的主流技术书实在太少。一方面是能写这种书的大牛懒得写，据我所知每卖出一本技术书，作者才拿到4块钱稿费。而中高级的受众面毕竟窄，即使热卖，卖出一万本，那么所拿稿费也不过是4万。然而有这种水平的开发者根本不缺这几万块钱。写书是一件特别耗费精力的事，除了内容还要排版，吃力不讨好。更何况还会面临盗版横行的情况，就更不值了。写书仅剩的吸引人的价值，恐怕是成书后带来的名声与成就感。由于这种客观情况的存在，市面上粗制滥造的书实在太多，我在此帮大家剔除糟粕，列举几本我觉得值得的书。

**推荐书单：**

**《Android开发艺术探索》任玉刚**

业界良心！里面的很多东西面试都被问到了！



image

**《App研发录》包建强**

[App研发录](https://read.douban.com/ebook/16178944/)：架构设计、Crash分析和竞品技术分析。这是一本非常务实的书，书里提到的内容都可以直接拿来用在项目上。这个作者原先是搞.net的，工作十多年经验非常丰富。

摘录一段读后感：

“落地”，是阅读此书的最大感受，大到技术架构、竞品分析、发布流程，小到每个Crash、员工座位安排、开展技术分享、简历筛选，作者总在寻求落地的解决方案，细细阅读，你不觉得这种方案有多么高大上，不觉得背后有多么华丽的理论支撑，但是你会觉得，一切都是踏踏实实、真真切切可落地实践的，为工作带来不少的实践指导。

**《Android内核设计思想》**

这本书写得很深，基本是这三本书里最接近底层的书了，读起来也比较晦涩。需要有一定的理解能力。 全书从操作系统的基础知识入手，全面剖析进程/线程、内存管理、Binder机制、GUI显示系统、多媒体管理、输入系统等核心技术在Android中的实现原理。书中讲述的知识点大部分来源于工程项目研发，因而具有较强的实用性，希望可以让读者“知其然，更知其所以然”。全书分为编译篇、系统原理篇、应用原理篇、系统工具篇共4篇22章，基本涵盖了参与Android开发所需具备的知识。

#### 4. 收藏几个博客，紧跟几个专家

有一阵我很迷茫，不知道继续进步的方向，也不知道我能达到什么样的程度。我甚至因为找不到安卓上进步的空间而横向求索，玩了一年的iOS开发。就在这时，我的偶像来了。偶然间一次搜索，翻到了 hukai.me。（当然发现作者长得帅也是一个激动的原因）

原来安卓还可以这么玩~！可以像胡凯一样紧跟谷歌的最新视频教程并翻译它们。 也可以像 [代码家](http://www.bingjie.me/2016/05/12/daimajia.com) 那样写控件造轮子。代码家甚至因为轮子造得好而被谷歌发现，发出了工作邀请。

具体你可以关注哪些安卓的优秀博客可以见 [这个链接](https://github.com/chaojimiaomiao/androidBlogCN)：

博客地址

* [柳志超博客](http://liuzhichao.com/)
* [代码家](http://blog.daimajia.com/)
* [胡凯](http://hukai.me/)
* [禅](http://www.bingjie.me/)
* [Trinea](http://www.trinea.cn/)
* [方杰](http://blog.fangjie.info/)
* [技术小黑屋](http://droidyue.com/)
* [程序亦非猿](http://yifeiyuan.me/)
* [脉脉不得语](http://www.inferjay.com/)
* [陈启超的博客](http://chenqichao.me/)

#### 5. 写自己的独立技术博客

见贤思齐焉。

有了那么几个专家榜样，我的视野突然变开阔了，也想变成像他们一样厉害的人。这个时候你已经阅读了几本书，也有了一定的积累，自然而然也会想发表自己的看法和见解。

总之，克服瓶颈期的最好的办法就是六个字——总结、归纳、演绎。把你的收获都写到自己的博客上吧。最好是独立的博客，一方面你会收获到折腾独立博客的乐趣，另一方面它也是你的个人品牌，假如你能长期积累、坚持的话。独立博客唯一的缺点是导流。当然，你对自己的技术总结，目标是为了精益求精，为了自己的进步成长，至于读者多不多，倒是其次的。这个后期在面试中也大有裨益。别人刚认识你的时候对你不了解，你扔出一个博客地址，胜过你的千言万语。

没事的时候就去写写博客吧，记住\_\_不要写很多生活琐事\_\_，而是\_\_记录进步\_\_。别人不是为了看你的个人秀而来到你的空间，而是搜索到你对技术新的见解，觉得对他可能会有帮助才过来的。牢骚什么的，写到你的社交网站上去。

有的人很勤奋，博客天天更新。有量而没有质，感觉很空虚。我主张坚持精品博文才是正道，，毕竟信息已经那么泛滥了，不要再为互联网制造过剩的垃圾辣。。。（真的不是教你懒） 想要写一个优秀的博客，不仅要有实际的项目积累和广泛的阅读积累以外，uml类图，流程图，顺序图等，也是你必不可少的好助手。假如再加上\_\_思维导图\_\_，不仅使你的思路更清晰，读者也能一目了然，有一个全局观。现在就开始试试这些工具吧，会让你的博客更严谨更正式，更熠熠生辉。

#### 6. 看源码

这里列举几个你必须要看的源码：

* Binder
* LruCache
* Handler
* AsyncTask
* EventBus

不要怀疑，这些在面试中一定会被问到的。前期你看过源码也会淡忘的，先有一个大致的印象就行。

#### 7. 提交自己的开源代码

写什么样的开源代码，有这么几个方向。

一是你博客中的例子。一个好的技术博客不仅有图有文字，假如有例子，更胜过苍白的自然语言。百闻不如一见，说得再多也不如实际一个例子。

二是自定义的控件基础库。这个主要集中在酷炫的动画，还有交互更简洁的自定义控件上。这方面的大牛比如很多国外的库，还有国内代码家博客上提到的开源库~

三是有关测试方面的工具。像leakCannary, blockCannary等，能让开发者更好地测试内存卡顿泄漏等。

四是一些反编译的工具。

五是一个完整的上架作品。

### 短期准备：

#### 1. 去NewCoder刷题

这个网站主要包括了几套公司真题，比如有 [Android的](https://www.nowcoder.com/contestRoom?mutiTagIds=642)，[算法讲解](http://www.nowcoder.com/live/2)，[数据结构](http://www.nowcoder.com/review#3)。

有时间的可以去上面刷刷题，题量较大

#### 2. 去极客学院阅读几本微书

没时间阅读厚重的大部头怎么办？有一些不出版的非纸质微书也是极好的选择。极客学院有[几个系列](http://wiki.jikexueyuan.com/list/android/) 写得不错，比如《深入理解 Android 卷》系列，《Java 集合学习指南》等。

做工程的开发者容易陷入埋头写代码的怪圈，实际上多看一些技术书，吸收别人的思想精华进步会更快。

#### 3. 看几篇分析源码的文章

在长期准备中，你已经接触过一些源码了；当然你可能跳槽时间比较紧，那不如关注一下这个开源项目吧，里面包括了大量的 android源码分析。

ANDROID SDK 源码解析：[AndroidSdkSourceAnalysis](https://github.com/LittleFriendsGroup/AndroidSdkSourceAnalysis)

如果你有时间，你也可以参与到这个项目中去。

#### 4. 梳理一下知识体系，找出薄弱环节

总结、总结，再总结。

总结是一个人非常重要的品质，在总结中思考自己的不足，完善自我。不仅适用于面试，也适用于职场，日常生活等等。常思考，才可能变成一个智慧的人。

具体有哪些可能的薄弱环节，可以看下一篇整理的30个发散性问题。

### 简历的准备：

写好简历也是一个重要的环节。程序员最好常备一份简历，以备不时之需。另一方面，更新简历的过程中也会明白，哪些经历是无用功，哪些是加分项。

#### 有几点

1、尽量避免主观表述，少一点语义模糊的形容词，除非是大公司大牛，已经有成果撑腰，否则慎用「熟悉…」、「使用过…」

2、多一点表意清楚，语气肯定的数量词、名词、成果描述。一定要将自己的优势和期望明晰地表达出来，便于招聘方能对候选人有更准确的定位：

* 介绍技术：最近几份工作经历中所参与过的产品、项目、角色
* 在工作中做的项目的技术细节
* 克服过的技术难点与细节
* 感兴趣的技术
* 在程序马拉松上参加的项目或者是业余的个人项目（+link）
* 如果领导过技术团队，写下带的团队的规模与管理风格
* 介绍自己：过往有特点经历、擅长的方向、对互联网的理解、职业发展规划

3、突出重点。重要的放前面，不重要的经历往后排。尽量注明每段经历的时间，不然技术官或hr会一头雾水。比如你面试 Android开发，iOS的经历就放后面一笔带过，另外时间越远的优先级也越低。对了，记得把联系方式写在开头，不然对方联系不到你，可就把你忘了。

3、试试用markdown语法，注意下排版，预览再提交，版面整洁、干净，也是加分项。

4、HR/技术负责人更喜欢看到一份显示「职业上升趋势」的简历。比如你做开发五年，跳过两三次槽。那么你的职位，你所做的技术难度应当是越来越具挑战，而不是依然罗列各种琐碎的需求实现。

5、牛人讲结果，普通人讲过程。话虽如此，你只要不太啰嗦，该列出的事情还是要列出来的。不仅如此，你在该项目中发挥了什么作用，有什么反思，也可以写出来。记住，你出卖的不仅是技术，还有能力。

#### 千万别…

1、真实的反映你的工作，别浮夸。 因为你的面试官之后一定会问起浮夸的事情，假如你不能自圆其说，会留下非常不好的印象。

2、记得最好不要出现“**精通**”等词汇。 很少有人能精通JAVA，精通数据库，这样的字眼基本等于自杀。好的办法是阐述事实。比如用JAVA开发过什么系统，在什么项目中深入接触数据库等等。

3、简历上的所有技术，你必须谙熟于心。 甚至你应当练习几遍，找出可能发散出去的所有问题。只有你非常熟悉，并且在面试中足够自信，你才有能力引导面试官问到你想要的坑中。

4、简历长度不要超过三页。最好是两页这样一目了然，当然如果从业经历比较长，也可以多写点。

5、别撒谎。天下没有不透风的墙。假如你不想让人知道什么，你可以不写；假如你在原公司的锻炼机会太少，那么你可以将经历写到自己的开源项目中去。

## 三十道android开放题

为姗姗来迟的安卓面试题表示抱歉。 大家可以先思考一下这些问题，答案会过一阵在后文放出。

### 基础题

* Activity生命周期和启动模式，以及使用场景
* Service的生命周期，如何启用/停用Service
* 可能导致OOM内存溢出的情况有哪些，怎么解决
* Android中的动画有哪几类，它们的特点和区别是什么
* 注册广播几种方式
* 如何自定义View
* 程序crash的所有可能原因
* android动画
* Android操作系统分层

### 中级题

* Android事件处理机制
* 内存泄露的案例
* 图片加载库的使用及比较，内在逻辑分析
* 网络库的使用和比较
* android线程间通信
* android进程间通信。Binder源码与Looper关系等
* 设计模式在android中的应用，手写几个典型模式
* annotation 以及java反射机制
* Java虚拟机
* Java几种线程池

### 高级题

* 热修复技术，原理及其应用
* react native
* 几种架构模式
* 新建一个工程，需要加入哪些库和模块，该怎么设计

有关测试：

* DDMS + MAT
* monkey test和logcat命令的常用过滤参数
* android作优化有哪些可以考虑的方向？

### 开放题

* 当我在浏览器中输入一个url，世界发生了什么。
* http和https的区别
* 几种加密方式区别，对称与非对称。
* 你还了解什么其他的语言？和Java对比一下。

原文链接：冰洁的技术博客，http://www.bingjie.me/