

20 | 考官面对面：我是如何面试程序员的？

2019-03-15 杨波

面试现场

[进入课程 >](#)



讲述：迈迈

时长 15:09 大小 13.88M



你好，我是杨波，拍拍贷研发总监，资深架构师。一晃在软件研发行业待了十五年多，不管是之前在一线参与实际项目研发，还是近年从事架构和管理工作，工程师招聘一直是我的工作重心之一。回想我面试过的程序员数量，即使没有一千，也一定不会少于八百，如果说我面人无数，我觉得一点也不夸张。

面试面得多了，自然会形成一个自己的面试体系，有自己的面试路数，最近几年我招人基本一直沿用这个路数。而通过我面试的程序员，事后证明他们的工作表现也比较出色，说明我的这个路数还是比较靠谱的。今天我想把这个路数以及其背后的逻辑分享给你，希望对年轻工程师准备面试提供一些参考。这个面试路数并不复杂，关键看三点：

编程能力（必须），考查候选人能不能实际动手写程序，有没有基本的编程思维。

技术技能经验（必须），考查候选人之前参与过哪些实际项目，积累了哪些实际的技能经验。

软性能力（参考），考查候选人的沟通合作能力如何，心理成熟度如何，管理和领导力如何。

下面，我就这三点展开分享。为简化说明，本文假定前提是面试 2~6 年经验的程序员，以 Java 技术栈为背景。如果在这个范围之外，虽然面试的内容、侧重点和能力要求会有所不同，但是核心考查点基本还是这三点。

另外也特别说明，由于我主要从事基础架构类研发工作，对基础架构类程序员的总体技术要求，会比一般应用类程序员高。

一、编程能力考查

我们招聘程序员主要是来干活的（说白了就是写代码），不管你之前多么有经验、有资历，我一定要考查你的实际编程动手能力，同时也考查你的编程思维和代码质量。下面是我常用的两个面试题示例，一起来看一下。

问题 1：请写出求二叉树深度的算法（我会给出二叉树图例和节点的定义）。

这个问题表面上是看候选人对二叉树数据结构的掌握程度，实际上我考查的重点是**递归思维**。因为这个问题如果用递归算法解的话，5~6 行代码就可以解决，但是没有掌握递归思维的候选人一般会陷进去，要么搞半天也写不出来，要么花半天弄出一个很复杂的非递归算法。

递归是一种很强大的编程武器，开发工作中一些看似非常复杂的技术问题（例如语言解析器、代码生成器等），都可以通过递归简单优雅地解决，而没有掌握递归的程序员则对此类问题常常束手无策。

最后，不管候选人有没有解出这个问题，我都会仔细看看他写的代码，看看程序组织、变量命名，甚至标点符号等细节，这些不仅可以看出他的代码质量，同时可以看他是否细心认真（程序员的基本素质）。写程序如绣花，一针一线不容马虎。有些候选人能把算法思路讲出来，或者把伪代码写出来，但是实际代码写不好，说明他写实战代码的能力还是比较生疏。

问题 2：用一台普通电脑对一个 100G 的大文件（内部是字符串记录）进行排序，本题只需给出算法思路。

有些候选人数据结构掌握较好，上面第 1 题能够轻松过关，这个时候我会进一步用问题 2 来考查。

问题 2 看上去规模很大，实际考查点也更多，但是核心考查的是**分治思维**（分而治之）：先将大问题分解为子问题，依次解决子问题，最后再合并子问题的解得到最终解。分治思维也很强大，能帮助我们解决很多表面上看起来很庞大的问题，例如一个中大规模系统的架构设计，甚至一个企业组织的架构设计都离不开分治思维。

所以我用一个技术问题考查候选人是否具有分治思维，也就是看他是否具备解决大规模问题的思维基础。另外，这个问题同时也考查候选人对计算机组织（内存）、文件操作、算法时空复杂度，以及各种排序算法（快排、归并等）的掌握程度，可以说是一举多得。

二、技术技能经验考查

技能经验本身不容易考查，行业的一般做法是先考查你对 Java 语言的掌握程度，因为如果你实际做过生产级的项目，那么对 Java 语言的核心知识点（Core Java）就应该掌握得比较好，下面列举两个我常问的 Java 面试题。

问题 3：介绍 Java 垃圾回收器的基本原理。

Java 语言最著名的特性就是自动垃圾回收（GC），很多生产上 Java 应用的问题（尤其是性能和内存问题），大都和 GC 有关。对于一个实战型的 Java 程序员来说，他在生产实践中一定会不止一次地遇到 GC 问题，所以对 GC 的理解程度，基本可以作为衡量一个 Java 程序员是否合格的门槛。

按我的要求，他至少要把 Java 的分代垃圾回收机制讲清楚，否则差不多就可以判定“不合格”了。对于有些能讲清楚的候选人，我还会进一步延伸，让他解释什么是 GC Root，解释 Java 程序中有哪些可以作为 GC Root？甚至深入到如何直接操作堆外内存（offheap）的程度，进一步挖掘这个候选人的技术广度和深度。

GC 的基本原理可以背诵，但是对于延伸的知识，如果没有拓展阅读或者实际动手实践过，你一般很难回答上来。

问题 4：介绍 HashMap 的实现原理。

在我们日常编程的中，除了 List/Array 以外，用得最多的数据结构基本上就是 HashMap 了。另外，在大学时 HashMap 也是数据结构课程中必学的一种基础结构（课本中称

为“散列表”）。因此，我基本上都会问 Java 的 HashMap 是如何实现的。按我的要求，候选人至少要把 HashMap 的内部数据结构实现和 get/put 算法给讲清楚，好一点的候选人要能把装载因子（load factor）和再哈希（rehashing）给讲清楚。

这个问题可以考查三个点：

1. 你是否真的在一线写代码，因为只要是写代码就一定绕不开 HashMap。
2. 对这么常用的数据结构，你是只停留在会用 API 的层面，还是说能够深入到内部算法实现？
3. 你在大学时有没有认真学习过数据结构，对这门核心计算机课程的态度如何？

对于能讲清楚 HashMap 基本实现的候选人，我还会进一步延伸，既可以向 Java 集合框架方向延伸，也可以向 Java 多线程方向延伸，其中涉及的内容很多，例如 HashTable、ConcurrentHashMap，甚至 TreeMap、ConcurrentSkipList 和 ThreadLocal 等。HashMap 的基本原理并不难，但是延伸的内容如果没有实际动手实践过，你一般很难回答上来。

除了 Core Java 之外，我也常常会问一些软件开发高级问题，例如和 Web 应用开发或者设计模式相关问题，下面同样给出两个例子。

问题 5：讲讲 Web 应用程序中的 Cookie 和 Session 是什么关系？

互联网应用大部分属于 Web 应用，开发人员或多或少会开发 Web 应用，而 Web 应用开发中一个很基本的问题是理解 Cookie 和 Session。有很多候选人简历中写精通 Web 开发，我就会问他们 Cookie 和 Session 是啥关系？实际上比较遗憾，真正能回答清楚的并不多。

这个问题需要一些底层 HTTP 协议知识，理解 HTTP 是无状态协议，理解服务端和客户端会话存储机制。对于能回答清楚的，我会进一步扩展，Session 可以存储在哪些地方？客户端？服务器端？缓存？还有数据库？这些存储机制有何利弊？粘性会话（sticky session）是什么意思？如何实现粘性会话？分布式会话如何实现？诸如此类，全面考查候选人 Web 应用开发和分布式架构的实战经验。

问题 6：你在生产实践中使用过哪些设计模式，是什么场景下使用的？为什么使用？

随着一个工程师项目经验的积累，他会逐渐体会到软件设计中有很多模式性解决方案，也就是所谓的软件设计模式（Design Pattern）。

设计模式是软件研发行业几十年实践后提炼出来的一些通用的设计方案，是比数据结构更高一层的抽象，学习和应用设计模式可以帮助我们提升软件设计能力。我认为，工程师能力提升的一个标志，就是会开始注重设计模式的学习和积累，并且在实际项目中应用模式解决具体问题，以提升设计效率和质量。

因此，我在面试中一般也会询问设计模式相关的问题，比如问候选人在实际项目中应用过哪些设计模式（因为工厂和单例太常见，我一般会要求两者除外的答案）？什么场景下使用的？为什么要使用？对于有模式使用经验的候选人，我会要求横向比较几种相关模式，例如策略（strategy）和模板方法（template method）模式的差异，代理（proxy）和装饰（decorator）模式的差异等，进一步挖掘他对模式掌握的全面性。

总之，对这个问题的回答，可以看出候选人项目经验是否充分，是否已经上升到模式性解决问题的高度。

问题 7：讲讲你最近开发的一个项目的业务背景和架构（或技术）。

这是一个开放性问题，考查目的并不在项目本身，而是看三点：

候选人近期是否还在一线参与实际项目？

候选人的沟通表达能力，能否用简单的语言，给一个完全没有上下文的人，讲清楚自己开发（或者参与开发）的一个项目的业务背景和架构。该问题也是后面软性能力考查的重要参考。

候选人的技术热情，但凡有技术热情的候选人，一讲到自己开发项目的架构或技术时，常常会两眼放光，滔滔不绝，充满激情。

三、软性能力考查

通过前面两关的考查，基本上可以说候选人的硬技能过关了，硬技能过关，说明候选人能独立干活，可以说是过了面试的门槛。最后一关是考查候选人的软性技能，也就软技能。

软技能考查主要有两个目的，一个是筛选掉那些明显有合作性问题或者心态问题的候选人；另一方面是考查候选人的管理和领导潜质。有些候选人不仅能独立干活，而且还能带团队，有潜力成为一个管理者，甚至是领导者。

当然，软技能考查的点非常多，我主要关注候选人的沟通能力、心理成熟度，以及管理和领导力等方面。

1. 沟通能力考查

其实在上面问技术问题的时候，我就已经开始观察候选人的沟通表达能力了（比如第 7 题）。同样一个问题，沟通表达好的候选人能够抓住重点，有条理地用简单的语言把问题讲清楚，而且面试过程中能够认真倾听面试官的问题，并且和面试官保持良好的互动。相反，沟通表达不好的候选人，要么表现非常被动，话讲不清楚，和面试官互动少，要么非常啰嗦且抓不住重点。

2. 心理成熟度考查

我们招一个人进入公司和团队，是要能够长期一起合作的（说白了是一起吃饭），需要心理成熟的成年人，而不是动不动就要小脾气的未成年人。

不少技术人员硬技能是有的，但心理不成熟。心理不成熟有很多表现，比如自负任性，喜欢单干合作性差，喜欢扯皮不愿意担责，言语经常传播负能量等。这样的候选人如果招入团队，可能造成一个烂苹果毁掉一篮筐的问题，所以必须谨慎！

在整个面试过程中，表面上我问的是技术问题，但时刻都在通过察言观色，对候选人进行心理成熟度的考查。有的时候，我会和候选人一起探讨问题，过程中看他的合作性和积极主动性；有的时候，我会故意否定或刁难（甚至带一点蔑视的态度），看候选人在压力环境下的反应。没有压力的时候，大家的表现都挺好，一旦在压力或逆境中，很多心理上的小毛病就都暴露出来了。如果在压力环境下仍然能做到镇定自若，冷静思考并解决问题，往往在一定程度上表明候选人心理成熟稳定，有一定的责任心和担当。

3. 管理能力考查

做工程搞开发，不仅需要良好的技术技能，同时也需要良好的计划性和时间管理能力，当我们做中大规模的项目，或者是作为管理者带团队的时候，这种能力就显得格外重要。

不管之前有没有做过项目管理或者是团队管理，我都会问候选人你是怎么管理你的时间和项目的？你每天 / 周 / 月 / 年是如何做计划的？自己有没有一套完善的计划和时间管理系统？比较可惜，只有非常少数的候选人能够清楚说明自己的计划和时间管理系统，这也可以说为什么能从程序员成长为架构师甚至管理者的人少之又少。

4. 领导力考查

我们招一个人，有能力能干活是基本要求，但是如果只能按部就班地执行，没有领导创新力，那么做老板的就会很累；相反，如果员工还能积极主动创造性地干活，时不时搞点意想不到的新东西出来，那老板可以省很多心，招到这样的员工是老板之福。

领导力并不容易考查，一方面通过面试过程中候选人的沟通表达和积极主动性来考查，另一方面也可以提问，比如让候选人讲讲在之前的职业经历中，在缺少资源和支持的情况下，自己主动发起项目，最终还把事情搞成了的案例。

总结

上述就是我面试程序员的主要路数，以及背后的面试逻辑，最后我再额外补充和总结几点：

- 1. 知之为知之，不知为不知，面试过程中如果有问题回答不上来，那就诚实讲不知道（或者自己没有相关经验”）。我们不可能每个问题都回答得上来（面试官也并不对此抱多大期望）。即使面试失败了，也建议你坦然面对，这不过是一次很好的学习机会，回去后查漏补缺，下次再看新机会即可。但是如果不懂装懂，就违反了诚实正直的价值观大忌，你在面试官心目中的形象会大打折扣。**
- 2. 整个面试重点考查候选人的实战能力（说白了就是干活、产出的能力），这个不太可能通过研究面试攻略搞突击来完成。你还是要平时一步一个脚印，多参与实际项目，多写代码，多阅读代码和图书，不断学习和提高。**
- 3. 数据结构和算法、编程语言（Core Java）、编程思维（递归和分治）、Web 开发和软件设计模式等，是我考查程序员硬技能的重点，我想也是很多其他公司面试候选人的重点。**
- 4. 面试过程中，要注意面试官同时也在考查你的软性技能，时刻保持谦虚谨慎，回答问题前先思考，再有条理地表达出来，过程中和面试官保持良好互动，同时表现自己的积极主动性和合作性，展示自己心理成熟、负责任、有担当的一面，也尽可能展示自己有管理甚至领导力的一面。**

说了这么多，希望你能面试成功，进入自己喜欢的公司。如果今天的文章让你有新的收获，也欢迎把它分享给你的朋友，一起探讨。



面试现场

面试只是起点，能力才是终局

白海飞

IBM 软件商务系统经理



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金奖励**。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 19 | 如何认识自己的优缺点？

下一篇 21 | 透过兴趣爱好，面试官可以看出什么？

精选留言 (16)

写留言



观弈道人

2019-03-15

8

非常棒的一篇，搞的我要去买他微服务视频课程了。

展开 ▼



果然如此

2019-03-16

6

这篇文章很值得踏实干活和想跳槽的程序员参考。

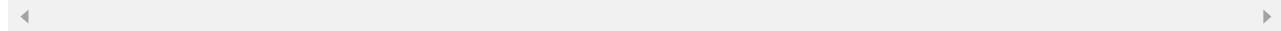
我的经验是面试需要学习的，只干活还不行，因为面试的问题多数干活中是用不到的，而且很多项目是关注结果，不关注过程，如果程序员没有经过大量的面试失败经验，大多数会倾向使用层面，而我也最近面试很多其他人。

所以，我认为如果想面试成功只是踏实工作不行（亲身经历），必须花大量额外时间学...

展开 ▼

作者回复: 你说得也很棒。

但是, 你觉得面试中的很多问题, 工作中是用不到的么?



大海

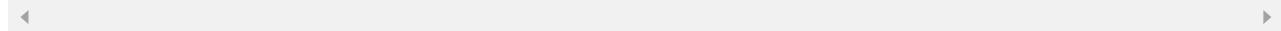
2019-03-15

1 6

这篇文章是目前为止最经典的文章没有之一, 凌晨一点竟然情不自禁的读了3遍, 您讲的考察细节对我来说非常重要, 虽然主要从事.net开发, 但是很多东西都是一个道理

展开 ▼

作者回复: 同感, 的确要感谢杨总的精辟总结。



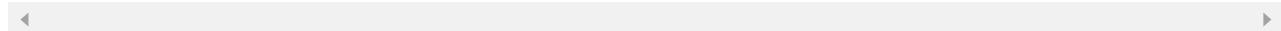
洛叶

2019-03-20

1 2

考察一个人的心理成熟度太重要了, 我领导根本就不成熟, 在他那我做什么都是错的, 好几次想离职。

作者回复: 根据领导的风格和个人的目标, 选择开诚布公地谈, 或者小心谨慎地干, 或者干脆利落地走。



果然如此

2019-03-28

1 1

回复:

去其他公司面试问的问题我们公司没有遇到过, 是因为我们的项目还没达到一定的访问量级或数据量级, 但是, 这不应该成为面试被拒的理由, 最主要还是靠自己平时学习和总结, 刚工作最初几年, 虽然求知若渴, 但不知道如何有效的学习、没抓住学习的方向。

展开 ▼



WizzXu

1 1

2019-03-15

虽然作为安卓开发，但是依然领会了特别多。基础，让我更明确了哪些是基础。思想，这点我感觉特别重要，一些思想是没有过经历的人无法编造的。总结，项目中的东西要总结，面试官要看到的是有条理的。规划，这一点确实有欠缺。今天真的学了很多，非常感谢大佬！



lamos 哟...

2019-03-15

1

涨知识了，感觉自己很多不足。非科班出身的我都挺关注这些知识点的，通过这篇面试让我有很多学习基础知识的决心...



不靠谱的琴...

2019-03-15

1

一般面多少年的需要了解jvm gc相关的，因为我遇到很多7-9年的都没有相关概念。

作者回复: 可见掌握技术的深度，跟年限没关系。再者，技术方向不同，用不到，也没有机会深入下去。

◀ ▶



Python d...

2019-05-12

1

大佬，我是应届毕业生之前实习接触过grpc zookeeper,现在面试刚好可以吹一下子哈哈。没想到又见到了杨波大神，我快看完你的20课程架构视频课程了但是160课程看起来有点吃力 我过一段时间再买。

杨波老师，是我的男神 表白。

作者回复: ↗

◀ ▶



刘哲

2019-05-06

1

看到上面100G文件的问题，我想问下，机械硬盘和固态硬盘多线程读写会加快速度么？机械硬盘和固态硬盘关于多线程的原理分别是怎样的？



AaronYu
2019-03-19



写的真好，这也是我们平时学习中要重点关注的点啊。

展开 ▼



张小盒...
2019-03-19



讲的好好 😊

展开 ▼



Elliot
2019-03-19



这篇文章堪称压轴之笔

展开 ▼



Elliot
2019-03-18



药石之言 当名肺腑

展开 ▼



LDxy
2019-03-15

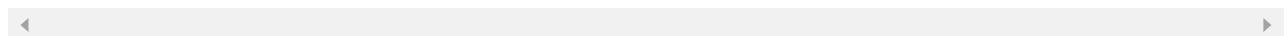


关于心里成熟度，这和人的性格有很大关系，是不容易改变或隐藏的，有时候就算知道怎样做会好但也没啥办法。就像一个内向的人是很难变成外向型的，反之亦然。

展开 ▼

作者回复: 嗯，相对于技能提高速度，心理成熟度是比较稳定的，不容易改变的，除非内因驱动、意识上发生改变。

工作中，我更多说的是职业成熟度。职业成熟度包括了职业技能和职业态度等方面，是在心理成熟度之上的表现。比如，能否协作、能否全局角度考虑问题、能否照顾长远利益等等。工作刚开始的五、六年是职业成熟度发展的关键期，





Feng

2019-03-15



老师总结中的第三点，就是自己特心虚的点，还是自己基础不够，希望自己可以进一步成长，并且在学习的过程中，通过让学习成果落地来提高转化率，分享读书和学习笔记，无形让自己保持总结，思考，分享的习惯。

展开▼