

你知道的越多，你不知道的越多

点赞再看，养成习惯

本文 **GitHub** <https://github.com/JavaFamily> 上已经收录，有一线大厂面试点思维导图，也整理了很多我的文档，欢迎Star和完善，大家面试可以参照考点复习，希望我们一起有点东西。

前言

你们都知道的，帅丙我本身肯定是达不到技术专家的地步的。

呸，敖丙你个渣男，又标题党！



我不是但是我身边很多人是呀，字节跳动，PDD，阿里等等的大佬我认识很多，所以以下是我问了他们的成长路径的总结，还是具有一定的代表性。

身边这些朋友都是在互联网领域多年的大牛了，因为生活和个人习惯的原因，很多都没有写博客的习惯，或者没有公开出来，但是他们却遇到了帅丙，我骚啊，我可以帮他们整理出来啊。

目的是为了很多跟我一样的新人，对一条完整的职业规划之路有一个清晰的认知，青铜到王者要经历些什么？



青铜-万恶之源

《计算机基础》，这是所有读者大学最开始都会上的课吧，我问了群里的仔，他们都说是的，我想大家也是。

在计算机基础中我们会学到计算机的历史、计算机的特点、进制转换、内存管理、线性数据结构、网络请求协议等等。

计算机基础真心很重要，无比重要，究极重要，为啥我这么说呢，里面涉及的计算机知识还有很多网络协议的知识，大家以后一定会用到，我可以负责的告诉你，面试也一定会问，什么Http、Https、Tcp/Ip、三次握手、四次挥手面试不要太常问。

Tip：这里有个小插曲就是我身边那个架构师团队Leader这周面试阿里p8岗位的时候，我以为问的都是什么源码，中间件的究极操作，我想大家跟我想的一样，但是大家错了，反而问的很多计算机的基础知识。

我聊天大概问了一下面试内容，有什么求二叉树的镜像，内核态和用户态的理解，计算机的缓存页等。

大家是不是惊了，这他*不是我校招的题目么？怎么P8级别的大佬还问这个？其实很好理解，大家想一下到了一定的工作年限技术的广度深度都有一定的造诣了，你写代码就这样了没办法优化了，机器配置也是最好的了，那还能优化啥？

底层，我们都知道所有语言到最后要运行都是变成机器语言的，最后归根究底都是要去跟机器交互的，那计算机的底层是不是最后还是要关注的东西了？

福利：他这周会来我们公司面试，届时我会详细问一下具体的面试内容，如果大家想看我可以到时候整理一下写出来，看看那种级别的强者的世界到底是怎么样的。



学完计算机基础你基本上对计算机的整个知识体系都一知半解了，对于编程或许还是很懵懂，但是你现在已经要踏入编程的门槛了。

白银-编程第一课

计算机基础学了，后面就要编程的入门课程了。

想必看我这篇文章有99%的小伙伴都是以下面这段代码开始的自己的程序人生的吧：

```
/**
 * 万恶之源
 *
 * @Author: 敖丙 JavaFamily
 */
public static void main(String[] args) {
    System.out.println("Hello World");
}
```

我还记得**帅丙**我就是大一开始学习的C语言，《C语言程序设计》谭浩强老师的课本，在第27页就是第一章，**HelloWord**，现在回忆起来还历历在目，老师在黑板上写下这两个庄严的单词，同桌的她和我露出无限的求知欲.....

我想所有的语言都是语法基础开始的，而且编程语言的基础语法也都是大同小异的。

If、For、While等关键字，List、Set、Map等集合，Int、float、Double等基础类型，大家第一次学的时候我想也不会知道，这些关键字将会陪伴我们走完各自的程序生涯吧？

接下去就难一点比如JDBC、IO、文件流啊什么的，帅丙依稀的记得当时还是**代码里面写SQL**然后查询，也不知道记错没有。

学到这里，你可能觉得你**JavaSE**无敌，出去可以找工作了，反正当时的帅丙，觉得自己可以闯出一片天了，不知天高地厚，四处炫技。

但是幼稚的我根本不知道，**未知的前方还有什么等着我**，我也不知道自己其实才刚入门，但是如果往WEB方向发展，这些倒是基本足够了。

但是你到现在为止学的东西都是服务端的东西呀，怎么让你的服务端东西展示给别人看呢？

这个时候我们就应该去做个丑不拉几的页面，进行简单的交互了，你这个时候要学**HTML、Servlet、MySQL、JavaScript、Tomcat、CSS**等。

目标呢就是写出第一个动态网站，也许只是个登陆功能，只能展示下个人资料，但这是很重要的一步，你要弄清楚的是，一个用户的点击产生的请求，是从哪里发起，哪里接收，哪里处理，哪里返回，你得理解浏览器和服务器的关系和分工，**Cookie**和**Session**，**Request**和**Response**。

Tip：我记得我刚出来面试就有面试题是这样的，一个Http请求从网页发起到服务端产生数据返回中间经历了些什么，或者Spring做了什么？大家可以思考一下

还有一个自己的项目真心很重要，敖丙我就是在大学期间做了项目，至今在校园内网上稳定运行着，出来实习面试的时候基本上也是一路披荆斩棘，吊打了同行面试的仔，反正大学有个项目贼加分，是读者的大学生记住了么？

黄金-初出茅庐

上面的只是WEB开发的学习初级阶段，这都是些JAVA诞生以来最原始的最官方的WEB开发技术，当然现在真正的项目很少直接采用这些技术了，现在都是**前后端分离**了，Vue、Node.js、React等前端的语言更新迭代速度非常快，帅丙我学了一点刚用舒服，前端的朋友告诉我又迭代了.....我他*不学了！

为了不断提高技术的易用性、开发效率和可维护性、可扩展性，无数开源项目都是在这些原始技术的基础上封装、改进。

所以这个阶段**不要盲目乐观地跑去找工作或者对实习挑三拣四**，你会被打击到的，又或者找个小公司浑浑噩噩几年过去感觉跟新的技术栈脱节。

有人跟你说小公司学的东西多，他害你的，刚开始能去多大的去多大的，越大越好，你想你去了大的你想去小的分分钟的事情，但是你小的要去大的就相对难了。

好，初始阶段完成，开始进入WEB开发的正题，首先是传说中的框架，SSM（Spring SpringMVC Mybatis）成为熟练的增删改查程序员是必须的，在这个阶段你还要理解为什么要用这仨而不是那些Jsp、Servlet、Jdbc，你要体会到写MVC、三层架构的好处。

这个阶段不要轻易质疑框架的价值，也许刚开始你会觉得麻烦，觉得他们有时候是多此一举，我最开始这么写的时候发现，真的恶心，什么dao，service，Controller等等，分这么多干嘛。

等过一两年后回过头来你会觉得少了这些框架你已经很难干活了，没有分层清晰的系统，你反而开口大骂项目的架构师的。

你*****



不过要提醒大家的是，这个阶段还要避免的心态是，能熟练地增删改查了，就自认为写程序不过如此，然后往简历上填个精通，这也是新手面试被批得体无完肤的原因之一。

浅尝即止，是新手的大忌，为啥这么写，这么写有啥好处，有啥坏处，多问几个为什么，你多年后会感谢敖丙的。

如果你学得好，这会应该能熟练地写个博客啊，小论坛啊之类的WEB项目了，也就是达到了就业的基本要求。

Tip：而且说一下作为一个应届生，你除了基础知识，如果你有自己的技术博客，还有像样的项目能展示给面试官看，是真的很加分，搭建个简单的项目，项目流程你也知道了，技术栈也熟悉了，还可以手机访问网页给面试官现场演示，很加分的。

像我身边的朋友Java3y，JavaGuide，我没有三颗心脏等，就是在这个阶段就已经开始写了自己的博客，并且各自都做出了成绩，这些以后都是你面试的议价能力的一部分，也是你的加分项，和你自身的财富。

我问了一下他们，都是靠博客就已经得到不少大公司的面试邀约了，反正有项目，有不错的履历，有不错的博客，都会给你加分，如果你什么都没有，现在动手去多学点，少打两局LOL等你到我这个年纪就会发现，索然无味。

铂金-遵义会议

这会是程序员生涯的一个**转折点**，把握好了人生起飞，把握不好**全盘皆输**，所以我用在我老家遵义举办的**遵义会议**，作为这阶段的标题，我觉得，**妙啊**！



慢慢的你会发现数据库性能实在不行，出于不甘现状，或者赶时髦，你该去了解**NoSQL**了，**Memcached**、**Redis**、**Mongodb**等非关系型数据库眼花缭乱，没关系，试着用用，能很好地缓解**MySQL**、**Oracle**之类关系数据库的压力。

出于公司某些业务需要，你可能会钻研一个特定技术领域，比如全文搜索技术**ElasticSearch**（以下简称**Es**），那你了解了**Es**，你又要想到数据库的数据怎么同步进**Es**呢？

你可能会接触到**Logstash**，**Canal**等中间件，然后发现可视化也是个问题，那**Kibana**就应运而生，用的时候发现欧洲人的分词习惯跟我们不一样，那**Ik**中文分词器又得了解啦。

使用之后你还会发现他底层的**Lucene**有很多坑要怎避免，时间多你还可以了解**Solr**等等。

总之知识就是一个体系，我经常跟群里的仔说，形成知识体系，你面试说了**Es**那相关的技术栈你一样要了解的，我题目取了叫《吊打面试官》就是噱头，你去面试如果面试官技术深度真的很深，我想被吊打的100%是你。

你甚至会开始对系统中一些比较特殊的存在感兴趣，比如**权限系统**，**单点登陆**之类的，又或者某些特定业务领域的**算法研究**，这些是你的加分项。

你还发现服务拆分后**Http**通信有诸多弊端，就开始接触优秀的**Rpc**框架还有消息队列中间件了，如**Dubbo**、**RocketMQ**等。

再再后来你发现呀，数据量大得一批，表顶不住了，几亿数据查出来要好几秒，那**分库分表**就出来，什么**水平拆分**，**垂直拆分**，还学习了**TDDL**、**Sharding-JDBC**、**DRDS**等这样的分库分表中间件。

但是你用了发现全局的唯一id生成又是个问题，或许中间件有自带的，但是你还还是要了解原理，什么**雪花算法**，**uuid**等等也得学。

再再再后来呀，你发现分库分表也顶不住了，业务体量爆炸式增长了，你可能就需要了解动态分库分表的解决思路和解决方案了，特别是**FaceBook**开源的一些方案。

再再再再后来呀，你发现动态分库分表也不行啊，很多离线的数据也很多啊，每天几个T，公司都要被败光了，那你就需要了解大数据场景的离线分析啊，数据缓存啊，数据清洗，数据可视化等等啊什么的。

那就需要学什么**ODPS**啊，**Hadoop**、**Hive**、**Hbase**等等中间件或者工具了。

再再再再再后来你开始发现你的代码很乱，久了以后自己都看不懂，重复的，难以重用的代码堆积如山，不想维护，BUG百出。

于是你要开始重视设计模式，合理地改造下自己的代码习惯，不再被僵化的SSH、MVC三层架构束缚住。

再再再再再再.....（敖丙你是不是没玩了？哈哈其实还真有我就不列举了）



到这里不知道你有没有体会到我每篇文章开头那句话的意思？敖丙就是工作之后发现自己越来越无知了，你再品一下下面这句话。

你知道的越多，你不知道的越多

Google和各种资料是你进步的动力，极少再遇到必须请教别人才能解决的问题，如果你这个阶段还老是问别人，你的技术生涯也就快到头了。

这个阶段，如果你**技术卓越**和**跟敖丙一样能吹**，你的收入将是白领水平，至少接近了，或者大幅领先同龄人了。

我觉得多数程序员在工作多年之后都能达到这个水平，有的人只需要一两年，有的人却要用上五六年，在这个阶段落伍的人，有的是出于天赋和思维所限，有的是出于对技术缺乏热情，有的是出于工作内容的制约。

等到年近中年，再也拼不过年轻人，被淘汰出局，只能在自嘲为屌丝和码农的无奈中黯然转行。

这是个很重要的分水岭，你能不能继续进步，能不能在30岁以后继续从事技术工作，能不能在公司里独当一面，我觉得就看你能不能超越这个阶段了。

很多烂公司里工作数年的项目经理，连这个层次都还没达到...

为了30岁的自己听到裁员完全不虚，为了家里的老母亲不再为了省电费不舍得开灯，为了让自己.....

我想这个阶段你应该要做好准备，这也是我大篇幅介绍这个阶段的原因。

砖石-扬帆远航



你要读读优秀开源项目的源码，你要搞懂一些当年不懂的基础知识，你开始理解《thinking in java》的精髓，你能写一些底层的代码，有时还会觉得自己封装的比某些开源工具更好用更简单。

当年看不懂的《深入分析JVM虚拟机》现在你已经可以对里面的知识点如数家珍，张口就来，并且能够将大量知识点带入到项目中优化，并且能够看到实质性的变化（暗示KPI）。

WEB的难点和重点永远都在于性能、负载能力上，而现在网络的发达造成了数据量和操作密度的大大上升，硬件却没有相应的进步。

你得试着更好地运用更多的服务器来协同工作，从WEB端到服务端到数据库，全都需要集群，需要分布式，需要合理的控制数据的流向，掌握好网站上下，一大堆机器的平衡，找出性能的瓶颈，找出稳定性和安全性的瓶颈，硬件出现故障，第三方技术出现错误，将被当成家常便饭融入到你的系统和代码里仔细考虑。

你会开始觉得方案无比地重要，一将无能累死千军将不断应验，一个不好的设计，一个不好的方案，会让一群优秀的程序员工作成果大打折扣。

你要关注架构知识，不能再满足于SSH三层架构到底。

领域驱动设计，面向事件开发，敏捷开发等等一系列的思想在关键时刻能决定你项目的生死，这个阶段不再有标准范例让你照抄，你只知道思想和原理，实践却需要自己不断尝试和改进。

多关注各种杂七杂八的开源技术，有些你可能前面已经接触过了，和通信有关的，和集成开发环境有关的，和架构有关的，各个领域你都应该能信口说上几个主流技术，虽然你可能只是听说过，了解。

但关键时刻你得知道如何去选择技术，并快速掌握它。

你还会去考虑尝试下别的语言，这里不是说转向什么C++ C#之类的，那和JAVA程序员不相干，我说的是一些**运行于JVM之上的语言**，比如scala和groovy，初识他们时你会觉得Java真的老了。

但当你回到一个综合性的大型项目中，又会觉得Java积累下来的整个体系技术是如此完善，就像一个工业化标准一样，你可能发现光是Spring家族的东西你都受用终身，无法完全通读。

你能把这个阶段实践好，胜任项目经理，乃至中小公司的技术总监，大厂的小团队Leader都是可以的。

大师-登峰造极

其实写到上面这个阶段的时候我觉得，这是很多人满足的地方了，都是小公司技术总监了，那我图啥？

但是未知的人生还在那等着你呢，我问了身边顶级的大神，**为啥要走出舒适区**？去像字节、阿里等这些压力大很多的地方呢？

他们给的回答都不太一样，但是一样的就是**挑战自我**吧，**谁也不知道哪里是自己的终点**，那为啥不多做点尝试，新的环境，新的技术栈，新业务场景挑战新的自我。

这个阶段你的一举一动可能都关系一个项目的生死存亡，一个错误或者正确的决策就可能改变整个项目的命运，水能载舟亦能覆舟，我想用在这里也恰到好处。

我身边这样级别的大佬凤毛麟角，但不是没有，他们在公司都是核心人物，大型项目或者项目紧要关头都是他们带领团队**冲冲冲**，除了问题也是能最快给出解决思路和方案的。

Tip：我现在的老大就是这样的人，双十一大家还手忙脚乱去追数据的时候，他上来一套操作，写了几个脚本就搞定了，卧*我们当时周围一群人，**从头到尾的知识盲区**，结束了还没反应过来，只能拍手叫666。

这就是强者的世界，我所向往的世界，当然我知道这样的世界，只有一步一个脚印才能涉足。



王者-泰山之巅

最后王者其实我不会写任何内容，能到这个领域的人本身就是屈指可数了，敖丙身边有，但是我觉得大家自己体会吧，一般就是人脉，交际，能力都到了一定的高度了，这个阶段的事情我也体会不到。

能想到的就只有先祖的诗句：指点江山，挥斥方遒。

算是给大家留下无限遐想的空间吧，未来或许你就是你所在领域的王者也说不定的呢对吧。

总结

不知不觉写了这么多了，以上是我个人眼里的一般向JAVA 程序员的发展线路，限于篇幅也不全面，实际个人成长路线可能因为工作内容的不同差异会很大，有的人偏向底层研究，有的人偏向业务需求设计，有的带有浓重的行业色彩，而且技术之外，还有很多知识也很重要，做JAVA没有轻松的方向，但一个对技术抱有兴趣的人，走到这一步时，仍然会对开发抱有热情，想要写出好的项目。

纯为了生计而工作的程序猿是走不到这一步的，这一行来都来了，大家一天都是24小时，为啥有差距，我想你我都明白的，知道为啥那就干出点名堂吧。



我是敖丙，一个在互联网苟且偷生的工具人。

创作不易，不想被白嫖，各位的「三连」就是丙丙创作的最大动力，我们下次见！

文章持续更新，可以微信搜索「三太子敖丙」第一时间阅读，回复【资料】【面试】【简历】有我准备的一线大厂面试资料和简历模板，本文 **GitHub** <https://github.com/JavaFamily> 已经收录，有大厂面试完整考点，欢迎Star。



你知道的越多，你不知道的越多