本文由 简悦 SimpRead 转码， 原文地址 www.imooc.com

作者：明明如月

1.1 自我介绍

首先我先进行简单的自我介绍，我叫明明如月，目前就职于某知名电商公司高级 Java 开发工程师，CSDN 博客专家，慕课专栏：解锁大厂思维：剖析《阿里巴巴 Java 开发手册》和 再学经典：《Effective Java》独家解析 专栏的作者。

作为经历过校招（校招网易）和社招，工作已经三年左右的过来人，本文将结合自己的经验和体会，分享一些干货本文将为大家解答 Java 入门到进阶的一些常见困惑。

1.2 技术选择

如果你想做一名后端开发工程师，岗位较多的是 Java 其次是 Go 语言和 Python 等（大家可以去 Boss 直聘 或者大型互联网公司的招聘官网上看下）。

虽然 Go 有一些优势，比如上手容易，性能高等优势。Python 上手也简单，但是从就业来看，对于大多数没有太多经验的同学来说，想学到能找到工作的水平难度较大。现在主流的互联网公司尤其是电商网站，大都以 Java 为主要编程语言，而且从招人的数量上也更多，如果想学后端，Java 是一个非常不错的选择。

1.3 担忧

也有一小部分人会担心 Java 会被淘汰。

从目前来看，近几年不太可能被淘汰。

- Java 本身也向其他优秀的编程语言（如 Kotlin）学习。
- 其次，众多大型互联网企业在使用 Java 作为核心业务的编程语言。
- 另外，Java 生态圈非常强大，围绕 Java 有大量的开源的，强大的中间件。
- 此外，知识都是相通的。精通一门编程语言的人往往可以快速掌握另外一门编程语言。
- 最后，学习是一种能力。大家在学习过程中如果能够提高自己快速学习和知识迁移的能力，不用担心一个语言是否被淘汰的问题。

手微信itit1122334

1.4 学习之路

1.4.1 自学 or 报班

可能会选择在线的技术网站课程；也有些同学更喜欢线下的编程培训机构；还有部分同学自学能力强，选择看视频或者看书自学。

如何选择根据自己的情况来决定。

科班的同学一般会选择自学或者看在线技术视频为主；转行的同学可能更倾向于参加培训机构。不过一般公司不太喜欢培训机构出来的学生，因为这类学生普遍突击为主，甚至过度包装。

如果是科班的同学，有条件接触一些有项目的老师或者有些校企合作项目（如果在校期间可以找到一些大公司的实习机会，那就更好了），建议可以主动参与进去，获得一些实践的机会。

1.4.2 我的成长之路

入门阶段

我当时入门的时候买了《Java 核心技术》和《Java 编程思想》，太厚了啃不动。后来主要看技术视频为主，主要跟着视频敲代码，发现这样更容易上手。大家入门的时候一定要动手敲代码，绝对不能只看视频。只看视频非常枯燥，而且印象不深刻。

开始的时候看着视频学习 Java 基础，学习如何使用 IDEA，学习一些调试技巧。

Java 基础学完以后，然后学习 MySQL 和三大框架 SSH（现在流行 Spring、Spring Boot、MyBatis）。

穿插着也学了点前端基础，入门了下缓存框架（Redis），最后跟着老师做一些项目（当时主要是单体应用为主）。

校招阶段

校招阶段我发现大多数人并不知道重点在哪里，不知道重点该看哪些书，怎么学，都是网上找面经，非常零散。刚开始面试的时候准备的一般都不太充分，错过了很多机会，非常可惜。如果能够知道哪些是重点，知道怎么学，会有更多的机会。

找工作前刷了下《剑指 Offer》，刷了一些常见的面试题，加上之前在学校里也做过单体应用的项目，学校里也拿了一些国奖，而且每次面试之后都会总结和反思，最终校招拿到了美团、网易等公司 Offer，最终选择网易。

工作阶段

初入职场有些忧虑和迷茫，问多了怕别人觉得自己能力不行，问少了很多东西不了解，不知道该如何进阶。

一方面重新温习操作系统方面知识；也会看一些微服务、中间件、架构方面的书来继续学习；一方面在工作中不断积累经验。比如掌握高级的调试技巧；积累排查问题的思路；学习老员工技术方案设计的一些好的思路；和已经工作的人多交流下心得体会。

从开始的安排啥做啥，产品怎么设计怎么做，到开始提出一些建议，有一些产品思维，学会 “以终为始”，根据目的来反推设计。

从完成任务为主，到考虑如何提高性能，考虑如何提高代码的可读性和可测试性（学技术需要有一定的追求）。

开始知识非常零散，通过一方面巩固专业基础，一方面不断 “归纳和演绎”（将共性的东西总结到一起，找出共性的本质，然后设计技术方案时可以逆向运用），现在知识更加系统化。

开始从纯开发的视角聊问题，开始转变为用产品能听得懂的话和产品沟通，用测试能听得懂的话和测试沟通。

学知识从记忆为主，会用为主，变为思考技术是为了解决什么问题，该技术的优缺点，作者为什么要这么设计，如果没有这个技术该怎么办等。学会采用 “先猜想，后验证” 的思维方式来学习，发现效果好很多。

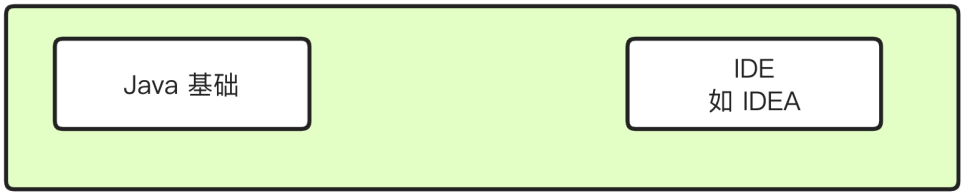
从纯输入为主，开始输出自己的学习方法论，通过博客输出自己的经验，通过技术群和一些优秀的朋友相互交流学习。

现在一直在不断进阶前行的路上。

下面介绍，从开始学 Java 到能够找到工作，再到工作两三年，再到后续的职业规划：

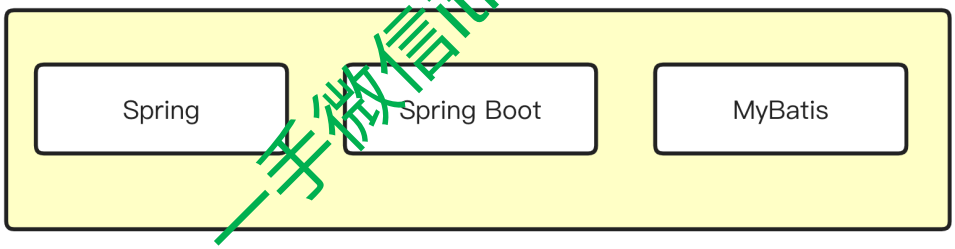
入门阶段，最重要的就是掌握 Java 基础（主要包括基本的语法，各种集合、文件读写、多线程等方面知识），此阶段一般通过视频来学习，然后自己动手多练习。

一手微信itu1122344



有了一定基础之后，要掌握 Linux 命令、数据库、版本管理、构建工具和 Web 服务器相关技术，这些都是工作必备的技能。

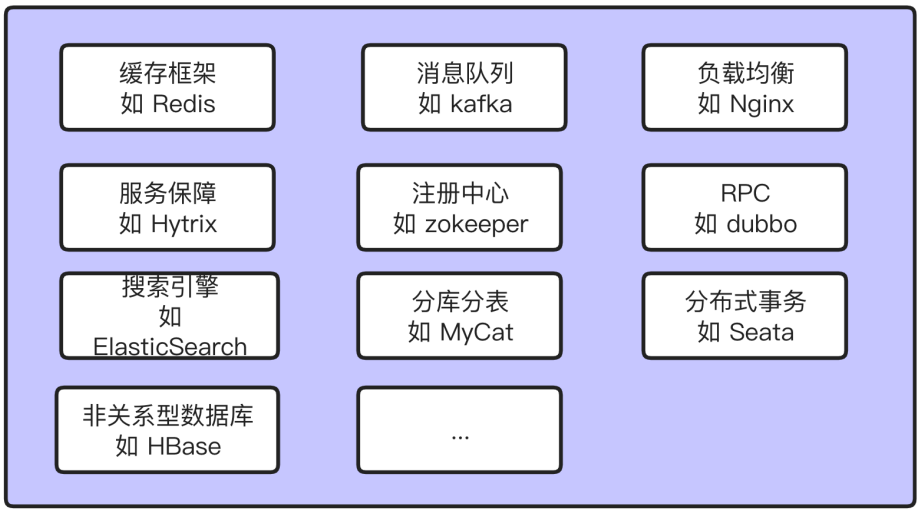
掌握好上述技术之后，想找 Java 相关工作，必然要掌握 Spring、 Spring Boot 和 MyBatis，这是很多互联网公司的“标配”。



一手微信itit1122344

一般来说掌握到第 3 个阶段基本可以开始考虑找工作了，此时如果能掌握部分第 4 阶段的内容，找工作会有些优势。

下面简单介绍第 4 部分的主要技术：



如果访问量较大，单机的承载量有限，就需要通过多台机器去分担请求，此时就需要**负载均衡服务**将请求分发到不同的机器上。

现在微服务比较流行，公司内不同服务之间通常会采用 **RPC** 实现跨机器的调用，因此要掌握 RPC 相关原理和技术（其中就包括注册中心）。

随着数据量和访问量越来越大，尤其是数据量达到千万级以后，查询的性能就会存在问题。因此可以利用化整为零的思想，将数据分摊到多台数据服务器来提高性能，因此就出现了**分库分表中间件。**

为了提升用户体验，提高响应时间，普遍会通过**缓存**的方式来加快查询速度。因为通常读缓存比直接查数据库更快，而且读缓存可以减少对数据库的压力。

消息队列是为了实现削峰、解耦和异步。稍微大一些的公司就会有多个二方服务，服务之间除了直接调用之外，通常会采用消息队列进行解耦。同一个服务内部也可以通过消息队列通过多实例加快处理速度，也可以通过消息队列处理耗时或者不需要同步的操作，来提高用户体验。

由于数据库搜索能力有限，有些场景下需要更强大的数据搜索能力，此时就要用到了**搜索引擎。**

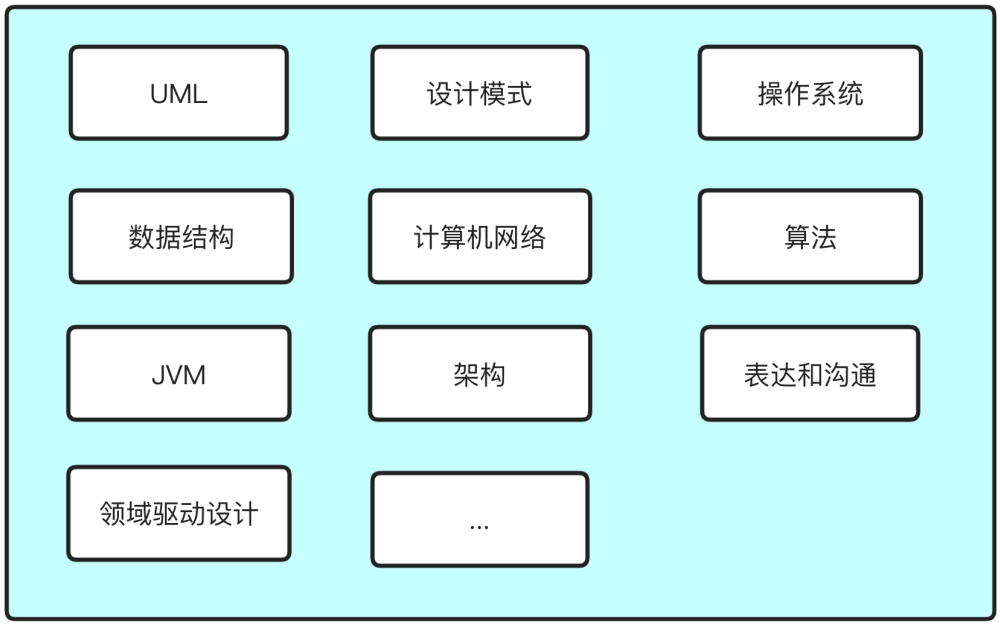
随着并发越来越高，业务场景也来越复杂，传统的关系型数据库局限就暴露出来，有时候需要引入**非关系型数据库**来解决问题。

上述技术都是电商类网站比较常用的技术（且不限于此）。

第 4 阶段的技术，有条件的话可以购买视频看效果更好，其次可以阅读后面推荐的相关图书，对找工作帮助也很大。

除了前面讲到的技术之外如果想进一步进阶，还要推荐掌握下面的知识，如 UML 作图、设计模式、领域驱动设计，还需要巩固好专业基础，还需要提供沟通表达能力。

一手微信it1122344



上图中 JVM 和设计模式推荐工作前就要掌握，面试可能会问到。其内容话，尽力而为。

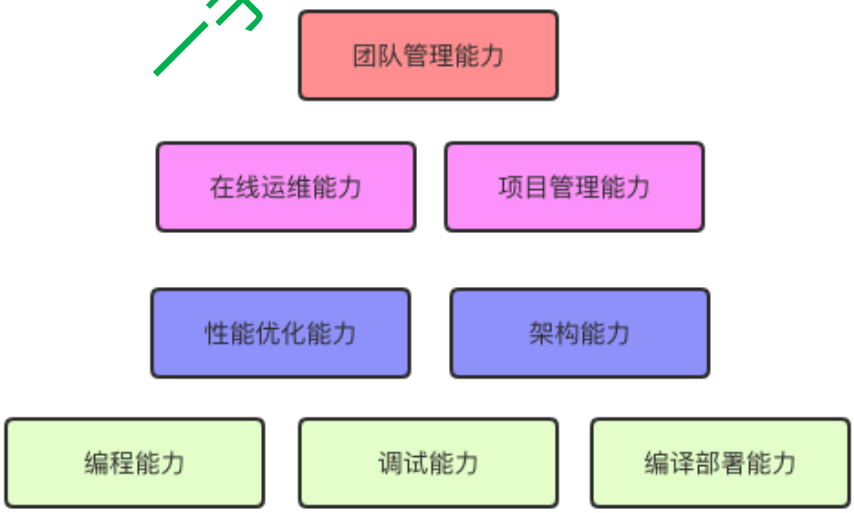
工作几年之后，如果还想从事这个行业，可以考虑下面几个主要方向：专精技术、转型管理、晋升为架构师。

想要成为优秀的架构师，需要掌握哪些能力呢?

架构师是一个 “定义规范” 和 “指导落地” 的人，要有全局观，善于学习和分享，能够很好地发现并解决问题的人。

结合美团技术的《成长为架构师途中的 11 个浮躁人一文的架构师能力图，将架构师的核心能力列举如下：

一手微信it1122344



编程能力、调试能力和编译部署能力是基础能力。如果不能精通掌握这三种基本能力，很难再性能优化能力和业务架构能力方面有所成就。具备了性能优化和业务架构能力之后，才能在运维能力和项目管理能力方面表现优越。团队管理能力是最高能力，它对项目管理能力的依赖性更大。

如果**想晋升管理岗**，除了有极强的专业能力之外，还要关注行业动态，学习团队、企业管理相关知识。要有较强的沟通协调能力，能够充分发挥团队成员的优势，并调动他们的积极性。

这些绝对不是单纯看几本书、看几个视频就可以做到。这个阶段这需要我们在工作中不断打磨技术的同时，不断提高抽象思维，提高沟通和管理能力。

对于初学者而言，建议看视频为主，对于有一定基础的同学可以侧重看书、读源码、看技术专栏。

3.1 视频

学习 Java 的视频，大家可以去慕课网等在线技术平台选择适合自己的免费的或者收费的在线课，这里就不多说了。

3.2 图书

前面也讲了，初学者建议看视频为主，不建议看书。

下面推荐大家在阶段 3 时，在找工作前可以读读下面几本书。

- 《Java 8 实践》(基础)
- 《Java 编程的逻辑》(基础)
- 《阿里巴巴 Java 开发手册》(基础)
- 《码出高效》(基础)
- 《深入理解 JVM》(进阶，面试必问)
- 《Java 多线程编程核心技术》(重要)
- 《Java 并发编程的艺术》(重要)
- 《设计模式之禅》(重要)
- 《剑指 Offer》(重要)
- 《MySQL 技术内幕》(选读)

上面推荐的图书，标注为 “基础” 的图书相对来说比较容易，有助于巩固基础。现在 Java 面试，虚拟机几乎是必须问的问题，一般都不出《深入理解 JVM》这本书，建议反复阅读。找工作如果出算法题，一般都在《剑指 Offer》或者 LeetCode 里。MySQL 也是面试的重点，可以读读《MySQL 技术内幕》或者购买相关专栏重点学习（尤其是聚簇索引，最左前缀原则，B+ 树，SQL 优化等）。

如果有条件，推荐求职前读读下面几本书，尤其是《Redis 深度历险》面试时问 Redis 相关问题一般不会超过这本书的范畴。

- 《Redis 深度历险》（必读）
- 《Elasticsearch 实践》（选读）
- 《Hbase 不睡觉书》（选读）

一般说来，前 3 个阶段的知识都能熟练掌握，上面的图书都读地不错，本身又热爱技术，面试问题不大。

找到工作以后，建议大家要重视编码风格，提高代码可读性、可维护性。这有助于帮助你写一手优雅的代码：

- 《阿里巴巴 Java 开发手册》
- 《重构 - 改善既有代码的设计》
- 《代码整洁之道》
- 《编写可读代码的艺术》
- 《修改代码的艺术》

找到工作之后，建议大家重视 Java 官方文档并且可以读读 Java 领域非常知名和经典的图书：

- 《Java 语言规范基于 JavaSE8》
- 《Java 虚拟机规范》（Java SE 8 版）
- 《Java 编程思想》
- 《Java 核心技术》
- 《Effective Java》

一手微信itt1223344

工作两年左右就要从更宏观的角度思考问题、设计方案，此时可以考虑学学架构。可以读读下面几本架构相关的书：

- 《微服务架构与实践》
- 《微服务设计》
- 《聊聊架构》
- 《架构整洁之道》
- 《演进式架构》
- 《微服务架构设计模式》

再往后进阶架构师或管理岗。架构和管理方面的书非常多，大家可以在豆瓣、京东或当当上查看排行榜，有选择性购买。

下面推荐几本书：

- 架构：《架构师修炼之道》、《一线架构师实践指南》、《架构真经》
- 思想：《高效能人士的七个习惯》、《第 3 选择》
- 管理：《可复制的领导力》、《领导梯队》
- 产品：《产品方法论》

在读各种经典的设计理念和架构基础上要形成自己的方法论，还要学习沟通和团队管理方面的知识，不断进阶。

3.3 专栏

有一定基础之后，大家也可以去购买一些专栏学习，专栏比书相对来说更接地气，更容易理解。

知道学什么之后，怎么学很关键。你会发现很多人因为学习方法不佳，入门和进阶非常缓慢。下面针对新手和有一定基础的同学给出一些方法论上的建议。

一手微信itit1122344

4.1 TO 新手

如果你是刚入门不久或者刚工作不久，这些建议大家可以参考下：

- **学习的时候电脑上备着一份 API 手册，随时查阅（此外可以多进源码里看看）。**
- 充分利用**搜索引擎**：谷歌、百度、必应等。此外特别推荐大家使用搜狗微信搜索，可以搜索微信公众号的文章，一般来说公众号里的文章为了保持对用户的吸引力，质量会比搜索引擎里搜到的更高，强烈推荐！
- **严格遵守代码规范，养成良好的编程习惯。**可参考《阿里巴巴 Java 编程规范》，建议买一份纸质的，虽然贵了一些，但是提醒自己用心多看。
- 学习技术最好去看**英文原版的官方文档**和官**方 GitHub 源码和示例。**
- **实践出真知**，如果看学习视频一定至少自己敲一遍。
- 关注一些非常好的**技术类公众号**（如：慕课网、阿里技术、架构师之路等）了解最新进展，了解别人的经历和经验。
- 可以加入一些**靠谱的 Java 学习群**和一些比自己**更优秀的人交流。**
- **维护一个技术博客**，把自己技术上遇到的难题和解决方案，自己遇到的一些坑分享上去，一方面加深自己的印象，一方面帮助其他人。
- **最经典的图书一定要看。**如《Java 核心技术》、《深入理解 Java 虚拟机：JVM 高级特性与最佳实践》、《 Java 并发编程的艺术》、《 Java 多线程编程核心技术》、《 Effective Java》、《深入剖析 Tomcat》等。
- **遇到的问题一定要记录下来**，可以记录到印象笔记、有道云笔记、为知笔记等，以后再遇到类似问题很容易找到之前的解决方案。也可以通过 CSDN 等平台采用博客的形式记录。
- 编程的时候感觉哪个类或者哪个方法感兴趣，可以直接去**查看源码**一探究竟。
- 要**有精益求精的精神。**如果可以优化尽量去优化代码。
- **制定学习计划和目标**，多久掌握某某框架，一年之后、两年之后要达到什么程度等等。

- 多去一些知名互联网公司校招网站查查自己**岗位所需的技能**，有针对性的复习；可以去 “牛客网” 去查看一些面经，针对题目有目的的学习或者复习。
- **心态上非常重要，不要被 “觉得挺难” 吓倒。**学习初期很多困惑没关系，不要以为有困惑就代表自己学的不好，先学会用，然后到后面会自然而然的懂了，不懂的话再去思考。
- **学习新技术**直接去 github 上搜入门 demo 或者官方 example，很快就可以上手。
- **学习总结排查错误的能力**，比如看错误堆栈信息，比如打日志，比如本地或者远程调试，比如分析源码等。

强烈建议新手遇到问题一定要先思考，再分析调试，再网上搜答案，否则很难快速成长。

4.2 TO 进阶中的小伙伴

很多小伙伴在学习进阶过程中会陷入：**读书看了记不住，记住不会用的困境。**

那么，如何才能突破这种困境，如何才能更好扎实地学好技术呢？

下面推荐几种自己实践过认为非常高效的学习技术的方法：

4.2.1 归纳和演绎

我们在学习过程中会发现不同技术之间有一些共性，那么这些共性就是通用的原理性的东西，就是更有价值的东西。

我们通过具体的知识之间的共性 “归纳” 出核心原理。

在实际的项目实战或工作中，将这些原理再灵活地运用到解决实际问题中，这才是真正地 “学以致用”，这才是学习的价值所在。

这么说你可能会觉得有些抽象，大家可以结合我自己总结的思维导图的其中一个分支，体会一下：

一手微信itit1122344

一手微信itit1122344

当我们发现不同的技术点的共性时，将其进行归类，思考他们的本质，比如这里的 “化整为零” 就是他们的原理。

通过不断地归纳，学习到知识最精华的部分。

在设计技术方案时，可以将这些核心的原理逆向运用来解决问题。

4.2.2 猜想和验证

很多人看书记不住，记住不会用，进阶速率较慢，是因为你一直在“背书”在“记忆”，缺乏思考。

这也是很多人看了很多源码却收获不大的重要原因。

我们在看源码和学习某个中间件的时候，一定要先猜想后验证！！

想一想作者为什么要这么设计；思考这个技术是为了解决什么问题；思考这个技术的优势和劣势是什么；思考如果是你会怎么设计；思考如果某种情况下，会怎样怎样；

然后和实际的源码和作者的设计进行对比，找出差别在哪里。

通过这种方式你能够意识到自己的不足，能够不断地纠正自己对知识的理解，进而可以更扎实地掌握知识。

4.2.3 以终为始（本质思维）

如果总是纠结于表面，纠结于细节，往往就会忽略做事的目的。

技术的本质还是为了解决问题，我们应该主动思考问题的本质，我们设计所要达到的目的是什么，据此来调整我们的方法和行为。

很多人设计技术方案的时候，经常会考虑“以前就是这么设计的”、“大家都是这么做的”，而不去考虑本质上是为了什么。就像很多人问技术问题的时候，你明显地可以感受到他在“走弯路”，当你帮助他梳理清楚他的核心目的之后，可以有更简单的方法来解决这个问题（也就是那句话“事出诡异必有妖”）。

我们学习某项技术的时候也是一样，要思考该技术的本质是什么？

比如 redis 缓存，就是一种缓存（和浏览器缓存、CDN 缓存、DNS 缓存甚至 CPU 缓存都是一致的），那么缓存的本质是什么？ 缓存的本质就是通过空间换时间的方式获取更好的性能。

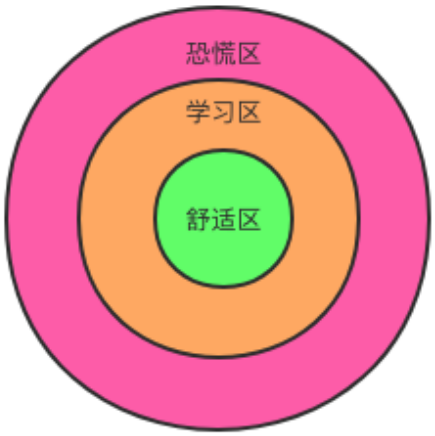
这样你对一种知识就有更宏观和更本质的理解，就更有可能灵活运用。

4.2.4 逃离舒适区

舒适区对一个人的阻碍是非常大的，因为人性害怕不确定性，更愿意呆在舒适区，这是很多人工作多年但是进步不大的重要原因。

如下图所示，舒适区主要是已经熟悉的知识（方法）。学习区是指通过学习可以掌握的知识；恐慌区是指目前的学习很难掌握的知识。

一手微信itw1122344



呆在舒适区就意味着很难成长，接触的都是已经掌握的东西，既不能提高知识的广度也不能提高知识的深度。

但是步子迈得太大，往往也容易受挫。

因此应该保持在学习区，不断学习新的东西 / 有一些挑战的东西 / 对自己发展有帮助的东西，让自己能过不断成长。

另外随着归纳和演绎方法不断深入，你会发现很多新的知识都是旧的知识的一个应用，学起来也会变得容易地多。

一手微信itt1122344

4.2.5 教是最好的学（教学相长 / 输出）

《礼记 · 学记》有言：虽有嘉肴，弗食，不知其旨也；虽有至道，弗学，不知其善也。是故学然后知不足，教然后知困。知不足，然后能自反也；知困，然后能自强也。故曰：教学相长也。《兑命》曰：“学学半。”其此之谓乎？

另外根据《学习金字塔理论》，主动学习的内容留存率比被动学习更高。其中将知识教授给他人，知识留存率约为 90%。

另外著名的费曼学习法也是通过教授知识给别人，发现问题，重新学习来不断精进的。

因此推荐大家可以尝试写博客、写公众号甚至做一些技术分享来输出。

在教的过程中不自然地要搜集各种资料，要进行梳理和思考。整理过程中会发现自己理解不好的地方，在分享之后也可以得到反馈，进而可以改进。（参考费曼学习法、PDCA 循环）

在输出的过程中，自己的知识更加系统化，得到反馈后也可以更快速地改进。

可以分享自己学习中遇到的坑，可以分享自己对某个技术（点）的思考等。

5.1 “适合自己的方法”

孤尽老师强调过：要重视学习如何学习的能力。

在大家学习 Java 和后续进阶的过程中，影响最大的一个因素就是学习方法。然而大多数人，大学用高中的方法，工作后用大学的方法，千年不变。

希望大家要敢于摆脱固有的低效的学习方法。

不仅要学习而且要学习牛人如何学习，学习如何高效学习。

从小就经常听说 “找到适合自己的学习方法”。**但是很多人潜意识里就会认为适合自己的方法其实是 “舒适” 的方法，但是舒适的方法往往是已经用过的方法。**

但是已经用过的很多方法往往并不是高效的方法，但是由于舒适区的原因，由于人们恐惧不确定性，导致很难做出改变，听到新的方法本能地抵触。

这也是很多人看了很多技术图书，知道了很多方法，进步却不大的原因。

不同的学习方法可以看作是数学中的不同的函数（如常数函数、幂函数、指数函数等），不同函数的增长速率是完全不同的。

其实我们不应该找 “适合自己的（舒适）的方法” 而更应该追求 “高效的方法”，这也符合以终为始的思想。

如果高效的方法不适合你，那么...

而且前面也提到以终为始，我们运用各种方法的目的是啥？难道不就是为了快速学习新的知识，快速成长吗？

一个人只有转变方法或思维方式，才能够有质的提升。

5.2 欲速则不达

5.2.1 专业基础

在学习进阶过程中，很多人会忽视专业基础，但是工作几年之后你会发现基础扎实才是你能够灵活运用知识的关键。

如果是非计算机专业的学生，推荐大家读读《深入理解计算机系统》这类书，甚至可以买来计算机专业操作系统、计算机网络、数据结构和算法、计算机组成原理的教材，补补基础。

5.2.2 解决问题

很多人解决问题急于搜到答案，然后就没有然后了。

这是很多新手进步不大的重要原因。

我们要先思考，通过日志等去验证自己的猜想，然后不断推理出原因。

在思考之后再去搜索引擎，StackOverflow 上寻找答案，再搞不定可以考虑请教别人。

只有通过不断地猜想和验证，不断巩固专业基础，不断积累经验，才能快速准确地定位问题。

5.3 技术之外

很多人认为只要学好技术就可以了，其实并非如此。

工作中就听说过有同学技术水平还可以，但是理解和沟通能力非常差，最后没有转正。

工作以后沟通和表达能力就显得非常重要。

建议大家可以读读《非暴力沟通》或者其他书籍，掌握良好的沟通技巧。

在工作中和不同角色的同事沟通时要有分别，比如和产品沟通时，多谈需求文档少谈技术术语；和领导沟通时，先说重点，说领导想知道的内容。

5.4 面试

面试可以讲的内容很多，可以单独写一篇文章甚至一个系列，篇幅优先，这里先谈几点。

一手微信11223344

很多人在准备面试的时候，由于缺乏经验，很难复习到重点。网上找一些面经，往往只有题目没有答案。即使有答案也未必全面有深度。

不管是平时的学习，还是准备面试，一定要有自己的思考。

> 举一个例子： 如果面试官问：**你知道哪些 HTTP 请求方法吗?**
>
> 要不然回答不全；要不然就千篇一律（GET、POST） 如果你还能回答出 PUT 、DELETE、TRACE、HEAD、OPTIONS 并说出其含义，就会比别人多一分机会。
>
> 再举个例子： 如果面试官问 ： **为什么 Redis 那么快?**
>
> 要不然回答不全；要不然就千篇一律（如基于内存、如非阻塞 IO 等就完了）
>
> 如果你能够回答更多一些，可能会让面试官刮目相看。 比如可以将动态字符串的内存预分配；讲惰性删除 + 集中删除；讲 WAL; 渐进式 rehash 等，就可以拉开差距。 如果能回答出这些性能优化背后体现出的核心思想，如空间换时间，同步转异步，化整为零，化零为整等，可能就更出彩了。

给出自己总结出的一个原则：

会的问题，回答的全面有深度；没见过的，说说自己的思路。

如果面试官问你一个第一次听说的问题，你学习过程中遵循 “猜想和验证” 或者 “归纳和演绎” 的学习方法。那么你很容易说出一些靠谱的思路，面试官就会认为你非常聪明有潜力。因为公司要的就是快速的学习能力，懂原理又能够解决实际问题的能力。

本文主要介绍了 Java 从入门到工作两三年，甚至再往后的路线图。希望能让大家对 Java 学习和进阶之路有一个清晰的了解。文中还介绍了自己的一些学习方法，并推荐了 Java 学习进阶的一些图书，并讨论了 Java 学习的一些误区。

俗话说：“授人以鱼不如授人以渔”，本文不仅讲了进阶路线，还提供了一些高效的学习方法。

如果你有技术热情，再掌握一些好的学习方法，后面技术提升会快很多。

以上这些都是自己从入门到工作这些年来的切身体会，希望对大家的学习进阶有帮助。

}

一手微信vx1122344