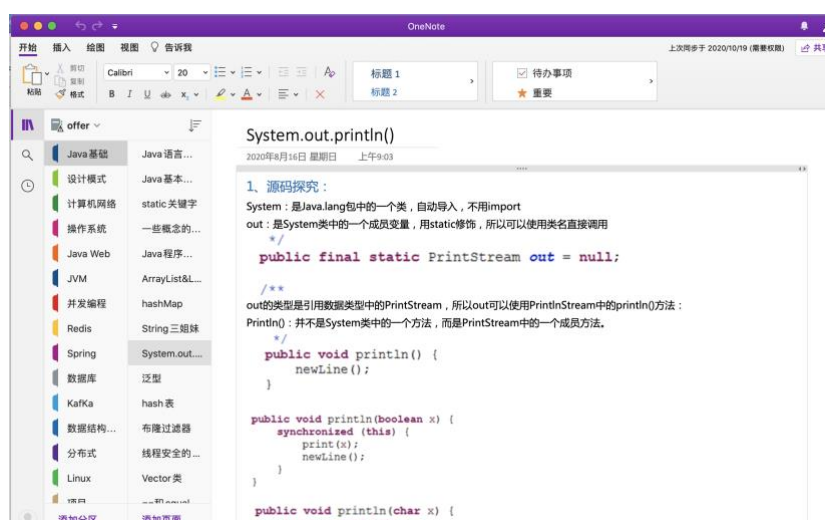


Java 各部分学习时间和学习方法安排与建议

开发工具和记录笔记 APP 的选择：

记录笔记推荐使用：

个人推荐 onenote 记笔记是因为 onenote 的分区效果非常好，根据不同的章、节进行分区，根据不同的笔记本记录不同的课程，在后续复习时候，一目了然。



(我用的是 mac 版的 onenote，win 版会略有区别，但是的应该也差不多，所以建议大家使用)

开发环境：

建议使用 IntelliJ Idea，因为现在基本大公司开发用的集成工具都是 idea，具体的下载方法，可以自行百度

IntelliJ IDEA - 百度百科



IDEA 全称 IntelliJ IDEA，是java编程语言开发的集成环境。IntelliJ在业界被公认为最好的java开发工具，尤其在智能代码助手、代码自动提示、重构、JavaEE支持、各类版本工具(git、svn等)、JUnit、CVS整合、代码分析、创新的GUI设计等方面的功能可以说是超常...

[特色功能](#) [优点](#) [版本概述](#) [快捷键](#) [系统要求](#) [总结](#)

baike.baidu.com/

JavaSe 部分：

这部分我推荐的视频的总时长是 60h，我们一般看视频记笔记，敲 demo 的话，学习时间基本上是视频时长的两倍，JavaSe 部分还是建议细看的，因为是整个 Java 的基础，但是 JavaSe 视频中有一些部分的内容算是计算机的入门内容（比如二进制之类的），如果之前有计算机基础的同学，看到这部分的时候可以相应的加速，但是整个视频中还是 java 内容占了大多数，所以建议大家不要倍速观看。还是学的踏实一点比较好！

60h 的视频，学习时间大概在 $60 \times 2 = 120h$ ，如果每天学 8h，大概 15 天能学完，加上学完之后复习 20h 的话，总时长在 140h 左右，合理的时间大概在 17 天左右，但是说实话，因为

大家都是转行的同学，在自己学科内可能还有学习任务和科研任务，基本很少有人能 17 天学完 se，所以我觉得合理的时间在一个月左右，大家根据自身学习情况和科研情况合理的安排好时间。

JavaSe 部分建议大家在学的时候一定要动手记笔记，虽然我也给大家提供了笔记，但是别人记的和自己记的还是有区分。建议大家在学习的时候，可以使用移动设备（非常建议 pad）来看视频，使用 pc 端（推荐 onenote）记笔记和敲 demo（建议使用 IntelliJ Idea，因为现在基本大公司开发用的集成工具都是 idea，具体的下载方法，可以自行百度）。

数据结构与算法：

我们在学习完 javase 之后，就完整的掌握了一门开发语言了，这个时候大家对于语言的熟练度和掌握程度可能都算是入门了。这时候建议大家可以开始学习数据结构与算法，这部分内容算是整个 java 学习过程中最考验智商的了，因为各种各样的算法确实有的时候很难理解，很难吃透，建议所有课堂上讲解到的算法和题目，大家都可以独立的敲出来，并弄懂器原理（特别是十大排序算法，几乎是面试必问），其次这里说明一下对于时间空间复杂度的问题，其实，我到现在都不是特别清楚时间复杂度的计算方法，这里大家其实只需要了解常见算法的时间复杂度即可（说白了就是背下来，因为基本面试的时候问的也都是这些常见的），但是学有余力的同学，可以尝试去把这个东西真正搞懂，肯定是更好的。

刷题：

学完数据结构与算法，大家就可以刷题了，刷题是一个持之以恒的过程，我建议大家刷 leetcode 的简单和中等难度即可，不需要刷困难（因为，你真的刷不出来）。

#	题名	题解	通过率	难度	出现频率
143	重排链表	436	58.0%	中等	🔒
1	两数之和	7210	49.7%	简单	🔒
2	两数相加	3946	38.7%	中等	🔒
3	无重复字符的最长子串	3863	35.8%	中等	🔒
4	寻找两个正序数组的中位数	1851	38.9%	困难	🔒
5	最长回文子串	2022	32.1%	中等	🔒

刷题量：

第一阶段：初期的 2~3 个月，建议大家每天刷 1~2 题，其实不要以为这很少，每天 1~2 题，基本就要花至少 1 个小时的时间，大家最开始刷的时候肯定没什么思路，如果没有好的方法，可以暴力求解，总之尽可能先通过，然后大家可以看别的大牛是如何解题的，自己理一理思路，慢慢的就有感觉了。其实大家如果能坚持 2~3 个月，基本力扣的题，你已经刷了将近 150 题了；

第二阶段：之后的 1~2 个月，大家可以增加题量，每天在 2~3 题，这样坚持的话，大概能

再刷 100 题左右，这样下来一共你已经刷了将近 300 题了，我觉得这个题量其实还可以了，当然对于自身要求较高的同学，可以在不同阶段增加题量。

第三阶段：最后 1~2 个月（在秋招之前），把你以前刷过的所有题，再拿出来重新刷一遍，同时建议刷一刷剑指 offer 的题（这部分题是比较常考的笔面试题，和 leetcode 有大部分重合，题目也比 LeetCode 简单一些，所以大家刷起来也不费劲）这一遍的速度会快很多。另外，在秋招过程中也建议不要停止刷题，这个时候你可以有针对性的去刷一刷各大公司的笔试常考题（自行百度）。

如何刷题：

建议大家先从简单部分刷起（刷题的顺序，可以直接按照题号来，也可以自己百度，有很多人也总结过这种刷题顺序，大家可以挑选最适合自己的顺序），当简单题型你看一眼就知道怎么写的时候，你就可以开始刷中等难度模式的题了。

刷题的时候建议大家不要图快，一道题真的要弄懂了，才算是你自己的东西，其实刷着刷着你会发现很多题目之间存在共性，所以一道题弄懂了，你可能可以解决很多其他类似的问题，总之，越刷越有感觉！

刷题也建议弄一个笔记本，把你自己的代码和刷题时的分析，心得等等进行记录，这样在你后续复习的时候会很有作用！

JavaWeb 部分：

同理，大家可以根据我所提供的根据视频时长计算学习时长的方法，评估一下学习 web 所需的时间。这里提一下，对于 web 部分的学习主要分三大块，1、数据库（非常重要，这部分建议一定要好好听，但是 web 所讲的数据库知识是数据库的基础知识，现在秋招的时候特别爱问数据库的知识，但是一般涉及的都是一些更深层次的问题，这个建议后期准备秋招之前进行一下系统的学习）；2、前端知识（很多同学问我，网上很多说后端程序员不用掌握前端知识，我觉得这么说的不太合适的，虽然后端程序员对前端没有要求，但是你将来要做项目，如果你一点前端知识都不懂，可能会出问题；况且现在 java 也越来越卷，保不齐将来到你面试的时候，就开始考察前端知识了呢，所以这部建议大家耐心的看完，这部分大家可以不用特别上心，但是建议一定要看完）；3、后端部分（这部分的重要性不言而喻了，对于我们后端开发人员来说，这是我们第一次接近真实的开发场景，所以大家一定要好好学习这部分，这部分可能 demo 会比较多，对于一些课程中的小项目，大家可以跟着老师一起敲，但是建议大家如果时间充裕的话，最好还是可以自己独立的把这些项目敲出来，并且尽可能的弄懂之间的原理和联系）。

这部分视频时长在 70h，我觉得就算学的再快的同学，应该也要差不多 180h 的时间，所以按照每天 8h，至少也需要 30 天，我觉得这部分学习在一个半月比较合理。

SSM 框架+SpringBoot 框架：

这部分的学习顺序是 Mybatis+Spring+SpringMVC+SpringBoot

这部分框架的学习主要是对 web 开发整体的封装，但是这几年也越来越爱问这部分内容，这里大家跟着视频，把该敲的 demo 敲好，该画的图，该看懂的内容看懂就行。

SpringBoot 比较常问，一些初始化之类的问题，这里我不细阐述了。

项目：


上述内容都学完之后可以开始做一个项目了：

我做的是一个牛客网网页开发的项目，其实，秋招的时候面试官也知道大家的项目都是网上的项目，最常见的是秒杀项目（我们可以都做同一个项目，但是大家需要学会给项目套壳，不然就真的千篇一律了，至于如何给项目套壳也是很简单的东西，大家有需要可以找我）。

下面我贴下我的项目，我的项目课视频是完整的教学视频，大家拿到资料之后，完全可以直接上手，所需的所有资料、源码都是完整的，但是项目是要收费的，因为项目也是我和管理员花钱在牛客官网上买的，价格肯定比收大家的钱要贵得多，大家有需要的可以联系我，项目付费的话，我会把资料费退给大家。

项目一共两个：秒杀+牛客网高级项目课（我和管理员做的都是这个，管理员两个都做了，我只做了一个，我觉得做完两个项目，简历是完全够用的了）

下图是我们根据项目写的简历和项目课的内容：



项目经历

智慧校园互动系统

2020.02—2020.05

项目描述：实验室与精伦电子有限公司合作开发项目，旨在为老师和同学搭建一个在线校园平台，实现课堂签到、布置作业、答题、交流区互动等功能。

责任描述：负责用户注册登录、交流区发帖互动、热帖排行、粉丝关注等模块。

1. 使用拦截器拦截用户请求，将用户信息绑定在 ThreadLocal 上，用以检查用户状态。
2. 通过 Redis 缓存用户的登录凭证、获赞数、粉丝列表，提高并发查询时的速度。
3. 使用 TrieTree 存储文件中的敏感词，实现发消息的敏感词过滤算法。
4. 对帖子的获赞数、评论数以及是否精华赋予不同的权重，计算帖子分数，实现热帖排行功能。


秒杀系统方案优化

2019.11—2020.01

项目描述：本项目主要利用 Redis 和 RabbitMQ 实现秒杀系统方案的优化，保障了在秒杀过程中的安全性、正确性、高并发和高可用性。

责任描述：

1. 使用双重 MD5 加密实现安全登录，并将随机生成的 token 存入缓存防止重复提交。
2. 通过 Redis 中间件缓存常用查询，预减库存量，减少对数据库的直接访问。
3. 通过 RabbitMQ 实现异步下单，达到削峰的目的，同时增强用户体验。
4. 使用 Jmeter 压测，50000 个线程请求的 QPS 由 1345 优化到 2992。



项目经历

校园留言板平台

2020.02—2020.05

项目功能：本项目旨在为机电学院师生提供一个可以进行学业讨论、生活分享的交流平台。机电学院师生可以通过注册登录，实现帖子发布、点赞、关注、热帖搜索等功能。

责任描述：负责开发用户的登录与注册，账户修改等基础功能，开发帖子发布和点赞、关注，热帖搜索和热帖排行等核心功能。

项目描述：

- 1、项目的编写语言是 java 语言；
- 2、项目整体框架使用了 SpringBoot 框架；
- 3、项目中使用了 Redis 进行缓存方面的优化，优化热帖排行、缓存验证码、粉丝列表等多功能；
- 4、项目使用 kafka 消息队列处理系统消息通知的发布问题；
- 5、项目中使用 Elasticsearch 搜索引擎完善帖子搜索的相关功能；
- 6、项目整体使用 Spring Security 进行权限控制和安全管理。



学完上面的东西，其实大家就已经把 Java 开发的主流知识都学完了，恭喜大家!!!

JVM 虚拟机:

这是 java 进阶知识中很重要的部分，建议大家好好学（据说对于一个 java 开发程序员最终能不能成长为一个高级的架构师，全靠它了），其实从操作系统开始，你就会发现，我们的学习会从之前的实操变为偏向理论的学习，其实说白了，如果是应付秋招面试的话，最好的方法就是理解+记忆，大家尽量去理解，实在理解不了的就背。

操作系统:

我提供的操作系统的学习视频是王道考研的视频，大家在看的时候可以考虑 1.5 倍速，因为

其实在面试的时候考察操作系统的东西比较少，而且操作系统中的一些重点内容和我们后面要学的 Java 并发编程有很大重合度，这也是为什么建议大家先学操作系统的原因。

Java 并发编程：

学完操作系统，就可以学习这部分知识了，其实这部分主要就是在将 java 的锁和并发问题，但是面试的时候真的非常爱问锁和并发。所以大家一定要好好学这部分。

计算机网络：

这是我们要学的最后一门课，因为这门课主要是背诵，所以放在了最后，大家学到这里估计也快要开始秋招了，所以大家可以边学边背。(计算机网络按照分层，分为应用层，**网络层**，**传输层**，数据链路层和物理层五个部分，其中标红的是重点，大家重点学习)

如果学完了上述知识，那么恭喜大家，所有的 Java 开发的基础知识大家都学完了!!!

但是，这些还远远不够，从此时开始，大家需要准备：

- 1、准备简历（有需要的可以联系我，我可以帮大家设计、修改简历）；
- 2、看面经，这是非常重要的部分，面经可以在牛客上看（类似在牛客搜索：美团 java 面经...等等），你会发现，牛客上的面经里的问题，你在学习过程中可能都没有见到过，你可能根本不知道该怎么回答，但是不要紧，这很正常，学习和应用本来就是两个过程，所以这个时候大家要疯狂的看面经，总结常见问题，然后自己百度答案（基本答案都在 csdn 社区），当然，这部分工作我已经做过一遍了，所以我把所有秋招可能会问的问题和面经都进行了总结，基本上是全覆盖，大家就不需要自己去总结，去查找了，但是，很遗憾，我是用 onenote 记的，暂时我还没有找到可以很好的分享给大家的方法，不过一定是可以分享的，所以大家有需求的可以找我来登记，等我可以分享之后分享给大家（收费）。
- 3、最后就是不断的投递简历，笔试，面试，等意向书，等正式 offer 了!!!（互联网秋招越来越早了，建议大家 6 月底就可以将上述所有问题解决完，然后开始投递，不用害怕，尽管投，大厂也可以试试，提前批的难度比正式批会小一些，有的还免笔试，所以大家完全可以积极尝试）

PS：大家平时有什么 java 相关的问题可以随时问我（大方向的问题，而不是一个 bug 怎么找这种问题，因为我的时间也有限，所以希望大家理解，而且一个程序员如果自己都没法解决 bug，那就不是一个合格的程序员，在解决问题的时候，大家其实也在不断的进步!!!）

最后想送给大家一句话，天道酬勤，没有一条路是好走的，大家既然选择走这条路，就请坚定的走下去!!! 我们江湖再见~~~

（版权所有，翻版必究：版权来自于与鹿）