

JAVA学习指南 - Tomihao - 博客频道 - CSDN

这一部分其实也算是今天的重点，这一部分用来回答很多群里的朋友所问过的问题，那就是LZ你是如何学习Java的，能不能给点建议？今天LZ是打算来点干货，因此咱们就不说一些学习方法和技巧了，直接来谈每个阶段要学习的内容甚至是一些书籍。这一部分的内容，同样适用于一些希望转行到Java的同学。

在大家看之前，LZ要先声明两点。

1、由于LZ本人是Java后端开发出身，因此所推荐的学习内容是Java Web和Java后端开发的路线，非Java Web和Java后端开发的同学请适当参考其学习思想即可，切勿照搬。

2、下面对于【第一部分】的推荐内容，目的是让你尽快成为一个可以参加工作的Java开发者，更适用于处于待业状态，准备转行Java的同学。如果你是在校学生，务必要在学好基础（比如计算机系统、算法、编译原理等等）的前提下，再考虑去进行下面的学习。

第一部分：对于尚未做过Java工作的同学，包括一些在校生以及刚准备转行Java的同学。

一、Java基础

首先去找一个Java的基础教程学一下，这里可以推荐一个地址，或者你也可以参照这个地址上去找相应的视频。

学习Java基础的时候，应该尽量多动手，很多时候，你想当然的事情，等你写出来运行一下，你就会发现不是这么回事儿，不信你就试试。

学完以上内容以后，你应该对Java有一个基本的了解了，你可以用Java语言写出一些简单的程序，并且你用的是最简单的编辑器，比如记事本。

这个时候，不要急于进入下一部分，留下几天好好写一些程序，尽可能熟悉这些基础内容。

二、Web开发

等你写上几天程序以后，你往往会比较迷茫，因为你写的东西似乎看起来毫无用处，比如实现一个简单的计算器，读取一个文件等。这个时候你就应该去学着写一些让你觉得有意思的东西了，所以你应该学习更多的知识。

这些内容主要是Web开发相关的内容，包括HTML/CSS/JS（前端页面）、Servlet/JSP（J2EE）以及MySQL（数据库）相关的知识。

它们的学习顺序应该是从前到后，因此最先学习的应该是HTML/CSS/JS（前端页面），这部分内容你可以去上面的那个runoob网站上找。你可以试着自己写一些页面，当然，你可以尽你最大的努力让它变得最漂亮。这部分内容对于后端Java来说，理论上不是特别重要，但至少要达到可以自己写出一些简单页面的水平。

接下来，你需要学习的是Servlet/JSP（J2EE）部分，这部分是Java后端开发必须非常精通的部分，因此这部分是这三部分中最需要花精力的，而且这个时候，你要学会使用开发工具，而不能再使用记事本了，可以选择eclipse。

当你下载安装好eclipse以后，请视频中的教程一步一步去学习，一定要多动手。关于Servlet/Jsp部分视频的选择，业界比较认可马士兵的视频，因此推荐给大家。当然了，LZ本人并没有看过他的视频，所以不好说的太绝对，如果大家自己有更好的选择，可以坚持自己的，不要被LZ干扰。

原本LZ也是打算出教学视频的，但是由于时间问题，还是决定放弃了。但是如果你看视频的过程中遇到了问题，欢迎来LZ的交流群提问，或者去斗鱼观看LZ的直播提出你的问题，直播地址和群号都在LZ的个人博客左侧。

最后一步，你需要学会使用数据库，mysql是个不错的入门选择，而且Java领域里主流的关系型数据库就是mysql。这部分一般在你学习Servlet/Jsp的时候，就会接触到的，其中的JDBC部分就是数据库相关的部分。你不仅要学会使用JDBC操作数据库，还要学会使用数据库客户端工具，比如Navicat, sqlyog，二选一即可。

三、开发框架

当你学会以上内容以后，这个时候你还不足以参加工作，你还需要继续深造。公司里为了提高开发的效率，会使用一些Java Web框架，因此你还需要学习一些开发框架。

目前比较主流的是SSM框架，即spring、springmvc、mybatis。你需要学会这三个框架的搭建，并用它们做出一个简单的增删改查的Web项目。你可以不理解那些配置都是什么含义，以及为什么要这么做，这些留着后面你去了解。但你一定要可以快速的利用它们三个搭建出一个Web框架，你可以记录下你第一次搭建的过程，相信我，你一定会用到的。

还要提一句的是，你在搭建SSM的过程中，可能会经常接触到一个叫maven的工具。这个工具也是你以后工作当中几乎是必须要使用的工具，所以你在搭建SSM的过程中，也可以顺便了解一下maven的知识。在你目前这个阶段，你只需要在网络上了解一下maven基本的使用方法即可，一些高端的用法随着你工作经验的增加，会逐渐接触到的。

关于学习SSM框架的地址给大家推荐一个，这里面有视频，大家可以去观看。

四、找工作

当你完成开发框架的学习以后，你就该找工作了，在校的找实习，毕业的找全职。与此同时，在找工作的同时，你不应该停下你的学习，准确的说，是你在以后都不能停下学习。

上面这些内容你只是囫圇吞枣的学会了使用，你可以逐步尝试着去了解更多的东西，网络是你最重要的老师。

第二部分：对于参加工作一年以内的同学。

恭喜你，这个时候，你已经拥有了一份Java的工作。这个阶段是你成长极快的阶段，而且你可能会经常加班。

但是加班不代表你就可以松懈了，永远记得LZ说的那句话，从你入行那一刻起，你就要不停的学习。在这一年里，你至少需要看完《Java编程思想》这本书。这本书的内容是帮助你对于Java有一个更加深入的了解，是Java基础的升级版。

这本书很厚，当初看这本书，LZ花了整整三个月。正常速度的话，应该可以在半年左右看完。LZ这里不要求过高，只要你在一年以内把这本书看完即可。当然了，LZ所说的看完，是充分吸收，而不是读一遍就完事了，因此有些内容你可能会看不止一遍。

总而言之，这个阶段的核心学习思想就是，在工作中实践，并且更加深入的了解Java基础。

第二部分：对于参加工作1年到2年的同学。

这部分时间段的同学，已经对Java有了一个更加深入的了解。但是对于面向对象的体会可能还不够深刻，编程的时候还停留在完成功能的层次，很少会去考虑设计的问题。

于是这个时候，设计模式就来了。LZ当时看的是《大话设计模式》这本书，并且写了完整版的设计模式博客。因此，LZ要求大家，最多在你工作一年的时候，必须开始写博客，而设计模式就是你博客的开端。

请记住，LZ所提的基本都是最低要求，因此不要有任何松懈的心理，否则五年后，你不要去羡慕别人高于你的工资，也不要羡慕别人进入了某公司。

这一年，你必须对于设计模式了如指掌，《大话设计模式》可以作为你的开端。当然了，你也可以去看LZ的个人博客去学习，地址是<http://www.cnblogs.com/zuoxiaolong/p/pattern26.html>。

此外，设计模式并不是你这一年唯一的任务，你还需要看一些关于代码编写优化的书。比如《重构 改善既有代码的设计》，《effective java》。

总而言之，这个阶段，你的核心任务就是提高你的代码能力，要能写出一手优雅的代码。

第三部分：对于参加工作2年到3年的同学

有的同学在这个时候觉得自己已经很牛逼了，于是忍不住开始慢慢松懈。请记住，你还嫩的多。

这个阶段，有一本书是你必须看的，它叫做《深入理解Java虚拟机》。这本书绝对是Java开发者最重要的书，没有之一。在LZ眼里，这本书的重要性还要高于《Java编程思想》。

这本书的内容是帮助你全面的了解Java虚拟机，在这个阶段，你一定已经知道Java是运行在JVM之上的。所以，对于JVM，你没有任何理由不了解它。LZ之前有写过JVM系列的知识，可以去看一下，地址是<http://www.cnblogs.com/zuoxiaolong/category/508918.html>。

另外，在过去2年的工作当中，你肯定或多或少接触过并发。这个时候，你应该去更加深入的了解并发相关的知识，而这部分内容，LZ比较推荐《Java并发编程实战》这本书。只要你把这本书啃下来了，并发的部分基本已经了解了十之六七。

与此同时，这个阶段你要做的事情还远不止如此。这个时候，你应该对于你所使用的框架应该有了更深入的了解，对于Java的类库也有了更深入的了解。因此，你需要去看一些JDK中的类的源码，也包括你所使用的框架的源码。

这些源码能看懂的前提是，你必须对设计模式非常了解。否则的话，你看源码的过程中，永远会有这样那样的疑问，这段代码为什么要这么写？为什么要定义这个接口，它看起来好像很多余？

由此也可以看出，这些学习的过程是环环相扣的，如果你任何一个阶段拉下来了，那么你就真的跟不上了，或者说是一步慢步步慢。而且LZ很负责的告诉你，LZ在这个阶段的时候，所学习的东西远多于这里所罗列出来的。因此千万不要觉得你已经学的很多了，LZ所说的这些都只是最低要求，不光是LZ，很多人在这个时间段所学习的内容都远超本文的范围。

如果你不能跟上节奏的话，若干年后，如果不是程序猿市场还不错的话，你很可能不仅仅是工资比别人低，公司没别人好，而是根本就找不到工作。

总而言之，这个阶段，你需要做的是深入了解Java底层和Java类库（比如并发那本书就是Java并发包java.concurrent的内容），也就是JVM和JDK的相关内容。而且还要更深入的去了解你所使用的框架，方式比较推荐看源码或者看官方文档。

另外，还有一种学习的方式，在2年这个阶段，也应该启用了，那就是造轮子。

不要听信那套“不要重复造轮子”的论调，那是公司为了节省时间成本编造出来的。重复造轮子或许对别人没有价值，因为你造的轮子可能早就有了，而且一般情况下你造出来的轮子还没有现存的好。但是对别人没有价值，不代表对你自己没有价值。

一个造轮子的过程，是一个从无到有的过程。这个过程可以对你进行系统的锻炼，它不仅考察你的编码能力，还考察你的框架设计能力，你需要让你的轮子拥有足够好的扩展性、健壮性。

而且在造轮子的过程中，你会遇到各种各样的难题，这些难题往往又是你学习的契机。当你把轮子造好的时候，你一定会发现，其实你自己收获了很多。

所以，这个阶段，除了上面提到的了解JVM、JDK和框架源码以外，也请你根据别人优秀的源码，去造一个任何你能够想象出来的轮子。

第四部分：参加工作3年到4年的同学

这个阶段的同学，提升已经是很难了，而且这个阶段的学习往往会比较多样化。

因为在前3年的过程中，你肯定或多或少接触过一些其它的技术，比如[大数据](#)、分布式缓存、分布式消息服务、分布式计算、软负载均衡等等。这些技术，你能精通任何一项，都将是未来面试时巨大的优势，因此如果你对某一项技术感兴趣的话，这个时候可以深入去研究一下。这项技术不一定是你工作所用到的，但一定是相关的。

而且在研究一门新技术时，切忌朝三暮四。有的同学今天去整大数据，搞搞[Hadoop](#)、[Hbase](#)一类的东西。过不了一段时间，就觉得没意思，又去研究分布式缓存，比如[Redis](#)。然后又过不了一段时间，又去研究分布式计算，比如整整Mapreduce或者storm。

结果到最后，搞得自己好像什么都会一样，在简历上大言不惭的写上大数据、分布式缓存、分布式计算都了解，其实任何一个都只是浮于表面。到时候面试官随便一问，就把你给识破了。

一定要记住，作为一个程序猿，平日里所接触的技术可能会很多，但是想要让一门技术成为你的优势，那么一定是你对这门技术的了解强过绝大多数人才行。

因此在这个阶段，你就不能再简单的去学习前3年的内容了，虽然前面的学习如果还不够深入的话依旧要继续，但这个时候你应该更多的考虑建立你的优势，也可以称为差异性。

差异性相信不难理解，就是让你自己变得与众不同。你前面三年的学习足够你成为一名基本合格的Java开发者，但你离成为一名优秀的Java开发者还有很大的距离。

所谓优秀，即能别人所不能。而你前三年所学习的内容，是很多做过几年的Java开发都能够掌握的。那么为了让自己有差异性，你就需要

另辟蹊径，找一个方向深入研究下去，以期在将来，你能够成为这个领域的专家，比如分布式计算领域的专家，大数据领域的专家，并发领域的专家等等。

此外，你除了建立你的差异性之外，还要去弥补你基础上的不足，直到现在，LZ都没有提及基础知识。原因是基础是很枯燥无味的，学的太早不仅容易懵逼，而且懵逼的同时还容易产生心理阴影，以至于以后再不想去研究这些基础。但基础又是你深入研究一些领域时所必须掌握的，比如你去研究分布式计算，你不懂算法你玩个毛毛？比如你去做分布式缓存，你对计算机系统的内存不了解，你如何去做缓存？如果你的基础本来就非常强，那么恭喜你，相信你在之前的工作中已经充分体会到了这些基础对你的帮助。但LZ相信大部分人的基础都很薄弱，哪怕是科班毕业的人，很多人也不敢说自己当初的基础学的多么强大，比如算法、计算机系统原理、编译原理这些。

但是每个人时间都是有限的，而且这些基础的书籍每一本读下来，没个一年半载的，还真拿不下来，因此还是要有有所取舍的。虽然艺多不压身，但问题是艺多是有代价的，是需要你付出时间和精力，而LZ个人更赞成在同等代价的情况下获取最大的收获。

首先，LZ比较推崇的基础书籍有三本，分别是《深入理解计算机系统》，《tcp/ip详解 卷一、二、三》，《数据结构与算法》。其中TCP/IP有三本书，但我们这里把这三本看成是一本大书。

这三本分别适合三种人，《深入理解计算机系统》比较适合一直从事Java Web开发和APP后端开发工作的人群。《tcp/ip详解 卷一、二、三》比较适合做网络编程的人群，比如你使用netty去开发的话，那么就要对TCP/IP有更深入的了解。而《数据结构与算法》这本书，则比较适合做计算研究工作的人，比如刚才提到的分布式计算。

另外，LZ要强调的是，这里所说的适合，并不是其它两本对你就没有用。比如你做Java Web和APP后端开发，《tcp/ip详解 卷一、二、三》这本书对你的作用也是很大的。这里只是分出个主次关系而已，你要是时间足够的话，能把三本都精读那当然最好不过了。但如果时间有限的话，那么就先挑对你帮助最大的书去读。

理论上讲，这一年你能把这三本其中一本精读下来，就已经非常厉害了。有了基础，有了前面的工作经验，你就可以去开拓属于你的领域了。

在这一年里，一定要规划好自己的领域，建立好自己的优势，制造出差异性。如果你对自己的领域不够清晰的话，随着你工作的时间日益增多，你接触的技术会越来越多，这个时候，你很容易被淹死在技术的海洋里，看似接触的技术越来越多，会用的也越来越多，但你毫无优势。

有的同学可能会问，“LZ，我也不知道我的领域是什么啊？怎么办呢？”

对于这种人，LZ只想说，“卧槽，这还问我？要不干脆我替你学习得了，好不好？”

第五部分：参加工作4年到5年的同学

经过前面一年的历练，相信你在自己所钻研的领域已经有了自己一定的见解，这个时候，技术上你应该已经遇到瓶颈了。

这个时候不要着急提高自己的技术，已经是时候提高你的影响力了，你可以尝试去一些知名的公司去提高你的背景，你可以发表一些文章去影响更多的人。当然，你也可以去Github创建一个属于你的开源项目，去打造自己的产品。这次的开源项目不同于之前的造轮子，你这个时候是真的要去尽量尝试造出真正对别人有价值的轮子。

技术学到这个阶段，很容易遇到瓶颈，而且往往达到一定程度后，你再深入下去的收效就真的微乎其微了，除非你是专门搞学术研究的。然而很可惜，大部分程序猿做不到这一步，那是科学家做的事情。

这个时候提高影响力不仅仅是因为技术上容易遇到瓶颈，更多的是影响力可以给你创造更多的机会。程序猿在某种程度上和明星很像，一个好的电视剧和电影就可以成就一批明星，程序猿有的时候也是，一个好的项目就可以成就一群程序猿。

比如国内几个脍炙人口的项目，像淘宝、支付宝、QQ、百度、微信等等。这每一个项目，都成就了一批程序猿。LZ敢说，这里面任何一个项目，如果你是它的核心开发，光是这样一个Title，就已经是你非常大的优势。更何况还不止如此，Title说到底也是个名头，更重要的是，这种项目在做的时候，对你的历练一定也是非常给力的。

而你如果想要参与这样的项目，除了靠运气之外，影响力也是很重要的一个手段。比如你在分布式计算领域有一定的影响力，那么如果有什么好的关于分布式计算的项目，对方就很可能邀请你。就算人家不邀请你，你自己主动去面试的时候，对方如果知道你在这个领域的影响力，也肯定会起到很大的作用，而这个作用，甚至可能会超过你现在的技术能力。

所以，在这个阶段，你最大的任务是提高自己的影响力，为自己未来的十年工作生涯那一天做准备。如果你能够靠你的影响力和以前积累的技术，参与到一个伟大的项目当中，那么你后面的五年也就有着落了。

当然了，LZ现在满打满算，做程序猿也就4年半不到，因此关于4年到5年这一部分，LZ的见解不一定是对的，就算是对的，也不一定是适合任何人的。所以，希望大家自己有的判断力，去决定到底该如何度过这一年。