

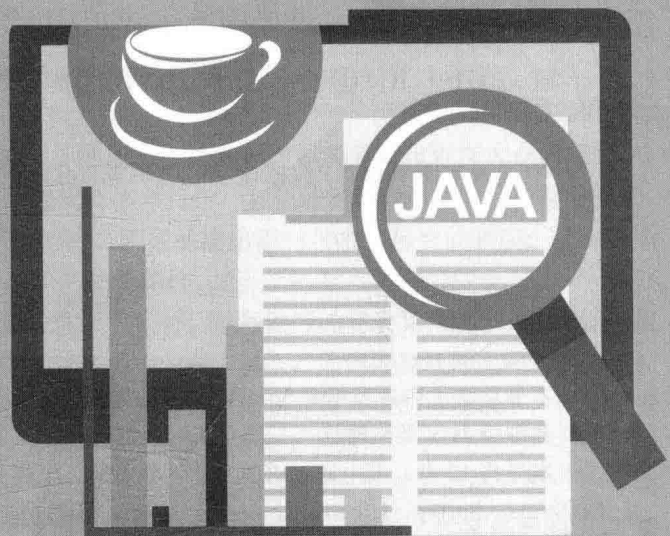
Java

王波 ● 编著

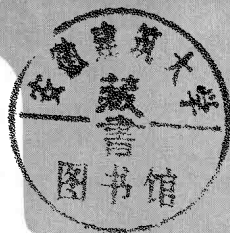
架构师指南

阿里、华为等知名企业工程师**联袂推荐**

资深Java专家多年经验总结，介绍从程序员进阶到架构师必备技能的完整教程
基于Java官方API的解读，从架构师的独特角度讲解Java知识和技能
企业级开发的经典示例，全程项目驱动，涵盖电商支付接口的综合调试



Java



架构师指南

王波 ● 编著

人民邮电出版社
北京

图书在版编目 (C I P) 数据

Java架构师指南 / 王波编著. — 北京 : 人民邮电出版社, 2018.6

ISBN 978-7-115-48066-8

I. ①J… II. ①王… III. ①JAVA语言—程序设计
IV. ①TP312.8

中国版本图书馆CIP数据核字(2018)第058309号

内 容 提 要

本书总结了作者多年来在 Java Web 方面的开发经验, 全面阐述了 Java 架构师所需掌握的知识和技能, 并围绕 Java 架构师这一主题介绍相关的内容。

本书共 12 章。书中通过讲解企业管理系统、电商系统、报表系统等项目的实际开发流程, 把流行的 Struts、Spring、Hibernate、Spring MVC、MyBatis 等框架整合起来, 再从代码层面讲述 Maven、WebService、POI 等技术, 让读者在学习 Java 架构师必备的专业技能的同时, 了解项目开发的整个过程。在项目运维方面, 本书还讲解了 SonarQube 和 Jenkins 开源组件, 以拓宽架构师的知识广度。

本书可以帮助不同技术层次的读者在短时间内掌握 Java 架构师必备的知识, 缩短从程序员到架构师的进阶时间。因为书中的每份代码都有详细的注释和解析, 很方便读者领会, 所以不论是刚步入职场的新手, 还是有一定工作经验的开发人员, 本书都同样适用。

◆ 编 著 王 波

责任编辑 杨海玲

责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京鑫正大印刷有限公司印刷

◆ 开本: 800×1000 1/16

印张: 25.75

字数: 643 千字

印数: 1—3 000 册

2018 年 6 月第 1 版

2018 年 6 月北京第 1 次印刷

定价: 89.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号



王波

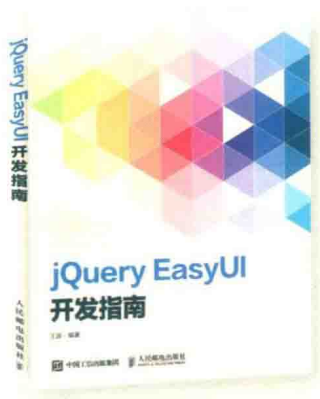
软件架构师。曾就职于中软国际。从业多年，成功地开发过基于通信、电商、数据管理等相关业务的项目，在Java Web领域拥有丰富的经验。擅长搭建以Struts、Spring框架为基础的项目架构、并且在此基础上选择合适的ORM框架。在日常工作中，坚持认为程序员应该以全栈作为要求，以架构师作为目标，才可以更好地将客户的需求用计算机语言描述出来；认为在软件行业未来的发展中，Java Web领域的开发仍然会以MVC模式为主，并且在此基础上不断演化，开发过程则会更加依赖于框架，所以架构师会成为不可缺少的人才。

程序员走向架构师是必经之路。本书基于Java官方API进行解读，从架构师的角度讲解Java知识和技能，并从搭建虚拟机开始，学习常用的Linux命令，力争使程序员能在较短的时间内成功迈入架构师的殿堂。

近年来编程领域的知识愈加丰富，使广大程序员应接不暇。如何在有限的时间内高效学习是一个棘手的问题。本书作者根据自身的丰富经验，精选了Java中必须掌握的核心技能，并以项目为驱动，循序渐进，详细讲解了企业级项目和电商平台，让读者在学习知识的同时以实战加深印象，可谓是高屋建瓴。

如何能在短时间内达到架构师的水平？面对难以入手的海量知识，读者往往一头雾水，相信通过阅读本书，每位读者都能从中找到答案。

好书分享



前言

互联网的发展带动了各行各业信息化的趋势，一大批高新企业如雨后春笋般出现在大众的视野中。于是，不同类型的软件项目应运而生。在这些琳琅满目的项目中，有企业管理、电商平台、财务报表、金融银行、医疗器械、智慧城市和大数据分析等类型。项目的层出不穷带来了巨大的利润，让高新企业不断地成长起来，与此同时，也带来了许多相关的就业岗位。

当然，要顺利地完成这些项目，就需要大量的软件工程师。这种硬性的需求又养活了一大批培训机构，从事软件行业的人员当初是凤毛麟角，现在依然是供不应求。那么，如何提高软件工程师的开发技能就成了一个无法回避的问题。诚然，公司可以不定期进行培训，提高开发人员的技能水平，但从更普遍、更直接的意义来说，提高技能水平的最佳方式还是系统地阅读相关书籍。

回到正题，项目从设计到完成的每一个环节，都需要精确地把控，如果这方面做不好，会让项目陷入困境，得不偿失！同时，在开发语言的选择上，也需要相当慎重。例如，大家熟悉的 Java 语言，它最大的优点就是跨平台运行。如果使用 Java 语言开发项目，程序员关注的无非是在某个系统环境下完成代码的编写和调试，至于最终需要用在哪儿，没有必要过多地关心，因为无论是在 Windows 系统还是在 Linux 系统，Java 程序都可以顺利地部署，流畅地运行。Java 跨平台的优点得到了很多公司的青睐，他们纷纷把自己公司的核心编程语言确定为 Java。这样的情况愈演愈烈，以至于 Java 语言在 J2EE 方向的发展非常迅速，成为企业级开发的首选。Java 与众多优质的第三方框架搭配起来使用，更是让项目的开发进入了一个非常高效、便捷、可复用的时代。举个经典的例子，大家熟知的 SSH 框架技术集合，就是使用 Java 语言开发，再把 Struts、Spring、Hibernate 三者结合起来，组成的一套成熟的开发框架。这套框架曾经风靡全球，引起了业界学习的浪潮，促使很多公司前赴后继。不过，在 2005 年 6 月的 JavaOne 大会上，Sun 公司发布标准版 Java SE 6 的时候，顺带将 J2EE 改成了 Java EE，但因为历史原因，J2EE 的提法仍然经常存在。

如果说 Java 语言的跨平台特性是很受欢迎的，那么 Java 语言的安全性也是非常让人放心的，这主要得益于 Java 语言中设计的沙箱安全模型。Java 代码的执行全部在类装载器、类文件检验器、Java 虚拟机内置的安全特性、安全管理器这 4 个组件的安全策略下完成，极大地保障了程序运行的安全。另外，Java 语言还提供了 AWT 和 Swing 方面的开发，这两者都是基于图形化用户界面的，也就是业界常说的 GUI 层面的开发。但是，Java 语言在 GUI 领域的优势并不那么明显，更多的开发人员仍然选择 C++，绕过虚拟机直接与操作系统交互。而与此同时，Java 在企业级方面的优势却越来越大，以至于出现了一枝独秀的局面。

计算机语言从机器语言、汇编语言发展到现在的高级语言，这个过程中诞生了很多种语言。有些语言已经逐步退出历史舞台，有些语言仍然在小众化的范围内存在。而 Java 语言，经历了二十多年的发展，仍然保持着旺盛的生命力，在编程语言排行榜中高居不下，Java 程序员的数量也与日俱增，这种现象主要是由 Java 自身的优势决定的。作为开发人员，需要关注的并不是底层的核心，更

多的是 Java 带给我们的简单、直观、易于使用的平台。因此，程序员不用关心虚拟机复杂的结构和每一步的运行情况，只需要关注项目业务的代码即可。这种易于接受的情形，让更多人把开发当成了一种乐趣。

最近，在业内流行起来的全栈工程师的定位更像是高级程序员，而架构师则需要站在更高的层面思考问题。作为 Java 架构师，不但要懂得前端插件化的开发理念，为项目选择合适的前端插件，还需要精通后端开发，为项目选择合适的框架，这样才能高效地完成工作。否则，极有可能出现事倍功半的情况。如果说需要弥补架构缺陷，最乐观的情况是通过加班实现，最糟糕的情况是直接导致项目失败。因为项目经理可能并不会深入了解具体的代码，他通常会参考架构师的意见，所以架构师的意见就显得极为重要。因此，本书在讲解架构师必备的知识技能的同时，也会穿插项目的管理知识。

Java 技术发展迅速，本书旨在结合最近几年流行的技术，带领读者见证从项目启动到收尾的全过程，力求在短时间内让读者掌握 Java 架构师必备的知识技能，并且能在日后的工作中做到游刃有余，既可以在掌握扎实的基础知识后，熟练地搭建框架，又可以为项目经理提供专业的参考意见。

内容特色

市场上的技术图书琳琅满目，令人难以选择。但是，这些书中讲解程序员进阶到架构师的过程的书却很少，这不得不说不是一件令人遗憾的事情。本书的出现将会带给读者全新的认知，帮助读者在短时间内掌握架构师必备的知识，缩短从程序员到架构师的进阶时间，早日达到架构师的高度。

另外，本书专注于 Java 企业级开发，从最基本的企业管理系统开始，到颇具特色的电商系统都有涉及，还附带了诸如报表系统、员工信息系统、代码扫描平台的开发等，基本上包含了业界常用的项目类型。书中的项目都是基于 BS 架构的，与 Java 程序员的技术成长趋势完全匹配。读者可以通过阅读本书，并结合提供的源码进行练习，以做到融会贯通。

本书结合实际、深入浅出，以项目为驱动，阐述了我多年来在 Java Web 方面的开发经验。同时，本书通俗易懂。虽然没有把 Java 中特别浅显的内容用独立的章节来讲解，但这些内容都会在本书的代码中出现，读者可以结合程序自行理解，或者通过阅读注释学习，都可以很容易地理解它们的意思。综合来看，本书不但适合刚步入职场的新手，还适合有一定工作经验的开发人员，因为书中的每份代码都会有详细的注释和代码解析，方便不同技术层次的读者领会代码的含义。本书通过讲解企业管理系统的开发过程，让读者全面掌握 Java EE 的精粹内容，之后再通过其他几章的讲解，让读者学习到电商系统、报表系统、员工信息系统、代码扫描平台的开发，不断地拓展 Java 架构师技能的广度和深度。

结构与组织

本书的核心内容是讲解 Java 架构师必备的知识技能。

- 第 1 章讲述编程基础，通过搭建简单的环境，开发第一个程序。
- 第 2 章从项目管理方面介绍需求调研的整个过程以及项目文档的撰写方法，让读者初步了解项目的启动、规划、执行、监控和收尾五大过程组。
- 第 3 章到第 9 章讲述企业管理系统的具体开发过程，重点讲述 Servlet、SSH、Spring MVC 等框架，并且加入了近年来流行的 Redis、EhCache、MongoDB 等新知识。

- 第 10 章详细讲解电商系统的开发，项目本身比较简单，最大的亮点是集成了支付接口，以帮助读者应对近些年来不同公司对程序员提出的需要有支付经验的痛点，帮 Java 架构师丰富知识，提高技能。未来做支付是必备的经验，但很多公司的项目并没有提供这样的机会。
- 第 11 章提出产品思维的概念，以此作为架构师的必备常识，旨在帮助架构师理解和建立产品思维，以方便在项目开发的同时设计出良好的产品雏形。与此同时，在管理系统中集成了 Bootstrap 和 ECharts 报表，以实例证明产品化的可行方案。
- 第 12 章讲的是项目运维，包含 Java 架构师需要具备的项目运维知识。另外，本章开发了 SonarQube 代码扫描平台并且初步配置了 Jenkins 持续集成，阐述了项目数据的迁移方案，讲述了 ETL 工具的使用。

目标读者

本书特别适合 Java Web 领域的开发人员以及刚步入职场的新手。本书通过讲述 Java 架构师必备的知识技能，让广大读者在原有知识的基础上更上一个台阶，争取早日实现架构师的梦想。

对于架构师的定义，每个人的看法都不尽相同，我结合自己多年的工作经验，也只是大致定义了一个范围，希望可以帮助到别人。读者可以结合自己的实际情况，通过阅读本书，不断地扩展和充实这种范围，以达到自己理想中的境界。“不想当将军的士兵，不是好士兵。”在软件行业中，也似乎有这样一句话：“不想当架构师的程序员，不是好程序员。”虽然这看似是一种调侃，但从学习的角度来说，成为架构师，显然是一个好的目标！人只有在心里有了目标，才会变得更加幸福。

如果你不希望一直停留在 Java 的初级阶段，想在未来成为架构师，那么本书非常适合用来全面提高自己的开发水平。如果你想转项目经理，那么本书同样适合你，因为书中的每个项目都是完整的迭代过程。

本书以项目为驱动，是非常科学的学习方法。读者需要自行下载源码，自己搭建环境，对照章节中讲解的过程，逐步深入地学习 Java 技术。一个项目的代码量多得超乎想象，如果读者从第一行代码开始就要逐个阅读和编写，那么等学习完这个项目可能就需要一年的时间。因此，本书合理地安排了章节，科学地封装了技术知识，让读者在轻松的氛围下对照源码即可完成技术水平的提升，这不得不说是个创新！相信大家在学习本书后，都能够学习到“干货”，极大地提高自己的 Java 水平，达到架构师的高度。

致谢

本书得以顺利出版，离不开我自己多年来的努力。正是因为我有作为 Java 架构师的觉悟，才让自己在平时从未停下脚步，并且积极学习。不论是在工作中还是在业余生活中，我都会认真总结、分析近年来 Java 技术领域的知识。所谓学习的诀窍，对每个人都是一样的，就是不停耕耘、努力奋进，在这个过程中，我们总会收获良多。另外，还要感谢家人、朋友对我的帮助，感谢人民邮电出版社和杨海玲编辑对我的信任和支持。

由于水平有限，书中难免有不足之处，恳请专家和读者批评指正。欢迎读者通过电子邮件(453621515@qq.com)与我交流。

资源与支持

本书由异步社区出品，社区（<https://www.epubit.com/>）为您提供相关资源和后续服务。

配套资源

本书提供如下资源：

- 本书源代码；
- 书中彩图文件。

要获得以上配套资源，请在异步社区本书页面中点击 [配套资源](#)，跳转到下载界面，按提示进行操作即可。注意：为保证购书读者的权益，该操作会给出相关提示，要求输入提取码进行验证。

提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，点击“提交勘误”，输入勘误信息，点击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的 100 积分。积分可用于在异步社区兑换优惠券、样书或奖品。

详细信息 写书评 提交勘误

页码: 页内位置 (行数): 勘误次数:

B I U

字数统计

扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。



与我们联系

我们的联系邮箱是 contact@epubit.com.cn。

如果您对本书有任何疑问或建议，请您发邮件给我们，并请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 www.epubit.com/selfpublish/submission 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 <https://www.epubit.com>。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术等。



异步社区



微信服务号

目 录

| | | | |
|--------------------|----|------------------|-----|
| 第 1 章 编程基础 | 1 | 3.4.2 销售数据查询报表 | 53 |
| 1.1 程序员进阶 | 1 | 3.5 月度版本 | 57 |
| 1.2 选择开发工具 | 3 | 3.6 小结 | 57 |
| 1.3 安装 JDK | 5 | 第 4 章 项目部署 | 58 |
| 1.4 安装 Tomcat 服务器 | 6 | 4.1 项目打包 | 58 |
| 1.5 Hello World 程序 | 8 | 4.2 项目发布 | 62 |
| 1.6 小结 | 9 | 4.3 构建工具 | 63 |
| 第 2 章 需求调研 | 10 | 4.3.1 Ant 环境搭建 | 63 |
| 2.1 搭建关系 | 10 | 4.3.2 Ant 经典实例 | 64 |
| 2.2 正式立项 | 11 | 4.3.3 Maven 环境搭建 | 66 |
| 2.3 需求调研 | 11 | 4.3.4 Maven 经典实例 | 66 |
| 2.4 输出文档 | 12 | 4.4 部署工具 | 70 |
| 2.5 技术选型 | 16 | 4.4.1 mstsc | 71 |
| 2.6 数据流图 | 16 | 4.4.2 VMware | 71 |
| 2.7 UML 建模 | 16 | 4.4.3 Xmanager | 75 |
| 2.8 项目开工会 | 17 | 4.4.4 WinSCP | 79 |
| 2.9 小结 | 18 | 4.4.5 JD-GUI | 80 |
| 第 3 章 项目开发 | 19 | 4.5 小结 | 81 |
| 3.1 定义范围和 WBS 分解 | 19 | 第 5 章 编程环境 | 82 |
| 3.2 企业管理系统框架搭建 | 20 | 5.1 Linux 系统介绍 | 82 |
| 3.3 Servlet 方式开发 | 20 | 5.2 Linux 系统安装 | 83 |
| 3.3.1 前端验证 | 22 | 5.3 Linux 常用命令 | 84 |
| 3.3.2 后端验证 | 25 | 5.3.1 基本命令 | 84 |
| 3.3.3 注册功能 | 28 | 5.3.2 高级命令 | 90 |
| 3.3.4 综合调试 | 39 | 5.3.3 部署命令 | 93 |
| 3.3.5 Servlet 注解 | 43 | 5.3.4 shell 脚本入门 | 100 |
| 3.4 不依赖框架的开发 | 45 | 5.4 DOS 介绍 | 101 |
| 3.4.1 销售数据导入报表 | 46 | 5.4.1 基本命令 | 102 |
| | | 5.4.2 高级命令 | 104 |

| | | | | | |
|-------|-------------------|-----|--------|----------------------|-----|
| 5.5 | SVN 与 Git 版本控制 | 105 | 6.5 | 文件与流 | 139 |
| 5.5.1 | SVN 常用操作 | 106 | 6.5.1 | File 类 | 139 |
| 5.5.2 | Git 常用操作 | 110 | 6.5.2 | 字节流 | 140 |
| 5.6 | Visio 画图 | 114 | 6.5.3 | 字符流 | 142 |
| 5.7 | Axure 原型设计 | 115 | 6.6 | 异常处理 | 145 |
| 5.8 | 代码编辑器 | 116 | 6.6.1 | try catch 捕获 | 145 |
| 5.9 | 小结 | 119 | 6.6.2 | throw throws 抛出 | 147 |
| 第 6 章 | 架构师思想 | 120 | 6.6.3 | 自定义异常 | 147 |
| 6.1 | 数据类型 | 120 | 6.7 | 代码调试 | 148 |
| 6.1.1 | Object | 121 | 6.7.1 | Web 调试方式 | 148 |
| 6.1.2 | byte 和 Byte | 122 | 6.7.2 | Java 调试方式 | 149 |
| 6.1.3 | short 和 Short | 122 | 6.8 | 多线程 | 150 |
| 6.1.4 | int 和 Integer | 122 | 6.8.1 | 线程创建 | 150 |
| 6.1.5 | long 和 Long | 123 | 6.8.2 | 线程调度 | 152 |
| 6.1.6 | float 和 Float | 123 | 6.8.3 | 线程同步 | 152 |
| 6.1.7 | char 和 Character | 124 | 6.9 | 监听器 | 155 |
| 6.1.8 | double 和 Double | 124 | 6.9.1 | 实现 Listener | 155 |
| 6.1.9 | boolean 和 Boolean | 125 | 6.9.2 | 配置 Listener | 155 |
| 6.2 | 类与对象 | 125 | 6.9.3 | 测试 Listener | 156 |
| 6.2.1 | 三大特性 | 126 | 6.10 | 过滤器 | 157 |
| 6.2.2 | 属性和方法 | 126 | 6.10.1 | 实现 Filter | 157 |
| 6.2.3 | 抽象类和接口 | 128 | 6.10.2 | 配置 Filter | 158 |
| 6.3 | 数组 | 129 | 6.10.3 | 测试 Filter | 159 |
| 6.3.1 | 创建数组 | 129 | 6.11 | 反射机制 | 160 |
| 6.3.2 | 数组的初始化 | 130 | 6.11.1 | ReflectDemo | 160 |
| 6.3.3 | 数组的排序 | 130 | 6.11.2 | InvokeDemo | 162 |
| 6.4 | 集合类 | 130 | 6.12 | XML | 163 |
| 6.4.1 | Collection 接口 | 131 | 6.12.1 | 创建 XML | 163 |
| 6.4.2 | Set 接口 | 132 | 6.12.2 | 解析 XML | 164 |
| 6.4.3 | List 接口 | 134 | 6.13 | WebService | 166 |
| 6.4.4 | Queue 接口 | 135 | 6.13.1 | 实现服务端 | 167 |
| 6.4.5 | Map 接口 | 136 | 6.13.2 | 实现客户端 | 168 |
| 6.4.6 | HashMap 实现类 | 136 | 6.14 | Ajax 传递 | 169 |
| 6.4.7 | TreeMap 实现类 | 137 | 6.14.1 | Ajax 是什么 | 169 |
| 6.4.8 | Hashtable 实现类 | 137 | 6.14.2 | Ajax 的 JavaScript 语法 | 171 |
| 6.4.9 | Iterator 迭代器 | 137 | 6.14.3 | Ajax 的 jQuery 语法 | 174 |
| | | | 6.15 | JSP 内置对象 | 175 |

| | | | |
|----------------------|-----|-------------------------------|-----|
| 6.15.1 request | 175 | 第 8 章 Struts Spring Hibernate | 226 |
| 6.15.2 response | 176 | 8.1 框架搭建 | 226 |
| 6.15.3 session | 177 | 8.1.1 整体规划 | 226 |
| 6.15.4 application | 179 | 8.1.2 MVC 理念 | 227 |
| 6.15.5 out | 180 | 8.2 框架集成 | 233 |
| 6.15.6 pageContext | 180 | 8.2.1 Struts 2 的集成 | 233 |
| 6.15.7 config | 181 | 8.2.2 Spring 3 的集成 | 234 |
| 6.15.8 page | 182 | 8.2.3 Hibernate 的集成 | 235 |
| 6.15.9 exception | 183 | 8.2.4 前端插件的集成 | 236 |
| 6.16 Log4j 配置 | 184 | 8.3 权限管理 | 236 |
| 6.16.1 配置 Log4j | 184 | 8.3.1 业务设计 | 237 |
| 6.16.2 配置 Logback | 186 | 8.3.2 程序设计 | 238 |
| 6.17 小结 | 188 | 8.3.3 数据库设计 | 247 |
| 第 7 章 数据库 | 189 | 8.4 架构设计 | 250 |
| 7.1 MySQL | 189 | 8.4.1 逻辑层 | 250 |
| 7.1.1 安装 | 189 | 8.4.2 业务层 | 255 |
| 7.1.2 命令 | 191 | 8.4.3 持久层 | 256 |
| 7.1.3 profiling | 196 | 8.4.4 架构优化 | 258 |
| 7.1.4 SQLyog | 198 | 8.4.5 架构拓展 | 259 |
| 7.2 Oracle | 199 | 8.4.6 配置文件 | 268 |
| 7.2.1 安装 | 200 | 8.5 报表导出 | 282 |
| 7.2.2 命令 | 202 | 8.5.1 POI 介绍 | 282 |
| 7.2.3 PLSQL | 205 | 8.5.2 POI 导出前端实现 | 283 |
| 7.3 NoSQL | 207 | 8.5.3 POI 导出后端实现 | 284 |
| 7.3.1 MongoDB | 207 | 8.5.4 下载 Excel 文件 | 289 |
| 7.3.2 Redis | 210 | 8.5.5 CSV 介绍 | 290 |
| 7.4 MyBatis | 212 | 8.5.6 CSV 导出前端实现 | 291 |
| 7.4.1 MyBatis 环境搭建 | 212 | 8.5.7 CSV 导出后端实现 | 292 |
| 7.4.2 MyBatis 配置参数 | 215 | 8.5.8 下载 CSV 文件 | 295 |
| 7.5 Hibernate | 219 | 8.5.9 导出功能 XML 文件配置 | 297 |
| 7.5.1 Hibernate 环境搭建 | 219 | 8.6 加入缓存机制 | 297 |
| 7.5.2 Hibernate 配置参数 | 222 | 8.6.1 Ehcache 的搭建 | 297 |
| 7.6 函数 | 222 | 8.6.2 Ehcache 的使用 | 300 |
| 7.7 游标 | 223 | 8.7 解决并发问题 | 301 |
| 7.8 存储过程 | 224 | 8.7.1 连接池 | 301 |
| 7.9 小结 | 225 | 8.7.2 Nginx | 302 |

| | | | |
|------------------|-----|---------------------|-----|
| 8.8 小结 | 306 | 10.4.1 开发账号 | 352 |
| 第9章 Spring MVC | 307 | 10.4.2 支付接口集成 | 354 |
| 9.1 框架搭建 | 307 | 10.4.3 支付接口调试 | 357 |
| 9.1.1 整体规划 | 307 | 10.5 JDBC 连接类 | 361 |
| 9.1.2 技术选型 | 308 | 10.6 小结 | 364 |
| 9.1.3 项目结构 | 309 | 第11章 产品思维 | 365 |
| 9.2 详细设计 | 309 | 11.1 何谓产品化 | 365 |
| 9.2.1 业务设计 | 310 | 11.1.1 三个标准 | 365 |
| 9.2.2 原型设计 | 310 | 11.1.2 软件服务 | 366 |
| 9.2.3 数据库设计 | 310 | 11.2 软件产品化 | 368 |
| 9.3 架构设计 | 311 | 11.2.1 开发文档 | 368 |
| 9.3.1 逻辑层 | 311 | 11.2.2 产品风格 | 368 |
| 9.3.2 业务层 | 315 | 11.2.3 前端框架 | 369 |
| 9.3.3 持久层 | 316 | 11.2.4 后端框架 | 371 |
| 9.3.4 配置文件 | 323 | 11.3 图表项目 | 372 |
| 9.4 POI 导入 | 334 | 11.3.1 Bootstrap 插件 | 372 |
| 9.4.1 POI 导入前端实现 | 334 | 11.3.2 ECharts 图表 | 374 |
| 9.4.2 POI 导入后端实现 | 336 | 11.4 小结 | 376 |
| 9.5 小结 | 342 | 第12章 项目运维 | 377 |
| 第10章 电商平台 | 343 | 12.1 平台维护 | 377 |
| 10.1 框架搭建 | 343 | 12.1.1 系统上线 | 378 |
| 10.1.1 整体规划 | 343 | 12.1.2 运维报告 | 380 |
| 10.1.2 技术选型 | 344 | 12.2 SonarQube 代码扫描 | 381 |
| 10.2 详细设计 | 344 | 12.2.1 环境搭建 | 381 |
| 10.2.1 业务设计 | 344 | 12.2.2 PMD 模板方式 | 381 |
| 10.2.2 原型设计 | 344 | 12.2.3 Java 自定义规则 | 386 |
| 10.2.3 数据库设计 | 345 | 12.3 Jenkins 自动化部署 | 394 |
| 10.3 架构设计 | 346 | 12.3.1 部署介绍 | 394 |
| 10.3.1 逻辑层 | 346 | 12.3.2 搭配使用 | 394 |
| 10.3.2 业务层 | 348 | 12.4 数据迁移 | 397 |
| 10.3.3 持久层 | 349 | 12.4.1 场景分析 | 397 |
| 10.3.4 数据通道 | 350 | 12.4.2 ETL 工具 | 397 |
| 10.4 支付接口 | 352 | 12.5 小结 | 399 |

第 1 章

编程基础

程序员到架构师的进阶之路是非常艰辛和漫长的，不但需要掌握很多高级的知识技能，还需要有过硬的基础知识。本章主要介绍 Java 程序员走向架构师的基础知识，还有开发环境的搭建。通过本章的学习，读者可以大致了解程序员的进阶之路，也可更加深刻地认识到程序员的发展方向。

1.1 程序员进阶

大学毕业后，初出茅庐的菜鸟经过千辛万苦，总算是找到了人生中的第一份工作。但是，随着工作的开展，菜鸟所面对的问题越来越多。有些人坚持了下来，有些人中途放弃，有些人则在职业生涯中选择了转型。作为一名程序员，不但需要编写大量的代码，还需要对自己的职业生涯做一个规划。结合前辈们所走过的道路，这个职业规划大致是图 1-1 所示的这个样子。

一般来说，从初级程序员到高级程序员需要经过 5 年的磨砺，这个时间段基本上是业界的共识了。而且，在众多招聘信息中也可以发现程序员的起点都是需要两年工作经验的。也许，有些天赋异禀的程序员可能经过 3 年的刻苦学习也能达到高级阶段，但是，他们的知识技能往往并不全面，可能只是在某些方面比较熟悉罢了。到了高级程序员的阶段，可供选择的方案就比较多，大概有图 1-2 所示的这 3 个走向。

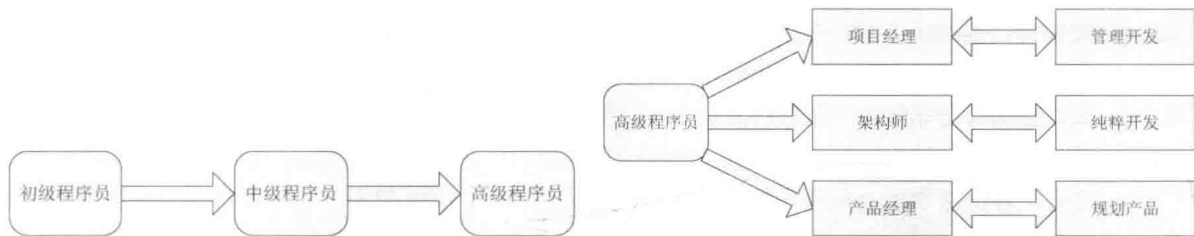


图 1-1 程序员职业生涯

图 1-2 程序员发展方向

如果高级程序员再向上进阶的话，会面临 3 个选择。第一种方案是成为项目经理，负责管理加上部分开发。因为高级程序员对公司的项目是非常了解的，对公司目前的开发过程也驾轻就熟。如

果本人有这方面的意愿，很容易胜任项目经理这个角色。而且，公司通常会从内部选择项目经理，空降项目经理的方式并不是常态，归其原因就是难以熟悉项目架构。

第二种方案是高级程序员可能更喜欢专著于技术，不喜欢出差和撰写大量的项目文档。在这种情况下，他可以成为一名架构师，专门负责维护公司的项目、产品方面的架构工作。如果公司有一定的规模，他可能会成为研发平台的负责人。当然，这种情况的前提是该程序员没有跳槽。

第三种方案是高级程序员可能经历了若干年的开发后，对写代码已经深恶痛绝，丝毫感受不到任何快乐了，但他对公司的项目和产品又非常熟悉，也有深厚的研发积累。在这种情况下，他可以彻底转型成为一名产品经理，纯粹负责公司产品的规划、设计、包装，甚至肩负一定的市场职责。当然，成为产品经理的前提是公司的项目已经产品化或者正在产品化之中。所谓的产品经理，通常就是向技术部提出一个原型设计：“看吧，这就是我想要的东西，至于怎么实现，你们看着办！”如果他懂代码还好说，但如果不懂代码，可能会让程序员陷入抓狂状态！

到了高级程序员的阶段，很多人就开始思考：究竟是去做项目经理？产品经理？还是继续写代码成为优秀的架构师呢？每个人的想法是不一样的，所作出的选择也是不一样的，这跟自己的能力和性格也有一定的关系。

- **项目经理**：在大型公司里，主要起协调资源的作用，再往上还有项目集经理。而在一些中小型公司里，项目经理不但要做好管理，还要兼备一部分代码的开发工作，但与此同时，也会有5年经验左右的项目组长，来管理不同的项目组。在软件行业中，经常有这样一个争论，项目经理到底应该不应该写代码？支持和反对的人都很多，但作者认为，这也是仁者见仁、智者见智的事情。首先，项目经理自身也是资源，是资源就有消耗，有些老板可能会认为：“我花这么多钱，请一个项目经理过来只为了写写文档，是不是太亏本了？”但到了数万人的大公司，该公司的项目通常特别多，就需要项目经理非常专注地管理项目，而不是分心去写代码。这种情况下，老板的思路就会转变，你写什么代码？好好地管理好公司的项目，不让它出乱子就可以了。
- **产品经理**：一般则是公司已经将项目过渡到了产品后，才能发挥更大的作用。如果公司一开始只有项目，则需要大量的时间来积累，最终实现产品化。在这个过程中，往往不是很需要专职的产品经理，可能项目中的每个人都会对项目献计献策，来使项目更加通用化。产品经理自身也是需要积累的，如果他成功地设计了一款App，并且在市场上取得了极大的成功，那么他的职业生涯可能会因此镀金，这个App将会成为他能力的体现。
- **架构师**：专注于公司的研发平台，管理框架方面的东西。例如，写核心代码，并且指导底下的开发人员合理地编码，维护代码库。在小公司里可能只有一两名架构师，但是在数万人的公司里，架构师会非常多。在这种情况下，架构师有可能会成为程序员级别称谓。例如，你在该公司待了8年，虽然你干的活一直是普通研发，并不负责实际上的架构，但是公司有正常的晋升渠道，你的级别就会从高级软件工程师上升到软件架构师。这种情况，在外包公司比较常见。
- **全栈工程师**：是最近才兴起的一个概念，但全栈工程师说到底还是程序员，类似于高级开发的角色，只不过是懂的东西比较多，前端和后端都可以做，技术比较全面。全栈工程师极大地拓展了自己的开发技能，成为了项目中的骨干成员，类似于技术专家的角色。一般而言，小公司比较喜欢这样的人，招募一个可以顶3个。但从学习的角度来说，全栈依然是不错的

目标。因为只有成了全栈工程师，才更能接近架构师。

每种开发语言，都有自己领域的架构师，如 C++ 架构师、PHP 架构师，当然也有 Java 架构师了。架构师需要对公司的整个研发平台了如指掌，清楚平台中细枝末节的东西。他极有可能是陪伴着这个公司成长起来的程序员；也极有可能是在别的公司工作多年后跳槽过来的。前者对公司的项目、产品非常熟悉，甚至还动手写过业务层。后者可能只是从大体上了解公司的研发平台，毫不深入，但这并不影响他的发挥，真正的架构师看到代码就有一种亲切感，可以很容易分析出隐藏在代码前后的业务过程。

Java 架构师，至少需要在 Java 领域有 5 年的开发经验。他需要掌握的内容很多，简单点可以分为前端、后端、数据库、服务器、中间件等。前端插件可以极大地提高开发效率，甚至在不需要美工的情况下做出时尚的界面，类似的插件有 AngularJS、Avalon、Bootstrap、ExtJS、jQuery EasyUI、jQuery UI 等，这些前端插件也可以称作前端组件或者前端框架，种种叫法也看人的习惯了，没必要吹毛求疵。除了这些前端插件外，还需要掌握 JavaScript、HTML 等技能。后端需要掌握的技能主要是 Java、JVM、Servlet、Struts、Spring、Hibernate、MyBatis 等，还有最近流行起来的 Spring MVC、Spring Boot 等。这些技能和框架只有综合起来使用，并且合理地搭配才能发挥出最好的效果。至于效果能够达到什么程度就需要看架构师的本事了。也许有的架构师可以把这个积木搭得很好，也许有的架构师在搭积木的过程中，这个积木就倒下了。数据库方面需要掌握的内容有 Oracle、MySQL、SQL Server，一般常用的数据库大概就是这 3 个。当然近年来，对于架构师需要掌握的数据库又有所增加，它们主要是代表了 NoSQL 的 MongoDB 等区别于传统关系型的数据库。但是，数据库相关的内容有不少，例如，需要熟练掌握 SQL 的各种语法，还需要掌握数据库性能的调优、备份和恢复。服务器并不是重点，但作为 Java 架构师，仍然需要有所了解。服务器包括物理服务器、云服务器，还有 Web 服务器，包括我们在开发中使用的 Tomcat。中间件在一些中小型项目中并不怎么常用，如 EJB 技术、消息中间件 ActiveMQ。当然，Web 服务器也可以算作中间件，如 Tomcat、Weblogic、WebShpere 和 JBoss 等。

只有熟练掌握这几个方面的技能后，才能算是一个初级架构师。如果想成为大神级别的架构师，还需要学习更多的知识。Java 架构师需要对这些技能非常熟悉，并且能像搭积木一样把他们整合在一起，构建出成熟的、完整的软件开发平台，以供底下的程序员在此基础上进行业务层的开发。但是，随着软件技术日新月异地发展，越来越多的框架进入眼帘，这对于我们来说既是好处又是坏处。好处是我们可以选择更好的、更合适的框架来提高项目的性能，降低开发难度，简化开发流程。坏处是可选择的框架太多，以至于让我们难以选择。所以，本书为大家精心挑选出了一名合格的架构师所必备的专业技能和开发思想，以供大家学习和参考，争取尽早地成为 Java 架构师。

1.2 选择开发工具

孔子曰：“工欲善其事，必先利其器。”这是一个千古不变的哲理，工匠想要使他的工作做好，一定要先让工具锋利，这样才能发挥出最大的效率。这个哲理告诉我们，不管做什么事情，都要选择合适的工具。那么在软件开发的道路上，选择一个合适的开发工具也是极其重要的事情了。Java 的开发工具有几种，这里不做太多的赘述，我们只需要对比它们的特点，即可从中选择出一款最适合自己的。