

软件架构师成长之路

# Java架构 完美设计

## 实战经典

颜廷吉 ○ 编著

本书附赠  
电子课件、  
微视频、源代码等  
丰富配套资源

一本帮你实现架构师梦想的图书  
这是成为架构师的必由之路！

软件架构师成长之路

# Java 架构之完美设计 ——实战经典

颜廷吉 编著



机械工业出版社

本书内容分为软件架构理论、软件架构高级技能、软件架构综合技能以及软件架构创新四大部分。第一部分是第1、2章,介绍了软件架构师练就的方法、软件架构设计模式、软件架构开发模型、软件架构模式、软件架构思维、软件架构编程思想以及软件架构设计原则等,为软件架构设计的基本理论;第二部分是第3~5章,介绍了软件架构师必备的工具体系、Java 机制技能要点专题、Java 后台技术栈技能要点专题以及日志架构设计、安全架构设计、权限架构设计、验证架构设计、异常架构设计、消息架构设计、阻塞架构设计、数据字典架构设计 Java 八大核心架构设计体系,为软件架构设计的高级技能体系;第三部分是第6~13章,以Spring 技术为脉络,介绍了 JavaWeb、SpringMVC、Spring Integration、Spring Boot、SOAP Webservice、REST Webservice 以及框架测试等架构设计相关的核心技术,为软件架构设计与实现相关的综合技能体系;第四部分是第14、15章,以开源框架以及自动化代码生成工具开发为中心,启发读者的创新思维,为软件架构的创新技能体系。

本书是“软件架构师成长之路”丛书的高级篇,适合软件工程师、架构师以及软件项目经理使用;还适合那些有志于成为软件架构师的其他软件从业人员自学使用;也可以作为高等院校相关专业师生参考教材;培训机构也可将本书作为软件架构等方面的培训教材。

## 图书在版编目(CIP)数据

Java 架构之完美设计:实战经典/颜廷吉编著. —北京:机械工业出版社, 2019. 8

(软件架构师成长之路)

ISBN 978-7-111-63512-3

I. ①J… II. ①颜… III. ①JAVA 语言-程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2019)第 184767 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:张淑谦 责任编辑:张淑谦

责任校对:张艳霞 责任印制:张博

三河市国英印务有限公司印刷

2019 年 9 月第 1 版·第 1 次印刷

184mm×260mm·26.25 印张·647 千字

0001-2500 册

标准书号:ISBN 978-7-111-63512-3

定价:109.00 元

电话服务

客服电话:010-88361066

010-88379833

010-68326294

封底无防伪标均为盗版

网络服务

机工官网:www.cmpbook.com

机工官博:weibo.com/cmp1952

金书网:www.golden-book.com

机工教育服务网:www.cmpedu.com

## 序

我与作者颜廷吉先生相识十余年，私交甚笃，也因此有幸亲历并参与了本丛书三部曲创作的全过程。如今历时数载而一朝梦圆，回首过往，受益良多，感慨万分。

颜廷吉赴日之初，正值金融危机肆虐，各行各业都陷入困境，尤以软件行业为甚，当时真可谓谈 IT 色变。然而在如此严酷的环境之下，颜廷吉居然可以从一家名不见经传的华人派遣公司社员一跃成为世界五百强企业 NTTDATA 的软件工程师。他跻身 IT 精英行列后并没有骄傲自满，而是更加勤奋好学，日积月累，最终完成了三部曲的创作。让人惊叹之余又羡慕不已。而在与颜廷吉深交之后，才明白这又是那么地理所当然。

不积跬步无以至千里，不积小流无以成江海。古圣先贤留下的千古名言，用来形容颜廷吉恰如其分。在东京的通勤电车上读东野圭吾的小说学习日语；将午饭时间从一小时缩短到三十分钟，为的是每天按时浏览各大 IT 技术网站的最新动态；每次开会之后都要把发言内容条分缕析，用以分析日本人的做事方法和思维逻辑；甚至在公司举办的年会上，在出外郊游的温泉旁，在烧烤聚会的篝火边，看到的都是颜廷吉沉思默想的背影。这种争分夺秒刻苦钻研的学习精神实为难能可贵。

颜廷吉说，做学问就像爬山，从山脚出发的人很多，能爬到山顶的却寥寥无几。有些人爬得很快，到达半山腰就盖了一所房子住在那里，对爬向山顶没有太大的兴趣。而能够登顶的人，除了要具备必要的技能外，更要有坚强的毅力，不被沿途景色所迷，不为声色犬马所扰。坚韧不拔，直至峰顶，终成正果。

坚持是有代价的。颜廷吉膝下三子，他却没有亲手为孩子换过一次尿布，女儿上二年级还没有和爸爸去过一次迪斯尼，回国看望父母从原来的一年两次到现在的两年一次，甚至结婚前对妻子许诺的蜜月之旅现在都没有成行。然而，从他家人的嘴里听不到一句怨言，有的只是理解和默默的支持。我被他们的奉献精神深深地感动。

我为这部内容详实的著作而赞叹，更为作者十余年如一日的拼搏精神所折服。望其再接再厉，秉志前行。

周伟鹏

2019年6月6日

# 前 言

软件架构师是每个程序员职业生涯中内功心法修炼的终极目标。要达到这个目标需要具备“十八般武艺，八十种技巧”，本书正是继《Java 代码与架构之完美优化——实战经典》《软件品质之完美管理——实战经典》之后，优秀软件架构师又一本必读书，也是“软件架构师成长之路”系列教程的第三部作品，亦是本系列的收官之作。本书总结了 JavaEE 软件架构师应该具备的架构设计相关技能体系，希望可以成为程序员朋友们架构师成长之路上的铺路石。

从形上看，架构是系统结构的骨架，支撑和链接各个部分；从神上看，架构是系统设计的灵魂，深刻体现了业务技术实现的本质。从纵向架构上看，架构涉及由客户端发送请求到服务器处理，再从服务器返回给客户端的各个主要步骤的具体处理细节；从横向架构上看，架构又关联到实现这种客户端-服务器端的架构模式。本书把与此纵横体系相关的技术进行了系统的总结与对比。另外，要成为一名优秀的软件架构师，还需要攻克以下三个难关：

- 1) 需要掌握各种技术的优缺点与特性，才能知道如何使用。
- 2) 站在架构设计者的角度，思考一款优秀的系统架构应该具备哪些要素。
- 3) 如何用“组合拳”来巧妙解决复杂问题。

本书尝试引导读者朋友来逐一攻破。

任何技术的学习都需要先调查研究，再模仿，最后再进行优化升级（也就是创新），架构设计亦是如此。本书首先介绍了架构设计理论与必备高级专题技能，然后介绍了各种常用架构模式以及对常用框架的模拟，最后介绍了开源框架设计以及自动化代码生成工具的相关技巧。其中重点介绍的部分配备了课后习题，也是对核心技能的进一步巩固与补充；也有点到为止的部分，目的是扩大大家的知识面，形成软件架构理论的必备知识。架构设计绝不是不着边际的假想，而应接地气，因此需要用一种具体的技术来实现其设计思想，所以本书以 Java 技术以及其最重要的开源框架 Spring 为核心，进行重点技术代码的演示。书中涉及架构实现代码部分，短的几十上百行，长的几百上千行，其中有些代码比较复杂，希望读者朋友能够静下心来慢慢体会。

“软件架构师成长之路”系列教程历时十年才陆续问世，之所以花费这么长的时间，一是因为涉及的知识与技巧非常多，二是体现了创作过程的“工匠精神”——作者除了倾尽自己的全力，本着务实严谨、精益求精的态度来创作，同时也时时思考：应该写哪些？不应该写哪些？系列教材内容广度与深度如何？应该以怎样的形式进行编排？如何贯穿前后知识点？如何体现综合技能？如何由点到线，由线到面，再由面到体，从而形成系统的知识与技巧体系？还有哪些部分可以优化？能否给读者带来实质性的帮助？有没有创新？如何启发读者朋友进行创新？有时候思考良久都迟迟不能下笔。

人们对软件架构的逐渐重视也是本书写作的一个契机，希望本系列教程能成为国内第一套软件架构师培训相关的原创系列图书，借此跟大家分享对架构的浅薄理解，同时亦介绍自己“刻意训练”的成长经历，权作抛砖引玉，希望读者朋友多来信交流。

一名优秀的软件架构师至少需要 5~10 年的一线实践。需练就一身本领，且能在项目中胜任架构师角色。如果想要成为一名优秀的架构师，就需要进一步精雕细琢自己的技能，且能培养架构师人才。虽然这条路充满艰辛，但是当一步一步攀登到峰顶时，那种收获的喜悦只有自己才可以体会得到……

## 本书与软件架构师

本书是作者多年 Java 软件架构实践与研究经验总结，包含架构设计的 8 项原则，21 种设计技巧，25 个温馨提示，39 个实战经典案例，架构测试以及开源软件设计、开发、发布等知识，同时还包括架构自动化代码工具的设计与开发技能。其内容详实、条理清晰、图文并茂、实战性强——一切都围绕提高读者软件架构实战技能。本书是培养具有高级软件架构设计与开发技能的优秀架构师所必备的武器之一。

365IT 学院规划的整个“软件架构师成长之路”系列教程分为初、中、高三部曲，如图 1 所示。第一部《Java 代码与架构之完美优化——实战经典》为品质管理“开发篇”，总结了代码品质与架构优化的相关技能；其姊妹篇《软件品质之完美管理——实战经典》为品质管理“管理篇”，总结了软件品质管理相关技能；第三部即本书，是这个培训系列教材的高级读本，总结了架构设计与实现的相关技能。三本书并不是孤立的，而是相辅相成的。例如，利用第三本书的架构设计技术进行系统设计之后，还需要借助第一本书的架构优化技巧进行优化，这样就形成一个由垂直整合的闭环生态链和一个横向扩展的开放生态链共同构成的架构技能生态体系。卓越的软件架构设计技能是程序员通往架构师神圣殿堂的必经之路，本书将是这条路上的一盏明灯，帮助读者早日实现软件架构师之梦。



图 1 软件架构师成长之路

## 本书与翻转课堂模式

本书采用翻转课堂模式 (The Flipped Classroom)。在内容安排上，首先抛出本章的关键问题，让读者进行思考；之后进行解释说明；章节最后根据内容深浅，适当加入练习题，以巩固核心内容的理解。这样在阅读技巧上进行了革新，可以让读者更好地吸收与理解本书内容。在

阅读过程中有任何疑问都可以和作者沟通，作者会给予及时的帮助与反馈。

## 本书配套教学视频与代码

本书配套教学视频与教程将同步出炉。配套视频可以在官网 ([www.365itedu.com](http://www.365itedu.com)) 免费下载，也可以在机械工业出版社计算机分社官方微信订阅号“IT 有得聊”上进行下载。在作者的引读下，希望读者朋友可以更好、更轻松学好本书所阐述的技能。

另外，本书配套代码也会放在 GIT 网 (<https://github.com/yantingji/java-architecture-book>)，以便随时更新。

## 本书特色

1) 授人以鱼，授之以渔：本书给出了架构设计技能要点、开源框架设计思想、自动化代码工具设计技巧等最基本的设计理念，希望读者能掌握并在未来架构设计中灵活运用。

2) 图解技术，形象生动：避免了乏味难懂的文字描述，使繁冗复杂的事物一目了然。

3) 综合能力，综合技巧：本书一部分内容是对 Java EE 领域技能的总结，因此给出的案例大部分具有一定的难度，希望以工程案例的形式，来引领读者领悟如何提高综合能力。

4) 设实分离，开源创新：本书主要论述的是设计思想以及架构相关技术，而关于架构设计的实现部分在本书的辅助代码资料中，也就是设计与实现进行分离。辅助代码将以开源的形式公开，不但会根据需求进行相应的升级，读者朋友亦可以进行优化。这种写作方式的大胆尝试，也是一种创新。本书不单独讲理论，而是以案例驱动的形式进行实战解析，这不仅是经验与理论的总结，更重要的是希望用最佳项目案例来说明技术应用。开源精神是软件领域的一大特色，作者希望能尽到一点绵薄之力。另外，本书介绍的各种案例也给读者留下了更大的创新与扩展空间。

## 本书所面向的读者

本书以提高读者的综合技能、启发读者的创新能力为目标，吸收国外架构设计思想并结合国内实际情况，从架构设计者的角度系统地介绍了 Java 领域架构设计思想与开发技术。如图 2 所示。

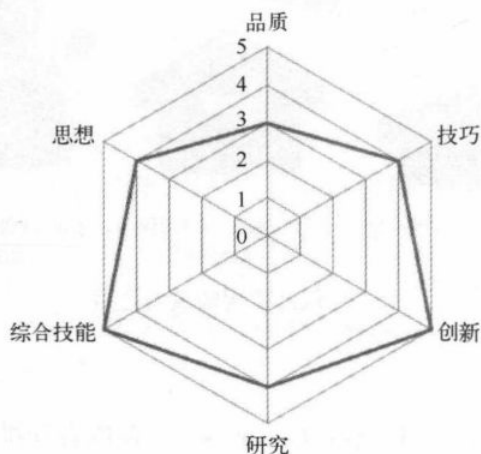


图 2 能力提高点设置

本书以实用为核心，所面向的读者主要是志在成为优秀软件架构师的程序员，也适用于以下读者朋友：

- 1) 走在架构师之路上的工程师。
- 2) 软件项目经理。
- 3) 对计算机技术感兴趣的在校学生。

## 如何最佳阅读本书

本书共 15 章，分为架构基本理论、架构高级技能、架构综合技能以及架构创新思维四部分，如图 3 所示。

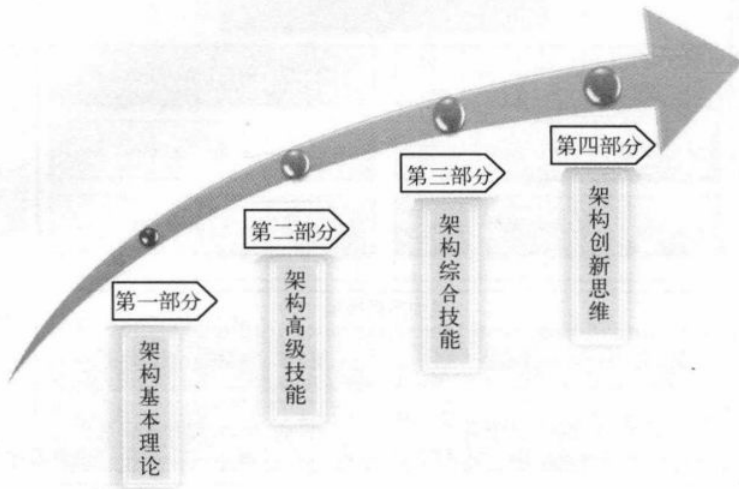


图 3 本书整体划分

第一部分包括第 1、2 章，介绍了架构模式、架构思维、架构编程思想、架构设计原则等，为软件架构设计的基本理论；第二部分包括第 3~5 章，介绍了架构师必备的工具体系、Java 机制技能要点专题、Java 后台技术栈技能要点专题以及 Java 八大核心架构设计体系，为软件架构设计的高级技能体系；第三部分包括第 6~13 章，以 Spring 技术为脉络，介绍了 JavaWeb、SpringMVC、SpringIntegration、Spring Boot、SOAPWebService、RESTWebService 以及框架测试等架构设计相关的核心技术，为软件架构设计与实现相关的综合技能体系；第四部分包括第 14、15 章，以开源框架以及自动化代码生成工具开发为中心，启发读者的创新思维并起到抛砖引玉的作用，为软件架构的创新技能体系。在整体内容的编排上，随着章节的展开，难度也逐步增加，如图 4 所示。

阅读时，希望读者朋友能够自己先思考并解决书中提出的问题。因为无法展示从零到最终完成的代码编写过程，读者朋友可以先运行这些代码，在 Debug 状态下进行跟踪执行流程，理解后再从零开始写。也可以利用逆向思维进行学习，也就是一部分一部分地删除案例中的辅助代码（先把宽度变窄，再把深度变浅），直到精简到几个一眼就可以看明白的类为止。在做好驱动测试的情况下，一步一步把代码输入一遍，直到自己完全理解。

成为软件架构师必须掌握的知识与技能非常多，但是为了在有限的篇幅内展示最重要的核心内容，本书把大部分代码全部以辅助教材的形式进行编排，希望读者朋友能够利用好这些代码示例。本书只是重点展示了架构设计思想以及最核心的部分代码。另外，架构相关技

能与架构设计的实现案例一部分安排在正文，另外一部分放在课后习题里，课后习题也是本书的重要组成部分，切莫忽视！

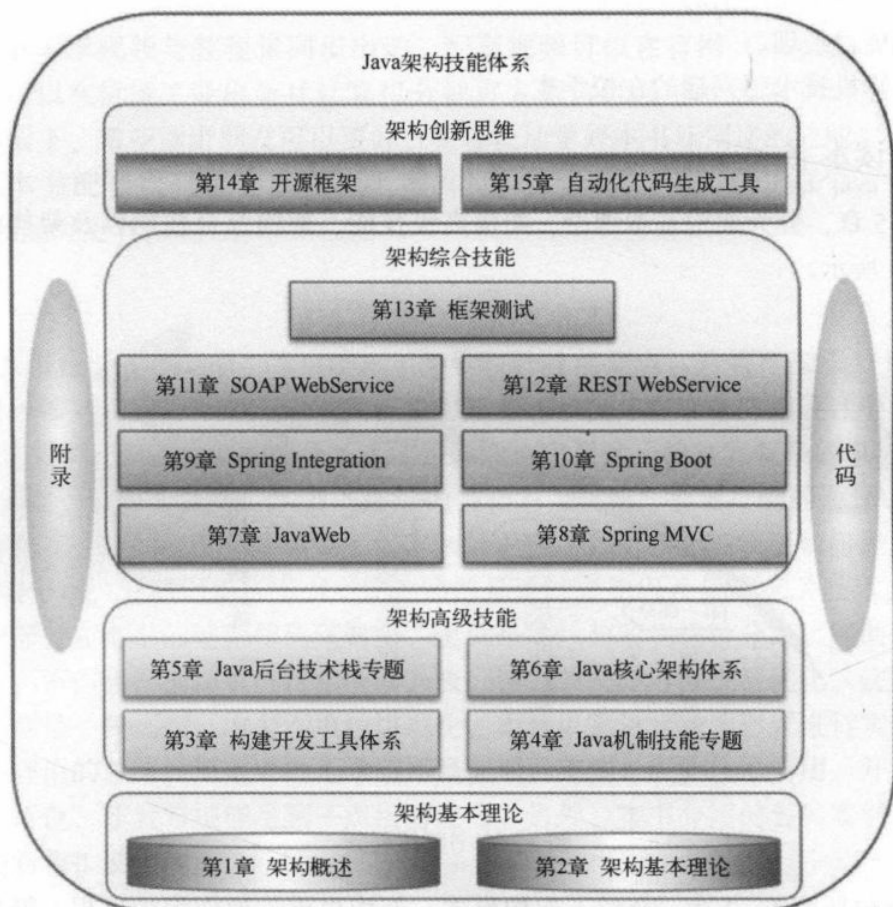


图4 Java 架构技能体系

书中对一些正文外的重要知识点用温馨提示（NOTE）的形式作了补充，内容安排亦分布在各个章节的最佳位置，以便读者学习，如图5所示。

**NOTE:** **Eclipse 无法启动**

如果修改 eclipse.ini 文件的 -Xmx 参数后，发现 Eclipse 无法启动。其原因之一就是 -Xmx 参数设置太大，适当调整数值后再次启动即可。

图5 温馨提示示例

阅读本书时一定要多问自己几个为什么。对于书中的技巧与案例，要问：为什么这么做？还有更好的解决方案吗？在实际工作中应该如何应用？是否还可以进一步扩展与创新？当对自己进行如此一番洗礼后，应该对架构设计与实现有“会当凌绝顶，一览众山小”之心境！

## 致谢

首先，感谢 NCIT 集团技术大咖杨小虎、庄茂为在百忙之中抽出宝贵时间参与技术内容的讨论与审核。

其次，感谢好友周伟鹏、尹勋成。十年来，他们和作者一起探讨了本系列教程的规划，还审核了从初级到高级所有版本的初稿。同样亦真心感谢若为投资执行董事吴冰，同事陈钢、杨晗和关宝林，以及山东老家的闫晓峰老师与兰霞老师细致耐心的审核。

再次，在作者调查研究整理的过程中，经常会把优秀的案例与文句进行记录，所以本书中的部分案例内容也参考了 51CTO、博客园、CSDN、ITEYE 等平台上的技术大咖博主的内容，这些作者的智慧同样为本书的写作带来了帮助，在此一并感谢。

最后，感谢爱人兰宁的大力支持。虽第三次身怀六甲，还为本书的完成给予了最大限度的支持。本系列教程的规划与完成历时近十年，没有家人的支持是不可能完成的。为了让本书早日和读者见面，作者几乎牺牲了所有的休息时间闭关写书，目的是希望能够早日完结，不辜负读者朋友的信赖。

## 勘误

本书如有勘误，会在 <https://github.com/yantingji/java-architecture-book> 上进行发布。由于作者能力有限，书中纰漏在所难免，希望能够与大家一起探讨，一起成长！如果大家在阅读此书时发现有任何错误或疑问，请来信（[yantingji@126.com](mailto:yantingji@126.com)）或者在 GIT 网站新建 issue 来讨论与切磋。另外，如果读者朋友感兴趣，请对本书提供的代码进行再次优化升级，从而激起设计灵感上的火花。

颜廷吉

2019 年 3 月 4 日于东京

# 目 录

序	
前言	
第 1 章 架构概述	1
1.1 架构来源	1
1.1.1 软件架构	1
1.1.2 架构单位	2
1.1.3 常见架构错误概念	2
1.2 架构的价值	4
1.3 架构发展历史	5
1.4 架构师	6
1.4.1 架构师分类	6
1.4.2 软件架构师职责	7
1.4.3 软件架构师素质要求	7
1.4.4 架构师练就方法	7
小结	9
习题	9
第 2 章 架构基本理论	10
2.1 架构品质	10
2.2 架构过程	11
2.2.1 架构流程	11
2.2.2 架构成果	11
2.2.3 架构团队	13
2.3 架构开发模型	14
2.3.1 开发模型概述	14
2.3.2 边做边改模型	14
2.3.3 迭代开发模型	14
2.3.4 Scrum 开发模型	15
2.4 架构设计常用 UML	22
2.4.1 概述	22
2.4.2 类图	24
2.4.3 序列图	27
2.4.4 构件图	28
2.4.5 部署图	29
2.5 架构常用设计模式	31
2.5.1 模式概述	31
2.5.2 设计模式概述	31
2.5.3 设计模式与框架	33
2.5.4 单例模式	34
2.5.5 工厂方法模式	35
2.5.6 建造者模式	36
2.5.7 适配器模式	37
2.5.8 外观模式	39
2.5.9 代理模式	40
2.5.10 模板方法模式	43
2.5.11 策略模式	44
2.5.12 责任链模式	45
2.6 架构模式	46
2.6.1 Java EE 规格与框架	46
2.6.2 模式与架构	48
2.6.3 MVC 架构模式	49
2.6.4 分层架构模式	56
2.6.5 管道-过滤器架构模式	57
2.7 架构思维	57
2.7.1 抽象思维	57
2.7.2 分层思维	58
2.7.3 分治思维	58
2.7.4 演化思维	59
2.7.5 架构思维之间的关系	59
2.8 架构编程思想	60
2.8.1 面向过程编程	60
2.8.2 面向对象编程	61
2.8.3 面向服务架构	67
2.8.4 架构思想之间的关系	71
2.9 架构设计原则	72
2.9.1 简单原则	73
2.9.2 合适原则	73

2.9.3 演化原则 .....	73	4.4.2 注解分类 .....	116
2.10 架构优化利器 .....	73	4.4.3 自定义注解 .....	117
2.10.1 重构带来的利益 .....	73	4.4.4 注解解析 .....	118
2.10.2 重构技巧 .....	74	4.4.5 适用场景 .....	119
2.10.3 重构工具 .....	75	4.5 Servlet .....	120
小结 .....	77	4.5.1 Servlet 机制 .....	120
习题 .....	77	4.5.2 Servlet 与 CGI .....	126
<b>第3章 构建开发工具体系</b> .....	<b>78</b>	4.5.3 Servlet 与 JSP .....	126
3.1 DevOps .....	78	4.5.4 Servlet 与普通 Java 类 .....	127
3.1.1 代码管理工具 .....	81	4.5.5 自定义标签 .....	127
3.1.2 项目构建工具 .....	83	4.6 监听器、过滤器与拦截器 .....	128
3.1.3 持续集成工具 .....	84	4.6.1 技术要点 .....	128
3.2 智能开发平台 .....	84	4.6.2 三者对比 .....	129
3.2.1 搭建智能开发平台 .....	84	4.7 Cookie 与 Session .....	130
3.2.2 Eclipse 对比 IntelliJ .....	85	4.7.1 Cookie 机制 .....	130
3.2.3 Eclipse 启动方式 .....	87	4.7.2 Session 机制 .....	133
3.2.4 Eclipse 程序启动参数 .....	87	4.8 Socket .....	137
3.2.5 Eclipse 插件安装方式 .....	88	4.8.1 Socket 通信机制 .....	137
3.3 工具快捷键 .....	96	4.8.2 三次握手建立连接 .....	140
3.3.1 快捷键的好处 .....	96	4.8.3 四次挥手释放连接 .....	140
3.3.2 快速键盘操作 .....	96	小结 .....	141
3.3.3 优化快捷键 .....	97	练习题 .....	141
3.4 工具优化与保养 .....	100	<b>第5章 Java 后台技术栈专题</b> .....	<b>142</b>
小结 .....	101	5.1 XML .....	142
习题 .....	101	5.1.1 概述 .....	142
<b>第4章 Java 机制技能专题</b> .....	<b>102</b>	5.1.2 DTD .....	144
4.1 通用处理方式 .....	102	5.1.3 XSD .....	144
4.1.1 串行与并行 .....	102	5.1.4 XML 解析 .....	148
4.1.2 同步与异步机制 .....	102	5.2 HTTP .....	149
4.1.3 在线处理与批处理 .....	103	5.2.1 概述 .....	149
4.1.4 系统间通信方式 .....	104	5.2.2 URL .....	150
4.2 核心机制 .....	105	5.2.3 HTTP 消息结构 .....	151
4.2.1 虚拟机机制 .....	105	5.3 JDBC .....	152
4.2.2 内存管理机制 .....	109	5.3.1 概述 .....	152
4.3 反射与自省 .....	112	5.3.2 JDBC 开发方法 .....	153
4.3.1 反射机制 .....	112	5.3.3 事务机制 .....	154
4.3.2 自省机制 .....	114	5.4 AOP .....	155
4.4 注解 .....	116	5.4.1 概述 .....	155
4.4.1 注解机制 .....	116	5.4.2 SpringAOP .....	157

5.5	字符集与乱码	160	6.5	异常架构设计	225
5.5.1	字符集	160	6.5.1	概述	225
5.5.2	乱码	169	6.5.2	异常分类	225
5.6	日期处理	181	6.5.3	异常处理方式	227
5.6.1	概述	181	6.5.4	异常处理目的	229
5.6.2	日期使用技巧	182	6.5.5	异常处理流程	233
5.7	翻页控件	184	6.6	消息架构设计	235
5.7.1	问题	184	6.6.1	概述	235
5.7.2	解决方案	184	6.6.2	设计范例	236
5.8	属性文件	190	6.7	阻塞架构设计	238
5.8.1	问题	190	6.7.1	概述	238
5.8.2	解决方案	191	6.7.2	设计技巧	238
5.9	重复提交	192	6.8	数据字典架构设计	239
5.9.1	问题	192	6.8.1	概述	239
5.9.2	解决方案	194	6.8.2	设计技巧	239
5.10	排他	199	6.9	体系架构之间的关系	240
5.10.1	问题	199	小结		241
5.10.2	解决方案	202	练习题		241
5.10.3	防止数据死锁	208	<b>第7章 JavaWeb</b>		242
小结		209	7.1	Web 服务器	242
练习题		209	7.1.1	服务器种类	242
<b>第6章 Java 核心架构体系</b>		211	7.1.2	Tomcat	243
6.1	日志架构设计	211	7.1.3	服务器集群	247
6.1.1	概述	211	7.2	框架对比	247
6.1.2	日志设计技巧	211	7.2.1	Struts1 对比 Struts2	247
6.1.3	日志框架	212	7.2.2	SpringMVC 对比 Struts2	248
6.2	安全架构设计	215	7.2.3	Hibernate 对比 MyBatis	249
6.2.1	安全设计原则	215	7.3	框架集成	249
6.2.2	安全注册	215	7.3.1	SSH	250
6.2.3	认证	216	7.3.2	SSI	251
6.2.4	SSL 与 HTTPS	219	7.3.3	SSM	251
6.2.5	CSRF	220	7.4	框架模拟	253
6.2.6	XSS	221	7.4.1	Struts2 框架模拟	253
6.3	权限架构设计	223	7.4.2	SpringMVC 框架模拟	254
6.3.1	概述	223	7.4.3	MyBatis 框架模拟	254
6.3.2	设计技巧	223	小结		259
6.4	验证架构设计	224	练习题		259
6.4.1	概述	224	<b>第8章 SpringMVC</b>		260
6.4.2	设计技巧	224	8.1	Spring 概述	260

8.1.1	Spring 与 Java EE	260	10.3.1	概述	298
8.1.2	Spring 技术栈	261	10.3.2	Spring Boot 与 Spring Cloud	299
8.1.3	Spring 5 新特性	262	10.3.3	微服务设计原则	299
8.2	SpringMVC 架构	264	10.3.4	微服务注册与发现	301
8.2.1	SpringMVC 概述	264	小结		305
8.2.2	SpringMVC 架构图	264	练习题		305
8.2.3	分派器	265	<b>第 11 章 SOAP Webservice</b>		306
8.2.4	处理器映射器	267	11.1	Webservice	306
8.2.5	控制器	268	11.1.1	技术要点	306
8.2.6	处理器适配器	270	11.1.2	富客户端	307
8.2.7	数据绑定器	270	11.1.3	RPC 与 Webservice	308
8.2.8	类型转换	272	11.2	SOAP 基本原理	309
8.2.9	数据验证	274	11.2.1	SOAP 概述	309
8.2.10	视图解析器	275	11.2.2	WSDL	310
8.2.11	视图	276	11.3	JAX-WS	311
8.2.12	异常处理	277	11.3.1	JAX-WS 概述	311
8.3	SpringBatch	278	11.3.2	JAX-WS-Server	312
8.3.1	SpringBatch 概述	278	11.3.3	JAX-WS-Client	314
8.3.2	SpringBatch 架构	279	11.3.4	SoapUI	315
小结		280	11.4	Spring-WS	316
练习题		280	11.4.1	Spring-WS 概述	316
<b>第 9 章 Spring Integration</b>		281	11.4.2	Spring-WS 架构	318
9.1	基本原理	281	11.4.3	WebserviceTemplate	319
9.2	核心组件	282	11.5	SOAP 框架	319
9.2.1	Message	282	小结		320
9.2.2	Channel	283	练习题		320
9.2.3	Endpoint	284	<b>第 12 章 REST Webservice</b>		321
9.3	可视化设计	287	12.1	REST 基本原理	321
小结		288	12.1.1	概述	321
练习题		288	12.1.2	WADL	321
<b>第 10 章 Spring Boot</b>		289	12.1.3	REST 架构	322
10.1	Spring Boot 概述	289	12.1.4	JAX-RS	323
10.1.1	特性	289	12.1.5	SOAP 与 REST	323
10.1.2	核心功能	290	12.1.6	JavaWeb 与 REST	324
10.1.3	Starter	291	12.2	REST-Server	324
10.2	Spring Boot 工程	294	12.2.1	Web 资源	324
10.2.1	SpringInitializr	294	12.2.2	URI	325
10.2.2	Gradle	295	12.2.3	HTTP 方法	326
10.3	Spring Cloud	298			

12.2.4 适当数据格式 .....	328	14.3.2 新开发 .....	359
12.2.5 适当 HTTP 状态码 .....	329	14.4 开源框架发布 .....	359
12.2.6 无状态通信 .....	330	14.4.1 版本号 .....	359
12.2.7 关联资源分层链接 .....	330	14.4.2 版本区别 .....	360
12.2.8 权限认证 .....	332	14.4.3 发布流程 .....	361
12.2.9 版本管理 .....	333	小结 .....	367
12.2.10 性能 .....	334	练习题 .....	367
12.3 REST-Client .....	335	<b>第 15 章 自动化代码生成工具</b> .....	368
12.3.1 RestTemplate .....	335	15.1 自动化工具概述 .....	368
12.3.2 测试工具 .....	337	15.1.1 分类 .....	368
12.4 SpringMVC 与 REST .....	339	15.1.2 重要性 .....	368
12.4.1 核心功能 .....	339	15.1.3 设计技巧 .....	369
12.4.2 架构图 .....	340	15.2 框架自动代码生成工具 .....	370
小结 .....	341	15.2.1 SSI 框架自动代码生成 工具 .....	370
练习题 .....	341	15.2.2 SSM 框架自动代码生成 工具 .....	374
<b>第 13 章 框架测试</b> .....	342	15.3 其他自动化工具 .....	375
13.1 自动化测试 .....	342	15.3.1 智能命名工具 .....	375
13.1.1 自动化测试准入标准 .....	342	15.3.2 SQL 自动生成工具 .....	379
13.1.2 自动化测试目标 .....	343	小结 .....	379
13.1.3 自动化测试框架 .....	343	练习题 .....	380
13.1.4 自动化测试工具 .....	344	附录 .....	381
13.1.5 持续集成与部署 .....	345	附录 A 原则一览 .....	381
13.2 性能测试 .....	347	附录 B 技巧一览 .....	381
13.2.1 性能测试概述 .....	347	附录 C 案例一览 .....	382
13.2.2 性能测试工具 .....	347	附录 D 温馨提示一览 .....	383
13.3 性能优化 .....	352	附录 E 常用快捷键 .....	384
13.3.1 性能优化原则 .....	352	E.1 Eclipse 常用快捷键 .....	384
13.3.2 应用层性能优化 .....	352	E.2 IntelliJ 常用快捷键 .....	385
13.3.3 数据库层性能优化 .....	352	附录 F 高级调试技巧 .....	387
13.3.4 框架层性能优化 .....	354	F.1 Eclipse 调试技巧 .....	387
13.3.5 JVM 层性能优化 .....	354	F.2 源代码导入技巧 .....	393
小结 .....	356	F.3 找不到文件分析技巧 .....	394
练习题 .....	356	F.4 问题定位技巧 .....	398
<b>第 14 章 开源框架</b> .....	357	附录 G 参考答案 .....	399
14.1 开源框架选择 .....	357	<b>参考文献</b> .....	403
14.2 开源框架使用 .....	358		
14.3 开源框架开发 .....	358		
14.3.1 再开发 .....	358		

# 第1章 架构概述

在阅读本章内容之前，首先思考以下问题：

1. 什么是软件架构？
2. 架构设计的目标是什么？
3. 什么是架构的单位？
4. 常见架构错误认识有哪些？
5. 架构能带来哪些利益？
6. 应用架构的发展大致经过了哪些过程？
7. 架构师分为哪些类别？
8. 架构师的素质要求有哪些？



## 1.1 架构来源

架构一词最初来源于建筑行业，指的是人们对一个结构内的元素及元素间关系的一种主观映射产物。所涉及的技术有地基、整体结构、外观、供水系统、供电系统、煤气系统、网络系统、安全系统等。设计师要把这些因素结合起来传达给施工者，用以指导建设，同时也给客户描绘出一个整体愿景图。

如今架构已经被广泛应用到企业管理和 IT 等领域，并演变成了各个领域的核心技术之一。

### 1.1.1 软件架构

软件架构是软件系统的顶层结构，是对重复性业务的抽象和未来业务拓展的前瞻，关注系统的可用性、信赖性、扩展性、安全性以及稳定性等非业务方面。

架构的设计目标是解决软件的复杂性，主要体现在以下两方面：

#### (1) 决定应用程序的结构

大中型企业的应用程序开发都具有一定的规模，因此会把系统中的一部分功能划分成很多组件。所以应用程序的结构应该从以下方面去考量：

- 1) 如何划分组件，要决定组件的划分原则与标准，即组件粒度大小。
- 2) 如何实现架构组件之间特定的协作与交互。
- 3) 如何使架构组件与业务组成一个系统。
- 4) 如何形成应用程序自己的架构风格。

#### (2) 决定应用程序的处理方式

处理方式指的是实现应用程序所提供的各种服务的通用机制。服务是指系统“业务

上能干什么”，即“**What**”，而处理方式指的是“技术上如何实现”，即“**How**”。例如：在处理由多个页面组成的数据登录业务时，经常会发生重复提交的问题，为解决此问题而选择的**Token**方案，就是架构的处理方式。

架构涉及一系列对设计和编码起约束作用的策略性的决定、规则和模式。架构的决定是最基本的决定，它的改变将产生很大影响，因此要做好充分的架构验证，如图1-1所示。

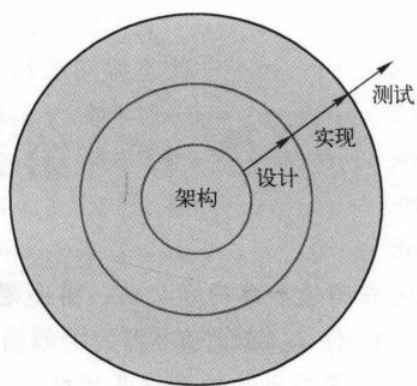


图 1-1 架构决定性作用

### 1.1.2 架构单位

《软件品质之完美管理——实战经典》一书中介绍了软件品质管理的基本单位是模块，而软件架构的单位是组件（Componet），即实现了架构的必要功能，可以用于分布、集成、交付、替换的模块单元，如图1-2所示。模块单元，不一定是一个单一的代码单元或单一的二进制文件，在Java中是由一个或多个类共同来完成其相应的服务。

模块化是指解决一个复杂问题时自顶向下逐层把软件系统划分成若干模块的过程。软件模块化技术是软件架构技术中最早讨论的技术之一，每个模块完成一个特定的功能，所有的模块按某种方法组装起来，就构成了整个系统所要求的功能。架构设计中所关注的“隐藏信息”“提高开放性”“高内聚”“低耦合”等亦是当今架构技术的重要原则，可以说模块化的结构设计技术是软件架构技术的根源。

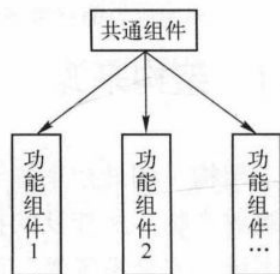


图 1-2 架构组成单位

模块与组件是实际工作中经常用到的术语，根据上下文不同，有不同的含义。一般把系统从逻辑角度拆分后得到的单元称为“模块”，而从物理角度拆分后得到的单元称为“组件”。划分模块的主要目的是职责分离，划分组件的主要目的是代码复用。

### 1.1.3 常见架构错误概念

#### 1. 架构=框架

框架（Framework）是整个或部分系统的可重用设计，表现为一系列组件或组件实例间的交互方法。框架实际上就是给应用开发者定制的“骨架”，包含一些共通处理和算法，类似生活中盖楼搭建起的“框架”，如Struts、Spring等框架。架构是一种设计指导思想，确定了系统整体结构、层次划分、不同组件之间的协作以及实现方式。如C/S架构、B/S架构等。所以，软件架构不等于框架，如图1-3所示。

框架比架构更具体，更偏重于技术细节，是架构思想实现的具体表现。确定框架后，软件体系结构也随之确定。而对于同一软件架构（如Web开发中的MVC），可以通过多种框架来实现。两者之间的区别见表1-1。