

• 开源中国创始人红薯先生 • 特赞CTO黄勇先生 • 恒生电子CTO范径武先生  
倾情作序并推荐!



# 企业级 Java EE 架构设计精深实践

罗果 著

**好的软件架构是“品”出来的，好的软件架构一定是简单的!**

全面涵盖缓存、文件处理、模板语言、服务层、数据库架构、数据库扩展、元数据、Web架构和前端界面等Java EE常用技术，涉及企业级应用架构的方方面面

提供大量的原创实践及原理性讲述，每章都提供实践性很强的开发案例，可以让读者对企业级Java EE架构中的一些技术及其解决方案有较为深入的理解

每章讲解一个Java EE领域的具体问题，采用问题背景→需求分析→解决思路→架构设计→实践示例→章节总结的顺序组织内容，注重知识的实用性和可操作性

清华大学出版社



# 企业级 Java EE 架构设计精深实践

罗果 著

清华大学出版社

北京

## 内 容 简 介

本书全面、深入介绍了企业级 Java EE 设计的相关内容，内容涵盖了 Java EE 架构设计的常见问题。本书每一章讲解一个 Java EE 领域的具体问题，采用问题背景、需求分析、解决思路、架构设计、实践示例和章节总结的顺序组织内容，旨在通过分析相关领域中的常用框架及存在问题，给出相应的解决方案，提高读者分析和解决问题的能力，并增强其架构设计的能力。

本书共 13 章。内容主要包括虚拟文件系统实践、缓存实践、文件处理框架实践、模板语言实践、数据库访问层实践、数据库扩展实践、服务层实践、流程引擎实践、元数据实践、展现层开发实践、Web 扩展实践、Tiny 统一界面框架实践和 RESTful 实践。附录中给出了相关学习资源和配置运行指南。

本书语言简洁，思路清晰，示例丰富、完整，适合具有一定 Java 基础的读者阅读，尤其适合从事企业级 Java EE 软件架构和设计的人员阅读。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

### 图书在版编目 (CIP) 数据

企业级 Java EE 架构设计精深实践 / 罗果著. —北京：清华大学出版社，2016  
ISBN 978-7-302-43470-2

I. ①企… II. ①罗… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2016) 第 078276 号

责任编辑：冯志强

封面设计：欧振旭

责任校对：徐俊伟

责任印制：刘海龙

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>, <http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社总机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈：010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印装者：清华大学印刷厂

经 销：全国新华书店

开 本：185mm×260mm 印 张：33 字 数：758 千字

版 次：2016 年 6 月第 1 版 印 次：2016 年 6 月第 1 次印刷

印 数：1~3000

定 价：99.80 元

产品编号：067076-01

# 序一

我不明白为什么相比于其他的编程语言，Java 的世界充满了各种框架和架构。可能是因为 Java 太过于灵活，也可能是因为官方对 Java EE 规范所提供的 API 都是非常底层的东西，很少牵扯到具体的业务需求。虽然这样可以在一定程度上保证规范本身的兼容性和适应性，但也因此导致了用 Java 开发一些具体业务应用时显得过于烦琐，不够轻便和快捷。于是出现了 Struts 等开源框架，再就是后来 Javaer 们耳熟能详的 SSH 三大框架，直至今天琳琅满目的各种开发框架。所有的这些框架，其唯一的目的就是简化业务逻辑的开发，其手法无不是利用各种各样的设计模式对 API 的各种层次进行封装。

我曾经发文公开反对初学者在对 Java 知之甚少的情況下学习各种框架。主要原因有两点：一是知其然而不知其所以然；二是更换框架后学习成本很高。因为先入为主的思想作怪，当本书作者（我们姑且就叫他的网名“悠然”吧）第一次将他的 Tiny 框架提交到“开源中国”的时候，我并没有过多的关注。哦，仅仅是又一个新的 Java EE 框架而已。

Tiny 再次进入我的视线是在“开源中国”社区围绕该项目越来越多的关注和讨论，而且不断出现在我们的各种开源项目榜单中。另外，Tiny 开源框架不同于其他开源框架的突出特点是代码提交非常活跃。可以毫不夸张地说是“开源中国”GIT 库中 Commits 提交最多的开源者。当然还有另外一个非常重要的原因是悠然写了一篇非常专业的“喷文”，该文章非常专业地指出我本人的一个开源项目 J2Cache 中存在的各种设计的不足，专业到让我无言以对，甚至颇有“挑衅”的意味。如此专业的设计思路及对代码近乎完美的追求，让我相信 Tiny 必然也会是在这种要求下的产物。此外，其团队利用该框架快速开发的几个应用类项目，也证明了其在开发效率上的提升是非常显著的。

这本书主要是通过对比各种流行的框架和技术来展示 Tiny 框架强悍又便利的优点。从前端模板展现、数据库访问、缓存，再到文件系统、服务分层、流程引擎、元数据和 RESTful，同时还延伸到了系统扩展，可以说是面面俱到。基本上你在开发 Java EE 应用时涉及的绝大多数技术层面的内容，本书都会详尽讲述。

前面我说过反对初学者一开始从框架入手进行学习。这本书在介绍 Tiny 框架的同时，对与其对应的一些 Java EE 开源框架或技术也做了简单的讲解，同时包括对同类产品的分析。以悠然如此善于深入分析各类产品的技术来看，可知对这些问题领域的分析及其解决方案便是本书的精华所在，不得不推荐！

最后我想提一下：没有最好的框架，只有最适合自己的框架。当你面对数百个 Java 框架一脸茫然的时候，考虑一下这么几个要素：是否打心眼里喜欢？是否足够简单和轻便，而不是充斥一堆你根本用不上的特性？项目本身在社区上是否足够活跃？作者本人长得帅不帅（开个玩笑）……如果上面所有的答案都是否定的，那就自己造一个吧！

开源中国创始人 红薯

## 序二

十年前，我上大学的时候，就听说 Java 这门计算机语言已经很流行了。Java 分为 J2SE、J2EE 和 J2ME 这三大平台，分别对应桌面、Web 和移动这三大领域。当时我为了找到一份高薪工作，放弃了自己所擅长的 C++，也学着别人开始学习 Java，尤其是学习当时市场需求量最大的 J2EE 技术。我学了 JSP、Servlet 和 EJB 这几种 J2EE 核心技术，也学了经典的 MVC 框架 Struts 和最流行的 ORM 框架 Hibernate。想当年，我可以熟练地使用 JSP+Struts+Hibernate 开发一个小型 Java Web 应用程序，轻松地应对毕业设计，自己也可以接点私活赚点外快。但对于大型企业级应用而言，首选技术还是 EJB。所谓企业级就是一个既安全又复杂的技术，因为它需要考虑到多方面的问题，所以会带来一系列的复杂问题。但凡用过 EJB 的人，都会感受到那是一场痛苦的经历。而 Spring 的到来，改变了这一切，给企业级 J2EE 带来了“春天”，Spring+Struts+Hibernate 三个框架组合也被广泛地应用于 Web 开发，并且大家称其为 SSH 组合。

十年后，Java 还能位居编程语言排行榜首，我真的庆幸自己当年的选择是正确的。以前的 J2EE，现已改名为 Java EE，也许这样的命名会更加有意义。而以前的 SSH，却显得有些重量级了。也许是因为它们自身的特性越来越多，或者是因为轻量级的框架越来越多，例如我开源的 Smart 框架，就是一款轻量级的 Java Web 框架，为了讲明白该框架的开发过程，我也写过一本叫做《架构探险》的书（一不小心就为自己做了一个广告，还是言归正传吧）。很多大型企业级应用会放弃掉一些技术，例如 Struts、Hibernate 和 JSP 等，但 Spring 目前仍然还是主流，尤其是 Spring MVC，足以取代 Struts。此外，轻量级 JDBC 框架 MyBatis（以前叫做 iBatis）也可以取代 Hibernate，所以现在企业级 Java EE 架构的首选变成了 Spring+Spring MVC+MyBatis，简称 SSM。

不管是 SSH 还是 SSM，都只是开发框架而已，并非技术解决方案。大家知道，企业级应用是一个相当复杂的应用系统，需要的不仅仅是一个开发框架，更是一系列的技术解决方案。这些解决方案包括虚拟文件系统、缓存、文件处理框架、模板语言、数据库访问层、数据库水平扩展、服务层、流程引擎、元数据、展现层、Web 层、统一界面框架、RESTful 和模块化等，这些才是企业级应用的核心需求。Tiny 框架将这些需求迎刃而解，代码完全开源，并且在国内有良好的技术社区网站（[tinygroup.org](http://tinygroup.org)）。更重要的是，Tiny 框架的作者是一位乐于分享的技术专家，在开源中国网站（[oschina.net](http://oschina.net)）上分享了大量的技术文章。现在 Tiny 团队将他们几年的研究成果汇集成书，并毫无保留地分享给各位技术爱好者。我也非常荣幸地提前看到了全书内容，相信本书一定会让您在架构和设计方面有所



收获。

使用 Tiny，可以让开发者从复杂的技术细节跳出来，而将精力集中到业务上，从而节省很多时间。热爱研究技术的朋友们也可以通过阅读本书获取技能上的提升，成为一名优秀的架构师。

祝愿 Tiny 的将来会越来越好！开源改变生活，开源改变世界

特赞 (tezign.com) CTO 黄勇  
2016 年 1 月 31 日于上海

## 序三

最近《三体》非常火，书中说到了三体文明将质子做维护展开的情形，甚至展开的质子可以覆盖整个星球。

其实在软件应用实践过程中，也存在多种不同层次的维度。例如，从用户角度来看，主要是人机交互上的一些感观；从业务角度来看，主要是各种不同的业务功能；从架构角度来看，软件又分属于不同的独立的体系，这些体系的相互作用最终构成了整个软件；从机器执行过程来看，又都是一些 0101 的二进制代码流或数据流……

如果直接使用基础开发语言和一些基础类库来进行开发，当软件比较小的时候没有任何问题。但是作为一个企业级应用来说，如果还是采用同样的模式，那么后果将会是不堪设想的。我见过许多业务上非常成功，但是由于没有良好的架构支撑而已经崩溃或走向崩溃的系统。真正优秀和成功的软件产品，必须有一个良好的架构支撑，才可以实现持续的发展和进步。

本书得益于作者的深厚功底及勇敢实践，并以企业应用中遇到的十多个真实应用场景作为主题进行架构实践，对这些主题按问题概述、分析问题和解决问题的过程进行了有益探索。虽然不能说所有的解决方案一定都是最优的，但是应该说都体现了作者在这方面所做的努力。“一花独放不是春，万紫千红春满园。”，也欢迎有更多、更好的企业级架构方面的实践书籍涌现出来，促进企业级 Java EE 应用的发展。

恒生电子股份有限公司执行总裁/CTO 范径武  
2016年1月25日于杭州



# 前 言

从本书的书名《企业级 Java EE 架构设计精深实践》可以看到几个关键词：企业级、Java EE、架构和实践。简单地说，本书是一本基于 Java EE 技术路线，面向企业级应用，解决企业级应用架构问题，并以实践为主的书籍。

目前市场上已经有了许多 Java EE 领域的书籍，涵盖了 Java EE 规范、Java EE 的各种流行前后台框架等。这些书籍有一些写得非常好，比较全面地覆盖了企业级 Java EE 应用的方方面面。因此本书不会讲解 Java 相关的基础知识，不会讲解 Java EE 规范方面的知识，也不会花大量篇幅介绍流行框架的使用及技巧，而是把重点放在需要解决的一些企业级应用中的问题及其解决方案和实践过程，让读者跟随笔者的思路一起实践。笔者不能保证书中所有的实践都是最优解，但一定是相当不错的解。

企业级应用是本书内容的核心，那么什么是企业级应用呢？这个问题本身就没有标准答案。但是企业级应用涉及的一些领域还是可以列举的。

- ❑ 数据持久化：这里的数据泛指结构化和非结构化数据。这些数据从产生到消亡有着漫长的创建、更新、查询和删除的生命周期，有的数据甚至要存在许多年。这些数据需要为避免丢失而进行存储操作，为高效利用而进行缓存加载操作，再加上数据自身的 CRUD 等操作，在上述过程中就需要解决各种各样的问题。
- ❑ 海量并发访问：对于小的应用系统，访问者可能就是一个或几个人，稍大点的系统访问者有上百人。但是对于企业级应用尤其是互联网级应用来说，其注册用户、在线用户和并发请求都是巨大的。应用请求数较少和应用请求数巨大的架构模式是有巨大差异的，由此也会引入许多的架构问题。
- ❑ 海量数据：对于企业级应用来说，百万级数据只不过是起步，千万甚至数十亿条数据都是常见的。在数据规模比较小的时候不存在问题，而在数据规模比较大的时候就会出现严重的性能问题。为了保证在海量应用的情景下也可以使用，这时原有的数据架构就需要进行重构。
- ❑ 应用规模庞大：应用的规模越大，开发、测试、集成和维护起来就越困难。笔者就亲眼见到过一个项目，业务方面是非常出色的，但是随着产品的规模越来越大，而项目的开发方式无法适应，导致整个项目无法继续进行，以失败而告终。所以如何让应用随着规模的变大，其开发、测试、集成和维护成本只是出现等比例增长，而不是随着应用规模的变大产生指数级增长，这就变得非常重要了。
- ❑ 界面需求复杂：随着应用规模的变大，企业级应用的界面也越来越多。而在企业

级应用中，又需要对使用权限、界面一致性和界面的易集成性等有非常高的要求。所以在企业级应用中，界面层有一个良好解决方案是非常有挑战性的。

- 与第三方系统协作：与小的应用不同，企业级应用的生成周期比较长，一个企业当中又有多个不同方面的系统同时存在，这些系统之间往往需要有数据或服务协作。但是由于不同的应用可能是来自于不同厂商采用不同开发语言和不同数据库开发的，所以这就对它们协同运作产生了许多问题。

总之，企业级应用可以理解成业务复杂、规模巨大、数据种类多、数据量大、访问请求大、生命周期长和并发请求巨大的应用。正是由于上述特点而产生了许许多多的问题，而这些问题的解决方案往往是有冲突的，因此如何艺术性地解决这些问题就往往需要从架构上想办法。

本书重点介绍的 Tiny 框架是 Tiny 开源小组历经 5 年业余时间，孜孜不倦地投入与坚持的产物。它是通过团队艰苦卓绝的努力所构建的体系完整、内容庞大的企业级开源框架。笔者和 Tiny 团队在解决这些问题时做了大量的尝试和实践，在诸多领域都有非常专业的解决方案，我们非常愿意通过本书与读者分享。

## 本书特色

### 1. 基于原创，开拓思路

本书有大量的原创实践及原理性讲述，不管读者用不用本书中的框架，都可以对企业级 Java EE 架构中的一些技术及其解决方案有较为深入的理解，这对于读者开拓思路，避免踩同样的“坑”有非常强的借鉴作用。

### 2. 涵盖广泛，讲解深入

本书介绍的内容都是当前主流框架和优秀技术，涵盖缓存、文件处理、模板语言、服务层、数据库架构、数据库水平扩展、元数据、Web 架构和前端界面等 Java EE 领域经常用到的知识，涉及企业级应用架构开发的方方面面。对这些内容的讲解不是泛泛而谈，而是有相当的深度。

### 3. 内容实用，注重实践

本书内容非常注重知识的实用性和可操作性，这些内容都是作者多年以来构建 Java EE 开发框架实践过程中的宝贵经验汇集而成的。本书每章都给出了实践性很强的开发案例供读者理解。

### 4. 思路清晰，文笔简练

本书每章按照问题背景、需求分析、解决思路、架构设计、实践示例和章节总结的顺序行文，丝丝入扣，符合人们的认知规律。而且本书行文风格朴实，文笔简练，知识的讲解由浅入深、循序渐进，即便是初学者，只要肯用心也会有所收获。

## 5. 与众不同，切中要害

本书有相当多的实践具有“反模式”的特质，也就是说与常见的解法不一样，再辅之以简单的示例，可以让读者非常容易理解其设计精要，有醍醐灌顶之感。

## 6. 理论与实践齐抓，局部与整体并重

本书注重问题的解决思路 and 开发细节，既介绍了企业级 Java EE 架构的设计思想，也兼顾了应用的具体实践，避免了学习过程中理论与实践的脱节，连贯而统一。本书每章都有独立的测试用例，最后还提供了完整的 Web 开发案例，这样通过先局部后整体的方式，可以让读者更好地理解 Java EE 开发架构。

## 7. 编排科学，适用面广

本书内容编排既可以让初学者对 Java EE 架构有整体的认知，也可以让那些学有余力、经验丰富的开发人员深入钻研自己感兴趣的内容。无论是哪个层次的读者，阅读完本书后都会有所收获，甚至对以前自己觉得深不可测的问题有豁然开朗的感觉。

## 本书内容

本书正文共 13 章，采用问题背景、需求分析、解决思路、架构设计、实践示例和章节总结的顺序来组织内容。下面简单介绍一下每章的内容。

### 第1章 虚拟文件系统实践

本章从 Apache VFS 框架存在的内存泄露 Bug 说起，引入企业级应用常见的虚拟文件系统的概念，并介绍了 Apache VFS 框架和 Tiny VFS 框架。另外，还讲解了 VFS 的设计思想，介绍了 VFS 管理器、虚拟文件接口和模式提供者等概念，并给出本地文件、JAR 和 FTP 的实践用例。

### 第2章 缓存实践

本章以 Java EE 应用遇到的性能问题作为话题，引导读者了解缓存技术的重要性。在讲解一个实际项目缓存的代码变迁过程中，自然而然地列举了对现有缓存方案的改进：业务代码与缓存框架分离；具体缓存实现的切换不影响业务开发；支持 XML 配置等。本章介绍了两种缓存架构：字节码缓存方案和动态代理缓存方案，并对比了两者的优缺点。希望读者通过阅读本章内容，能够对开发企业级应用有新的心得体会。

### 第3章 文件处理框架实践

本章介绍了文件处理框架，可以有效地解决 Java EE 应用模块化导致的各种配置文件分散在不同模块而带来的处理这些资源难度大的问题。在介绍 Tiny 文件处理框架的过程中，请仔细阅读文件扫描器接口、文件扫描器主入口和定时扫描器的介绍，理解框架如何

把文件的扫描、文件的变化、文件的遍历等与文件的实际处理分离，体会“开-闭”原则的具体运用。

## 第4章 模板语言实践

本章介绍了模板语言及其在 Java EE 领域的实践开发。为了便于读者理解，笔者先介绍模板语言的概念、原理和应用场景，并列举了常见的模板语言 Velocity、FreeMarker 和 Tiny，比较了三者的优势和不足。特别是在设计开发的章节，按模板语言的架构、语法解析和渲染机制的顺序，从零开始讲解如何构建一种模板语言。通过对本章内容的学习，有能力的读者完全可以尝试扩展模板语言。

## 第5章 数据库访问层实践

本章首先介绍了应用框架的三层架构，然后引入了数据库访问层的相关概念。为了便于读者理解数据库访问层，笔者讲解了业界常用的数据库访问层框架 Hibernate、Ibatis 和 JPA，以及 DSL 风格的数据库访问层 JEQUEL、JOOQ 和 Querydsl。通过比较它们的优缺点，分析了 Tiny 团队开发 TinyDsl 的原因，然后详细说明 TinyDsl 的实践过程。最后通过具体示例说明各种数据库访问层框架的用法和配置。

## 第6章 数据库扩展实践

本章侧重介绍数据库水平扩展方案。首先从互联网的发展所导致的业务应用压力日益增加，从而让读者认识到数据库水平扩展方案对 Java EE 企业级应用是非常合理的需求。论证了水平扩展的合理性之后，列举了常见的扩展层次：DAO 层、DataSource 层、JDBC 层和 Proxy 层，重点讲解了 Tiny 分库分表的技术架构和设计思路，并讲解了分库分表、读写分离和集群事务的处理，还与开源软件 Routing4DB 做了详细对比。最后通过具体示例演示数据库水平扩展的配置和增删改查的结果。

## 第7章 服务层实践

本章首先介绍了传统的服务层定义，以及与表现层、业务逻辑层的关系，然后讲解了 Tiny 对服务层的改进，通过重新定义服务概念，引入服务定义、服务注册和服务中心的设计思路。基于 Tiny 服务体系，应用可以不必关心服务提供方的具体信息。同时，由于远程调用与本地调用在代码层面并没有区别，所以系统根据不同场景进行多机部署或者单机部署时，无须对代码进行调整。远程调用体系由核心服务中心来组织整个调用网络，水平扩展极为容易。最后列举了本地场景、远程场景、单中心场景和多中心场景下，读者如何配置部署服务与服务中心。

## 第8章 流程引擎实践

本章介绍了面向对象编程的不足，引入了面向组件编程的概念。流程引擎框架（后文亦称之为 Flow）是一款基于面向组件开发的组件流程执行框架。目前 Flow 支持两种流程：逻辑流程和页面流程。在设计部分，读者可以通过组件设计、流程管理、流程配置和流程

执行了解流程引擎的强大。在实践部分笔者介绍了如何通过组件化开发算术表达式的流程。

## 第9章 元数据实践

本章介绍的是基本的元数据元素，包含方言模板、标准数据类型、业务数据类型和标准字段这几类基础元数据。最后的示例演示使用 Eclipse 插件设计元数据，包括通过工具动态生成业务 Java 代码和 SQL 语句。

## 第10章 展现层开发实践

本章首先介绍了 Java EE 展现层的常用代表技术 Servlet、JSP 和模板语言，并详细分析了三者的特性及优缺点。接着讲解了展现层方案设计，列举了设计人员常见的做法和技巧。然后给出了 Tiny 框架的展现层架构思路：采用模板语言做前端展示，同时引入组件包的概念，通过资源合并和压缩等框架手段，提升展现层的性能。最后演示了 Tiny 展现层的告警框和文本输入框的示例。

## 第11章 Web扩展实践

本章主要讲解了 Tiny Web 层框架的实践过程。首先介绍了 Tiny Web 层框架的由来及设计思想与设计原理。然后详细介绍了框架内置过滤器与处理器的使用方式。最后通过具体示例讲解 Tiny Web 层框架的开发过程，从而加强对 Tiny Web 层框架的理解。

## 第12章 Tiny统一界面框架实践

本章重点介绍了 Java EE 领域的界面开发设计，并通过介绍问题由来，归纳用户需求，提出 UIML 解决方案。本章的侧重点是介绍 UIML 设计思路和开发细节，让读者了解 UI 组件化开发带来的好处与便利。在实践小节中介绍了图形编辑器的使用，最后还列举了 UIML 配置开发示例，归纳和总结了开发过程中的常见问题。

## 第13章 RESTful实践

本章讲解了业界流行的 RESTful 的背景和开发方式。首先介绍了 Spring RESTful 的开发方式，这种方式比较适用于新项目开发，而不适用于已经开发完毕的项目。然后讲解了 Tiny RESTful 如何解决这个问题。本章详细介绍了 Tiny RESTful 的设计思想与实现过程，最后的 Web 层示例演示了 RESTful 风格配置 Web 映射。

## 本书读者对象

- 有一定 Java EE 基础的编程人员；
- 对 Java EE 架构设计感兴趣的开发人员；
- 构建企业级应用的架构师；
- 学有余力，乐于尝试新事物的初学者；
- 想通过本书学习分析问题和解决问题思路的人员。

## 本书源文件获取方式

本书涉及的案例源文件需要读者自行下载。请登录清华大学出版社的网站 (<http://www.tup.com.cn>)，搜索到本书页面，然后单击“资源下载”模块中的“课件下载”或“网络资源”按钮即可下载。

## 本书作者

本书由罗果主笔编写，其他参与编写的人员有严诚、陈佼、任辉、李强强、王维煜、张程浩、王玲珑、严文杰、葛强燕、陈超、陈锴、陈佩霞、陈锐、黎华、李鹏钦、李森、李奕辉、李玉莉、刘仲义、卢香清、鲁木应、马向东、麦廷琮、米永刚、欧阳昉、綦彦臣、冉卫华、宋永强、滕科平、王秀丽、王玉芹、魏莹、魏宗寿、乐西萍。

限于笔者及团队能力水平，书中可能会存在一些疏漏或对软件领域一些有失偏颇的理解，对于有些框架的优缺点总结也不一定完全到位，我们给出的一些实践与希望达到的完美程序也许会有有一定的差距。欢迎各位读者向我们提出意见和建议，也欢迎对我们的方案提出批评与指正，给出更好的思路，以便及时进行优化和完善。联系我们请发邮件至 [bookservice2008@163.com](mailto:bookservice2008@163.com)。

最后祝各位读者读书快乐，学习进步！

编著

# 目 录

第 1 章 虚拟文件系统实践	1
1.1 背景介绍	1
1.2 什么是 VFS	1
1.3 VFS 对比	2
1.3.1 Apache VFS	2
1.3.2 Tiny VFS	2
1.4 VFS 框架设计思想	3
1.5 VFS 实现讲解	5
1.5.1 VFS 管理器	5
1.5.2 SchemaProvider 模式提供者	7
1.5.3 FileObject 虚拟文件	7
1.5.4 FileObjectFilter 过滤接口	10
1.6 VFS 应用示例	12
1.6.1 本地文件	13
1.6.2 Jar 文件	13
1.6.3 FTP 文件	14
1.6.4 ZIP 文件	14
1.7 本章总结	15
第 2 章 缓存实践	16
2.1 缓存简介	16
2.1.1 问题的提出及其解决方案分析	16
2.1.2 用户需求	19
2.1.3 Tiny 缓存解决思路	19
2.2 字节码缓存设计	23
2.2.1 字节码操作工程	23
2.2.2 预编译工程	27
2.2.3 缓存实现工程	28
2.2.4 技术特点	31
2.3 动态代理缓存设计	31
2.3.1 缓存接口定义	32



2.3.2	切面缓存工程	33
2.3.3	技术特点	43
2.4	缓存方案实践	43
2.4.1	字节码方案配置	43
2.4.2	字节码方案示例	44
2.4.3	动态代理方案配置	46
2.4.4	动态代理方案示例	48
2.5	本章总结	54
2.5.1	关键点：缓存实现方案的可替换性	54
2.5.2	关键点：缓存代码与业务代码的解耦	54
2.5.3	关键点：模板语言的应用	55
<b>第 3 章</b>	<b>文件处理框架实践</b>	<b>56</b>
3.1	概述	56
3.1.1	FileProcessor 接口	56
3.1.2	FileResolver 接口	58
3.1.3	FileMonitorProcessor 类	60
3.2	基础文件扫描器	60
3.2.1	XStreamFileProcessor 类	61
3.2.2	I18nFileProcessor 类	63
3.2.3	Annotation 扫描器	63
3.2.4	SpringBeansFileProcessor 类	65
3.3	完整示例	66
3.3.1	单独使用	66
3.3.2	通过配置文件配置	66
3.4	本章总结	67
<b>第 4 章</b>	<b>模板语言实践</b>	<b>69</b>
4.1	模板语言简介	69
4.1.1	模板语言构成	69
4.1.2	模板语言应用场景	70
4.2	常见的模板语言	71
4.2.1	Velocity 模板语言	71
4.2.2	FreeMarker 模板语言	71
4.2.3	Tiny 模板语言	72
4.3	Tiny 模板语言设计	73
4.3.1	Tiny 模板语言的构建原因	73
4.3.2	模板语言执行方式	73
4.3.3	模板语言架构	73

4.3.4	Tiny 模板语言实现与扩展	74
4.3.5	模板语言语法解析	88
4.3.6	模板语言渲染机制	94
4.4	模板语言的使用	99
4.4.1	依赖配置	99
4.4.2	模板语言的配置	99
4.4.3	模板语言的 Eclipse 插件	101
4.4.4	Hello, TinyTemplate	104
4.5	模板语言语法介绍	105
4.5.1	变量	106
4.5.2	取值表达式	107
4.5.3	Map 常量	107
4.5.4	数组常量	108
4.5.5	其他表达式	109
4.5.6	索引表达式	111
4.5.7	#set 指令	111
4.5.8	条件判断	112
4.5.9	==相等运算	113
4.5.10	AND 运算	114
4.5.11	OR 运算	114
4.5.12	NOT 运算	114
4.5.13	循环语句	114
4.5.14	循环状态变量	116
4.5.15	循环中断: #break	116
4.5.16	循环继续: #continue	117
4.5.17	while 循环	117
4.5.18	模板嵌套语句#include	117
4.5.19	宏定义语句#macro	118
4.5.20	宏引入语句#import	120
4.5.21	布局重写语句#layout #@layout	120
4.5.22	停止执行#stop	122
4.5.23	返回指令#return	122
4.5.24	行结束指令	123
4.5.25	读取文本资源函数 read 和 readContent	123
4.5.26	解析模板 parser	123
4.5.27	格式化函数 fmt、format 和 formatter	123
4.5.28	宏调用方法 call 和 callMacro	124
4.5.29	实例判断函数 is、instanceOf 和 instance	124