

# 精通

# Spring —

## Java 轻量级架构开发实践

孟劼 编著

615 个示例，帮助读者全面掌握 Spring

详尽剖析代码，方便读者学习

详细讲解 Spring 和其他流行 Java 技术的整合应用



人民邮电出版社  
POSTS & TELECOM PRESS



CD-ROM

# 精通

# Spring —

## Java 轻量级架构 开发实践

孟劼 编著

人民邮电出版社

## 图书在版编目 (CIP) 数据

精通 Spring: Java 轻量级架构开发实践 / 孟劼编著. —北京: 人民邮电出版社, 2006.10

ISBN 7-115-15029-X

I. 精... II. 孟... III. ①计算机网络—程序设计 ②JAVA—程序设计 IV. ①TP393②TP312  
中国版本图书馆 CIP 数据核字 (2006) 第 080582 号

## 内 容 提 要

本书由浅入深、循序渐进地介绍了开源应用框架 Spring 的开发思想和实践。全书共分为 20 章, 内容涵盖了 Spring 的基础知识体系、获取和部署方法、背后的先进核心理念、相关的部分设计模式和 J2EE 核心模式、核心组件 Bean 工厂的使用、Spring 风格的面向方面编程、各种 J2EE 企业级应用框架和解决方案 (包括持久数据访问、事务管理、EJB、邮件服务、时程安排、Web MVC 框架、表现层视图技术、JMX 远程方案等)、Spring 和单元测试等内容。

此外本书也剖析了 Spring 宠物店示例, 最后还讨论了宠物店和 Martin Fowler 企业架构模式 (POEAA) 的关系。

本书适用初中级读者学习, 对高级用户也有很好的参考价值。

### 精通 Spring——Java 轻量级架构开发实践

◆ 编 著 孟 劼

责任编辑 屈艳莲

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京隆昌伟业印刷有限公司印刷

新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 38.75

字数: 939 千字

印数: 1—5 000 册

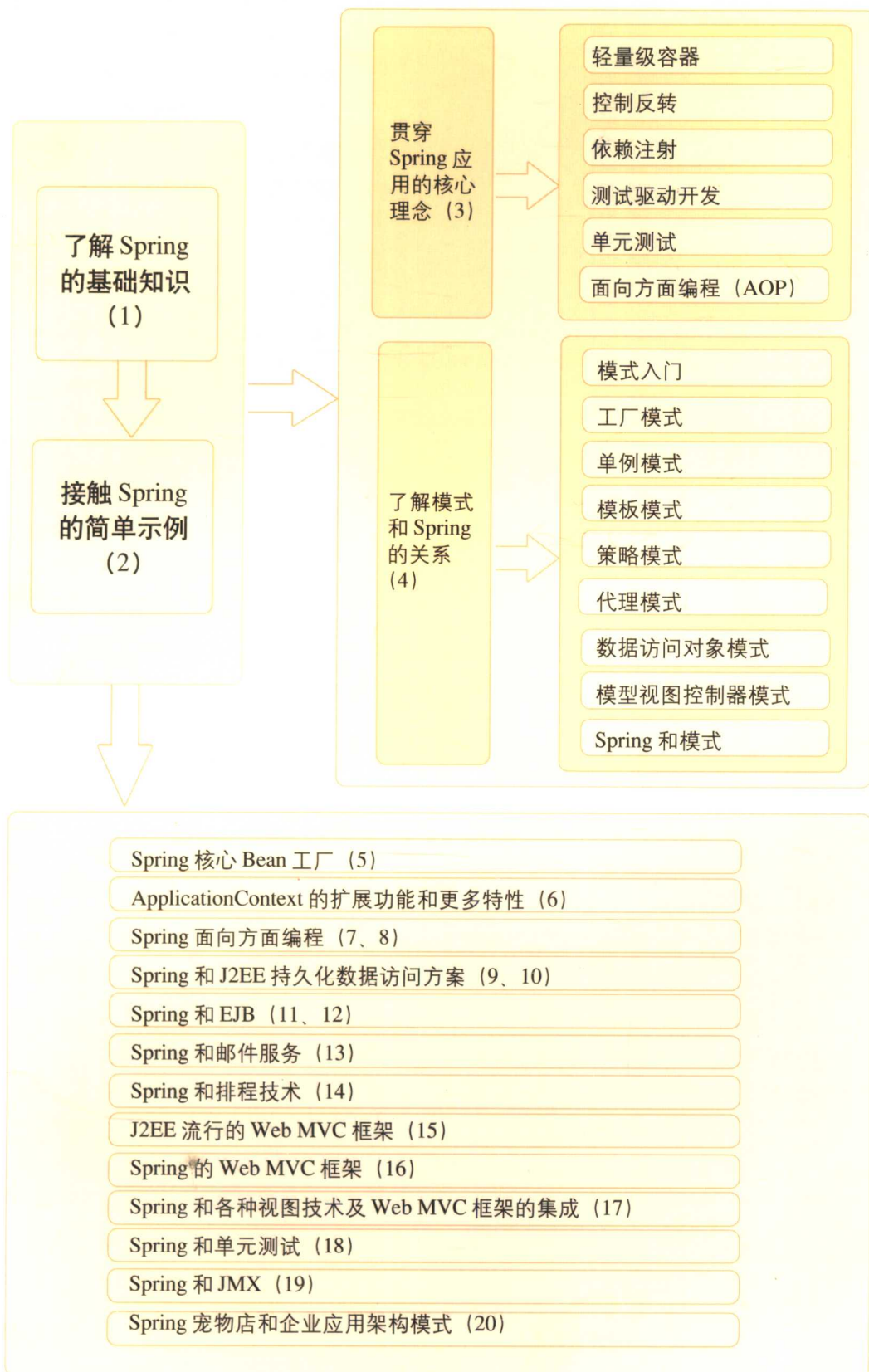
2006 年 10 月第 1 版

2006 年 10 月北京第 1 次印刷

ISBN 7-115-15029-X / TP · 5567

定价: 68.00 元 (附光盘)

读者服务热线: (010)67132692 印装质量热线: (010)67129223



# 流行的Java

## 技术整合应用

### 贴近实际 指导实践

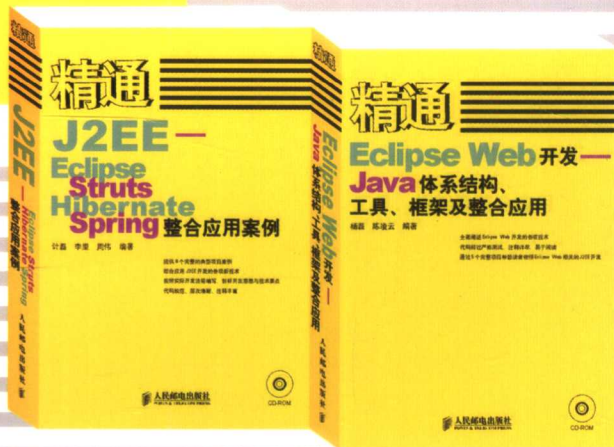


精通 Struts ——  
Java 流行服务器、框架、工具及整合应用  
书号: 15014  
定价: 58 元

精通 Spring ——  
Java 轻量级架构开发实践  
书号: 15029  
定价: 65 元

精通 J2EE ——  
Eclipse、Struts、Hibernate Spring  
整合应用案例  
书号: 15033  
定价: 69 元

精通 Eclipse Web 开发 ——  
Java 体系结构、工具、框架及整合应用  
书号: 15034  
定价: 69 元



<http://www.ptpress.com.cn>



## 前 言

J2EE 非常复杂, 往往使人们难以把握, Spring 的出现大大简化了传统的 J2EE 开发模式。作为一款基于控制反转、依赖注射的轻量级容器, Spring 在广阔的 J2EE 世界中筑起了一座开源应用框架的里程碑。

本书帮助广大国内开发者由浅入深, 全面了解并掌握 Spring。无论对于初涉或已经熟悉 Spring 的读者, 通过阅读本书, 都将大获裨益。

本书在内容的编排和目录组织上可谓匠心独具, 力争完美。读者通过全书丰富真实、趣味盎然的示例, 可以迅速掌握 Spring 的开发及其背后的核心理念。另外, 本书也是一本软件模式和各类流行 J2EE 框架的精要指南。

本书有如下特点。

(1) 框架源码, 剖析到位。

本书对 Spring 框架的部分核心源码进行了剖析, 同时也剖析了 Struts 框架的核心源码。本书可以为那些对框架源码有兴趣的读者带来一定的启发。这部分知识在本书的第 4 章和第 15 章进行了详细的讲解。

(2) 示例详尽, 趣味实用。

本书源码和注解说明非常详尽, 富有趣味性并贴近实际应用。这些代码都经过了严格的测试。此外, 各章节的源码都有明确含义的包定位, 非常便于索引。

(3) 内容丰富, 新颖前瞻。

本书不仅着重于介绍 Spring 框架自身以及它和众多流行的 J2EE 框架或技术的整合。同时也围绕着 Spring 背后的一些先进思想展开介绍, 如轻量级容器、依赖注射/控制反转、面向方面编程、敏捷实践(包括单元测试、测试驱动)等。这些软件思潮都是近年来 J2EE 社群所涌现出的最新产物, 它们具有足够的优雅内涵和前瞻性。所以说, 学会了 Spring, 也就等于学会了如何使用大量其他的 J2EE 框架和技术, 更重要的是学会了一种看待 J2EE 开发的独特视点。本书写作之际, Spring 已经推出了 2.0 版本, 甚至还有了 .NET 的版本, 相信学习一个足够优秀、活跃、前瞻的开源应用框架, 肯定物超所值。

(4) 结合模式, 标新立异。



在本书中，笔者以独特的视点诠释了多种和 Spring 相关的模式，其中不乏对模式本质的探讨、设计模式、J2EE 核心模式、框架模式以及企业架构模式（POEAA），这是同类 Spring 书籍中所罕有的，也是一次浅尝辄止的尝试。笔者始终认为模式是 Spring 前行的路标。

## （5）敏捷实践，落在实处。

敏捷软件开发是一种面临迅速变化的需求，快速开发软件的能力，它是软件领域中一个历久弥新的课题。本书的一大特点就是抛开对敏捷的空谈，深入讲解其中的关键实践：单元测试和测试驱动。在本书多个章节中，示例的演示都采用了小步敏捷。

## （6）设计重构，融会贯通。

本书中一个很重要的特点就是经常重构同一个示例，比如从一种技术切换至另一种技术实现（如 AspectJ 到 Spring AOP）、从不使用到使用 Spring（如 Spring 和 EJB）产生的效果、从一个简陋的架构过渡到一个灵活的架构（Spring 和邮件服务）等。通过这些重构过程，不仅对各种技术做了横向比较，也加深了读者对应用程序设计的认识。此外本书结合了多种技术，对 Spring 宠物店示例进行了深入讲解和部分重构。

## （7）讲解透彻，事半功倍。

Spring 的功能非常丰富，本书目的不在于列出面面俱到的功能和模棱两可的示例，而是针对每个概念，力求使读者吃透。通过本书，相信读者对其他同类技术的学习也将事半功倍。

## 本书包括的内容

第 1 章介绍了 Spring 的基础知识和概貌，它是一个用于简化 J2EE 开发的优秀开源框架。

第 2 章介绍了 Spring 发布包的组织结构及相关依赖软件包。随后给出了 Hello World 重构示例以及 Spring Web MVC 示例，展示了 Spring 应用的最小集合。

第 3 章介绍了蕴含在 Spring 背后的一些核心理念，它们分别是轻量级容器、反向控制（IoC）、依赖注射（DI）、面向方面编程（AOP）、单元测试和测试驱动开发。

第 4 章介绍了部分软件模式知识，其中涉及了设计模式和 J2EE 核心模式，并且也尝试了 Spring 和模式的结合讲解。

第 5 章深入了 Spring 的核心框架，围绕着其中的核心组件 BeanFactory，展示了 Spring 丰富的依赖装配方式和应用机制，另外对其他的一些核心组件也做了细致的介绍。

第 6 章介绍了 Spring 核心 Bean 工厂的扩展子接口 ApplicationContext 以及其他更多特性，如 Bean 包装器、属性编辑器、验证等。

第 7 章介绍了 Spring AOP 框架的核心概念、功能、应用及原理。

第 8 章介绍了 Spring AOP 的高级引介特性、自动代理以及它在 J2EE 企业级应用方面提供的部分功能，如事务管理等。

第 9 章关注了一些常见和目前流行的 J2EE 持久化数据访问框架和解决方案。在 J2EE 应用中，持久层的数据访问永远是一个重头戏。本章主要围绕了几种持久策略和一个持久模式而展开，它们分别是基于 JDBC 的手工原始策略、基于 SQL 语句映射的小步 ORM 策略、基于完备 O/R 映射的透明持久化策略以及关联所有这些策略的数据访问对象模式。

第 10 章介绍了 Spring 的持久之道的展开，从中可以看到，Spring 不但提供了自己的 JDBC 抽象以及 DAO 支持，而且也提供了对各种持久化方案的优雅整合，通过 Spring 来进行企业级开发更是如鱼得水，事半功倍。

第 11 章回顾了 EJB 的概念，结合 JBoss 对 EJB 的部署和开发做了介绍。

第 12 章讲述了 Spring 和 EJB 的结合，详细剖析了 Spring 的 EJB 抽象访问层。

第 13 章介绍了 Spring 和邮件服务的结合。

第 14 章介绍了 Spring 和排程技术的结合。

第 15 章介绍了目前 J2EE 中比较常用的一些 Web MVC 框架，剖析了使用这些 Web MVC 框架之间的利弊得失。

第 16 章介绍了 Spring 的 Web MVC 框架。

第 17 章介绍了 Spring 和各类视图及 Web MVC 框架的集成。

第 18 章介绍了 Spring、单元测试以及一些更高级的软件测试技术。

第 19 章介绍了 Spring 和 JMX 技术。

第 20 章结合企业架构模式，再次深入剖析了 Spring 宠物店示例。

### 本书推荐的学习方法

对于本书所介绍的 Spring，推荐的学习方法如下。

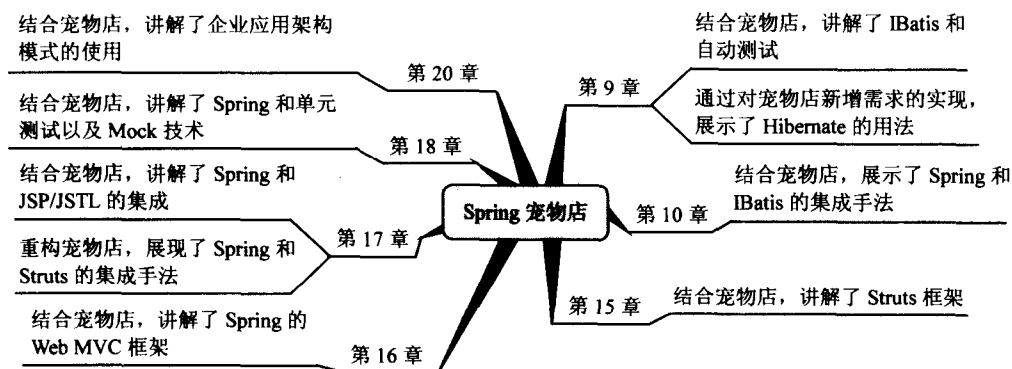
(1) 通过第 1 章和第 2 章对 Spring 可以有个简单的了解，其中给出了两个 Spring 的最小应用，这些应用示例并不简单，可见 Spring 门槛不低。

(2) 第 3 章和第 4 章深入讲解了 Spring 背后的核心理念以及 Spring 与模式的知识，这些章节体现了部分 Spring 的处世哲学。初级读者在阅读这些章节时，如果暂时不能消化，可直接跳过。

(3) 从本书的第 5 章到第 19 章，陆续介绍了 Spring 框架的核心以及 Spring 和各种 J2EE 技术的结合运用，这些章节面向各种层次的读者。

(4) 对于高级读者，本书也保证了相当的深度、广度和自由度，如第 3 章关于轻量级容器、IoC/DI、AOP、TDD 等的启发式讲解，第 4 章关于软件模式的独特视点，第 20 章关于 Spring 宠物店结合企业架构模式（POEAA）的讲解，第 4 章和第 15 章对 Spring 和 Struts 框架核心源码的解读等。

此外，本书也是宠物店学习的好资料，下面给出了 Spring 宠物店的学习地图：



### 本书的约定

1. 本书的源代码有些有编号，有些没有。有编号的代码是指实际可运行的或者在 Spring 发布包中可找到的。而没有编号的代码，它们只是简单的示意说明，或者是那些已编号代码



的延续、修改或新增，读者在阅读时要注意。

2. 阅读本书需要读者熟悉 UML 的基本知识。

本书由孟劼统筹编写，同时参与编写的还有戎伟、张双、郝彩红、温丽、尹亮亮、马叶兵、白自利、李朋、候丽宝、赵志升、郭军军、张丽仙、梁倩、杜文婧、陈海慧、仲维俊、孙勇、马军、王妍、刘会神、刘小东、于召亮、郑林、周宇炜、公芳亮、李臻、赵艳铎、单红等人，在此一并表示感谢。

书中的不足之处敬请读者批评指正。欢迎读者通过发送 E-mail 至 [quyanlian@ptpress.com.cn](mailto:quyanlian@ptpress.com.cn) 与本书责任编辑交流。

编者

2006 年 7 月

## CD-ROM 光盘使用说明

### 1. 工程环境

- Java 环境是 JDK1.5.0\_06, IDE 环境是 MyEclipse 4.0.0+Eclipse 3.1.0。
- 使用了笔者自定义的 User Library——SpringMagicBox, 它指向了 lib 下的所有包, 读者在使用时, 请自行改变包的引用路径。
- 除了上述 Library 以外, Buildpath 中还引入了 Eclipse 自带的 J2EE 1.4 Library。
- 本书使用了 Tomcat 5.5.4 作为 (不涉及 EJB 的) JSP/Servlet 容器, 而 EJB 容器则采用了 JBoss 4.0.3 SP1。

### 2. 工程目录

#### 第 1 章 (\chapter5)

- .myeclipse: myeclipse 的相关设定
- .settings: Eclipse 的相关设定
- bin: 编译后类的存放路径
- conf: 相关的配置文件
- db: 数据库脚本, 其中包含了宠物店、Quartz、Hibernate
- deploy: EJB 发布包的目录
- images: 相关图片
- lib: 所有的开源包
- other: 一个 Eclipse 插件和 Jboss Bug 补丁
- petstore-src: 修改过的宠物店源码
- src: 核心源代码
- test: 测试代码, 它也是本书核心源代码的一部分
- webapps: 所有的 Web 工程及相关发布包
- .classpath: Eclipse 的 classpath 相关设定
- .mymetadata: 工程的一些自定义属性设定

## ■ .project: 工程属性相关设定

### 3. 本书所附的源码

#### 第2章 (\chapter2)

■ \getstart\\*

Hello World 的重构示例

■ \getstart\web\\*

Spring Web MVC 的最简示例

■ webapps\ch2helloworld.war

Spring Web MVC 最简示例的 Web 发布包

#### 第3章 (\chapter3)

■ \iocdi\\*

依赖注射/控制反转的示例

■ \aop\\*

未采用 AOP 手法编写的传统业务逻辑

■ \aop\proxy\\*

采用动态代理实现的 AOP 重构

■ \aop\aspectj\\*

采用 AspectJ 实现的 AOP 重构

■ \junit\\*

单元测试的示例

■ \tdd\\*

测试驱动的示例

#### 第4章 (\chapter4)

■ \pattern\\*

模式概念的示例 1

■ \pattern\better\\*

模式概念的示例 2

■ \pattern\factory\strongdep\\*

未采用工厂模式, 产生强依赖的示例

■ \pattern\factory\dipioc\\*

导入依赖倒置/控制反转的示例

■ \pattern\factory\\*

引出工厂模式的示例

■ \pattern\singleton\\*

单例模式的相关示例

■ \pattern\template\\*

模板模式的相关示例

■ \pattern\strategy\\*

策略模式的相关示例

■ \pattern\proxy\\*

代理模式的相关示例

■ \pattern\dao\\*

数据访问对象模式的结构

■ \pattern\dao\sample

一个真实的数据访问对象模式示例

■ \pattern\mvc\\*

MVC 模式和命令模式的结合示例

■ webapps\ch4patternmvc.war

MVC 模式结合命令模式的 Web 发布包

#### 第5章 (\chapter5)

■ \beanfactory\\*

初始 Bean 工厂示例

■ \beanfactory\staticfactory\\*

使用静态工厂方法创建 Bean 示例

■ \beanfactory\instancefactory\\*

使用实例工厂方法创建 Bean 示例

■ \beanfactory\singletonornot\\*

是否使用 Singleton 示例

■ \beanfactory\coredi\\*

Bean 工厂的使用方法和细节示例

■ \beanfactory\beannature\\*

定制 Bean 的原生属性示例

■ \beanfactory\childbean\\*

父和子 Bean 示例

## ❑ \beanfactory\postprocessor\\*

使用后处理器示例

## 第 6 章 (\ chapter6)

### ❑ \beanfactory\appcontext\\*

ApplicationContext: 资源国际化、底层资源及其他示例

### ❑ \beanfactory\event\\*

ApplicationContext: 事件传播示例

### ❑ \beanfactory\cstmpropeditor\\*

ApplicationContext: 使用自定义属性编辑器示例

### ❑ \beanfactory\beanwrapper\\*

包装器和属性编辑器使用示例

### ❑ \validate\\*

使用 Spring 验证的一个示例

## 第 7 章 (\ chapter7)

### ❑ \getstart\\*

采用 Spring AOP 重构传统业务逻辑的示例

### ❑ \beforeadvice\\*

Spring AOP 前置建议示例

### ❑ \afterreturningadvice\\*

Spring AOP 后置建议示例

### ❑ \aroundadvice\\*

Spring AOP 环绕拦截建议示例

### ❑ \throwsadvice\\*

Spring AOP 抛出建议示例

### ❑ \pointcutadvisor\\*

Spring AOP 切入点点和顾问的使用示例

## 第 8 章 (\ chapter8)

### ❑ \introduction\\*

Spring AOP 引介建议和自动代理示例

### ❑ \enterprise\\*

Spring AOP 企业级服务 (事务管理) 示例

## 第 9 章 (\ chapter9)

### ❑ \overview\jdbc\\*

使用 JDBC 持久概览

### ❑ \overview\ibatis\\*

使用 IBatis 持久概览

### ❑ \overview\hibernate\\*

使用 Hibernate 持久概览

### ❑ \ibatis\\*

IBatis 使用的详细示例和测试

### ❑ \hibernate\\*

Hibernate 结合 Spring 宠物店使用的示例和测试

## 第 10 章 (\ chapter10)

### ❑ \jdbc\\*

Spring JdbcTemplate 的使用示例

### ❑ \jdbc\object\\*

Spring 的对象化 RDBMS 使用示例

### ❑ \spring\hibernate\\*

Spring 整合 Hibernate 的使用示例

Spring 整合 IBatis 的示例请参考宠物店

## 第 11 章 (\ chapter11)

### ❑ \ejb\jboss\\*

在 JBoss 下开发部署 EJB 的示例

### ❑ \ejb\sbs\\*

无状态会话 Bean 的开发和部署示例

### ❑ \ejb\sfsb\\*

有状态会话 Bean 的开发和部署示例

## 第 12 章 (\chapter12)

- ❑ \ejb\spring\sbsb\\*
- ❑ webapps\ch12localsbsbclient.war
- ❑ \ejb\spring\sfsb\\*
- ❑ webapps\ch12localsfsbclient.war

Spring 整合本地无状态会话 Bean 的示例

Spring 整合本地无状态会话 Bean 示例的 Web 发布包

Spring 整合本地有状态会话 Bean 的示例

Spring 整合本地有状态会话 Bean 示例的 Web 发布包

## 第 13 章 (\chapter13)

- ❑ \javamail\\*
- ❑ \javamail\spring\\*
- ❑ \javamail\spring\mime\\*

使用 Spring 邮件抽象层发送简单文本邮件的示例

使用 Spring 更灵活地实现邮件发送

使用 Spring 发送 Multipart HTML MIME 邮件的示例

## 第 14 章 (\chapter14)

- ❑ \timer\\*
- ❑ \timer\spring\\*
- ❑ \quartz\\*
- ❑ \quartz\spring\\*
- ❑ \quartz\spring\persistence\\*

JDK 的 Timer 使用示例

Spring 结合 Timer 的使用示例

Quartz 的使用示例

Spring 结合 Timer 的使用示例

Quartz JDBC 持久的示例

## 第 15 章 (\chapter15)

- ❑ \webwork\\*
- ❑ webapps\ch15webwork.war
- ❑ \tapestry\\*
- ❑ webapps\ch15tapestry.war

Webwork2 的使用示例

Webwork2 示例的 Web 发布包

Tapestry4 的使用示例

Tapestry4 示例的 Web 发布包

Spring Web MVC 的示例请参考宠物店

## 第 17 章 (\chapter17)

- ❑ \springjspjstl\\*
- ❑ \springxslt\\*
- ❑ webapps\ch17springxslt.war
- ❑ \springpdfxls\\*
- ❑ webapps\ch17springpdfxls.war
- ❑ \springwebwork\\*
- ❑ webapps\ch17springwebwork.war
- ❑ \springtapestry\\*
- ❑ webapps\ch17springtapestry.war

Spring 集成 JSP/JSTL 的示例

Spring 集成 XSLT 的示例

Spring 集成 XSLT 示例的 Web 发布包

Spring 集成 PDF/Excel 的示例

Spring 集成 PDF/Excel 示例的 Web 发布包

Spring 集成 WebWork2 的示例

Spring 集成 WebWork2 示例的 Web 发布包

Spring 集成 Tapestry4 的示例

Spring 集成 Tapestry4 示例的 Web 发布包

## 第 18 章 (\chapter18)

- ❑ \springmock\\*
- ❑ \easymock\\*

Spring Mock 的使用示例

Easy Mock 的使用示例

## ❑ \springandeasymock\\*

结合使用 Spring Mock 和 Easy Mock 的示例

### 第 19 章 (\chapter19)

#### ❑ \jmx\\*

JMX HTML 控制台的示例

#### ❑ \jmx\standardmbean\\*

JMX 标准 MBean 编程示例

#### ❑ \jmx\notification\\*

JMX 通知模型编程示例

#### ❑ \jmx\dynamic\\*

JMX 动态 MBean 编程示例

#### ❑ \jmx\connector\\*

JMX 连接器编程示例

#### ❑ \jmx\spring\\*

JMX 结合 Spring 使用的示例

#### ❑ \jmx\spring\autodetect\\*

Spring 自动侦测并注册 MBean 的示例

#### ❑ \jmx\spring\objectname\\*

JMX ObjectName 命名示例

#### ❑ \jmx\spring\connector\\*

Spring 结合使用 JSR-160 连接器的示例

### 第 20 章 (\petstore-src)

#### ❑ \org.springframework.samples.jpstore\dao\\*

数据访问对象接口

#### ❑ \org.springframework.samples.jpstore\dao\ibatis\\*

数据访问对象接口的 IBatis 实现

#### ❑ \org.springframework.samples.jpstore\dao\ibatis\maps\\*

领域对象的映射文件

#### ❑ \org.springframework.samples.jpstore\dao\domain\\*

领域对象

#### ❑ \org.springframework.samples.jpstore\web\spring\\*

Web 表单和处理器的 Spring 实现

#### ❑ \org.springframework.samples.jpstore\web\struts\\*

Web 表单和处理器的 Struts 实现

#### ❑ webapps\jpstore.war

Spring 宠物店的 Web 发布包



## 目 录

|                                         |    |
|-----------------------------------------|----|
| 第 1 章 Spring 基础知识.....                  | 1  |
| 1.1 Spring 的来历和目标.....                  | 1  |
| 1.2 Spring 概览.....                      | 2  |
| 1.3 Spring 的应用场景.....                   | 3  |
| 1.4 Spring 和 J2EE .....                 | 5  |
| 1.4.1 经典 J2EE 体系架构.....                 | 5  |
| 1.4.2 轻量级容器 .....                       | 6  |
| 1.4.3 Spring 和 J2EE 的关系 .....           | 6  |
| 1.5 小结 .....                            | 6  |
| 第 2 章 开始 Spring 之旅.....                 | 7  |
| 2.1 获取 Spring.....                      | 7  |
| 2.1.1 两大开源网站简介 .....                    | 7  |
| 2.1.2 获取 Spring 发布包和源代码.....            | 7  |
| 2.2 Spring 的标准发布包和相关软件包 .....           | 9  |
| 2.2.1 Spring 标准发布包.....                 | 9  |
| 2.2.2 Spring 相关软件包.....                 | 10 |
| 2.2.3 Spring 自带范例.....                  | 12 |
| 2.3 部署 Spring.....                      | 13 |
| 2.4 Spring 应用初体验.....                   | 14 |
| 2.4.1 Hello World 遇到的问题.....            | 14 |
| 2.4.2 使用 Spring 对 Hello World 进行重构..... | 20 |
| 2.4.3 实现依赖注入 (DI) .....                 | 22 |

|                                                |    |
|------------------------------------------------|----|
| 2.4.4 重构总结 .....                               | 24 |
| 2.4.5 简单的 Web 应用程序 Hello World .....           | 24 |
| 2.5 小结 .....                                   | 26 |
| 第 3 章 贯穿 Spring 应用的核心理念 .....                  | 27 |
| 3.1 轻量级容器再探 .....                              | 27 |
| 3.1.1 容器和轻量级容器 .....                           | 27 |
| 3.1.2 需要容器的理由 .....                            | 28 |
| 3.1.3 轻量级容器的优势 .....                           | 28 |
| 3.2 控制反转 (IoC) .....                           | 28 |
| 3.2.1 一个简单的例子 .....                            | 29 |
| 3.2.2 引入问题 .....                               | 31 |
| 3.2.3 使用控制反转 (Inversion of Control) 模式 .....   | 32 |
| 3.2.4 总结 .....                                 | 33 |
| 3.3 依赖注射 .....                                 | 34 |
| 3.3.1 依赖注射的几种形式 .....                          | 35 |
| 3.3.2 使用 Spring 进行设值方法和构造函数注射 .....            | 35 |
| 3.3.3 设值方法注射和构造函数注射的使用场合 .....                 | 36 |
| 3.4 面向方面编程 (Aspect Oriented Programming) ..... | 37 |
| 3.4.1 AOP 的历史来源 .....                          | 37 |
| 3.4.2 为什么需要 AOP .....                          | 38 |
| 3.4.3 AOP 的重要概念 .....                          | 39 |
| 3.4.4 理解“横切” .....                             | 41 |
| 3.4.5 AOP 的实现策略 .....                          | 43 |
| 3.4.6 认识 AspectJ .....                         | 44 |
| 3.4.7 安装 AspectJ 的运行环境 .....                   | 44 |
| 3.4.8 用 J2SE 动态代理实现 AOP .....                  | 47 |
| 3.4.9 用 AspectJ 实现 AOP .....                   | 51 |
| 3.4.10 深入理解 AspectJ .....                      | 54 |
| 3.5 单元测试 (Unit Test) .....                     | 58 |
| 3.5.1 单元测试和 JUnit 框架 .....                     | 59 |
| 3.5.2 JUnit 框架的 3 个核心概念 .....                  | 60 |
| 3.5.3 准备测试 .....                               | 60 |
| 3.5.4 添加测试代码并进行测试 .....                        | 62 |
| 3.5.5 取得更有意义的出错信息 .....                        | 63 |
| 3.5.6 捕捉错误信息 .....                             | 64 |
| 3.5.7 单元测试技巧 .....                             | 65 |
| 3.6 测试驱动开发 (Test Driven Development) .....     | 67 |
| 3.6.1 传统测试的问题 .....                            | 67 |

|                                                                            |           |
|----------------------------------------------------------------------------|-----------|
| 3.6.2 测试驱动开发的应对策略 .....                                                    | 68        |
| 3.6.3 一个简单的示例 .....                                                        | 68        |
| 3.6.4 总结测试驱动开发的优势 .....                                                    | 73        |
| 3.7 小结 .....                                                               | 74        |
| <b>第4章 模式：Spring 前行的路标 .....</b>                                           | <b>75</b> |
| 4.1 模式（Pattern）入门 .....                                                    | 75        |
| 4.1.1 什么是模式 .....                                                          | 75        |
| 4.1.2 不用模式产生的问题 .....                                                      | 76        |
| 4.1.3 通过实例理解模式本质 .....                                                     | 79        |
| 4.1.4 小结 .....                                                             | 81        |
| 4.2 工厂模式（Design Pattern: Factory Method）的精髓 .....                          | 81        |
| 4.2.1 引入问题 .....                                                           | 82        |
| 4.2.2 解决方法 .....                                                           | 83        |
| 4.2.3 工厂模式和依赖倒置的关系 .....                                                   | 85        |
| 4.3 单例模式（Design Pattern: Singleton） .....                                  | 88        |
| 4.3.1 单例模式的实现 .....                                                        | 88        |
| 4.3.2 单例注册表 .....                                                          | 89        |
| 4.4 模板模式和策略模式（Design Pattern: Template Method And Strategy） .....          | 91        |
| 4.4.1 模板模式 .....                                                           | 92        |
| 4.4.2 策略模式 .....                                                           | 95        |
| 4.5 代理模式（Design Pattern: Proxy） .....                                      | 100       |
| 4.5.1 第一个代理模式的例子 .....                                                     | 100       |
| 4.5.2 虚拟代理（Virtual Proxy） .....                                            | 103       |
| 4.6 数据访问对象模式（J2EE Pattern: Data Access Object） .....                       | 105       |
| 4.7 模型视图控制器模式（Architecture/Framework Pattern: Model View Controller） ..... | 113       |
| 4.7.1 Model 1 和 Model 2 简介 .....                                           | 114       |
| 4.7.2 一个 Model 2 框架示例 .....                                                | 115       |
| 4.8 框架与模式的关系 .....                                                         | 120       |
| 4.9 Spring 和工厂模式 .....                                                     | 121       |
| 4.9.1 Spring 工厂体系的另类视图 .....                                               | 121       |
| 4.9.2 Spring 工厂核心原理 .....                                                  | 122       |
| 4.10 Spring 和单例模式 .....                                                    | 123       |
| 4.11 Spring 的模板以及策略模式 .....                                                | 125       |
| 4.11.1 Spring 模板模式 .....                                                   | 125       |
| 4.11.2 Spring 策略模式 .....                                                   | 126       |
| 4.12 Spring 和代理模式 .....                                                    | 128       |
| 4.13 Spring 和数据访问对象模式 .....                                                | 129       |
| 4.14 Spring 和 MVC 模式 .....                                                 | 131       |