

每一个开发者，都是软件世界里的列兵！
每一项关键技术，都是前进道路上的堡垒！

开发者突击：

精通 AOP

整合应用开发 (AspectWerkz+AspectJ+Spring)



刘中兵
飞思科技产品研发中心

编著
监制



AOP可以改进OOP系统架构、优化系统性能，让我们的代码看起来更加简单、清晰！
用AOP画龙点睛，Java将如虎添翼！

本书突出技术性、深入性、实用性，从内在原理和实际应用两个方面讲解AOP

以理论为基础：AOP是3个框架（AspectWerkz+AspectJ+Spring）的基础，IoC是Spring应用AOP的基础

层层递进：AOP—AspectWerkz—AspectJ—SpringAOP，AspectJ 5是AspectWerkz与AspectJ的合并版本，Spring又是以AspectJ为基础的框架，可以实现与AspectJ的集成

注重实战：在讲解AOP及3个框架（AspectWerkz+AspectJ+Spring）的应用技术时，以实用案例为主进行讲解，让初学者也能够驾轻就熟



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

随书光盘内容为
书中实例源文件

开发者突击:

精通 AOP

整合应用开发

[AspectWerkz+AspectJ+Spring]

刘中兵

飞思科技产品研发中心

编著

监制



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以 AOP 基础理论为主线, 首先讲解 AOP 的产生与发展、为什么要应用 AOP、AOP 的核心概念, 然后详细讲解 AspectWerkz、AspectJ、Spring 框架的 AOP 应用开发技术。

- AspectWerkz: 讲解 AspectWerkz 的 3 种开发方式 (命令行、Eclipse+Ant、Eclipse+AspectWerkz 插件)、两种核心技术 (Aspect 类与关联配置), 并在实际的 Java EE 项目中实现 3 种 AspectWerkz 功能 (日志记录器、性能分析器、权限检查拦截器);
- AspectJ: 讲解在 Eclipse 下开发 AspectJ 的 3 种方式 (命令行、Ant、AJDT)、两种核心技术 (Aspect 类与 Annotation 开发), 并讲解 AspectJ 对 Java 5 新特性的支持, 最后在实际的 Java EE 项目中实现 5 种 AspectJ 功能 (日志记录器、性能分析器、权限检查拦截器、异常监视器、在线用户监视器);
- Spring AOP: 以 IoC 为基础讲解 Spring 下的 AOP 开发, 讲解了 3 种 AOP 的开发方式, 即使用 @Aspect 注释符、基于 Schema 的配置的开发方式和 Spring API 的开发方式, 最后在实际的 Java EE 项目中实现 5 种 Spring AOP 功能 (日志记录器、性能分析器、权限检查拦截器、异常监视器、在线用户监视器), 并讲解 Spring 与 AspectJ 的集成。

随书附赠的光盘内容为本书开发的案例程序包。本书内容循序渐进, 通俗易懂, 覆盖了 AOP 开发的最流行框架技术与应用案例。无论是对于 Java EE 架构设计还是软件开发, 本书都是精通开发 Java Web 应用的必备的实用手册。

本书的读者对象: Java 和 Java EE 的软件架构师与设计师、软件开发工程师、项目及技术经理。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目 (CIP) 数据

开发者突击: 精通 AOP 整合应用开发: AspectWerkz+AspectJ+Spring / 刘中兵编著.

北京: 电子工业出版社, 2008.10

ISBN 978-7-121-07283-3

I. 开… II. 刘… III. Java 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 129368 号

责任编辑: 杨 鹄

印 刷: 北京智力达印刷有限公司

装 订: 北京中新伟业印刷有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×1092 1/16 印张: 29.5 字数: 755.2 千字

印 次: 2008 年 10 月第 1 次印刷

印 数: 5 000 册 定价: 59.00 元 (含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

AOP (Aspect Oriented Programming), 即面向方面编程, 是施乐公司帕洛阿尔托研究中心 (Xerox PARC) 在 20 世纪 90 年代发明的一种编程方式。AOP 是从 OOP 中抽象出来的“方面”的概念, 目的是为了打破对象的封装性。它以“方面”的方式对原有的模块进行重组, 抽取那些与业务无关却为整个系统所通用的功能, 并将其最终封装在一起。

在 Java 的世界里, AOP 已经走向比较成熟的应用。从 AOP 体现的能力上来说, AspectWerkz、AspectJ、Spring 已经渐趋成熟, Dynaop、JAC、JBoss AOP 等也有各自的优势, 它们在日志记录 (Logging)、权限管理 (Authentication)、错误处理 (Error Handling)、事务处理 (Transactions)、持久化 (Persistence) 等方面都取得了很好的应用。

AOP 在企业应用中正逐渐体现其自身的价值, 它的作用更多地是关注于系统的某一方面。AOP 不是对 OOP 的颠覆, 而是对 OOP 的改进与优化, 它能够改进 OOP 系统的以下问题: 代码重复、可读性差、可重用性差。因此本书的目标也是应用 AOP 来改进 OOP 系统架构, 让我们的代码看起来更加简单、清晰。

一、本书特色

本书结合技术性、深入性、实用性, 从内在原理和实际应用两个方面讲解 AOP, 是一本全面的参考书。本书具有如下特点。

- 以理论为基础: AOP 是 3 个框架的基础, IoC 是 Spring 应用 AOP 的基础;
- 层层递进: AOP→AspectWerkz→AspectJ→SpringAOP, AspectJ5 是 AspectWerkz 与 AspectJ 的合并版本, Spring 又是以 AspectJ 为基础的框架, 可以实现与 AspectJ 的集成;
- 实例讲解: 在讲解 AOP 及其他 3 个框架的应用技术时, 处处以实例进行讲解, 让初学者也能够驾轻就熟。

二、本书的组织结构和主要内容

本书的内容总体上分为 4 个部分, 从第一篇 AOP 基础理论开始, 逐步介绍各种 AOP 框架的应用开发技术, 并结合实际的案例进行讲解。各部分内容如下:

1. AOP 基础篇——AOP 的产生与基础理论

- (1) 讲解 AOP 的产生与发展过程, 概述 AOP 能够做什么。
- (2) 通过实例直观地讲解 AOP 的入门知识、应用开发与核心技术, 为应用篇的讲解做好理论准备。

2. AspectWerkz 应用篇——AspectWerkz 技术讲解与 3 个应用案例

- (1) 了解 AspectWerkz 的简介、主要特性和版本发展, 以及安装 AspectWerkz 的开发环境, 包括 JDK、AspectWerkz、Eclipse 与 MyEclipse。
- (2) 通过实例的形式讲解 AspectWerkz 的 3 种开发方式: 使用命令行进行开发、使用 Eclipse+Ant 进行开发、使用 Eclipse+AspectWerkz 插件进行开发, 并演示了在

Eclipse+AspectWerkz 插件方式下开发的一个实际应用。

(3) 讲解 AspectWerkz 核心类——Aspect 方面类的开发, 包括编写 Aspect 方面类、编写 Advice 通知函数、JoinPoint 输入参数。

(4) 讲解 Aspect 类的匹配模式语言、匹配表达式, 并讲解了两种 Aspect 关联方式: Annotation 和 XML 定义方式。

(5) 重点讲解 3 个最具代表性的 AOP 应用的开发, 包括 AOP 日志记录器、AOP 性能分析器、AOP 权限检查拦截器。

3. AspectJ 应用篇——AspectJ 技术讲解与 5 个应用案例

(1) 讲解 AspectJ 的简介、产生与发展, 以及安装 AspectJ 的开发环境并测试安装。

(2) 通过实例的形式讲解 AspectJ 的两种开发方式, 即命令行开发方式和 Ant 开发方式, 包括 4 个命令和 3 种 Ant 任务。

(3) 通过实例讲解如何使用 AJDT 进行方面类的开发, 以及 AJDT 的高级功能, 最后简单讲解了 AspectJ 与 JBuilder、NetBeans、Emacs 和 JDeveloper 的继承插件。

(4) 详细讲解 Aspect 方面类、连接点函数和通知函数的编写方法, 然后讲解使用 Annotation 编写 Aspect 类的方法, 并给出开发的实例。

(5) 讲解 AspectJ 对 Java 5 的新特性的支持, 包括注释符、泛型、自动装箱与拆箱、类型覆盖、可变变量和枚举。

(6) 在 Java EE 实例项目中开发基于 AspectJ 的 AOP 功能, 包括日志记录器、性能分析器、权限检查拦截器、异常监视器、在线用户监视器, 最后形成 AOP 应用案例包 demo_ssh_oa_aspectj.zip。

4. Spring AOP 应用篇——Spring 技术讲解与 5 个应用案例

(1) Spring 与 IoC 入门基础: 讲解 Spring 的 IoC 基础知识, 为 AOP 的讲解做好准备。

(2) Spring 的安装与配置: 讲解搭建 Spring 的开发环境, 并通过实例演示搭建的过程。

(3) AOP 的 3 种方式:

- 第一种方式——基于@Aspect 注解: 对于新的应用程序, 如果用户使用 Java 5 开发, 我们推荐用户使用@Aspect 风格。
- 第二种方式——基于模式 (schema-based) 的方式: 对于不支持 Java 5 的应用开发, 可以使用这种模式。
- 第三种方式——使用 Spring 的 API 开发 AOP。

(4) Spring AOP 在 Java EE 项目中的应用: 在 Java EE 实例项目中开发基于 Spring 的 AOP 功能, 包括日志记录器、性能分析器、权限检查拦截器、异常监视器、在线用户监视器, 最后形成 AOP 应用案例包 demo_ssh_oa_spring.zip。

(5) Spring 与 AspectJ 的集成: 本章基于 demo_ssh_oa_spring.zip 进行修改, 实现 Spring 与 AspectJ 的集成, 最后形成新的软件包 demo_ssh_oa_spring2aspectj.zip。

三、本书范例程序

本书从第 1 部分到第 4 部分逐层深入, 每一个篇章都是先讲解技术然后讲解实际的应

用，在技术的讲解中也贯穿案例程序。

- AspectWerkz 应用范例程序：demo_ssh_oa_aw.zip；
- AspectJ 应用范例程序：demo_ssh_oa_aspectj.zip；
- Spring 应用范例程序：demo_ssh_oa_spring.zip；
- Spring 与 AspectJ 集成应用范例程序：demo_ssh_oa_spring2aspectj.zip。

四、光盘使用说明

本书配套光盘包含以下目录。

1. software 目录

在该目录下仅包含了本书所需要的软件安装包的下载存放目录，并在 software\readme.txt 中详细记录了每一个软件的官方网站、所需版本和该版本的下载链接地址，请读者在学习本书前到指定的网址中去下载各个软件。

说明：考虑到版权问题，我们只提供了这些软件的下载链接地址。这些下载地址在本书写作时都已经经过了验证确认，有些链接也许需要您注册登录后才能够下载。如果下载的连接地址不存在，则表明该网站已经移除了该链接，请您在指定的官方网站上找到下载的页面来下载对应版本的文件。

本书所使用的所有软件的安装程序包括：

- JDK 的安装软件 (JDK 1.5.0)；
- MySQL 服务器的安装软件 (MySQL 5.1.22, SQL-Front 3.3)；
- Tomcat 的安装软件 (Tomcat 5.5.25)；
- Eclipse 的中文插件 (Eclipse 3.1.1)；
- Ant 软件 (Ant 1.7.0)；
- AspectWerkz (AspectWerkz 2.0RC2)；
- Jad 软件 (Jadnt 158)；
- AspectJ (aspectj-1.5.4.jar)；
- Spring (spring-framework-2.0.6.zip)。

2. source 目录

在该目录下包含了本书开发的所有案例程序、系统及数据库，并且还包含了 3 部分源代码。

首先是 Java EE 样例项目。

- demo_ssh_oa.zip：样例项目；
- database.zip：样例项目的 MySQL 数据库文件及其建表 SQL。

下面为 3 个框架的案例程序，以及在样例项目中实现 AOP 后的系统。

(1) AspectWerkz (位于子目录 aw 下)

- AWHelloWorld1.zip：使用命令行方式开发的演示程序；
- AWHelloWorld2.zip：使用 Eclipse+Ant 方式开发的演示程序；
- AWHelloWorld3.zip：使用 Eclipse+AspectWerkz 插件方式开发的演示程序；

- AWHelloWorld.zip: 使用 Eclipse+AspectWerkz 插件方式开发的实际应用程序;
- demo_ssh_oa_aw.zip: 基于 Java EE 项目 demo_ssh_oa.zip 开发 AOP 后的项目包, 实现了日志记录器、性能分析器、权限检查拦截器的 AOP 实现。

(2) AspectJ (位于子目录 aspectj 下)

- AspectJHelloWorld.zip: 使用 AJDT 插件方式开发的实际应用程序;
- AnnotationHelloWorld.zip: 使用 Annotation 开发的实际应用程序;
- demo_ssh_oa_aspectj.zip: 基于 Java EE 项目 demo_ssh_oa.zip 开发 AOP 后的项目包, 实现了日志记录器、性能分析器、权限检查拦截器、异常监视器、在线用户监视器的 AOP 实现。

(3) Spring (位于子目录 spring 下)

- SpringTest.zip: 安装 Spring 后的测试实例。
- SpringAnnotation.zip: 基于 @Aspect 开发的 Spring AOP 实际应用程序;
- SpringSchema.zip: 基于 Schema 开发的 Spring AOP 实际应用程序;
- SpringAPIDemo.zip: 基于 Spring 的 API 开发的 AOP 实际应用程序;
- demo_ssh_oa_spring.zip: 基于 Java EE 项目 demo_ssh_oa.zip 开发 Spring AOP 后的项目包, 实现了日志记录器、性能分析器、权限检查拦截器、异常监视器、在线用户监视器的 Spring AOP 实现;
- demo_ssh_oa_spring2aspectj.zip: 基于 demo_ssh_oa_spring.zip 进行修改, 实现 Spring 与 AspectJ 的集成, 最后形成新的软件包 demo_ssh_oa_spring2aspectj.zip。

提示: 在 source 目录中共提供了 10 个演示程序、4 个实际程序包。请读者按照书中的讲解配置好开发环境, 导入这些项目即可运行。首先将要运行的程序包解压缩到 Eclipse 的工作目录 (如 D:\eclipse\workspace), 单击 Eclipse 左侧的项目工作区空白处, 然后单击鼠标右键, 在弹出的右键菜单中选择“导入”, 再在弹出的导入窗口中选择“从现有项目到工作空间中”, 并选择刚才解压缩后的程序目录, 最后单击“确定”按钮即可导入到 Eclipse 中。

六、致谢

本书在编写过程中得到了 Apache 软件组织和 SUN 公司在技术上的大力支持, 并且由飞思科技产品研发中心负责监制工作, 在此表示衷心的感谢! 尽管我们尽了最大努力, 但本书难免会有不妥之处, 欢迎各界专家和读者朋友批评指正。

编 著 者
飞思科技产品研发中心

联系方式

咨询电话: (010) 68134545 88254160

电子邮件: support@fecit.com.cn

服务网址: <http://www.fecit.com.cn> <http://www.fecit.net>

通用网址: 计算机图书、飞思、飞思教育、飞思科技、FECIT

第 1 部分 AOP 基础篇

在目前的 Java 应用中, 大多数的软件项目都选择了 OOP 的编程方式。的确, OOP 已经表明了它处理一般行为的能力, 但是, OOP 不能很好地处理横越多个 (经常是不相关的) 模块的行为, 相比之下, AOP 填补了这个空白, 它是编程方法学发展的又一个里程碑。

第 1 章 AOP 的产生 3

软件开发经过了结构化(SA)、面向对象(OO)与设计模式(Design Pattern)、组件化(Component)、面向方面(AOP)直至面向服务(SOA)这条技术发展之路, AOP 是为了解决业务核心代码与日志、权限、事务等通用功能代码混杂的问题而产生的。

1.1 软件开发方法的演进 3

1.1.1 结构化方法 SP (Structured Programming) 3

1.1.2 面向数据结构的软件开发方法 DOP (Data-Oriented Programming) 4

1.1.3 面向问题的分析法 PAM (Problem Analysis Method) 5

1.1.4 面向对象编程 OOP (Object-Oriented Programming) 5

1.1.5 面向界面的可视化开发方法 7

1.1.6 组件化编程 CP (Component Programming) 8

1.1.7 面向方面编程 AOP (Aspect-Oriented Programming) 8

1.1.8 面向服务架构 SOA (Service-Oriented Architecture) 9

1.2 AOP 产生的前奏——把系统看做一批关注点 11

1.2.1 拆分关注点 11

1.2.2 拆分需求 12

1.3 AOP 产生的理由——传统方法存在的问题 13

1.3.1 多维需求一维实现 13

1.3.2 用传统方式实现 13

1.3.3 传统方式带来的问题 14

1.3.4 传统方式导致的结果 14

1.3.5 现有的 3 种解决办法 14

1.3.6 AOP 能更好地解决问题 15

1.4 下一个主题: AOP 基础理论 15

第 2 章 AOP 基础理论 17

通过实例直观地讲解 AOP 的入门知识、应用开发与核心技术, 为应用篇讲解 AspectWerkz、AspectJ、Spring AOP 做好理论准备。

2.1 AOP 入门知识 17

2.1.1 AOP 简介 17

2.1.2 AOP 是设计模式的延续 18

2.1.3 第一个 AOP 程序 HelloWorld 18

2.1.4 使用 AOP 的优点 19

2.1.5 使用 AOP 需要考虑的问题 20

2.2 AOP 应用开发 21

2.2.1	AOP 的开发过程.....	21
2.2.2	AOP 的实现语言.....	22
2.2.3	AOP 的应用范围.....	22
2.2.4	AOP 最佳实践——为 HelloWorld 添加日志功能.....	23
2.2.5	AOP 最佳实践——为 HelloWorld 添加权限控制功能.....	24
2.3	AOP 核心技术.....	24
2.3.1	AOP 名词解释.....	25
2.3.2	AOP 通知类型.....	26
2.3.3	AOP 织入过程.....	26
2.3.4	AOP 织入策略.....	27
2.3.5	AOP 织入时机.....	28
2.4	AOP 开源框架.....	30
2.4.1	AspectWerkz.....	31
2.4.2	AspectJ.....	31
2.4.3	Spring.....	32
2.4.4	其他 AOP 框架.....	33
2.5	下一个主题: AspectWerkz 应用篇.....	33

第 2 部分 AspectWerkz 应用篇

AspectWerkz 是一个基于 Java 的简单易用、功能强大、免费开源的 AOP 框架，既强大又简单，有助于用户更容易地使 AOP 集成到新的或已存在的项目中。AspectWerkz 在 Eclipse 下也有辅助的开发插件。

第 3 章 AspectWerkz 入门基础.....37

从本章开始，我们将逐步揭开 AspectWerkz 的神秘面纱，了解 AspectWerkz 概要、主要特性和版本发展，安装 AspectWerkz 的开发环境，包括 JDK、AspectWerkz、Eclipse 与 MyEclipse。

3.1	认识 AspectWerkz	37
3.1.1	AspectWerkz 简介.....	38
3.1.2	AspectWerkz 主要特性.....	39
3.1.3	AspectWerkz 版本发展.....	40
3.2	安装 AspectWerkz 的开发环境	45
3.2.1	安装与配置 JDK	46
3.2.2	安装与配置 AspectWerkz.....	47
3.2.3	安装与配置 Eclipse.....	48
3.2.4	安装 MyEclipse	50
3.3	下一个主题: AspectWerkz 应用开发详解.....	52

第 4 章 AspectWerkz 应用开发详解.....53

AspectWerkz 有 3 种开发方式：使用命令行进行开发、使用 Eclipse+Ant 进行开发、使用 Eclipse+AspectWerkz 插件进行开发，这 3 种方式从简单到复杂，从原始到自动化，越来越容易开发 AOP。

4.1	第 1 种：使用命令行进行开发——实例演示 AWHelloWorld1	54
4.1.1	新建目标类 com.demo.HelloWorld.java	55
4.1.2	新建方面类 com.demo.MyAspect.java	56
4.1.3	定义 XML 关联文件 aop.xml	56
4.1.4	织入与运行	57
4.1.5	编写 Annotation 方面类 com.demo.MyAspectWithAnnotations.java	58
4.1.6	编写 Annotation 的 XML 关联文件 aopWithAnnotations.xml	60
4.1.7	织入与运行 Annotation 的类	60
4.2	第 2 种：使用 Eclipse+Ant 进行开发——实例演示 AWHelloWorld2	62
4.2.1	编写 Ant 的构建文件 build.xml	63
4.2.2	编写属性文件 aw.properties	67
4.2.3	从 AWHelloWorld1.zip 中添加类代码	68
4.2.4	运行构建任务	68
4.2.5	程序的调试	70
4.3	第 3 种：使用 Eclipse+AspectWerkz 插件进行开发—— 实例演示 AWHelloWorld3	71
4.3.1	安装 AspectWerkz 插件	71
4.3.2	从 AWHelloWorld1.zip 中添加类代码	72
4.3.3	运行 AOP 程序	73
4.3.4	Aspect 查看器	74
4.3.5	编写带 Java 5 注释符的方面类	75
4.4	织入方面类前后的对比	77
4.4.1	下载并安装反编译工具	77
4.4.2	织入前的 HelloWorld.class 反编译结果	77
4.4.3	织入后的 HelloWorld.class 反编译结果	78
4.5	实际应用案例 AWHelloWorld	80
4.5.1	新建目标类 com.demo.HelloWorld.java	80
4.5.2	新建方面类 com.demo.HelloWorldHijacker.java	81
4.5.3	新建 XML 文件 META-INF/aop.xml	81
4.6	下一个主题：开发 Aspect 核心类	82
第 5 章	AspectWerkz 核心技术——开发 Aspect 核心类	83
	AspectWerkz 应用开发中的重要元素——Aspect 方面类,它是 AspectWerkz 应用开发的核心元素。 Aspect 方面类的开发,包括编写 Aspect 方面类、编写 Advice 通知函数、JoinPoint 输入参数。	
5.1	编写 Aspect 方面类	83
5.1.1	Aspect 类的修饰类型	84
5.1.2	Aspect 类的构造函数	84
5.1.3	Aspect 类的继承	85
5.2	编写 Advice 通知函数	86
5.2.1	Advice 函数的修饰类型	86

5.2.2 通知的类型和 Advice 函数形式	87
5.3 JoinPoint 输入参数	88
5.3.1 JoinPoint 类型的输入参数	88
5.3.2 StaticJoinPoint 类型的输入参数	89
5.4 下一个主题：关联 Aspect 核心类	90
第 6 章 AspectWerkz 核心技术——关联 Aspect 核心类	91
开发了 Aspect 核心类，要让这些 Aspect 类发挥作用，就需要通过一定的方法来将 Aspect 类与目标类关联起来，那么如何将 Aspect 类与众多的 Java 目标类进行关联呢？AspectWerkz 为我们提供了标准的匹配表达式规范，用来将 Aspect 类与 Java 目标类进行匹配，可以将不同的 Aspect 类与不同的 Java 目标类进行关联。	
6.1 匹配模式语言	92
6.1.1 通配符	92
6.1.2 类的匹配	92
6.1.3 方法的匹配	93
6.1.4 构造函数的匹配	94
6.1.5 变量的匹配	95
6.1.6 子类的匹配	95
6.1.7 数组的匹配	96
6.2 匹配表达式	96
6.2.1 匹配函数	96
6.2.2 匹配运算符	98
6.2.3 如何引用匹配表达式	98
6.3 第 1 种关联方式：使用 Annotation 定义关联	99
6.3.1 使用 @Aspect 设置 Aspect 部署模式	100
6.3.2 使用 @Expression 自定义一个表达式	100
6.3.3 设置 Advice 通知函数	102
6.3.4 在 XML 中申明 Aspect 类	105
6.4 第 2 种关联方式：使用 XML 文件定义关联	105
6.4.1 使用 <aspect> 定义 Aspect 类	106
6.4.2 使用 <pointcut> 定义匹配表达式	106
6.4.3 使用 <advice> 定义 Advice 通知函数	107
6.4.4 使用 <param> 传递参数到 Aspect 类中	109
6.4.5 使用 <package> 定义包	109
6.4.6 使用 <exclude> 和 <include> 设置包的范围	110
6.4.7 定义 <system>	110
6.4.8 添加 DTD	111
6.4.9 完整的 XML 样例	111
6.4.10 XML 命名方式	112
6.5 下一个主题：AspectWerkz 在 Java EE 项目中的应用	112

第 7 章 AspectWerkz 在 Java EE 项目中的应用	113
在 Java EE 实例项目中开发基于 AspectWerkz 的 AOP 功能, 包括日志记录器、性能分析器、权限检查拦截器, 最后形成 AOP 应用案例包 demo_ssh_oa_aw.zip。	
7.1 安装 Java EE 项目的开发环境	114
7.1.1 安装并运行数据库 MySQL	114
7.1.2 安装 Tomcat	116
7.1.3 配置 Tomcat 与 Eclipse 的集成	119
7.2 运行 Java EE 项目	120
7.2.1 将项目导入到 Eclipse 中	121
7.2.2 发布项目到 Tomcat 中并启动 Tomcat	124
7.2.3 登录该项目的系统界面	126
7.3 开发前的准备工作	127
7.3.1 分析项目包结构	127
7.3.2 添加 AspectWerkz 库	129
7.4 应用一: 开发 AOP 日志记录器	129
7.4.1 创建日志记录器类 Logger.java	130
7.4.2 在 aop.xml 中配置该日志记录器	133
7.4.3 织入日志记录器到 Action 类中	133
7.4.4 发布并运行项目查看运行结果	136
7.5 应用二: 开发 AOP 性能分析器	137
7.5.1 创建性能分析器类 Timer.java	138
7.5.2 在 aop.xml 中配置该性能分析器	139
7.5.3 织入性能分析器到 Action 类中	140
7.5.4 发布并运行项目查看运行结果	142
7.6 应用三: 开发 AOP 权限检查拦截器	143
7.6.1 创建权限检查拦截器类 AuthorizationInterceptor.java	143
7.6.2 在 aop.xml 中配置该权限检查拦截器	145
7.6.3 织入权限检查拦截器到 Action 类中	146
7.6.4 发布并运行项目查看运行结果	147
7.7 程序结构图	148
7.8 下一个主题: AspectJ 应用篇	149

第 3 部分 AspectJ 应用篇

AspectJ 是 AOP 的 Java 语言的实现, 它为 Java 语言增添了新的语法规则。初次接触 AspectJ 的读者看到这些资料 (或者语法手册), 会感到 AspectJ 有些神秘。AspectJ 是对 Java 程序的无缝的、面向方面的扩展, 与原有的 Java 程序完全兼容, 并且较容易学习和使用。

第 8 章 AspectJ 入门基础	153
目前最成熟、功能最丰富的 AOP 框架当数 AspectJ, AspectJ 已成为大多数框架跟从的标准。首先我们需要了解 AspectJ 为何物, 以及 AspectJ 的安装与配置过程。	
8.1 AspectJ 概述	153
8.1.1 AspectJ 简介	154
8.1.2 AspectJ 的产生	154
8.1.3 AspectJ 的版本发展	155
8.2 AspectJ 安装与配置	155
8.2.1 下载 AspectJ 安装程序	156
8.2.2 安装 AspectJ	157
8.2.3 认识 AspectJ 安装后的目录	161
8.2.4 配置 AspectJ	161
8.2.5 测试 AspectJ 环境	162
8.3 下一个主题: AspectJ 应用开发详解	162
第 9 章 AspectJ 应用开发详解	163
AspectJ 提供了两种开发与编译的方式, 第一种方式是使用 AspectJ 自带的脚本命令, 第二种方式是使用 AspectJ 安装包中提供的 Ant 构建任务。	
9.1 使用 AspectJ 命令行开发	164
9.1.1 类路径的属性变量	164
9.1.2 ajc——编译时和编译后织入类	165
9.1.3 aj——加载时织入类	171
9.1.4 ajdoc——生成 Javadoc 文档	172
9.1.5 ajbrowser——AspectJ 浏览器	174
9.2 使用 Ant 任务进行开发	179
9.2.1 AjcTask(ajc)——执行类的编译与运行	179
9.2.2 Ajc11CompilerAdapter(javac)——切换 javac 的编译器执行编译	186
9.2.3 Ajc10(ajc)——执行类的编译与运行	189
9.3 下一个主题: 在 Eclipse 中使用 AJDT 开发 AspectJ 应用	191
第 10 章 在 Eclipse 中使用 AJDT 开发 AspectJ 应用	193
前面讲解的使用命令行和使用 Ant 工具进行开发的两种方式, 都是手工操作的方式。为了进行更高效的可视化开发, AspectJ 同时提供了与各种 IDE 软件的集成插件, 包括与 Eclipse、JBuilder、NetBeans、Emacs 和 JDeveloper 的集成, AspectJ 与 Eclipse 的集成是比较流行的开发方式。	
10.1 AJDT 下载与安装	194
10.1.1 AJDT 简介	194
10.1.2 AJDT 的发布版本	195
10.1.3 AJDT 的下载安装包	195
10.1.4 AJDT 的两种安装方式	197
10.2 AJDT 应用开发详解——开发实例 AspectJHelloWorld	200
10.2.1 创建 AspectJ 项目的两种方式	201

10.2.2	配置 AspectJ 项目 AspectJHelloWorld.....	205
10.2.3	新建一个目标类 HelloWorld.java	207
10.2.4	新建一个方面类 MyAspect.aj	207
10.2.5	在 MyAspect.aj 中添加通知函数	209
10.2.6	对比 HelloWorld.class 织入前后的不同	210
10.2.7	运行目标类 HelloWorld.java 查看结果	212
10.3	AJDT 的高级功能	213
10.3.1	目标类视图——查看 HelloWorld.java	213
10.3.2	方面类视图——查看 MyAspect.aj	214
10.3.3	AspectJ 图形化视图	215
10.4	AspectJ 与其他 IDE 的集成.....	216
10.4.1	AspectJ 与 JBuilder 的集成插件	216
10.4.2	AspectJ 与 NetBeans 的集成插件	217
10.4.3	AspectJ 与 Emacs and JDEE 的集成插件	218
10.4.4	jdeveloperaop 插件	219
10.5	下一个主题: AspectJ 核心语法详解	220
第 11 章	AspectJ 核心语法详解.....	221
	根据前面的讲解, 我们已经对 AspectJ 的开发过程有了直观了解, 并了解到 AspectJ 应用开发中的重要元素——Aspect 方面类, 它是 AspectJ 应用开发的核心元素。本章就来讲解 AspectJ 核心类——Aspect 方面类的开发。	
11.1	编写 Aspect 方面类.....	222
11.1.1	Aspect 类的定义	222
11.1.2	Aspect 类的继承	223
11.1.3	Aspect 类的抽象	223
11.2	编写 Aspect 连接点函数.....	224
11.2.1	定义连接点函数.....	224
11.2.2	匹配表达式的函数类型	225
11.2.3	匹配表达式的通配符.....	228
11.2.4	匹配表达式的组合	229
11.2.5	在连接点中传递上下文参数.....	229
11.3	编写 Aspect 通知函数.....	230
11.3.1	通知函数的语法格式.....	231
11.3.2	通知函数的 5 种类型	231
11.3.3	通知函数的异常处理.....	234
11.3.4	使用 JointPoint 访问上下文信息.....	235
11.4	使用 Annotation 编写 Aspect 类.....	236
11.4.1	定义 Aspect 的注释符 @Aspect.....	236
11.4.2	定义连接点的注释符 @Pointcut.....	237
11.4.3	定义通知的 5 种注释符.....	238

11.4.4	定义声明的 3 种注释符.....	241
11.5	Annotation 开发实例——AnnotationHelloWorld	242
11.5.1	创建项目 AnnotationHelloWorld.....	242
11.5.2	使用@Aspect 修饰方面类 MyAspect.aj.....	243
11.5.3	使用@Pointcut 修饰连接点函数 methods().....	245
11.5.4	使用@Before 修饰通知函数 before().....	245
11.5.5	使用@Around 修饰通知函数 around()	246
11.5.6	使用@After 修饰通知函数 after().....	247
11.5.7	运行目标类 HelloWorld.java 查看结果	248
11.6	下一个主题: AspectJ 对 Java 5 新特性的支持	249
第 12 章	AspectJ 对 Java 5 新特性的支持	251
	AspectJ 为了支持 Java5 的新特性,进行了许多改进,包括注释、泛型、自动装箱与拆箱、类型覆盖、可变变量和枚举。	
12.1	使用 Java 5 注释符 (Annotation)	251
12.1.1	Java 5 中的注释符.....	251
12.1.2	在 Aspect 中使用 Java 5 注释符.....	253
12.1.3	在匹配表达式中匹配 Java 5 注释符	255
12.1.4	为目标代码注入 Java 5 注释符	257
12.1.5	为目标函数注入错误信息.....	258
12.1.6	为目标类设置继承类和接口	258
12.1.7	设置 Aspect 类的执行顺序.....	258
12.2	使用 Java 5 泛型 (Generics)	259
12.2.1	Java 5 中的泛型知识.....	259
12.2.2	在 Aspect 中应用泛型.....	261
12.2.3	为目标类设置参数化的继承类和接口	264
12.2.4	定义参数化的 Aspect 类.....	264
12.3	使用 Java 5 自动装箱 (AutoBoxing) 与 拆箱 (Unboxing)	265
12.3.1	Java 5 中的自动装箱.....	265
12.3.2	Java 5 中的自动拆箱.....	266
12.3.3	在 Aspect 中使用装箱与拆箱.....	266
12.4	Java 5 中的类型覆盖 (Covariance)	267
12.4.1	Java 5 中的类型覆盖是什么.....	267
12.4.2	在 Aspect 中进行匹配.....	268
12.5	使用 Java 5 可变变量类型 (Varargs)	268
12.5.1	Java 5 中的可变变量类型.....	268
12.5.2	在 Aspect 中匹配可变变量类型.....	269
12.6	使用 Java 5 枚举类型 (Enum)	270
12.6.1	Java 5 中的枚举类型.....	270
12.6.2	在 Aspect 中使用枚举类型.....	273

12.7 下一个主题: AspectJ 在 Java EE 项目中的应用	273
第 13 章 AspectJ 在 Java EE 项目中的应用	275
在 Java EE 实例项目中开发基于 AspectJ 的 AOP 功能, 包括日志记录器、性能分析器、权限检查拦截器、异常监视器、在线用户监视器, 最后形成 AOP 应用案例包 demo_ssh_oa_aspectj.zip。	
13.1 开发前的准备工作	276
13.1.1 运行该 Java EE 项目	276
13.1.2 分析项目包结构	276
13.1.3 添加 AspectJ 库	278
13.1.4 将项目转换为 AspectJ 项目	278
13.2 应用一: AOP 日志记录器	279
13.2.1 创建日志记录器类 Logger.aj	280
13.2.2 记录用户登录信息	281
13.2.3 记录用户退出信息	283
13.2.4 记录用户操作信息	285
13.2.5 查看 Logger.aj 连接点导航图	287
13.2.6 发布并运行项目, 查看运行结果	289
13.3 应用二: AOP 性能分析器	290
13.3.1 创建性能分析器类 Timer.aj	290
13.3.2 记录用户操作时间信息	292
13.3.3 查看 Timer.aj 连接点导航图	294
13.3.4 发布并运行项目, 查看运行结果	295
13.4 应用三: AOP 权限检查拦截器	297
13.4.1 创建权限检查拦截器类 AuthorizationInterceptor.aj	297
13.4.2 验证用户的操作	299
13.4.3 查看 AuthorizationInterceptor.aj 连接点导航图	302
13.4.4 发布并运行项目, 查看运行结果	303
13.5 应用四: AOP 异常监视器	305
13.5.1 创建异常监视器类 ExceptionMonitor.aj	305
13.5.2 捕捉用户操作的异常	307
13.5.3 查看 ExceptionMonitor.aj 连接点导航图	309
13.5.4 发布并运行项目, 查看运行结果	310
13.6 应用五: AOP 在线用户监视器	311
13.6.1 创建在线用户监视器类 OnlineUser.aj	311
13.6.2 监视用户的操作	313
13.6.3 查看 OnlineUser.aj 连接点导航图	315
13.6.4 发布并运行项目, 查看运行结果	317
13.7 程序结构图	317
13.8 下一个主题: Spring AOP 应用篇	318

第4部分 Spring AOP 应用篇

Spring 框架中目前最吸引人的也是该应用框架最具特色的地方就是名为控制反转 (IoC, Inverse of Control) 或者依赖注入 (DI, Dependence Injection) 的设计思想, 这是一种相当优秀的设计思想, 即“好莱坞”原则: 不用你主动来找我, 我会通知你。但是, 仅仅凭借着这样一个单纯的设计模式并不能使得 Spring 如此成功, Spring 最成功的地方, 还是目前使用最为广泛的 AOP 应用, 也就是 Spring 中基于 AOP 实现的业务管理机制, 也正是由于这一点, 使得 Spring AOP 成为应用框架中极其闪光的一个亮点。

第14章 Spring 与 IoC 入门基础 321

Spring 是一个轻量的控制反转和面向切面的容器框架。Spring 框架所提供的众多功能之所以能成为一个整体, 正是建立在 IoC 的基础之上, 当然本书要讲解的 AOP 也是构建在 IoC 基础之上的, 因此对这一内涵简单、外延丰富的技术我们有必要进行详细介绍。

14.1 了解 Spring 框架 321

14.1.1 Spring 的产生与发展 321

14.1.2 Spring 框架结构图 322

14.1.3 Spring 组件包详解 323

14.2 了解 IoC 控制反转 326

14.2.1 IoC 与 DI 的基本概念 327

14.2.2 IoC 模式的实例讲解 327

14.2.3 利用 Java 反射机制实现 IoC 容器 332

14.2.4 Spring 中 IoC 容器的实现方式 338

14.2.5 IoC 模式的优缺点 338

14.3 依赖注入的 3 种类型 340

14.3.1 Type1 接口注入 340

14.3.2 Type2 构造注入 341

14.3.3 Type3 设值注入 341

14.3.4 3 种类型对比总结 342

14.4 Spring 的 IoC 容器装载机制 343

14.4.1 BeanFactory 的使用步骤 344

14.4.2 配置 <bean> 的属性 346

14.4.3 配置 <bean> 的子元素 350

14.4.4 ApplicationContext 354

14.5 下一个主题: Spring 的安装与配置 357

第15章 Spring 的安装与配置 359

要进行基于 Spring 的功能开发, 就需要添加 Spring 的开发支持环境。所谓的 Spring 支持环境, 就是在 Java Web 应用中所添加的 Spring 所需要的 jar、tld、xml 文件, 有了这些文件我们就可以开发 Spring 相关的功能代码。

15.1 使用 MyEclipse 自动添加 Spring 360

15.1.1 使用 MyEclipse 添加 Spring 开发环境 360