

<<开发者突击>>

图书基本信息

书名：<<开发者突击>>

13位ISBN编号：9787121072833

10位ISBN编号：7121072831

出版时间：2008-10

出版时间：电子工业出版社

作者：刘中兵 著

页数：449

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<开发者突击>>

前言

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, AspectWcrkz、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。

<<开发者突击>>

内容概要

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，然后再详细讲解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Java5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），共讲解Spring与AspectJ的集成。

随书附赠的光盘内容为《开发者突击：精通AOP整合应用开发（AspectWerkz+AspectJ+Spring）》开发的案程序包。

《开发者突击：精通AOP整合应用开发（AspectWerkz+AspectJ+Spring）》内容循序渐进，通俗易懂，覆盖了AOP外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，《开发者突击：精通AOP整合应用开发（AspectWerkz+AspectJ+Spring）》都是精通开发Java Web应用的必备的实用手册。

《开发者突击：精通AOP整合应用开发（AspectWerkz+AspectJ+Spring）》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、项目及技术经理参考。

书籍目录

第1部分 AOP基础篇第1章 AOP的产生1.1 软件开发方法的演进1.1.1 结构化方法SP (Structured Programming) 1.1.2 面向数据结构的软件开发方法DOP (Data-Oriented Programming) 1.1.3 面向问题的分析法PAM (Problem Analysis Method) 1.1.4 面向对象编程OOP (Object-Oriented Programming) 1.1.5 面向界面的可视化开发方法1.1.6 组件化编程CP (Component Programming) 1.1.7 面向方面编程AOP (Aspect-Oriented Programming) 1.1.8 面向服务架构SOA (Service-Oriented Architecture) 1.2 AOP产生的前奏——把系统看做一批关注点1.2.1 拆分关注点1.2.2 拆分需求1.3 AOP产生的理由——传统方法存在的问题1.3.1 多维需求一维实现1.3.2 用传统方式实现1.3.3 传统方式带来的问题1.3.4 传统方式导致的结果1.3.5 现有的3种解决办法1.3.6 AOP能更好地解决问题1.4 下一个主题：AOP基础理论第2章 AOP基础理论2.1 AOP入门知识2.1.1 AOP简介2.1.2 AOP是设计模式的延续2.1.3 第一个AOP程序HelloWorld2.1.4 使用AOP的优点2.1.5 使用AOP需要考虑的问题2.2 AOP应用开发2.2.1 AOP的开发过程2.2.2 AOP的实现语言2.2.3 AOP的应用范围2.2.4 AOP最佳实践——为HelloWorld添加日志功能2.2.5 AOP最佳实践——为HelloWorld添加权限控制功能2.3 AOP核心技术2.3.1 AOP名词解释2.3.2 AOP通知类型2.3.3 AOP织入过程2.3.4 AOP织入策略2.3.5 AOP织入时机2.4 AOP开源框架2.4.1 AspectWerkz2.4.2 AspectJ2.4.3 Spring2.4.4 其他AOP框架2.5 下一个主题：AspectWerkz应用篇第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AspectWerkz3.1.1 AspectWerkz简介3.1.2 AspectWerkz主要特性3.1.3 AspectWerkz版本发展3.2 安装AspectWerkz的开发环境3.2.1 安装与配置JDK3.2.2 安装与配置AspectWerkz3.2.3 安装与配置Eclipse3.2.4 安装MyEclipse3.3 下一个主题：AspectWerkz应用开发详解第4章 AspectWerkz应用开发详解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AWHelloWorld4.1.1 新建目标

类com.demo.HelloWorld.java4.1.2 新建方面类com.demo.MyAspect.java4.1.3 定义XML关联文件aop.xml4.1.4 织入与运行4.1.5 编写Annotation方面类com.demo.MyAspectWithAnnotations.java4.1.6 编写Annotation的XML关联文件aopWithAnnotations.xml4.1.7 织入与运行Annotation的类4.2 第2种：使用Eclipse+Ant进行开发——实例演示AWHelloWorld4.2.1 编写Ant的构建文件build.xml4.2.2 编写属性文件aw.properties4.2.3 从AWHelloWorld1.zip中添加类代码4.2.4 运行构建任务4.2.5 程序的调试4.3 第3种：使用Eclipse+AspectWerkz插件进行开发——实例演示AWHelloWorld4.3.1 安装AspectWerkz插件4.3.2 从AWHelloWorld1.zip中添加类代码4.3.3 运行AOP程序4.3.4 Aspect查看器4.3.5 编写带Java 注释符的方面类4.4 织入方面类前后的对比4.4.1 下载并安装反编译工具4.4.2 织入前的HelloWorld.class反编译结果4.4.3 织入后的HelloWorld.class反编译结果4.5 实际应用案例AWHelloWorld4.5.1 新建目标

类com.demo.HelloWorld.java4.5.2 新建方面类com.demo.HelloWorldHijacker.java4.5.3 新建XML文件META-INF/aop.xml4.6 下一个主题：开发Aspect核心类第5章 AspectWerkz核心技术——开发Aspect核心类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5.1.1 Aspect类的修饰类型5.1.2 Aspect类的构造函数5.1.3 Aspect类的继承5.2 编写Advice通知函数5.2.1 Advice函数的修饰类型5.2.2 通知的类型和Advice函数形式5.3 JoinPoint输入参数5.3.1 JoinPoint类型的输入参数5.3.2 StaticJoinPoint类型的输入参数5.4 下一个主题：关联Aspect核心类第6章 AspectWerkz核心技术——关联Aspect核心类开发了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？

<<开发者突击>>

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6.1.1 通配符6.1.2 类的匹配6.1.3 方法的匹配6.1.4 构造函数的匹配6.1.5 变量的匹配6.1.6 子类的匹配6.1.7 数组的匹配6.2 匹配表达式6.2.1 匹配函数6.2.2 匹配运算符6.2.3 如何引用匹配表达式6.3 第1种关联方式：使用Annotation定义关联6.3.1 使用@Aspect设置Aspect部署模式6.3.2 使用@Expression自定义一个表达式6.3.3 设置Advice通知函数6.3.4 在XML中申明Aspect类6.4 第2种关联方式：使用XML文件定义关联6.4.1 使用定义Aspect类6.4.2 使用定义匹配表达式6.4.3 使用定义Advice通知函数6.4.4 使用传递参数到Aspect类中6.4.5 使用定义包6.4.6 使用和设置包的范围6.4.7 定义6.4.8 添加DTD6.4.9 完整的XML样例6.4.10 XML命名方式6.5 下一个主题：AspectWerkz在Java EE项目中的应用第7章 AspectWerkz在Java EE项目中的应用在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7.1.1 安装并运行数据库MySQL7.1.2 安装Tomcat7.1.3 配置Tomcat与Eclipse的集成7.2 运行Java EE项目7.2.1 将项目导入到Eclipse中7.2.2 发布项目到Tomcat中并启动Tomcat7.2.3 登录该项目的系统界面7.3 开发前的准备工作7.3.1 分析项目包结构7.3.2 添加AspectWerkz库7.4 应用一：开发AOP日志记录器7.4.1 创建日志记录器类Logger.java7.4.2 在aop.xml中配置该日志记录器7.4.3 织入日志记录器到Action类中7.4.4 发布并运行项目查看运行结果7.5 应用二：开发AOP性能分析器7.5.1 创建性能分析器类Timer.java7.5.2 在aop.xml中配置该性能分析器7.5.3 织入性能分析器到Action类中7.5.4 发布并运行项目查看运行结果7.6 应用三：开发AOP权限检查拦截器7.6.1 创建权限检查拦截器类AuthorizationInterceptor.java7.6.2 在aop.xml中配置该权限检查拦截器7.6.3 织入权限检查拦截器到Action类中7.6.4 发布并运行项目查看运行结果7.7 程序结构图7.8 下一个主题：AspectJ应用篇第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8.1.1 AspectJ简介8.1.2 AspectJ的产生8.1.3 AspectJ的版本发展8.2 AspectJ安装与配置8.2.1 下载AspectJ安装程序8.2.2 安装AspectJ8.2.3 认识AspectJ安装后的目录8.2.4 配置AspectJ8.2.5 测试AspectJ环境8.3 下一个主题：AspectJ应用开发详解第9章 AspectJ应用开发详解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9.1.1 类路径的属性变量9.1.2 ajc——编译时和编译后织入类9.1.3 aj——加载时织入类9.1.4 ajdoc——生成Javadoc文档9.1.5 ajbrowser——AspectJ浏览器9.2 使用Ant任务进行开发9.2.1 AjcTask(iajc)——执行类的编译与运行9.2.2 Ajc11CompilerAdapter(javac)——切换javac的编译器执行编译9.2.3 Ajc10(ajc)——执行类的编译与运行9.3 下一个主题：在Eclipse中使用AJDT开发 AspectJ应用第10章 在Eclipse中使用AJDT开发AspectJ应用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10.1.1 AJDT简介10.1.2 AJDT的发布版本10.1.3 AJDT的下载安装包10.1.4 AJDT的两种安装方式10.2 AJDT应用开发详解——开发实例AspectJHelloWorld10.2.1 创建AspectJ项目的两种方式10.2.2 配置AspectJ项目AspectJHelloWorld10.2.3 新建一个目标类HelloWorld.java10.2.4 新建一个方面类MyAspect.aj10.2.5 在MyAspect.aj中添加通知函数10.2.6 对比HelloWorld.class织入前后的不同10.2.7 运行目标类HelloWorld.java查看结果10.3 AJDT的高级功能10.3.1 目标类视图——查看HelloWorld.java10.3.2 方面类视图——查看MyAspect.aj10.3.3 AspectJ图形化视图10.4 AspectJ与其他IDE的集成10.4.1 AspectJ与JBuilder的集成插件10.4.2 AspectJ与NetBeans的集成插件10.4.3 AspectJ与Emacs and JDEE的集成插

<<开发者突击>>

件10.4.4 jdeveloperaop插件10.5 下一个主题：AspectJ核心语法详解第11章 AspectJ核心语法详解根据前面的讲解，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11.1.1 Aspect类的定义11.1.2 Aspect类的继承11.1.3 Aspect类的抽象11.2 编写Aspect连接点函数11.2.1 定义连接点函数11.2.2 匹配表达式的函数类型11.2.3 匹配表达式的通配符11.2.4 匹配表达式的组合11.2.5 在连接点中传递上下文参数11.3 编写Aspect通知函数11.3.1 通知函数的语法格式11.3.2 通知函数的5种类型11.3.3 通知函数的异常处理11.3.4 使用JointPoint访问上下文信息11.4 使用Annotation编写Aspect类11.4.1 定义Aspect的注释符@Aspect11.4.2 定义连接点的注释符@Pointcut11.4.3 定义通知的5种注释符11.4.4 定义声明的3种注释符11.5 Annotation开发实例——AnnotationHelloWorld11.5.1 创建项目AnnotationHelloWorld11.5.2 使用@Aspect修饰方面类MyAspect.aj11.5.3 使用@Pointcut修饰连接点函数methods() 11.5.4 使用@Before修饰通知函数before() 11.5.5 使用@Around修饰通知函数around() 11.5.6 使用@After修饰通知函数after() 11.5.7 运行目标类HelloWorld.java查看结果11.6 下一个主题：AspectJ对Java 新特性的支持第12章 AspectJ对Javac5新特性的支持第13章 AspectJ在JavacEE项目中的应用第4部分 SpringAOP应用篇第14章 Spring与IoC入门基础第15章 Spring的安装与配置第16章 基于@Aspect注释符的AOP开发第17章 基于Schema的AOP开发第18章 基于SpringAPI的AOP开发第19章 SpringAOP在JavacEE项目中的应用第20章 Spring与AspectJ的集成

<<开发者突击>>

章节摘录

AOP是本书中要讲解的框架的核心，因此本部分将首先对AOP进行引入，让读者了解AOP的产生与发展。我们为什么要应用AOP, AOP的核心技术等。

我们将分2章进行讲解。

AOP的产生：分析由结构化开发面向数据结构的开发，面向问题的分析方法，面向对象编程、可视化编程，组件化编程到AOP, SOA的发展历程，并进一步分析AOP产生的实际需求。

由此让我们了解到，AOP是为了更好地解决OOP中需求混合交织的问题而出现的； AOP基础理论：通过实例直观地讲解AOP的入门知识，应用开发与核心技术，并引入几个AOP的重要框架

—AspectWerkz、AspectJ, Spring，为应用篇的讲解做好理论准备。

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，我们一会儿会看到（或许你已经感觉到了），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。

本章接下来就从软件开发方法的演进讲起，概述不同的软件开发方法所能解决的不同侧面的问题。

然后根据现有方法所存在的问题，寻找到解决方案——那就是AOP。

<<开发者突击>>

编辑推荐

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个框架的基础，IOC是spring应用AOP的基础，层层递进：AOP-AspectWekz—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。

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。

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！

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！

<<开发者突击>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>