

<<项目实践精解>>

图书基本信息

书名：<<项目实践精解>>

13位ISBN编号：9787121067181

10位ISBN编号：7121067188

出版时间：2008-7

出版时间：电子工业出版社

作者：梁立新

页数：624

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>

## <<项目实践精解>>

### 内容概要

本书通过一个完整的电子商务应用的设计和开发项目，引领读者学习Struts-Spring-Hibernate架构下完整的开发体系，以及Struts-Spring-Hibernate的各个组件在项目开发中的实际应用。通过对该项目各个功能模块需求分析、系统分析设计、部署实现及测试等全过程的分析，使读者能够快速掌握Struts-Spring-Hibernate中最核心的组件，并能够依据书中提供的项目案例定制所需的功能，开发专业的Java应用系统。

本书可作为欲学习基于Struts-Spring-Hibernate的Java应用开发学习者的指导用书，也可作为Java应用开发的培训教材，还可作为Struts-Spring-Hibernate开发人员的参考书。

## 书籍目录

第一篇 整体介绍第1章 Struts-Spring-Hibernate概述1.1 框架概述1.2 Struts、Spring和Hibernate概述1.2.1 表示层1.2.2 持久层1.2.3 业务层1.2.4 域模块层第二篇 项目分析设计第2章 项目开发流程与分析设计概述2.1 项目开发流程2.1.1 项目生命周期2.1.2 项目开发阶段2.2 UML概述2.2.1 用例图2.2.2 类图2.2.3 对象图2.2.4 组件图2.2.5 部署图2.2.6 序列图2.2.7 协作图2.2.8 状态图2.2.9 活动图第3章 AscentWeb电子商务项目概述3.1 项目需求分析3.2 项目系统分析和设计3.2.1 面向对象分析设计3.2.2 数据库设计 (Data Model) 3.3 Struts 2.0-Spring 2.0-Hibernate 2.13.3 实现3.3.1 开发工具与环境3.3.2 开发工具集成步骤第三篇 Struts-Spring-Hibernate核心技术第4章 Struts技术4.1 MVC模式概述4.2 MVC与Struts映射4.3 Struts的工作流程和配置文件4.3.1 Struts的工作流程4.3.2 Struts的配置文件4.4 创建Controller组件4.4.1 FilterDispatcher4.4.2 Action的开发4.4.3 属性驱动和模型驱动4.5 创建Model组件4.5.1 JavaBean概述4.5.2 Enterprise JavaBean (EJB) 概述4.5.3 Web服务 (Web Services) 概述4.6 创建View组件4.6.1 Struts常用标签库 (Struts Tag Library) 4.6.2 Struts中的表达式语言4.6.3 国际化支持4.7 Struts的异常机制4.8 转换器4.9 数据验证 (Validation) 4.9.1 使用Action的validate()方法4.9.2 使用Struts .0的校验框架4.10 拦截器 (Interceptor) 4.10.1 拦截器概述4.10.2 已有的拦截器4.10.3 自定义拦截器4.11 文件上传4.12 Eclipse下Struts 2.0项目开发4.12 步骤第5章 Hibernate技术5.1 Hibernate概述5.2 Hibernate单表的对象/关系5.2 数据库映射5.2.1 持久化对象 (Persistent Object) 5.2.2 DAO (Data Access Object) 5.3 Hibernate多表的对象/关系数据库映射5.3.1 多对一/一对多关系5.3.2 一对一关系5.3.3 多对多关系5.4 HQL语言 (Hibernate Query Language) 5.5 Hibernate过滤器 (filters) 5.6 对象状态管理5.7 继承映射 (Inheritance Mapping) 5.8 组件 (Component) 映射5.9 性能优化5.9.1 抓取策略 (Fetching Strategy) 5.9.2 缓存管理5.9.3 理解集合性能 (Understanding Collection Performance) 5.9.4 监测性能 (Monitoring Performance) 5.10 批量处理 (Batch Processing) 5.10.1 批量插入 (Batch Insert) 5.10.2 批量更新 (Batch Update) 5.10.3 大批量更新/删除 (Bulk Update/Delete) 5.11 最佳实践 (Best Practices) 5.12 Hibernate实战开发步骤第6章 Spring技术6.1 Spring 2.0的新特性6.1.1 简介6.1.2 控制反转 (IoC) 容器6.1.3 面向切面编程 (AOP) 6.1.4 中间层6.1.5 Web层6.1.6 其他特性6.1.7 移植到Spring 2.06.1.8 更新的样例应用6.1.9 改进的文档6.2 Spring控制反转IoC (Inversion of Control) 6.2.1 IoC原理6.2.2 BeanFactory6.2.3 ApplicationContext6.3 Spring AOP面向方面编程原理6.3.1 AOP概念6.3.2 Spring的切入点 Spring的通知类型6.3.4 Spring中的advisor6.3.5 用ProxyFactoryBean创建AOP代理6.4 事务处理6.4.1 声明式事务处理6.4.2 编程式事务处理6.5 Spring其他特性6.5.1 Spring对JDBC的集成功能6.5.2 Spring MVC Web框架6.5.3 Spring对EJB的集成6.5.4 Spring对时序调度的支持6.5.5 使用Spring邮件抽象层发送E-mail6.5.6 通过Spring使用远程访问和Web服务6.5.7 JMS支持6.5.8 集成表现层6.6 Struts-Spring-Hibernate集成6.6.1 环境搭建和基本配置6.6.2 Spring和Hibernate集成6.6.3 Spring和Struts集成第四篇 测试与其他相关技术第7章 测试概述7.1 软件测试概述7.2 常用测试技术7.3 JUnit单元测试简介7.3.1 安装JUnit7.3.2 测试流程7.3.3 Eclipse与JUnit7.4 JMeter系统测试7.4.1 JMeter简介7.4.2 JMeter测试流程7.4.3 JMeter总结7.5 测试跟踪工具Bugzilla7.5.1 Bugzilla安装7.5.2 Bugzilla操作说明7.5.3 Bugzilla管理员操作指南第8章 CVS的使用8.1 CVS介绍8.2 CVS服务器的安装和配置8.3 CVS Eclipse客户端的配置和使用8.4 CVS在Eclipse下的冲突处理第9章 Ant工具的使用9.1 Ant的安装使用9.2 项目中Ant的使用9.2.1 build.xml9.2.2 build.properties9.3 Eclipse中使用Ant第10章 Log4j日志工具10.1 Log4j日志概述10.2 项目中使用Log4j的实例第五篇 参考文档和技术细节附录A Struts与Struts对比附录B Struts 技术介绍附录C Spring和Hibernate整合的分页方案结束语参考文献

## 章节摘录

第1章 Struts-Spring-Hibernate概述 目前,国内外信息化建设已经进入以Web应用为基础核心的阶段。

Java语言应该算得上是开发Web应用的**最佳语言**。

然而,就算用Java建造一个不是很烦琐的Web应用系统,也不是件轻松的事情。

有很多东西需要仔细考虑,比如要考虑怎样建立用户接口?

在哪里处理业务逻辑?

怎样持久化数据?

幸运的是,Web应用面临的一些问题已经由曾遇到过这类问题的开发者建立起相应的框架(Framework)解决了。

事实上,企业开发中直接采用的往往并不是某些具体的技术,比如大家熟悉的Core Java、JDBC、Servlet、JSP等,而是基于这些技术之上的应用框架(Framework),Struts、Spring、Hibernate就是其中最常用的几种。

1.1 框架概述 在介绍软件框架(Framework)之前,我们首先要明确什么是框架和为什么要使用框架。

这要从企业面临的挑战谈起,如图1.1所示。

我们可以看到,随着项目的规模和复杂性的提高,企业面临前所未有的各个方面的挑战。根据优先级排序,主要包括高可靠性(High Availability)、低成本(CostEffective)、可扩展性(Scalability)、投放市场快速性(Time to Market)、安全性(Secure)、性能(Good Performance)、可集成性(Ability to integrate)以及多平台支持(Multi—channel)等。

那么,我们如何面对并且解决这些挑战呢?

这需要采用通用的、灵活的、开放的、可扩展的软件框架,由框架来帮助我们解决这些挑战,之后再**在框架基础之上开发具体的应用系统**,如图1—2所示。

这种基于框架的软件开发方式和传统的汽车生产方式是很类似的,如图1—3所示。

那么,到底什么是软件框架呢?

框架(Framework)的定义如下: 是应用系统的骨架,将软件开发中反复出现的任务标准化,以可重用的形式提供使用; 大多提供了可执行的具体程序代码,支持迅速地开发出可执行的应用;但也可以是抽象的设计框架,帮助开发出健壮的设计模型; 好的抽象、设计成功的框架,能够大大缩短应用系统开发的周期; 在预制框架上加入定制的构件,可以大量减少编码量,并容易测试; 分别用于垂直和水平应用。

框架具有以下特点: 框架具有很强(大粒度)的可重用性,远远超过了单个类;它是一个功能连贯的类集合,通过相互协作为应用系统提供服务 and 预制行为; 框架中的不变部分,定义了接口、对象的交互和其他不变量; 框架中的变化部分(应用中的个性)。

一个好的框架定义了开发和集成组件的标准。

为了利用、定制或扩展框架服务,通常需要框架的使用者从已有框架类继承相应的子类,以及通过执行子类的重载方法,用户定义的类将会从预定义的框架类获得需要的消息。

这会给我们带来很多好处,包括代码重用性和一致性,对变化的适应性,特别是它能够让开发人员专注于业务逻辑,从而大大减少了开发时间。

图1.4对是否使用框架对项目开发所需工作量(以人月来衡量)的影响进行了对比。

从图1.4中我们不难看出,对于没有使用框架的项目而言,开发所需工作量(以Man days,即人月来衡量)会随着项目复杂性的提高(以Business function,即业务功能来衡量)以几何级数递增;而对于使用框架的项目而言,开发所需工作量会随着项目复杂性的提高以代数级数递增。

举个例子:假定开发团队人数一样,一个没有使用框架的项目所需的周期为6~9个月的话,那么同样的项目如果使用框架则只需要3~5个月。

1.2 Struts、Spring和Hibernate概述 在本书中,我们会具体讨论如何使用3种著名的框架Struts、Spring和Hibernate来使你的应用程序在保证质量前提下得以快速开发。

## &lt;&lt;项目实践精解&gt;&gt;

在软件架构设计中，我们往往建立软件系统的高层结构，常常用到分层架构模式。

(1) 分层模式是一种将系统的行为或功能以层为首要的组织单位来进行分配（划分）的结构模式。

?通常在逻辑上进行垂直的层次Layer划分； ?在物理上则进行水平的层级Tier划分。

(2) 分层要求：层内的元素只信赖于当前层和之下的相邻层中的其他元素。

注意，这并非绝对的要求。

大部分的Web应用在职责上至少能被分成4层：表示层（Presentation Layer）、持久层（Persistence Layer）、业务层（Business Layer）和域模块层（Domain Model Layer）。

每个层在功能上都应该是十分明确的，而不应该与其他层混合。

每个层要相互独立，通过一个通信接口而相互联系。

下面将分别详细地介绍这4层，讨论一下这些层应该提供什么，不应该提供什么。

这里讨论一个使用3种开源框架的策略：表示层用Struts；业务层用Spring；而持久层则用Hibernate，如图1.5所示。

<<项目实践精解>>

编辑推荐

可作为欲学习基于Struts-Spring-Hibernate的Java应用开发学习者的指导用书，也可作为Java应用开发的培训教材，还可作为Struts-Spring-Hibernate开发人员的参考书。

#### 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>