



# 轻量级 Java EE 企业应用实战— Struts 2+Spring+Hibernate 整合开发

新东方IT培训广州中心 李刚 编著  
软件教学总监

疯狂源自梦想

技术成就辉煌

疯狂源自梦想

技术成就辉煌





# 轻量级 Java EE 企业应用实战 Struts 2+Spring+Hibernate 整合开发

新东方IT培训广州中心 李刚 编著  
软件教学总监

电子工业出版社  
Publishing House of Electronics Industry  
北京·BEIJING

## 内 容 简 介

本书是《轻量级 J2EE 企业应用实战》的第二版，同时还融合了《整合 Struts+Hibernate+Spring 应用开发详解》理论部分。实际上，本书凝结了前两本书的精华部分。

本书介绍了 Java EE 领域的三个开源框架：Struts 2、Spring 和 Hibernate。其中 Struts 2 升级到 2.1.2，Spring 升级到 2.5.5，Hibernate 升级到 3.2.6。本书保留《轻量级 J2EE 企业应用实战》里关于 Tomcat、Jetty 的介绍，新增了关于 Ant、CVS 的内容。

本书内容主要包括三部分，第一部分介绍 Java EE 开发的基础知识，包括如何搭建开发环境。第二部分详细讲解了 Struts 2.1、Spring 2.5 和 Hibernate 3.2 三个框架用法，介绍三个框架时，通过 Eclipse IDE 的使用来上手，一步步带领读者深入三个框架的核心。这部分内容是笔者讲授“疯狂 Java 实训”的培训讲义，既包含了笔者多年开发经历的领悟，也融入了丰富的授课经验。第三部分示范开发了一个包含 7 个表，表之间具有复杂的关联映射、继承映射等关系，且业务也相对复杂的工作流案例，让读者理论联系实际，将三个框架真正运用到实际开发中去。该案例采用目前最流行、最规范的 Java EE 架构，整个应用分为领域对象层、DAO 层、业务逻辑层、MVC 层和视图层，各层之间分层清晰，层与层之间以松耦合的方法组织在一起。

本书不再介绍 Struts 1.x 相关内容，如果读者希望获取《轻量级 J2EE 企业应用实战》第一版中关于 Struts 1.x 的知识，请登录 <http://www.crazyjava.org> 下载。当读者阅读此书遇到技术难题时，也可登录 <http://www.crazyjava.org> 发帖，笔者将会及时予以解答。

阅读本书之前，建议先认真阅读笔者所著的《疯狂 Java 讲义》一书。本书适合有较好的 Java 编程基础，或有初步 JSP、Servlet 基础的读者阅读。尤其适合对 Struts 2、Spring、Hibernate 了解不够深入，或对 Struts 2+Spring+Hibernate 整合开发不太熟悉的开发人员阅读。

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

轻量级 Java EE 企业应用实战: Struts 2+Spring+Hibernate 整合开发 / 李刚编著. —北京: 电子工业出版社, 2008.11  
ISBN 978-7-121-07471-4

I. 轻… II. 李… III. ①JAVA 语言—程序设计②软件工具—程序设计 IV. TP312 TP311.56

中国版本图书馆 CIP 数据核字 (2008) 第 151616 号

责任编辑: 葛 娜

印 刷: 北京天宇星印刷厂

装 订: 三河市皇庄路通装订厂

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: 850×1168 1/16 印张: 52.25 字数: 1771 千字

印 次: 2008 年 11 月第 1 次印刷

印 数: 4000 册 定价: 89.00 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 [zltz@phei.com.cn](mailto:zltz@phei.com.cn)，盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线: (010) 88258888。



# 前 言

经过多年实践证明，J2EE（Java 2 Enterprise Edition）的确是最优秀的企业信息化开发平台。由于 J2EE 的名称容易引起误解，Sun 将 J2EE 更名为 Java EE，如今，Java EE 平台已经成为电信、金融、电子商务、保险、证券等各行业的大型应用系统的首选开发平台。

在实际的 Java EE 平台中，使用 Struts 2+Spring+Hibernate 进行整合开发是最为流行的开发平台，这种开发平台也被称为轻量级 Java EE 企业应用平台。与经典的、以 EJB 为核心的 Java EE 应用相比，轻量级 Java EE 应用简化了经典 Java EE 应用的难度，在保留经典 Java EE 应用架构、高度可扩展性、高度可维护性的基础上，降低了 Java EE 应用的开发、部署成本。

本书融合《轻量级 J2EE 企业应用实战》、《整合 Struts+Hibernate+Spring 应用开发详解》两本书的理论部分，并将 Struts 2 升级到 Struts 2.1.2，Spring 升级到 2.5.5，Hibernate 升级到 3.2.6。新增了 Struts 2、Spring 和 Hibernate 三个框架的“零配置”特性的内容，并重点介绍了 Spring 2.x 的 AOP 支持，详细介绍了 Spring 2.x 中 Scheme 配置所支持的 util、aop、tx 等命名空间。由于 Spring 2.x 的 AOP 需要 AspectJ 支持，所以本书还简要讲解了 AspectJ 的相关内容。

## 本书主要内容



本书内容主要包括三部分，第一部分介绍 Java EE 开发的基础知识，包括如何搭建开发环境。第二部分详细讲解了 Struts 2.1、Spring 2.5 和 Hibernate 3.2 三个框架用法，介绍三个框架时，通过 Eclipse IDE 的使用来上手，一步步带领读者深入三个框架的核心。这部分内容是笔者讲授“疯狂 Java 实训”的培训讲义，既包含了笔者多年开发经历的领悟，也融入了丰富的授课经验。第三部分示范开发了一个包含 7 个表，表之间具有复杂的关联映射、继承映射等关系，且业务也相对复杂的工作流案例，让读者理论联系实际，将三个框架真正运用到实际开发中去。该案例采用目前最流行、最规范的 Java EE 架构，整个应用分为领域对象层、DAO 层、业务逻辑层、MVC 层和视图层，各层之间分层清晰，层与层之间以松耦合的方法组织在一起。

## 本书创作感言



到现在为止，笔者已出版了不少软件开发技术的图书，比如 Spring、Ajax、Struts 2 和 Ruby On Rails 等，笔者所一直致力于软件开发技术的最前沿写作，这正是笔者写书的目的：传播知识，希望能将国际流行的、实用的软件开发技术带给国内读者。

笔者现阶段主要从事软件开发培训，而且一直坚持一线授课工作，因为这份工作既是激发笔者创作的直接原因，也给笔者的创作带来巨大的帮助。在长达 2 年多的授课生涯中，笔者接触了大量大学毕业生，以及一些做了 2~3 年开发、但工资依然不高的程序员，笔者对他们的彷徨、苦痛感同身受，因为笔者曾经也经历过学习技术的苦痛：当我们面临一个又一个无法解决的技术难题时，那种巨大挫败感、压抑感，那种夜不能寐、彻夜思索的经历，只有真正的程序员才懂——笔者完全可以体会，因为这些经历就是笔者的过往。

有感于此，所以笔者创作图书时主要关心如下几点：

- 读者是否可以看懂、理解各知识点的讲解？是否可以运用这些知识解决实际开发问题？
- 书中的案例是否可以运行？这些案例是否采用实际开发的应用架构？
- 读者是否可参考本书来开发实际项目？读完本书后是否能获得实用的软件开发技巧？

笔者相信这些也是读者所真正关心的，所以笔者尽量使用简短的语句、通俗的说法来介绍那些所谓的“深奥知识”。实际上，笔者有一种感觉：当真正达到融会贯通的境界后，再去看那些所谓的“深奥知识”时会产生一种“不过如此”的感觉。比如笔者上课介绍 HashSet 的存取时，笔者告诉学生：实际上就像你妈妈教你“不同的东西放在不同的位置”，这样才能最快地找到它，你可以把你妈妈的那套存取规则理解为 Hash 算法。他们感叹：原来如此简单，可惜以前从来没有人这样跟我讲过。

很多看似深奥的知识，可能只是因为你看了一些翻译得过于生僻的专业术语，或者是因为你看了一段过于晦涩的讲解，更甚至于只是你的畏难心理作祟。抛开这些障眼的浮云，你会发现：编程领域的许多知识和日常生活中的常识有极高的相似性。

由于这个缘故，有人评价笔者的书过于简单，主要针对初学者——那么，笔者愿意把简单作为其创作特色，笔者不愿故作高深，宁愿以简单方式来解决实际问题。笔者希望读者认真看完一本书后，实际开发问题得到了解决，并切实学会一种实用的开发技术，而不是空洞的吹嘘理论。

有时候忍不住想：中国传统的风水理论、玄学理论、中医理论，也许未必真的那么深奥，是不是曾经被人为地复杂化，人为地高深化了，所以导致这些知识没有更好地流传开来？如果我们用通俗、易懂的方式来传播它们，也许不是今天这个局面。

比如极个别的 Java 技术论坛，如果一个 Java 学习者去提问，得到的结果往往是几句看似高深的提示，甚至是几句嘲笑，但实际问题依然没有解决——这种解决问题的方式，让故作高深的人获得了一种精神上的、虚无的优越感，对普通学习者，却是一场灾难。有感于此，笔者亲自创办了 <http://www.crazyjava.org> 论坛，希望带给广大技术学习者更好的交流平台。

## 本书有什么特点



本书保持了《轻量级 J2EE 企业应用实战》一书简单、实用的优势，同样坚持让案例说话、以案例来介绍知识点的风格。本书最后同样示范开发了企业工作流案例，希望读者通过该案例真正步入实际企业开发的殿堂。

本书依然保留了《轻量级 J2EE 企业应用实战》第一版的以下三个特色。

### 1. 经验丰富，针对性强

笔者既担任过软件开发的技术经理，也担任过软件公司的培训导师，还从事过职业培训的专职讲师。这些经验影响了笔者写书的目的，这不是一本学院派的理论读物，而是一本实际的开发指南。

### 2. 内容实际，实用性强

本书所介绍的 Java EE 应用范例，采用了目前企业流行的开发架构，绝对严格遵守 Java EE 开发规范，而不是将各种技术杂乱地糅合在一起号称 Java EE。读者参考本书的架构，完全可以身临其境地感受企业实际开发。

### 3. 高屋建瓴，启发性强

本书介绍的几种架构模式，几乎是时下最全面的 Java EE 架构模式。这些架构模式可以直接提升读者对系统架构设计的把握。

与《轻量级 J2EE 企业应用实战》第一版相比，程序中关键代码以粗体字标出，可以让读者更迅速地找到这些程序的关键点。书中代码的注释更加详细，几乎每两行代码就有一行注释，这些注释也添加在光盘的源代码中，并且给出了书中代码和光盘源代码的对应关系，尽量降低读者阅读光盘源码的难度。

## 本书写给谁看



如果你已经掌握 Java SE 内容，或已经学完了《疯狂 Java 讲义》一书，那你非常适合阅读此书。

如果你已有初步的 JSP、Servlet 基础，甚至对 Struts 2、Spring 2.5、Hibernate 3.2 有所了解，但希望掌握它们在实际开发中的应用，本书也将非常适合你。

如果你对 Java 的掌握还不熟练，则建议遵从学习规律，循序渐进，暂时不要购买、阅读此书。

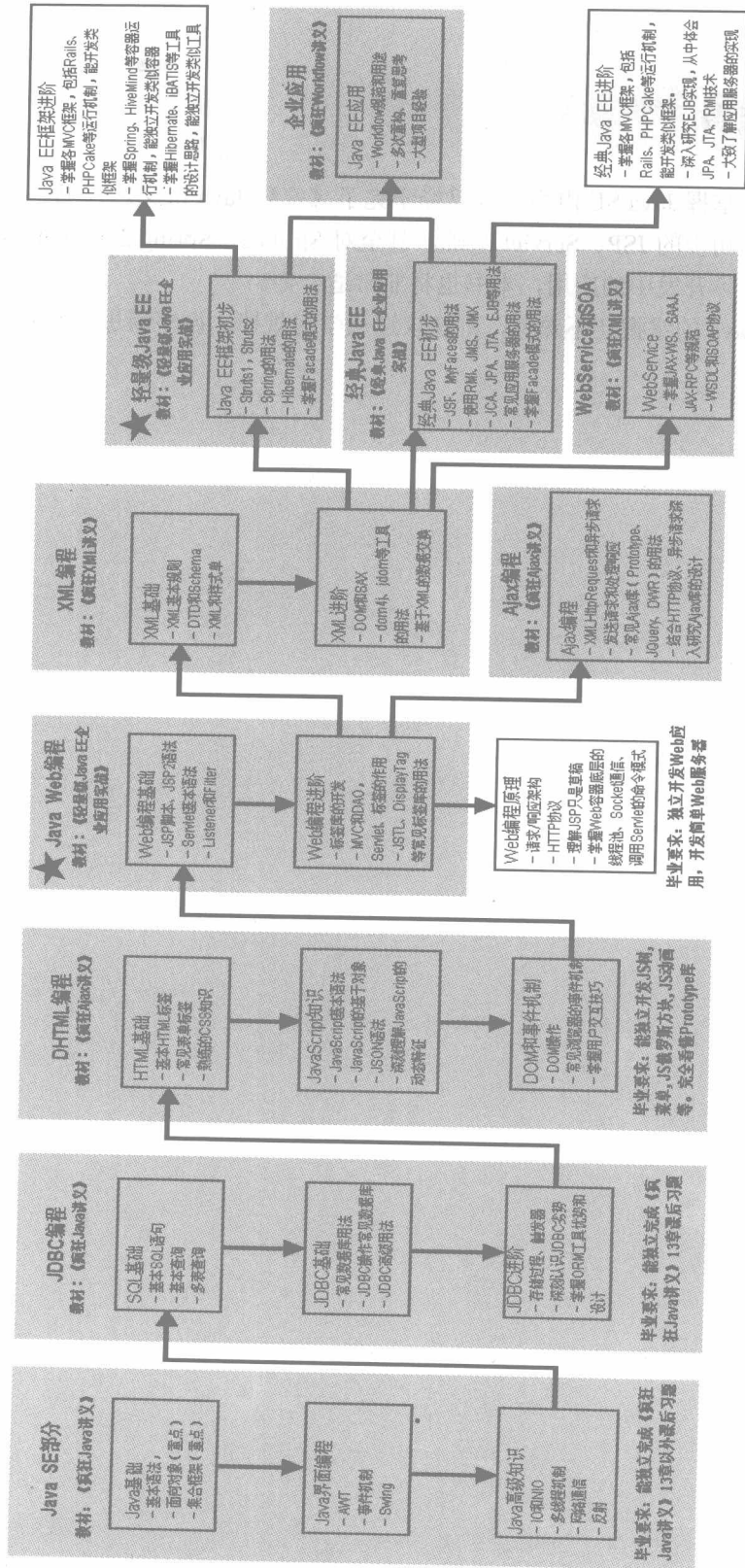
2008-8-20





# 疯狂 Java 学习路线图

笔者就自己对 Java EE 体系的理解，对 Java EE 学习者给出一个粗略路线图：



备注:

1. 没有灰色覆盖的区域稍有难度，请谨慎尝试。
2. 本学习路线图不涉及设计模式、软件方法学等概念，但希望大家能从开发中悟道。
3. 本人并不认为 Spring、Hibernate 很复杂，只要基础扎实，掌握框架是水到渠成的。



# 光盘说明

## 一、光盘内容

本光盘是《轻量级 Java EE 企业应用开发实战——Struts 2+Spring+Hibernate 整合开发》一书的配书光盘，书中的代码按章、按节存放，即第 2 章 2.1 节所使用的代码放在 codes 文件夹中的 02\2.1 文件夹下，依此类推。

另：书中每份源代码也给出与光盘源文件的对应关系，方便读者查找。

本光盘 codes 目录下有 10 个文件夹，其内容和含义说明如下：

① 文件夹 01~10 对应于《轻量级 Java EE 企业应用开发实战——Struts 2+Spring+Hibernate 整合开发》中的章名，即第 2 章所使用的代码放在 codes 文件夹中的 02 文件夹下，依此类推。

② 文件夹 10 下有 HRSystem 和 HRSystem\_Eclipse 两个文件夹，它们是同一个项目的源文件，其中 HRSystem 是 IDE 平台无关的项目，使用 Ant 来编译即可；而 HRSystem\_Eclipse 是该项目在 Eclipse IDE 工具中的项目文件。

③ codes\03\3.2\Struts2Demo 目录、codes\05\5.2\HibernateDemo 目录、codes\07\7.2\myspring 目录和 codes\10\HRSystem\_Eclipse 目录下有.classpath、.mymetadata、.project、.springBeans 等文件，它们是 Eclipse 项目文件，请不要删除。

## 二、运行环境

本书中的程序在以下环境调试通过：

① 安装 jdk-6u6-windows-i586-p.exe，安装完成后，添加 CLASSPATH 环境变量，该环境变量的值为“;%JAVA\_HOME%/lib/tools.jar;%JAVA\_HOME%/lib/dt.jar”。如果为了可以编译和运行 Java 程序，还应该在 PATH 环境变量中增加“%JAVA\_HOME%/bin”。其中，JAVA\_HOME 代表 JDK（不是 JRE）的安装路径。

② 安装 Apache 的 Tomcat 6.0.16，不要使用安装文件安装，而是采用解压缩的安装方式。

安装 Tomcat 请参看第 1 章。安装完成后，将 Tomcat 安装路径的 lib 下的 jsp-api.jar 和 servlet-api.jar 两个 JAR 文件添加到 CLASSPATH 环境变量之后。

③ 安装 apache-ant-1.7.0。将下载的 Ant 压缩文件解压缩到任意路径，然后增加 ANT\_HOME 环境变量，让变量的值为 Ant 的解压缩路径。并在 PATH 环境变量中增加 %ANT\_HOME%/bin 环境变量。

④ 安装 MySQL 5.0 或更高版本，安装 MySQL 时选择 GBK 的编码方式。

⑤ 安装 Eclipse 3.2，并安装 MyEclipse 6.0 插件。

关于如何安装上面工具，请参考本书的第 1 章。

## 三、注意事项

① 独立应用程序的代码中都包括 build.xml 文件，在 DOS 或 Shell 下进入 build.xml 文件所在路径，执行如下命令：

```
ant build -- 编译程序
```

```
ant run --运行程序
```



② 对于 Web 应用，将该应用复制到%TOMCAT\_HOME%/webapps 路径下，然后进入 build.xml 所在路径，执行如下命令：

ant build -- 编译应用

启动 Tomcat 服务器，使用浏览器即可访问该应用。

③ 对于 Eclipse 项目文件，导入 Eclipse 开发工具即可。

④ 第 10 章的案例，请参看项目下的 readme.txt 文件。

⑤ 代码中有大量代码需要连接数据库，读者应修改数据库 URL 以及用户名、密码，让这些代码与读者运行环境一致。如果项目下有 SQL 脚本，导入 SQL 脚本即可；如果没有 SQL 脚本，系统将在运行时自动建表，读者只需创建对应数据库即可。

⑥ 在使用本光盘中的程序时，请将程序拷贝到硬盘上，并去除文件的只读属性。

#### 四、技术支持

如果您在使用本光盘中遇到什么问题，可以登录如下网站与我们联系：

<http://www.leegang.org>

或者发 E-mail 到：[kongyeeku@163.com](mailto:kongyeeku@163.com)

# 目 录 CONTENTS

## 第 1 篇 基础知识

第 1 章 Java EE 应用和开发环境	1
1.1 Java EE 应用概述	2
1.1.1 Java EE 应用的分层模型	2
1.1.2 Java EE 应用的组件	3
1.1.3 Java EE 应用结构和优势	4
1.1.4 常用的 Java EE 服务器	4
1.2 轻量级 Java EE 应用相关技术	5
1.2.1 JSP、Servlet 和 JavaBean 及替代技术	5
1.2.2 Struts 2.1 及替代技术	5
1.2.3 Hibernate 3.2 及替代技术	6
1.2.4 Spring 2.5 及替代技术	6
1.3 Tomcat 的下载和安装	7
1.3.1 安装 Tomcat 服务器	7
1.3.2 配置 Tomcat 的服务端口	9
1.3.3 进入控制台	9
1.3.4 部署 Web 应用	11
1.3.5 配置 Tomcat 的数据源	12
1.4 Jetty 的下载和安装	13
1.4.1 Jetty 的下载和安装	14
1.4.2 Jetty 的基本配置	15
1.4.3 部署 Web 应用	16
1.4.4 配置 JNDI 绑定	17
1.5 Eclipse 的安装和使用	19
1.5.1 Eclipse 的下载和安装	20
1.5.2 在线安装 Eclipse 插件	20
1.5.3 手动安装 Eclipse 插件	21
1.5.4 安装 MyEclipse 插件	22
1.5.5 使用 Eclipse 开发 Java EE 应用	25
1.5.6 导入 Eclipse 项目	28
1.5.7 导入非 Eclipse 项目	29
1.6 Ant 的安装和使用	30
1.6.1 Ant 的下载和安装	30
1.6.2 使用 Ant 工具	31
1.6.3 定义生成文件	32
1.6.4 Ant 的任务 (task)	37
1.7 使用 CVS 进行协作开发	39
1.7.1 安装 CVS 服务器	40
1.7.2 配置 CVS 资源库	42
1.7.3 安装 CVS 客户端	43
1.7.4 发布项目到服务器	44
1.7.5 从服务器下载项目	46
1.7.6 同步 (Update) 本地文件	47
1.7.7 提交 (Commit) 修改	48
1.7.8 添加文件和目录	48
1.7.9 删除文件和目录	50
1.7.10 查看文件的版本变革	51
1.7.11 提取文件以前版本的内容	51
1.7.12 从以前版本重新开始	52
1.7.13 创建标签	52
1.7.14 创建分支	53
1.7.15 沿着分支开发	54
1.7.16 使用 Eclipse 作为 CVS 客户端	55
1.8 本章小结	58
第 2 章 JSP 相关技术详解	59
2.1 Web 应用和 web.xml 文件	60
2.1.1 构建 Web 应用	60
2.1.2 配置描述符 web.xml	61
2.2 JSP 的基本原理	62
2.3 JSP 注释	66
2.4 JSP 声明	67
2.5 输出 JSP 表达式	68
2.6 JSP 脚本	69
2.7 JSP 的 3 个编译指令	72
2.7.1 page 指令	72
2.7.2 include 指令	76
2.8 JSP 的 7 个动作指令	77
2.8.1 forward 指令	77
2.8.2 include 指令	79
2.8.3 useBean、setProperty、getProperty 指令	81
2.8.4 plugin 指令	84
2.8.5 param 指令	86
2.9 JSP 脚本中的 9 个内置对象	86
2.9.1 application 对象	88

2.9.2 config 对象	94
2.9.3 exception 对象	96
2.9.4 out 对象	98
2.9.5 pageContext 对象	99
2.9.6 request 对象	101
2.9.7 response 对象	108
2.9.8 session 对象	112
2.10 Servlet 介绍	114
2.10.1 Servlet 的开发	114
2.10.2 Servlet 的配置	116
2.10.3 Servlet 的生命周期	117
2.10.4 load-on-startup Servlet	117
2.10.5 访问 Servlet 的配置参数	118
2.10.6 使用 Servlet 作为控制器	120
2.11 JSP2 的自定义标签	125
2.11.1 开发自定义标签类	126
2.11.2 建立 TLD 文件	126
2.11.3 使用标签库	127
2.11.4 带属性的标签	128
2.11.5 带标签体的标签	132
2.12 Filter 介绍	135
2.12.1 创建 Filter 类	135
2.12.2 配置 Filter	136
2.13 Listener 介绍	139
2.13.1 创建 Listener 类	139
2.13.2 配置 Listener	141
2.14 JSP 2 特性	141
2.14.1 配置 JSP 属性	142
2.14.2 表达式语言	144
2.14.3 Tag File 支持	152
2.15 本章小结	155

## 第 2 篇 整合开发

第 3 章 Struts 2 的基本用法	156
3.1 MVC 思想概述	157
3.1.1 传统 Model 1 和 Model 2	157
3.1.2 MVC 思想及其优势	158
3.2 Struts 2 的下载和安装	159
3.2.1 为 Web 应用增加 Struts 2 支持	159
3.2.2 在 Eclipse 中使用 Struts 2	161
3.2.3 增加登录处理	161
3.3 Struts 2 的流程	165

3.3.1 Struts 2 应用的开发步骤	165
3.3.2 Struts 2 的流程	166
3.4 Struts 2 的常规配置	167
3.4.1 常量配置	167
3.4.2 包含其他配置文件	173
3.5 实现 Action	173
3.5.1 Action 接口和 ActionSupport 基类	175
3.5.2 Action 访问 Servlet API	177
3.5.3 Action 直接访问 Servlet API	180
3.6 配置 Action	182
3.6.1 包和命名空间	182
3.6.2 Action 的基本配置	185
3.6.3 使用 Action 的动态方法调用	186
3.6.4 指定 method 属性及使用通配符	189
3.6.5 配置默认 Action	195
3.7 配置处理结果	195
3.7.1 理解处理结果	195
3.7.2 配置结果	196
3.7.3 Struts 2 支持的结果类型	198
3.7.4 plainText 结果类型	199
3.7.5 redirect 结果类型	201
3.7.6 redirect-action 结果类型	202
3.7.7 动态结果	203
3.7.8 Action 属性值决定物理视图资源	204
3.7.9 全局结果	206
3.8 Struts 2 的“零配置”支持	207
3.8.1 自动搜索 Action	207
3.8.2 使用 Annotation 指定命名空间、 父包和结果映射	208
3.9 配置 Struts 2 的异常处理	211
3.9.1 Struts 2 的异常处理机制	211
3.9.2 声明式异常捕捉	212
3.9.3 输出异常信息	214
3.10 使用 Struts 2 的国际化	216
3.10.1 Struts 2 中加载全局资源文件	216
3.10.2 访问国际化消息	216
3.10.3 输出带占位符的国际化消息	219
3.10.4 加载资源文件的方式	221
3.10.5 加载资源文件的顺序	225
3.11 使用 Struts 2 的标签库	226
3.11.1 Struts 2 标签库概述	226
3.11.2 使用 Struts 2 标签	227

3.11.3 Struts 2 的 OGNL 表达式语言	227	4.3.8 校验顺序和短路	327
3.11.4 OGNL 中的集合操作	230	4.3.9 内建校验器	327
3.11.5 Lambda ( $\lambda$ ) 表达式	231	4.3.10 基于 Annotation 的输入校验	339
3.11.6 控制标签	231	4.3.11 手动完成输入校验	340
3.11.7 数据标签	242	4.4 使用 Struts 2 的 Ajax 支持	344
3.11.8 主题和模板	252	4.4.1 JSON 的基本知识	345
3.11.9 自定义主题	254	4.4.2 安装 Struts 2 的 JSON 插件	347
3.11.10 表单标签	255	4.4.3 实现 Action 逻辑	348
3.11.11 非表单标签	270	4.4.4 实现 JSP 页面	349
3.12 本章小结	273	4.5 使用 Struts 2 控制文件上传	351
<b>第 4 章 深入使用 Struts 2</b>	<b>274</b>	4.5.1 文件上传的准备	351
4.1 详解 Struts 2 的拦截器机制	275	4.5.2 Struts 2 的文件上传	352
4.1.1 拦截器在 Struts 2 中的作用	275	4.5.3 实现文件上传的 Action	353
4.1.2 Struts 2 内建的拦截器	276	4.5.4 配置文件上传的 Action	355
4.1.3 配置拦截器	277	4.5.5 手动实现文件过滤	358
4.1.4 使用拦截器	279	4.5.6 拦截器实现文件过滤	359
4.1.5 配置默认拦截器	280	4.5.7 输出错误提示	361
4.1.6 实现拦截器类	282	4.5.8 文件上传的常量配置	362
4.1.7 使用拦截器	283	4.6 使用 Struts 2 控制文件下载	363
4.1.8 拦截方法的拦截器	285	4.6.1 实现文件下载的 Action	363
4.1.9 拦截器的执行顺序	287	4.6.2 配置 Action	364
4.1.10 拦截结果的监听器	288	4.6.3 下载前的授权控制	365
4.1.11 覆盖拦截器栈里特定拦截器的 参数	290	4.7 本章小结	366
4.1.12 使用拦截器完成权限控制	291	<b>第 5 章 Hibernate 基本用法</b>	<b>268</b>
4.2 详解 Struts 2 的类型转换	293	5.1 ORM 和 Hibernate	369
4.2.1 Struts 2 内建的类型转换器	294	5.1.1 对象/关系数据库映射 (ORM)	369
4.2.2 基于 OGNL 的类型转换	295	5.1.2 基本映射方式	370
4.2.3 指定集合元素的类型	298	5.1.3 流行的 ORM 框架简介	371
4.2.4 自定义类型转换器	300	5.1.4 Hibernate 概述	372
4.2.5 注册类型转换器	303	5.2 Hibernate 入门	372
4.2.6 简化的自定义类型转换器	304	5.2.1 Hibernate 下载和安装	372
4.2.7 处理 Set 集合	305	5.2.2 Hibernate 的数据库操作	373
4.2.8 类型转换中的错误处理	308	5.2.3 在 Eclipse 中使用 Hibernate	378
4.3 使用 Struts 2 的输入校验	313	5.3 Hibernate 的体系结构	383
4.3.1 编写校验规则文件	314	5.4 深入 Hibernate 的配置文件	385
4.3.2 国际化提示信息	316	5.4.1 创建 Configuration 对象	385
4.3.3 使用客户端校验	317	5.4.2 hibernate.properties 文件与 hibernate.cfg.xml 文件	388
4.3.4 字段校验器配置风格	320	5.4.3 JDBC 连接属性	388
4.3.5 非字段校验器配置风格	321	5.4.4 数据库方言	389
4.3.6 短路校验器	323	5.4.5 JNDI 数据源的连接属性	390
4.3.7 校验文件的搜索规则	324	5.4.6 Hibernate 事务属性	391

5.4.7 二级缓存相关属性	391	6.3.2 批量更新	484
5.4.8 外连接抓取属性	391	6.3.3 DML 风格的批量更新/删除	485
5.4.9 其他常用的配置属性	392	6.4 使用 HQL 查询	486
5.5 深入理解持久化对象	392	6.4.1 HQL 查询	486
5.5.1 持久化类的要求	392	6.4.2 HQL 查询的 from 子句	489
5.5.2 持久化对象的状态	393	6.4.3 关联和连接	489
5.5.3 改变持久化对象状态的方法	394	6.4.4 HQL 查询的 select 子句	492
5.6 深入 Hibernate 的映射文件	397	6.4.5 HQL 查询的聚集函数	493
5.6.1 映射文件结构	397	6.4.6 多态查询	493
5.6.2 映射主键	400	6.4.7 HQL 查询的 where 子句	494
5.6.3 映射普通属性	402	6.4.8 表达式	495
5.6.4 映射集合属性	406	6.4.9 order by 子句	497
5.6.5 集合属性的性能分析	418	6.4.10 group by 子句	497
5.6.6 有序集合映射	419	6.4.11 子查询	498
5.6.7 映射数据库对象	422	6.4.12 命名查询	499
5.7 映射组件属性	424	6.5 条件查询	499
5.7.1 组件属性为集合	427	6.5.1 关联和动态关联	502
5.7.2 集合属性的元素为组件	429	6.5.2 投影、聚合和分组	503
5.7.3 组件作为 Map 的索引	431	6.5.3 离线查询和子查询	506
5.7.4 组件作为复合主键	434	6.6 SQL 查询	507
5.7.5 多列作为联合主键	436	6.6.1 标量查询	508
5.8 使用注释	438	6.6.2 实体查询	509
5.9 本章小结	441	6.6.3 处理关联和继承	512
<b>第 6 章 深入使用 Hibernate</b>	442	6.6.4 命名 SQL 查询	512
6.1 Hibernate 的关联映射	443	6.6.5 调用存储过程	514
6.1.1 单向 N-1 关联	443	6.6.6 使用定制 SQL	515
6.1.2 单向 1-1 关联	449	6.7 数据过滤	517
6.1.3 单向 1-N 关联	452	6.8 事务控制	520
6.1.4 单向 N-N 关联	455	6.8.1 事务的概念	520
6.1.5 双向 1-N 关联	456	6.8.2 Hibernate 的事务	520
6.1.6 双向 N-N 关联	461	6.8.3 事务和 Session	521
6.1.7 双向 1-1 关联	464	6.8.4 上下文相关的 Session	523
6.1.8 组件属性包含的关联实体	468	6.9 事件机制	524
6.1.9 传播性持久化	471	6.9.1 拦截器	524
6.2 继承映射	472	6.9.2 事件系统	527
6.2.1 采用 subclass 元素的继承映射	476	6.10 本章小结	531
6.2.2 采用 joined-subclass 元素的继承映射	477	<b>第 7 章 Spring 的基本用法</b>	532
6.2.3 采用 union-subclass 元素的继承映射	480	7.1 Spring 的起源和背景	533
6.3 Hibernate 的批量处理	482	7.2 Spring 的下载和安装	534
6.3.1 批量插入	483	7.2.1 在应用程序中使用 Spring	534
		7.2.2 在 Web 应用中使用 Spring	535
		7.2.3 在 Eclipse 中开发 Spring 应用	535

7.3 Spring 的核心机制：依赖注入.....	541	7.8.2 使用 util Schema.....	608
7.3.1 理解依赖注入.....	541	7.9 本章小结.....	610
7.3.2 设值注入.....	542	<b>第 8 章 深入使用 Spring</b> .....	611
7.3.3 构造注入.....	546	8.1 两种后处理器.....	612
7.3.4 两种注入方式的对比.....	547	8.1.1 Bean 后处理器.....	612
7.4 Spring 容器和被管理的 Bean .....	548	8.1.2 Bean 后处理器的用处.....	616
7.4.1 Spring 容器.....	548	8.1.3 容器后处理器.....	616
7.4.2 使用 ApplicationContext.....	550	8.1.4 属性占位符配置器.....	618
7.4.3 ApplicationContext 的国际化支持.....	550	8.1.5 重写占位符配置器.....	619
7.4.4 ApplicationContext 的事件机制.....	552	8.2 Spring 的“零配置”支持.....	621
7.4.5 Bean 的基本定义.....	554	8.2.1 搜索 Bean 类.....	621
7.4.6 容器中 Bean 的作用域.....	556	8.2.2 指定 Bean 的作用域.....	624
7.4.7 Spring 的 Bean 和 JavaBean .....	559	8.2.3 使用@Resource 配置依赖.....	624
7.4.8 获得容器的引用.....	560	8.2.4 自动装配和精确装配.....	625
7.5 Bean 实例的创建方式及对应配置 .....	562	8.3 资源访问.....	628
7.5.1 使用构造器创建 Bean 实例.....	562	8.3.1 Resource 实现类.....	628
7.5.2 使用静态工厂方法创建 Bean .....	564	8.3.2 ResourceLoader 接口和	
7.5.3 调用实例工厂方法创建 Bean .....	567	ResourceLoaderAware 接口.....	634
7.6 深入理解容器中的 Bean.....	569	8.3.3 使用 Resource 作为属性.....	636
7.6.1 使用抽象 Bean .....	569	8.3.4 在 ApplicationContext 中使用资源.....	638
7.6.2 使用子 Bean .....	570	8.4 Spring 的 AOP.....	642
7.6.3 Bean 继承与 Java 继承的区别.....	572	8.4.1 为什么需要 AOP.....	642
7.6.4 定制生命周期行为.....	572	8.4.2 使用 AspectJ 实现 AOP.....	643
7.6.5 协调作用域不同步的 Bean .....	578	8.4.3 AOP 的基本概念.....	648
7.6.6 容器中的工厂 Bean .....	581	8.4.4 Spring 的 AOP 支持.....	649
7.6.7 获得 Bean 本身的 id.....	583	8.4.5 基于 Annotation 的“零配置”方式.....	650
7.7 深入理解依赖关系配置.....	584	8.4.6 基于 XML 配置文件的管理方式.....	666
7.7.1 配置依赖.....	584	8.5 Spring 的事务.....	673
7.7.2 设置普通属性值.....	586	8.5.1 Spring 支持的事务策略.....	673
7.7.3 配置合作者 Bean .....	588	8.5.2 使用 TransactionProxyFactoryBean	
7.7.4 组合属性名称.....	589	创建事务代理.....	679
7.7.5 注入嵌套 Bean .....	589	8.5.3 Spring 2.X 的事务配置策略.....	682
7.7.6 注入集合值.....	590	8.6 Spring 整合 Struts 2.....	687
7.7.7 注入其他 Bean 的属性值.....	594	8.6.1 启动 Spring 容器.....	687
7.7.8 注入 Field 值.....	596	8.6.2 MVC 框架与 Spring 整合的思考.....	689
7.7.9 注入方法返回值.....	598	8.6.3 让 Spring 管理控制器.....	690
7.7.10 强制初始化 Bean .....	601	8.6.4 使用自动装配.....	694
7.7.11 自动装配.....	602	8.7 Spring 整合 Hibernate .....	696
7.7.12 依赖检查.....	605	8.7.1 Spring 提供的 DAO 支持.....	697
7.8 基于 XML Schema 的简化配置		8.7.2 管理 Hibernate 的 SessionFactory.....	698
方式.....	606	8.7.3 使用 HibernateTemplate .....	699
7.8.1 使用 p 名称空间配置属性.....	607	8.7.4 使用 HibernateCallBack .....	703



8.7.5 实现 DAO 组件	705
8.7.6 使用 IoC 容器组装各种组件	707
8.7.7 使用声明式事务	710
8.8 本章小结	711

### 第 3 篇 应用实践

第 9 章 企业应用开发的思考与策略	712
9.1 企业应用开发面临的挑战	713
9.1.1 可扩展性、可伸缩性	713
9.1.2 快捷、可控的开发	714
9.1.3 稳定性、高效性	715
9.1.4 花费最小化, 利益最大化	715
9.2 如何面对挑战	715
9.2.1 使用建模工具	715
9.2.2 利用优秀的框架	716
9.2.3 选择性地扩展	718
9.2.4 使用代码生成器	718
9.3 常见设计模式精讲	718
9.3.1 单例模式	719
9.3.2 简单工厂	720
9.3.3 工厂方法和抽象工厂	727
9.3.4 代理模式	730
9.3.5 命令模式	736
9.3.6 策略模式	738
9.3.7 门面模式	741
9.3.8 桥接模式	745
9.3.9 观察者模式	748
9.4 常见的架构设计策略	751
9.4.1 贫血模型	752
9.4.2 领域对象模型	755
9.4.3 合并业务逻辑对象与 DAO 对象	757
9.4.4 合并业务逻辑对象和 Domain Object	759
9.4.5 抛弃业务逻辑层	760

9.5 本章小结	761
第 10 章 简单工作流系统	762
10.1 项目背景及系统结构	763
10.1.1 应用背景	763
10.1.2 系统功能介绍	763
10.1.3 相关技术介绍	764
10.1.4 系统结构	765
10.1.5 系统的功能模块	765
10.2 Hibernate 持久层	766
10.2.1 设计持久化对象 (PO)	766
10.2.2 创建持久化类	767
10.2.3 映射持久化类	772
10.3 实现 DAO 层	777
10.3.1 DAO 组件的定义	778
10.3.2 实现 DAO 组件	784
10.3.3 部署 DAO 层	788
10.4 实现 Service 层	791
10.4.1 业务逻辑组件的设计	791
10.4.2 实现业务逻辑组件	791
10.4.3 事务管理	797
10.4.4 部署业务逻辑组件	798
10.5 实现任务的自动调度	800
10.5.1 使用 Quartz	800
10.5.2 在 Spring 中使用 Quartz	804
10.6 实现系统 Web 层	807
10.6.1 Struts 2 和 Spring 的整合	807
10.6.2 控制器的处理顺序	808
10.6.3 员工登录	809
10.6.4 进入打卡	811
10.6.5 处理打卡	813
10.6.6 进入申请	814
10.6.7 提交申请	816
10.6.8 使用拦截器完成权限管理	818
10.7 本章小结	820

# 第 1 章

## Java EE 应用和开发环境

### 本章要点

- ✎ Java EE 应用的基础知识
- ✎ Java EE 应用的模型和相关组件
- ✎ Java EE 应用的结构和优势
- ✎ 轻量级 Java EE 应用的相关技术
- ✎ Tomcat 的下载和安装
- ✎ Tomcat 的相关配置
- ✎ Jetty 的下载和安装
- ✎ Jetty 的相关配置
- ✎ 下载和安装 Eclipse
- ✎ 安装 Eclipse 插件
- ✎ 使用 Eclipse 开发项目
- ✎ Ant 的下载和安装
- ✎ 使用 Ant
- ✎ 定义 Ant 生成文件
- ✎ CVS 服务器的下载和安装
- ✎ CVS 服务器的简单配置
- ✎ WinCvs 的下载和安装
- ✎ 使用 WinCvs 发布项目
- ✎ 使用 WinCvs 下载项目
- ✎ 使用 WinCvs 同步、提交文件
- ✎ 在 WinCvs 中创建标签、创建分支
- ✎ 使用 Eclipse 作为 CVS 的客户端

时至今日，轻量级 Java EE 平台在企业开发中具有绝对的优势，Java EE 应用以其稳定的性能、良好的开放性及严格的安全性，深受企业应用开发者的青睐。实际上，对于信息化要求较高的行业，如银行、电信、证券及电子商务等行业，都不约而同地选择了 Java EE 开发平台。

对于一个企业而言，选择 Java EE 构建信息化平台，更体现了一种长远的规划：企业的信息化是不断整合的过程，在未来的日子里，经常会有不同平台、不同系统的异构系统需要整合。Java EE 应用提供的跨平台性、开放性及各种远程访问的技术，为异构系统的良好整合提供了保证。

2006 年，Sun 提出了 Java EE 的概念，与之同步出现了两个主要规范：JSF 1.2 和 EJB 3.0。但应用依然不如 SSH (Struts+Spring+Hibernate) 组合的应用广泛，SSH 组合是一种轻量级的 Java EE 平台，具有高度的实用性、高度的可扩展性。基于轻量级 Java EE 平台的应用可以运行在普通 Web 容器中，无须 EJB 容器的支持，且一样具有稳定的性能和极高的可扩展性、可维护性。

本书作为《轻量级 J2EE 企业应用实战》的第二版，将全面升级 SSH 组合里三个开源框架的版本：Struts 将全面升级到 2.1，Spring 将升级到 2.5，Hibernate 将升级到 3.2，尽量让读者走在技术的最前沿。

## 1.1 Java EE 应用概述

今天我们所说的 Java EE 应用，往往超出了 Sun 所提出的经典 Java EE 应用规范，而是一种更广泛的开发规范。经典 Java EE 应用往往以 EJB (企业级 Java Bean) 为核心，以应用服务器为运行环境，所以通常开发、运行成本较高。本书所介绍的轻量级 Java EE 应用具备了 Java EE 规范的种种特征，例如面向对象建模的思维方式、优秀的应用分层以及良好的可扩展性、可维护性。轻量级 Java EE 应用保留了经典 Java 应用的架构，但开发、运行成本更低。

### 1.1.1 Java EE 应用的分层模型

不管是经典的 Java EE 架构，还是本书所介绍的轻量级 Java EE 架构，大致上都可分为如下几层：

- Domain Object (领域对象) 层：此层由系列的 POJO (Plain Old Java Object, 普通的、传统的 Java 对象) 组成，这些对象是该系统的 Domain Object，往往包含了各自所需要实现的业务逻辑方法。
- DAO (Data Access Object, 数据访问对象) 层：此层由系列的 DAO 组件组成，这些 DAO 实现了对数据库的创建、查询、更新和删除 (CRUD) 等原子操作。
- 业务逻辑层：此层由系列的业务逻辑对象组成，这些业务逻辑对象实现了系统所需要的业务逻辑方法。这些业务逻辑方法可能仅仅用于暴露 Domain Object 对象所实现的业务逻辑方法，也可能是依赖 DAO 组件实现的业务逻辑方法。
- 控制器层：此层由系列控制器组成，这些控制器用于拦截用户请求，并调用业务逻辑组件的业务逻辑方法，处理用户请求，并根据处理结果转发到不同的表现层组件。
- 表现层：此层由系列的 JSP 页面、Velocity 页面、PDF 文档视图组件组成。此层负责收集用户请求，并将显示处理结果。

大致上，Java EE 应用的架构如图 1.1 所示。