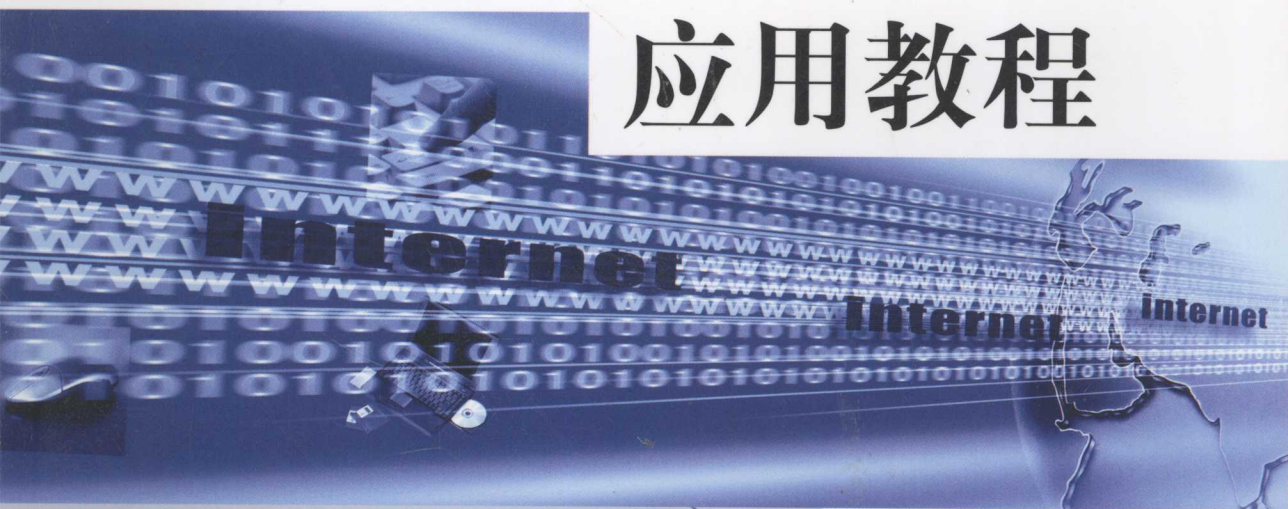


21世纪高职高专系列规划教材·计算机类专业
高职高专“十二五”规划教材

JSP基础与

应用教程



JSP JICHUYU
YINGYONGJIAOCHENG

主 编◎周 超
副主编◎董清潭 张洪生
王东燕

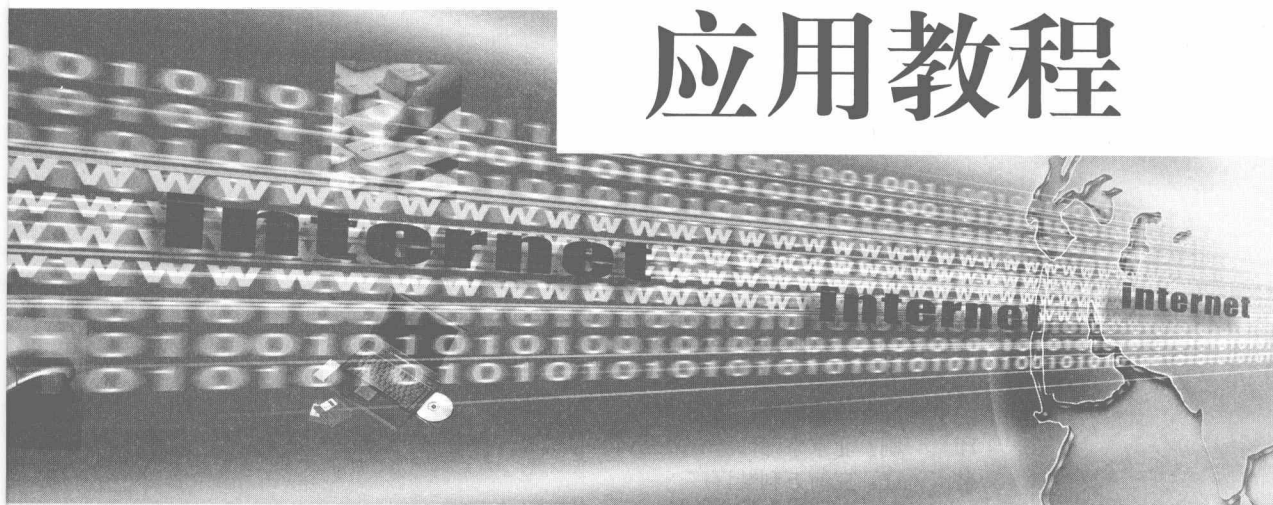


北京师范大学出版集团
BEIJING NORMAL UNIVERSITY PUBLISHING GROUP
北京师范大学出版社

21世纪高职高专系列规划教材·计算机类专业
高职高专“十二五”规划教材

JSP基础与

应用教程



JSP JICHU YU
YINGYONG JIAOCHENG

主 编◎周 超
副主编◎董清潭 张洪生
 王东燕
参 编◎童锡骏 何 凤
 葛宝利 陈 洁
 姚 策 杨 森
 (排名不分先后)



北京师范大学出版集团
BEIJING NORMAL UNIVERSITY PUBLISHING GROUP
北京师范大学出版社

图书在版编目(CIP) 数据

JSP基础与应用教程/周超主编. —北京: 北京师范大学出版社, 2011.9

(21 世纪高职高专系列规划教材)

ISBN 978-7-303-12828-0

I. ①J… II. ①周… III. ①JAVA 语言—网页制作工具—高等职业教育—教材 IV. ①TP312 ②TP393.092

中国版本图书馆 CIP 数据核字 (2011) 第 121348 号

223082

出版发行: 北京师范大学出版社 www.bnup.com.cn

北京新街口外大街 19 号

邮政编码: 100875

印 刷: 北京京师印务有限公司

装 订: 三河万利装订厂

经 销: 全国新华书店

开 本: 184 mm × 260 mm

印 张: 18

字 数: 270 千字

版 次: 2011 年 9 月第 1 版

印 次: 2011 年 9 月第 1 次印刷

定 价: 32.00 元

策划编辑: 周光明 责任编辑: 周光明

美术编辑: 高 霞 装帧设计: 国美嘉誉

责任校对: 李 菡 责任印制: 孙文凯

版权所有 侵权必究

反盗版、侵权举报电话: 010-58800697

北京读者服务部电话: 010-58808104

外埠邮购电话: 010-58808083

本书如有印装质量问题, 请与印制管理部联系调换。

印制管理部电话: 010-58800825

前言

随着互联网技术的发展,网络的应用越来越普及,掌握一种 Web 开发技术是能够适应时代潮流、迎接信息社会挑战的重要本领。JSP 是当前主流的 Web 开发技术,它是一种基于 Java 的服务器语言。

JSP 技术继承了 Java 语言的强大功能,由于 JSP 的强大功能和显著优势,JSP 技术已经成为网络时代的宠儿。应用 JSP 技术不仅能制作出具有各种功能的 Web 页面,也可以开发出基于 B/S 结构的先进、安全和跨平台的商务系统,并且开发出的系统具有良好的扩充性。

本书对 JSP、JavaScript 编程基础及 JSP 与 Servlet、JavaBean、数据库等技术的结合使用进行了详细介绍,并通过两个课程实例设计来使读者进行综合运用,从而能够深入地感受到它们的魅力所在,全书一共分为 12 章。

第 1 章主要介绍 JSP 的相关发展历史、优势;同时也对 Java 的主要开发工具及如何安装配置开发环境作了介绍。

第 2 章主要讲解了 Java 语言程序设计中涉及的 Java 语言基础、Java 语言面向对象编程基础、异常处理。

第 3 章主要通过介绍 JavaScript 的语法基础、程序结构以及常用的函数、事件和对象,来使读者掌握 JavaScript 的编程方法。

第 4 章主要介绍 JSP 页面的基本结构:包括 JSP 的注释、编译指令元素、动作指令元素和 JSP 代码元素。

第 5 章主要介绍 Servlet 与 JSP 的关系和工作原理;通过介绍 Servlet 的生命周期的几个方法,了解它的特点用法;如何开发 Servlet 和使用 Servlet 接口和类。

第 6 章主要介绍 JavaBean 的目标任务,以及如何创建 JavaBean 和设定 JavaBean 的相关属性,介绍 JavaBean 标签及如何使用 JavaBean。

第 7 章主要介绍 JSP 实用组件:通过 JSPSmartUpload,了解如何进行文件的上传;通过学习 JSPmail,了解如何进行邮件的收发;通过 JFreeChart,了解如何生成图形。

第 8 章主要介绍关系数据库的概念,SQL 基本操作;常见数据库管理系统;JSP 中数据库的使用。

第 9 章主要介绍 JSP 与 Ajax 技术的综合运用;EL 表达式及标签的使用;自定义标签的使用以及 JSP 框架技术。

第 10 章主要介绍一些常见的 Web 系统的安全问题,并分类介绍一些 Web 页面的

测试方法，以及如何优化 Web 系统的一些方法。

第 11 章是课程设计，主要通过一个在线投票系统实例的设计与实现来掌握 JSP、JavaBean 在 Java Web 项目中的应用以及 Java Web 项目中数据库的连接与应用。

第 12 章也是课程设计，通过设计一个留言板，来综合掌握 JSP 的应用。

本书通过理论知识与实例相结合的教学方法，深入浅出，循序渐进，最后通过两个课程设计来使读者对所学知识有更全面的掌握。

本书不仅可以作为高等院校、高职院校相关专业“十二五”规划教材，也可以作为 Web 技术开发人员的参考书。

本书由周超主编，董清潭、张洪生、王东燕担任副主编。编写人员分工如下：周超负责全书统稿稽核并编写了第 3、6、8 章；董清潭负责全书体系结构设计并编写了第 2、12 章；张洪生编写了第 1、4、5 章；天津职业大学的陈洁、天津轻工职业技术学院的姚策共同编写了第 7 章；童锡骏、王东燕共同编写了第 9 章；王东燕、葛宝利和天津电子信息职业技术学院的杨森共同编写了第 10 章；北京信息职业技术学院的何凤编写了第 11 章。

本书编写过程中，也得到了北京师范大学出版社领导和编辑及周光明的老师指导和帮助，谨此向他们表示最真挚的感谢！

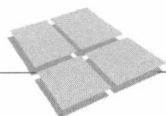
由于计算机技术发展十分迅速以及作者水平有限，加之时间仓促，书中难免存在疏漏和错误，敬请广大读者不吝赐教。

周超

2011 年 4 月

目 录

第 1 章 JSP 概述	(1)	3.3 JavaScript 的对象	(66)
1.1 JSP 简介	(1)	本章小结	(72)
1.2 JSP 所具有的优势	(1)	课后习题	(72)
1.3 JSP 开发模式	(3)	第 4 章 JSP 页面元素及内置对象	(73)
1.4 Java 开发工具	(4)	4.1 JSP 注释	(73)
1.5 Java Web 中间件介绍	(6)	4.2 JSP 编译指令	(75)
1.6 本书开发环境的安装与配置	(7)	4.3 JSP 动作指令	(79)
1.7 测试运行环境	(11)	4.4 隐含对象概述	(86)
本章小结	(12)	4.5 与 I/O 有关的隐含对象	(86)
课后习题	(12)	4.6 与上下文有关的隐含对象	(96)
第 2 章 Java 语言基础	(13)	4.7 与 Error 有关的隐含对象	(108)
2.1 标识符、注释、关键字	(13)	4.8 cookie 对象	(110)
2.2 基本数据类型及基本数据类型间的转换	(14)	本章小结	(112)
2.3 变量与常量	(16)	课后习题	(112)
2.4 数组	(16)	第 5 章 JSP 与 Servlet	(113)
2.5 字符串	(18)	5.1 Servlet 简介	(113)
2.6 运算符	(22)	5.2 Servlet 的生命周期	(115)
2.7 流程控制语句	(25)	5.3 如何开发 Servlet	(117)
2.8 Java 面向对象编程基础	(28)	5.4 使用 Servlet 接口和类	(120)
2.9 Java 实用集合	(36)	本章小结	(135)
2.10 网络编程基础	(43)	课后习题	(135)
本章小结	(49)	第 6 章 JSP 与 JavaBean	(136)
课后习题	(49)	6.1 JavaBean 概述	(136)
第 3 章 JavaScript 编程基础	(51)	6.2 创建 JavaBean	(137)
3.1 了解 JavaScript	(51)	6.3 JavaBean 的使用	(141)
3.2 JavaScript 编程基础	(53)		



6.4	JavaBean 的应用	(144)
	本章小结	(150)
	课后习题	(150)
第 7 章	JSP 实用组件	(151)
7.1	JSPSmartUpload	(151)
7.2	JSPmail	(159)
7.3	JFreeChart	(162)
	本章小结	(165)
	课后习题	(165)
第 8 章	JSP 与数据库	(166)
8.1	数据库基础	(166)
8.2	SQL 基本操作	(168)
8.3	常见数据库管理 系统	(172)
8.4	JDBC	(179)
8.5	JSP 中数据库的使用 ..	(184)
8.6	JSP 数据库编程实例 ..	(188)
	本章小结	(190)
	课后习题	(191)
第 9 章	JSP 高级程序设计	(192)
9.1	JSP 与 Ajax 技术	(192)
9.2	EL 表达式及标签	(197)
9.3	自定义标签	(208)
9.4	JSP 框架技术	(214)
	本章小结	(218)

	课后习题	(219)
第 10 章	Web 系统的安全测试与 优化	(220)
10.1	Web 系统的安全问题	(220)
10.2	Web 系统的测试	(223)
10.3	Web 系统的优化	(227)
	本章小结	(231)
	课后习题	(231)
第 11 章	课程设计一：在线投票 系统	(232)
11.1	系统开发的背景	(232)
11.2	系统的设计思路	(232)
11.3	详细设计	(234)
	本章小结	(248)
第 12 章	课程设计二：简单留 言板	(249)
12.1	留言板系统功能 分析	(249)
12.2	系统开发平台	(249)
12.3	系统详细设计	(249)
12.4	系统运行	(278)
	本章小结	(278)
	参考文献	(279)

第 1 章 JSP 概述

本章主要介绍 JSP 的相关发展历史、优势, 以及使用 JSP 技术开发 Web 应用程序的两种架构模式: JSP+JavaBean 模式和 JSP+JavaBean+Servlet 模式; 同时也对 Java 的主要开发工具、Java Web 中间件及如何安装配置开发环境作了介绍。

1.1 JSP 简介

JSP 是 Java Server Pages 的缩写, 是由 SUN Microsystems 公司于 1999 年 6 月开发出来的一种动态网页标准。它在传统的 HTML 代码中加入 Java 程序片段, 使用丰富的 JSP 指令构成 JSP 页面。程序片段可以操作数据库、重新定向网页, 以及发送 E-mail 等, 这就是建立动态网站所需要的功能。Web 服务器在遇到访问 JSP 网页的请求时, 首先执行其中的程序片段, 然后将执行结果以 HTML 格式返回给客户。所有程序操作都在服务器端执行, 网络上传送给客户端的仅是得到的结果, 对客户浏览器的要求最低。

在 SUN 正式发布 JSP 之后, 这种新的 Web 应用开发技术很快引起了人们的关注。JSP 为创建高度动态的 Web 应用提供了一个独特的开发环境。JSP 使得我们能够分离页面的静态 HTML 和动态 HTML 部分。HTML 可以用任何通常使用的 Web 制作工具编写, 编写方式也和原来的一样: 动态部分的代码放入特殊标记之内, 大部分以“<%”开始, 以“%>”结束。

1.2 JSP 所具有的优势

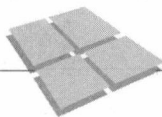
JSP 技术是由 Servlet 技术发展起来的, 自从有了 JSP 后, 在 Java 服务器端编程中普遍采用的就是 JSP, 而不是 Servlet。因为 JSP 在编写表示页面时远远比 Servlet 简单, 并且不需要手工编译(由 Servlet 容器自动编译), 目前 Servlet 主要用做视图控制器、处理后台应用等。由于 JSP 构建在 Servlet 上, 所以它有 Servlet 所有强大的功能。

在开发 JSP 规范的过程中, Sun 公司与许多主要的 Web 服务器、应用服务器和开发工具供应商积极进行合作, 不断完善技术。JSP 基于强大的 Java 语言, 具有良好的伸缩性, 与 Java Enterprise API 紧密地集成在一起, 在网络数据库应用开发领域具有得天独厚的优势。从这几年的发展来看, JSP 已经获得巨大的成功, 它通过和 EJB 等 Java EE 组件进行集成, 可以编写出处理伸缩性大、负载高的企业级应用。JSP 技术在多个方面加速了动态 Web 页面的开发。

JSP 在跨平台、执行速度等特性上具有很大的技术优势, 主要体现在以下方面。

1. 将内容的生成和显示进行分离

使用 JSP 技术, Web 页面开发人员可以使用 HTML 或者 XML 标识来设计和格式化最终页面。使用 JSP 标识或者小脚本来生成页面上的动态内容。生成内容的逻辑被



封装在标识和 JavaBean 组件中,并且捆绑在脚本中,所有的脚本在服务器端运行。如果核心逻辑被封装在标识和 Bean 中,那么其他人(如 Web 管理人员和页面设计者)能够编辑和使用 JSP 页面,而不影响内容的生成。

在服务器端,JSP 引擎解释 JSP 标识和小脚本,生成所请求的内容(例如,通过访问 JavaBean 组件,使用 JDBC 技术访问数据库或者包含文件),并且将结果以 HTML 或者 XML 页面的形式发送回浏览器。这有助于作者保护自己的代码,而又保证任何基于 HTML 的 Web 浏览器的完全可用性。

2. 生成可重用的组件

绝大多数 JSP 页面依赖于可重用的、跨平台的组件(JavaBean 或者 Enterprise JavaBean 组件)来执行应用程序所要求的更为复杂的处理。开发人员能够共享和交换执行普通操作的组件,或者使得这些组件为更多的使用者或客户团体使用。基于组件的方法加速了总体开发过程,并且使得各种组织在他们现有的技能和优化结果的开发努力中得到平衡。

3. 采用标识简化页面

Web 页面开发人员不一定是熟悉脚本语言的编程人员。JSP 技术封装了许多功能,这些功能是在易用的、与 JSP 相关的 XML 标识中进行动态内容生成时所需的。标准的 JSP 标识能够访问和实例化 JavaBean 组件,设置或者检索组件属性,下载 Applet,以及执行其他方法更难于编码或耗时的功能。通过开发定制标识库,JSP 技术是可扩展的。今后,第三方开发人员和其他人员可以为常用功能创建自己的标识库。这使得 Web 页面开发人员能够使用熟悉的工具和如同标识一样执行特定功能的构件来工作。

4. JSP 能提供所有的 Servlet 功能

与 Servlet 相比,JSP 能提供所有的 Servlet 功能,它比用 println 书写和修改 HTML 更方便。此外,可以更明确地进行分工,Web 页面设计人员编写 HTML,只需要留出空间让 Servlet 程序员插入动态部分即可。

5. 稳定的存储管理和安全性

由于 JSP 页面的内置脚本语言是基于 Java 语言编写的,而且所有的 JSP 页面都被编译成为 Java Servlet,JSP 页面就具有 Java 技术的所有优点,包括稳定的存储管理和安全性。

6. 一次编写,随处运行

作为 Java 平台的一部分,JSP 拥有 Java 编程语言“一次编写,随处运行”的特点。随着越来越多的供应商将 JSP 支持添加到他们的产品中,用户可以使用自己所选择的服务器和工具,但并不影响当前的应用。

7. JSP 的平台适应性更广

这是 JSP 比 ASP 优越之处。几乎所有平台都支持 Java JSP+JavaBean,它们可以在任何平台下通行无阻。Windows NT 下的 IIS 通过一个插件就能支持 JSP,使用 JRUN 或者 ServletExec,著名的 Web 服务器 Apache 已经能支持 JSP。由于 Apache 广泛应用在 Windows、UNIX 和 Linux 上,因此 JSP 有更广泛的运行平台。从一个平台移植到另一个平台,JSP 和 JavaBean 甚至不用重新编译,因为 Java 字节码都是标识的

字节码，与平台无关。

8. Java 中连接数据库的技术是 JDBC(Java Database Connectivity)

很多数据库系统带有 JDBC 驱动程序，Java 程序通过 JDBC 驱动程序与数据库相连，执行查询、提取数据等操作。SUN 公司还开发了 JDBC-ODBC Bridge，用此技术 Java 程序就可以访问带有 ODBC 驱动程序的数据库，目前大多数数据库系统都带有 ODBC 驱动程序，所以 Java 程序能访问诸如 Oracle、Sybase、MS SQL Server 和 MS Access 等类型的数据库。

9. 简单易学

随着 JSP2.0 中标记语言的出现，程序员即使不懂 Java，也能编写出功能完善的 JSP 应用。

1.3 JSP 开发模式

使用 JSP 技术开发 Web 应用程序，有两种架构模式可供选择，通常称为模式 1 和模式 2。下面我们将对这两种模式进行介绍，并说明两种架构模式各自的优缺点及它们应用的场合。

1.3.1 JSP+JavaBean 模式

模式 1 使用 JSP+JavaBean 技术将页面显示和业务逻辑处理分开。JSP 实现页面的显示，JavaBean 对象用来保存数据和实现业务逻辑。在模式 1 中，JSP 页面独自响应请求并将处理结果返回给客户，所有的数据通过 JavaBean 来处理，JSP 实现页面的显示。模式 1 的结构如图 1-1 所示。

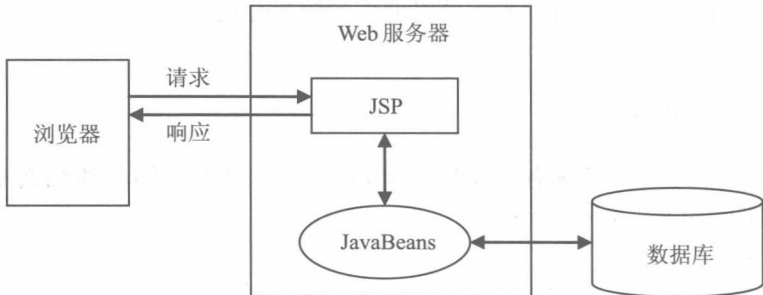


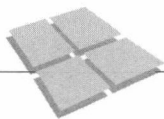
图 1-1 模式 1 的 JSP 结构

模式 1 实现了页面显示和业务逻辑的分离。然而这种方式需要在 JSP 页面中嵌入流程控制和调用 JavaBean 的代码，当需要处理的业务逻辑非常复杂时，这种情况会变得更加糟糕。在 JSP 页面中嵌入大量的 Java 代码将会使程序变得异常复杂，对于前端页面设计人员来说，大量的嵌入代码使他们无从下手。

模式 1 不能满足大型应用的需要，但是对于小型应用，它却是一个不错的选择。因为该模式简单，不涉及过多的要素，从而可以很好地满足小型应用的需要。

1.3.2 JSP+JavaBean+Servlet 模式

在模式 1 中，JSP 页面嵌入了流程控制代码和部分逻辑处理代码，我们可以将这部



分代码提取出来，放到一个单独的角色中，这个角色就是控制器，而这样的 Web 架构就是所谓的模式 2，它符合 MVC 架构模式。

在模式 2 中，控制器的角色由 Servlet 来实现，视图的角色由 JSP 页面来实现，模型的角色由 JavaBeans 来实现。模式 2 的架构如图 1-2 所示。

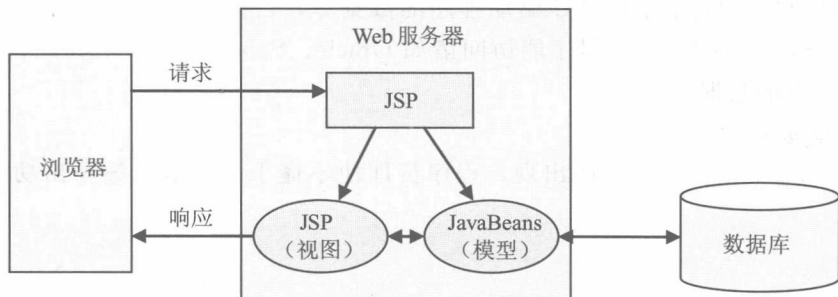


图 1-2 模式 2 的架构

采用模式 2 的架构，可以将页面显示、业务逻辑处理和流程控制很清晰地区分开，JSP 负责数据显示，JavaBean 负责业务逻辑处理，Servlet 负责流程控制。采用模式 2 架构的 Web 应用程序很容易维护和扩展，因为作为视图的 JSP 页面之间没有直接的关联。另外，在大型项目的开发过程中，采用模式 2 的架构，可以充分利用前端的页面设计人员和后端的 Java 开发人员，由页面设计人员发挥自己的美术和设计才能来表现页面，由程序编写人员发挥自己的业务逻辑处理思维，实现项目中的业务处理。

在实际应用中，具体采用哪种模式要根据实际业务需求来确定。通常情况下，对于小型的、业务逻辑处理不多的应用，采用模式 1 比较合适。如果应用有着较复杂的逻辑，并且返回的视图也不同，那么采用模式 2 较为合适。

1.4 Java 开发工具

JSP 技术常用的开发工具有：JDK、Eclipse、Jbuilder、JRun 和纯文本编辑器等，本书只介绍 JDK 和 Eclipse。

1.4.1 JDK

JDK(Java Development Kit)是 SUN Microsystems 为 Java 开发人员提供的软件开发工具包，是整个 Java 的核心，包括 Java 运行环境(Java Runtime Environment, JRE)、Java 工具及 Java 基础类库(rt.jar)等。自从 Java 推出以来，JDK 已经成为使用最广泛的 Java SDK(Software Development Kit)。

JDK 包含的基本组件有：

- javac，编译器，将原程序转换成字节码。
- jar，打包工具，将相关的类文件打包成一个文件。
- javadoc，文档生成器，负责从源代码注释中提取帮助文档。
- jdb，调试工具。
- java，运行编译后的 java 程序(.class 后缀的)。

- appletviewer, 小程序浏览器, 一种执行 HTML 文件上的 Java 小程序的 Java 浏览器。

- Javah, 产生可以调用 Java 过程的 C 过程, 或建立能被 Java 程序调用的 C 过程的头文件。

- Javap, Java 反汇编器, 显示编译类文件中的可访问功能和数据, 同时显示字节代码含义。

- Jconsole, Java 进行系统调试和监控的工具 JDK 中还包括各种例程, 用以展示 Java API 中的各部分。

JDK 一般有三种版本:

- J2SE(Java 2 Standard Edition, Java2 标准版), 主要用于创建典型桌面与工作站应用的 Java 平台。

- J2ME(Java 2 Micro Edition, Java2 微型版), 主要用于创建嵌入式应用程序(如 PDA、电子仪表等), 需要 J2SE 平台的支持。

- J2EE(Java 2 Enterprise Edition, Java2 企业版), 提供一个平台独立的、便携式、多用户、安全和标准的企业级平台。

可以从 SUN 的官方网站 <http://java.sun.com> 上下载最新版本的 JDK 工具包。

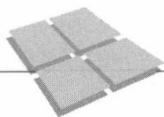
1.4.2 Eclipse

Eclipse 是一个开放源代码的, 基于 Java 的可扩展的软件开发项目, 其目标是为高度集成的工具开发提供一个全功能的、具有商业品质的工业平台。其主要由 Eclipse 项目、Eclipse 工具项目和 Eclipse 技术项目三部分组成, 本节中只对 Eclipse 开发工具的特点进行介绍。

Eclipse 最初由 OTI 和 IBM 两家公司的 IDE 产品开发组于 1999 年 4 月创建。IBM 提供了最初的 Eclipse 代码基础, 包括 Platform、JDT 和 PDE。目前围绕着 Eclipse 项目已经发展成为一个由 IBM 牵头、150 多家软件公司参与其中的庞大的 Eclipse 联盟, 其中不乏 Borland、Sybase、Rational Software、Oracle 及 Red Hat 这样的知名软件企业。

就 Eclipse 本身而言, 它只是一个框架和一组服务, 用于通过插件组件构建开发环境。Eclipse 附带了一个标准的插件集和插件开发环境(Plug-in Development Environment, PDE), 插件集中包括 Java 开发工具(Java Development Tools, JDT), PDE 主要针对希望扩展 Eclipse 的软件开发人员, 允许他们构建与 Eclipse 环境无缝集成的工具。由于 Eclipse 中的每样东西都是插件, 对于给 Eclipse 提供插件以及给用户提供一致和统一的集成开发环境而言, 所有工具开发人员都具有同等的发挥场所。而且这种平等和一致性并不仅限于 Java 开发工具。Eclipse 框架还可以用来作为与软件开发无关的其他应用程序类型的基础, 比如内容管理系统。基于 Eclipse 的应用程序的突出例子是 IBM 的 WebSphere Studio Workbench, 它构成了 IBM Java 开发工具系列的基础。例如 WebSphere Studio Application Developer 添加了对 JSP、Servlet、EJB、XML、Web 服务和数据库访问的支持。

Eclipse 的插件结构非常类似于 Web 浏览器, 都是要解决扩展性问题。通过 E-



clipse 的插件模式，我们可以安装不同的插件来进行 HTML 的编辑、数据库的设计、Java 开发、C++ 开发等，所有这些开发工作完全在一个 Eclipse 安装平台上进行。理解 Eclipse 插件的关键是首先要理解 Eclipse 扩展点，当 Eclipse 插件实现了一个扩展点，就创建了一个扩展。此外，扩展还可以创建自己的扩展点，这种插件模式的扩展和扩展点是递归的。实际上，Eclipse 本身可以被看作是一系列的插件。

Eclipse 的插件体系架构决定了 Eclipse 的高度可扩展性。作为 IDE，在各种插件的帮助下，它不仅可以支持各种语言(Java、C/C++、PHP、Perl、Python 等)的开发，还支持软件开发过程中的各种开发活动(设计建模、测试、编译构建、插件开发、J2EE 开发、GUI 开发、数据库设计等)；此外，Eclipse 还可以用作通用的应用平台——富客户端平台。

Eclipse 的下载和安装非常简单，读者可以到 <http://www.eclipse.org> 下载，这里不再赘述。

1.5 Java Web 中间件介绍

常用 Java 的中间件(服务器)种类很多，如 Jakarta 项目组的 Apache Tomcat 服务器、Caucho 的 Resin 服务器、BEA 公司的 WebLogic 服务器以及开源项目 SourceForge 的 JBoss 服务器和 IBM 的 WebSphere 服务器等，下面分别对各种服务器进行简单的介绍。

1.5.1 Tomcat

Tomcat 是一款比较优秀的 JSP 服务器，不但支持 JSP 和 Servlet 的开发，而且还具备了商业 Web 应用容器的特征，如 Tomcat 管理和控制平台、安全域管理以及 Tomcat 阀等。Tomcat 已成为当前开发企业 Java Web 应用程序的服务器选择之一。但由于 Tomcat 不包含 EJB 容器，所以不能支持 EJB 组件的部署和运行。

1.5.2 WebLogic

WebLogic 服务器美国 BEA 公司出品的一个 WebServer(是构建网站的必要软件，用来解析发布网页等)，是一款功能丰富、基于标准的应用服务器，支持面向企业级的、多层次的和完全分布式的 Web 应用，是一种用于开发、集成、部署和管理大部分分布式 Web 应用、网络应用及数据可应用的 Java 应用服务器。确切地说，WebLogic Server 是一个基于 J2EE 架构的中间件，拥有处理关键 Web 应用系统问题所需的性能、可扩展性和高可用性，与 BEA WebLogic Commerce Server 配合使用，WebLogic Server 可为部署适应于个性化的电子商务应用系统提供完善的解决方案。相对于其他种类的服务器，WebLogic 具有快速开发、部署灵活等特点。

1.5.3 Resin

Resin 服务器平台提供了最快的 JSP/Servlet 运行平台。在 Java 和 JavaScript 的支持下，Resin 可以为任务灵活选择合适的开发语言。如果选择 JSP 作为 Internet 商业站点的支持，那么速度、价格和稳定性都要考虑，Resin 在上述方面的表现十分出色，充分具备商业软件的要求。Resin 是全免费的，而且从站点下载的就是完整的版本。相对

于其他的服务器, Resin 是一个比较简单的服务器, 设置的地方比较少, 所带的附件比较少, 运行速度快。目前 Resin 可以支持 SUN 的 J2EE, 而 J2EE 是基于 Java 服务器端大系统的基础, 并且 Resin 支持 JSP 和在服务器端编译的 JavaScript。

1.6 本书开发环境的安装与配置

1.1 节和 1.2 节所介绍的开发工具和 Web 中间件, 任意一对组合起来都可以开发 JSP 和 Servlet 程序, 常用的如记事本(JDK)+Tomcat、Eclipse+Tomcat、Eclipse+Resin、Jbuilder+WebLogic 等。

Tomcat 是 Jakarta 项目中一个重要的子项目, 也是 SUN 公司官方推荐的 Servlet 和 JSP 容器, 比较适合于初学者, 也越来越受到软件公司和开发人员的喜爱。Servlet 和 JSP 的最新规范都可以在 Tomcat 最新版本中实现。本书采用的操作系统是 Windows XP, 开发环境是 JDK+Tomcat。

1.6.1 安装和配置 JDK

可以从 SUN 的官方网站 <http://java.sun.com> 上下载最新版本的 JDK 工具包。安装过程非常简单, 在此恕不赘述。

在安装好 JDK 后, 还需要进行一些配置才能继续后面的应用程序开发。具体配置步骤如下:

在 Windows 桌面上, 右击【我的电脑】图标, 在弹出菜单中选择【属性】, 将弹出【系统属性】对话框。

在【系统属性】对话框中选择【高级】选项卡, 单击【环境变量】按钮将弹出【环境变量】对话框。

在【环境变量】对话框中单击【系统变量】选项区【新建】按钮, 将打开【新建系统变量】对话框。

在【新建系统变量】对话框中的【变量名】文本框中输入 JAVA_HOME, 在【变量值】文本框中输入 c:\program Files\java\jdk1.6.0_07(注: 输入项为用户的安装路径), 如图 1-3 所示。

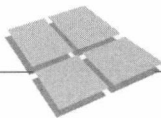


图 1-3 添加名为 JAVA_HOME 的系统环境变量

(1)按照同样方式创建系统变量 CLASSPATH, 变量值为:

.;C:\Program Files\java\jdk1.6.0_07\lib\dt.jar;D:\Program Files\java\jdk1.6.0_01\lib\tools.jar;

其中, 首字符“.”表示在当前目录下, 而“;”是各部分的分隔符。



(2)另外,要在已有的系统变量 PATH 的变量之中添加以下变量值:

C: \ Program Files \ java \ jdk1.6.0_07 \ bin;

注意,最后的“;”不要忘记输入。

这样, JDK 在 Windows XP 下的安装配置就完成了,在 Windows2000/NT 系统上的安装配置过程完全相同。接下来就可以安装 Tomcat 了。

1.6.2 下载和安装 Tomcat

Tomcat 的版本很多,最新版本为 6.0 版。读者可以从 <http://tomcat.apache.org> 上下载该版本。Tomcat 的安装非常简单,安装过程中均选择默认值即可,这里不再赘述。

1. 测试 Tomcat

安装完毕后,读者可以打开网页浏览器,在地址栏中输入 <http://localhost:8080>,如果看到如图 1-4 所示的 Tomcat 相关信息,则表示 Tomcat 已成功安装。

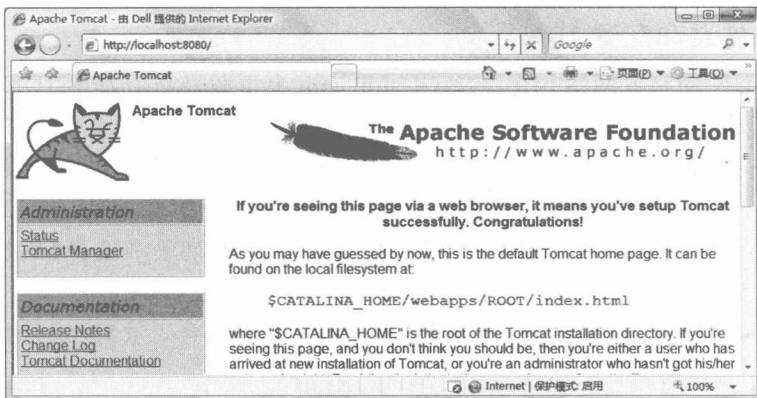


图 1-4 Tomcat 成功安装后的相关信息界面

2. Tomcat 目录结构介绍

Tomcat 成功安装后的目录层次结构如图 1-5 所示,下面对各目录进行简单说明。

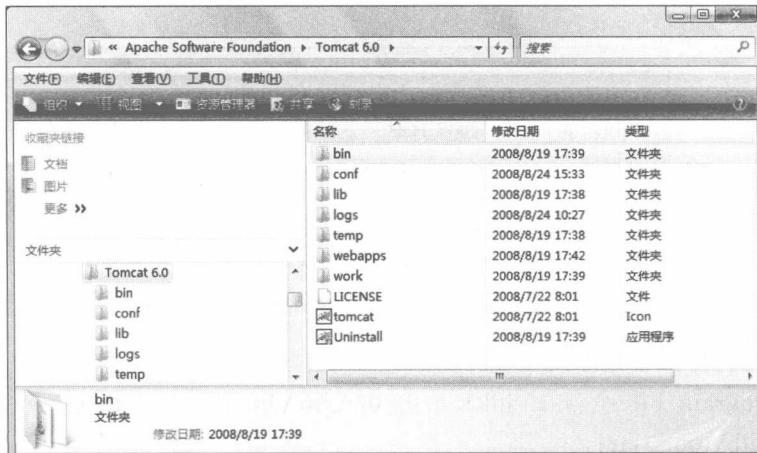


图 1-5 Tomcat 目录层次结构

- bin, 用于存放启动和关闭 Tomcat 的脚本文件。
- conf, 用于存放 Tomcat 服务器的各种配置文件, 其中包括 Server.xml(Tomcat 的主要配置文件)、tomcat-user.xml 和 web.xml 等配置文件。
- lib, 用于存放 Tomcat 服务器及所有 Web 应用程序都可以访问的 JAR 文件。
- logs, 用于存放 Tomcat 的日志文件。
- temp, 用于存放 Tomcat 运行时产生的临时文件。
- webapps, 用于存放要发布的 Web 应用程序的目录及其文件。
- work, 用于存放 JSP 生成的 Servlet 源文件和字节码文件。

需要说明的是, 对后面要介绍的 JavaWeb 应用程序, 在它的 Web_INF 目录下也可以有 lib 子目录, 用以存放各种 JAR 文件, 但与直接存放在图 1-5 所示的 lib 目录下的 JAR 文件不同, 这些 JAR 文件只能被当前的 Web 应用程序所访问。

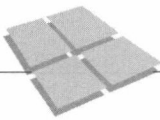
3. Tomcat 配置文件

Tomcat 提供了一系列的配置文件帮助用户配置自己的 Tomcat 服务器, 其中多数文件是基于 XML 的, 如 Server.xml、web.xml 等, 只有 catalina.properties 和 logging.properties 等少数文件是传统的配置文件, 这也为用户更加方便地修改配置文件提供了可能。下面介绍 Tomcat 的主要配置文件。

(1) Server.xml。

在 Server.xml 文件中包含了许多元素, 主要包括:

- Server, 它是整个的 Catalina Servlet 容器。Tomcat 提供了 Server 接口的一个默认值实现, 通常不需要用户自己去实现。在 Server 容器中可以包含一个或多个 Service 组件。
- Service, 它是存活于 Server 中的内部组件, 它将一个或多个连接器(Connector)组件绑定到一个单独的引擎(Engine)上。在 Server 中, 可以包含一个或多个 Service 组件。Service 很少由用户定制, Tomcat 提供了 Service 接口的默认实现。
- Connection, 称为连接器, 用于处理与客户端的通信, 负责接收客户请求, 向用户返回响应结果。在 Tomcat 中有多个连接器可用。
- Context, 一个 Context 表示一个 Web 应用程序, 运行在特定的虚拟主机中, 它可以在多个供应商提供的实现了 Servlet 规范的 Web 容器中运行。一个 Host 可以包含多个 Context, 每一个 Context 都有一个唯一的路径, 用户通常无须创建自定义的 Context, 因为 Tomcat 给出的 Context 接口实现类(StandardContext)提供了重要的附加功能。
- Host, 它表示一个虚拟主机, 一个引擎可以包含多个 Host。用户通常不需要创建自定义的 Host, 因为 Tomcat 提供的 Host 实现类(StandardHost)提供了重要的附加功能。
- Engine, 在 Tomcat 中每个 Service 只能包含一个 Servlet 引擎(Engine)。Engine 表示一个特定 Service 的请求处理流水线。作为一个 Service 可以有多个连接器, 引擎从连接接收和处理所有的请求, 将响应返回给适当的连接器, 通过连接器传输给用户。用户可以通过实现 Engine 接口提供自定义引擎。
- Logger, 表示日志调试和错误信息。
- Realm, 表示存放用户名、密码及角色的数据库。



● Value，功能与 Logger 相似，其 prefix 和 suffix 属性的解释和 logger 中的一样。

Server. xml 文件元素包含了一些属性，这些属性用于配置 Tomcat。表 1-1 为各元素的属性及其说明。

表 1-1 Server. xml 元素属性及其说明

元素名	属性	说明
server	port	指定一个端口，负责监听关闭 Tomcat 的请求
	shutdown	指定向端口发送的命令字符串
service	name	指定 Service 的名字
	port	指定服务器端所要创建的端口号，并在这个端口监听来自客户端的请求
	minThreads	服务器启动时创建的用于处理请求的最小线程数
	maxThreads	服务器启动时创建的用于处理请求的最大线程数
	enablelookups	如果为 True，则可以通过调用 Request.getRemoteHost() 进行 DNS 查询来得到远程客户端的实际主机名；若为 False，则不进行 DNS 查询，而返回其 IP 地址
	redirectPort	指定服务器正在处理 Http 请求时，若收到一个 SSL 传输请求，将重定向的端口号
	acceptCount	指定当所有可以使用的处理请求线程都被使用后，可以放到等待处理队列中的请求数上限，超过上限的请求将不予处理
	connectionTimeout	指定连接时的超时时间，单位毫秒
	defaultHost	指定默认的处理请求的主机名，它至少应该与其中的一个 host 元素的 name 属性值是相同的
Host	name	指定主机名
	appBase	应用程序基本目录，即存放应用程序的目录
	unpackWARS	如果为 True，则 Tomcat 自动将 War 文件解压；否则不解压，直接从 War 文件中运行应用程序
Logger	className	指定 Logger 使用的类名，此类必须实现 org.apache.catalina.logger 接口
	prefix	指定 log 文件的前缀
	suffix	指定 log 文件的后缀
	timestamp	如果为 true，则 log 文件命中要加入时间，如 localhost_log.2008-08-08.txt
Realm	classname	指定 realm 使用的类，此类必须实现 org.apache.catalina.realm 接口
Value	directory	指定 log 文件存放的位置
	pattern	取值为 common 时，记录远程主机或 IP 地址、用户名、日期、第一行请求的字符串、Http 响应代码、发送的字节数 取值为 combined 时，将获得比 common 方式更多的记录信息