

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

JSP 程序设计

JSP Programming

范立锋 乔世权 程文彬 主编

- 经典案例，大量习题，理论结合实践
- 循序渐进，合理架构，便于学习理解
- 推陈出新，扩展思路，提升实用价值



高校系列

人民邮电出版社
POSTS & TELECOM PRESS

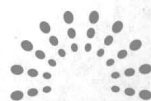
21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

JSP 程序设计

JSP Programming

范立锋 乔世权 程文彬 主编



高校系列

人民邮电出版社

北京

图书在版编目 (CIP) 数据

JSP程序设计 / 范立锋, 乔世权, 程文彬主编. —北京:
人民邮电出版社, 2009. 5
21世纪高等学校计算机规划教材. 高校系列
ISBN 978-7-115-19780-1

I. J… II. ①范…②乔…③程…III. JAVA语言—主页制作
—程序设计—高等学校—教材 IV. TP393.092

中国版本图书馆CIP数据核字 (2009) 第026774号

内 容 提 要

本书系统地介绍了 JSP 技术的概念、方法与实现过程, 包括 JSP 运行环境、JSP 语法与组成元素、JSP 内置对象、JSP 对数据库的操作、JSP 对 JavaBean 和 Servlet 的调用、JSP 核心表达式与标签、Web 网页模板技术、JSP 实用组件技术和 MVC 设计模式等, 最后还介绍了论坛和购物车两个 JSP 综合实例。通过本书的学习, 读者可以系统地掌握 JSP 技术相关概念、方法、编程思路和技巧。本书不求面面俱到, 也不追求博大精深, 仅要求能够使读者在最短的时间内学会用 JSP 开发动态网站的编程技术。

本书可作为普通高等院校计算机及相关专业“Web 程序设计”、“JSP 程序设计”、“动态网站设计”等课程的教材, 也可作为 JSP 初学者及网站开发人员的参考用书。

21 世纪高等学校计算机规划教材——高校系列

JSP 程序设计

-
- ◆ 主 编 范立锋 乔世权 程文彬
责任编辑 滑 玉
执行编辑 刘 博
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 17.25
字数: 447 千字 2009 年 5 月第 1 版
印数: 1—3 000 册 2009 年 5 月北京第 1 次印刷

ISBN 978-7-115-19780-1/TP

定价: 29.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

出版者的话

现今社会对人才的基本要求之一就是应用计算机的能力。在高等学校,培养学生应用计算机的能力,主要是通过计算机课程的体制改革,即计算机教学分层、分类规划与实施;密切联系实际,恰当体现与各专业其他课程配合;教学必须以市场需求为导向,目的是培养高素质创新型人才。

人民邮电出版社经过对教学改革新形势充分的调查研究,依据目前比较成熟的教学大纲,组织国内优秀的有丰富教学经验的教师编写一套体现教学改革最新形势的“高校系列计算机教材”。在本套教材的出版过程中,我社多次召开教材研讨会,广泛听取了一线教师的意见,也邀请众多专家对大纲和书稿做了认真的审读与研讨。本套教材具有以下特点。

1. 覆盖面广,突出教改特色

本套教材主要面向普通高等学校(包括计算机专业和非计算机专业),是在经过大量充分的调研基础上开发的计算机系列教材,涉及计算机教育领域中的所有课程(包括专业核心骨干课程与选修课程),适应了目前经济、社会对计算机教育的新要求、新动向,尤其适合于各专业计算机教学改革的特点特色。

2. 注重整体性、系统性

针对各专业的特点,同一门课程规划了组织结构与内容不同的几本教材,以适应不同教学需求,即分别满足不同层次计算机专业与非计算机专业(如工、理、管、文等)的课程安排。同时本套教材注重整体性的策划,在教材内容的选择上避免重叠与交叉,内容系统完善。学校可根据教学计划从中选择教材的各种组合,使其适合本校的教学特点。

3. 掌握基础知识,侧重培养应用能力

目前社会对人才的需要更侧重于其应用能力。培养应用能力,须具备计算机基础理论、良好的综合素质和实践能力。理论知识作为基础必须掌握,本套教材通过实践教学与实例教学培养解决实际问题的能力和知识综合运用能力。

4. 教学经验丰富的作者队伍

高等学校在计算机教学和教材改革上已经做了大量的工作,很多教师在计算机教育与科研方面积累了相当多的宝贵经验。本套教材均由有丰富教学经验的教师编写,并将这些宝贵经验渗透到教材中,使教材独具特色。

5. 配套资源完善

所有教材均配有PPT电子教案,部分教材配有实践教程、题库、教师手册、学习指南、习题解答、程序源代码、演示软件、素材、图书出版后要更新的内容等,以方便教与学。

我社致力于优秀教材的出版,恳请大家在使用的过程中,将发现的问题与提出的意见反馈给我们,以便再版时修改。

人民邮电出版社

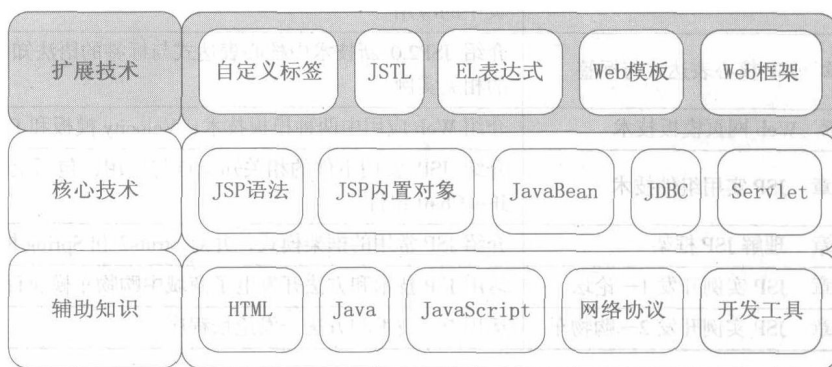
前言

JSP (Java Server Pages) 是近年来发展最迅速、最引人注目的 Web 应用开发技术之一,它是 Java Platform, Enterprise Edition (Java EE, Java 企业版) 的重要技术。JSP 将 Java 语言的跨平台和开放性、Servlet 的强大功能与 HTML 以及脚本语言等简单易用的元素结合起来,解决了过去 Web 开发技术存在的各种不足和局限。

本书是作者在总结了多年开发经验与成果的基础上编写的。书中全面、翔实地介绍了 JSP 开发所需的各种知识和技巧。通过本书的学习,读者可以快速、全面地掌握使用 JSP 开发 Web 应用程序的方法,并且可以达到融会贯通、灵活运用之目的。

JSP 知识体系

JSP 的知识体系如下图所示。



JSP 知识体系图

本书特点

(1) 知识体系结构合理。知识点的安排强调整体性和系统性,知识点的表达强调层次性和有序性,便于读者学习和理解。

(2) 理论与应用紧密结合。在每一章节,首先对相关知识点进行解释,然后用所学知识点和技术构建应用,通过实例来理解知识点,通过应用来熟悉技术。

(3) 语言通俗易懂,读者容易理解。书中采用程序结构、页面交互图、流程图、表格等多种方式描述问题及讲解解决问题的过程,读者可以从多个角度理解技术和知识点。

(4) 新的知识点。与其他教材不同的是,本书还加入了 Web 应用新的知识点。例如,本书第 9 章中介绍的 Web 模板技术,主要介绍了两种流行的模板技术:Velocity 模板和 FreeMarker 模板。模板语言在现代软件开发中占据着重要的地位,它的功能强大,而且学习起来又非常简单,即使不熟悉编程的人也能很快地掌握它。

(5) 扩展知识点。本书在介绍数据库连接池时,不仅介绍了连接池的原理与应用,而且还介绍了一些常用的连接池组件,使读者扩展思路,了解连接池不仅可以通过单一技术实现,而且可以通过多种技术进行实现。

(6) 案例的实用性。本书最后两个案例是 Web 典型应用:购物车和论坛,前者用到的技术是“Servlet+MYSQL 5.0+JSP 表达式”,后者用到的技术是“Servlet+SQL Server 2000+JSTL”。通过两个不同的数据库介绍了两个案例的开发,使读者直接获取项目的开发经验。

本书结构

本书围绕 JSP 动态网页制作循序渐进地介绍相关知识，书中给出了许多实例，而且每章后附有习题，用来巩固所学内容。全书共分 13 章，各章具体内容如下表所示。

章 名	描 述
第 1 章 JSP 初步	介绍 JSP 技术、优点、缺点及应用
第 2 章 JSP 辅助知识	介绍学习 JSP 之前必须掌握的辅助知识，主要包括 HTML、JavaScript、网络协议
第 3 章 JSP 语法详解	介绍在 JSP 页面编写 Java 代码的语法知识
第 4 章 JSP 内置对象详解	介绍 JSP 页面中 9 个内置对象的方法与应用，并且给出相关实例
第 5 章 JavaBean 组件技术	介绍 JavaBean 在 Web 开发过程中的实质，并介绍 JavaBean 绑定属性的特点与应用
第 6 章 Servlet 核心技术	介绍 Servlet 相关的知识点，并同时讲解 Servlet 过滤器与监听器在实际开发中的作用
第 7 章 JSP 数据库核心技术	介绍 JDBC 操作数据库中 SQL 语句的方法，并且介绍 JDBC 连接池在网页中的应用
第 8 章 JSP 核心表达式与标签	介绍 JSP2.0 新技术中核心表达式与标签的语法知识，并且给出各种标签的相关实例
第 9 章 Web 网页模板技术	介绍 Web 应用中两种模板技术：Velocity 模板和 FreeMarker 模板
第 10 章 JSP 实用组件技术	介绍 JSP 实用组件的相关知识点与应用，包括上传组件、Email 组件、JfreeChart 组件
第 11 章 理解 JSP 框架	介绍 JSP 常用的框架模式，并对 Struts2 和 Spring 框架有一个初步的了解
第 12 章 JSP 实例开发 1—论坛	运用 JSP 技术和方法开发电子商城中购物车模块程序
第 13 章 JSP 实例开发 2—购物车	运用 JSP 技术和方法开发论坛程序

本书面向的读者

本书可作为普通高等院校计算机及相关专业“Web 程序设计”、“JSP 程序设计”、“动态网站设计”等课程的教材，同时也可作为 JSP 初学者及网站开发人员的参考用书。

如果读者具备以下方面的知识，学习起来将会事半功倍。

- ★ 熟悉 HTML
- ★ 熟悉 Java 语言
- ★ 熟悉 Web 开发流程

本书作者

本书由范立锋主编，参加本书编写的主要人员有：王毅、李世欣、于琦、程峰、吕正超、吴新伟、霍晶馨、唐瑶等。

本书实例的开发环境

本书用到了 J2SE、Apache Tomcat、Eclipse 等软件，读者可以到相关网站下载。书中实例使用的是 SQL Server 2000 和 MYSQL 5.0 数据库，因此要查看或修改数据库，计算机上必须安装这两个数据库。

技术支持

本书实例开发中用到的程序源代码，可以在“人民邮电出版社教学服务与资源网（www.ptpedu.com.cn）”上免费下载，供读者学习和使用。

编 者

2008 年 12 月

目 录

第 1 章 JSP 初步	1	小结	34
1.1 认识 JSP	1	习题	34
1.2 JSP 技术特性	2	第 3 章 JSP 语法详解	35
1.3 JSP 工作原理	3	3.1 JSP 文件的组成	35
1.4 搭建 JSP 的运行环境	5	3.1.1 JSP 页的创建	35
1.4.1 JDK 的安装与配置	5	3.1.2 JSP 文件的组成元素	36
1.4.2 Tomcat 的安装、运行与目录结构	7	3.1.3 JSP 的转义字符	36
1.4.3 Eclipse 的安装、运行与特性	11	3.2 JSP 注释方式	37
1.5 JSP 程序初步	12	3.2.1 HTML 注释	37
1.5.1 创建 JSP 页	12	3.2.2 JSP 隐藏注释	37
1.5.2 部署 JSP 程序	14	3.2.3 脚本段注释	38
小结	15	3.3 JSP 脚本元素	38
习题	15	3.3.1 声明语句	39
第 2 章 JSP 辅助知识	17	3.3.2 脚本段	39
2.1 JSP 中的 HTML 元素	17	3.3.3 JSP 表达式	40
2.1.1 HTML 文本结构	17	3.4 JSP 指令元素	41
2.1.2 表单元素设置	18	3.4.1 页面指令元素: page	41
2.1.3 其他元素设置	21	3.4.2 包含指令元素: include	42
2.2 JSP 中的 JavaScript 语言	22	3.4.3 提供动作指令元素: taglib	43
2.2.1 JavaScript 语言概述	23	3.5 JSP 动作元素	44
2.2.2 网页中的 JavaScript	23	3.5.1 包含文件: <jsp:include>	44
2.2.3 基本语法	23	3.5.2 请求转发: <jsp:forward>	45
2.2.4 常用语句	24	3.5.3 声明使用 JavaBean: <jsp:useBean>	47
2.2.5 对象	26	3.5.4 设置 JavaBean 属性值: <jsp:setProperty>	50
2.2.6 事件	27	3.5.5 获取 JavaBean 属性值: <jsp:getProperty>	51
2.3 Web 应用程序体系结构	30	3.5.6 声明使用 Java 插件: <jsp:plugin>与 <jsp:fallback>	52
2.3.1 三层架构	30	3.5.7 参数传递: <jsp:params>与 <jsp:param>	53
2.3.2 二层架构	31	3.5.8 其他动作元素	53
2.3.3 JSP 技术支持的架构	32	小结	54
2.4 应用服务器	32	习题	54
2.4.1 Web 服务器	32		
2.4.2 JSP 引擎和 Servlet 引擎	33		
2.5 HTTP	33		

第 4 章 JSP 内置对象详解55	习题.....72
4.1 请求对象: request.....55	第 5 章 JavaBean 组件技术73
4.1.1 获取请求参数.....55	5.1 JavaBean 简介.....73
4.1.2 在作用域中管理属性.....55	5.1.1 为什么要使用 JavaBean.....73
4.1.3 获取 Cookie 对象.....58	5.1.2 JavaBean 的形式和要素.....74
4.1.4 获取客户端信息.....59	5.2 JavaBean 属性.....75
4.2 响应对象: response.....59	5.2.1 简单属性.....75
4.2.1 客户端与服务器端的交互.....60	5.2.2 索引属性.....76
4.2.2 页面重定向.....60	5.2.3 束缚属性.....77
4.2.3 缓冲区的输出.....60	5.2.4 限制属性.....77
4.2.4 response 对象的常用方法.....61	5.3 JavaBean 的作用域.....78
4.3 会话对象: session.....62	5.3.1 page 作用域.....78
4.3.1 理解 session.....62	5.3.2 request 作用域.....78
4.3.2 内置对象对通信的控制.....62	5.3.3 session 作用域.....78
4.3.3 创建与获取客户端 session.....63	5.3.4 application 作用域.....78
4.3.4 移除指定 session 中的对象.....63	5.3.5 JavaBean 获取作用域数据.....79
4.3.5 session 销毁.....63	5.4 使用 JavaBean 计算圆的周长与面积.....79
4.3.6 session 超时管理.....64	小结.....81
4.3.7 session 实现局部网页计数器.....64	习题.....81
4.4 多客户端共享对象: application.....64	第 6 章 Servlet 核心技术82
4.4.1 application 对象的作用范围.....65	6.1 Servlet 基础.....82
4.4.2 application 对象的常用方法.....65	6.1.1 Servlet 技术功能.....82
4.4.3 application 实现全局网页计数器.....65	6.1.2 Servlet 特征.....82
4.5 页面对象: page.....66	6.2 Servlet 生命周期.....83
4.5.1 page 对象的常用方法.....66	6.2.1 加载并初始化 Servlet.....83
4.5.2 page 对象的转换类型.....66	6.2.2 处理客户端请求.....84
4.6 页面上下文对象: pageContext.....67	6.2.3 卸载 Servlet.....84
4.6.1 pageContext 对象的常用方法.....67	6.3 使用 Servlet.....85
4.6.2 pageContext 对象获取作用域的值.....68	6.3.1 认识第 1 个 Servlet.....85
4.7 输出对象: out.....68	6.3.2 使用 HttpServlet.....86
4.8 配置对象: config.....69	6.4 获取运行环境信息.....87
4.8.1 config 对象的常用方法.....69	6.4.1 获取 Servlet 信息.....87
4.8.2 config 对象获取初始化参数.....69	6.4.2 获取服务器端信息.....88
4.9 异常对象: exception.....70	6.4.3 获取客户端信息.....90
4.9.1 exception 错误机制.....70	6.5 Servlet 中的会话设置.....92
4.9.2 exception 对象的常用方法.....70	6.5.1 获取 HttpSession 对象.....93
4.9.3 exception 设置指定错误页面.....70	6.5.2 在 HttpSession 对象中保存数据.....93
4.9.4 exception 对象指向空指针错误.....71	6.5.3 在 HttpSession 对象中读取数据.....93
小结.....72	

6.6 Servlet 中异常设置	95	7.7.1 执行静态 SQL 语句的实例	121
6.7 Servlet 过滤器	97	7.7.2 执行预处理 SQL 语句的实例	124
6.7.1 Servlet 过滤器工作原理	98	7.7.3 执行存储过程的实例	125
6.7.2 Servlet 过滤器配置	98	7.7.4 获取数据表信息	126
6.7.3 Servlet 过滤器典型应用	99	7.7.5 JDBC 事务的应用	128
6.8 Servlet 监听器	101	7.8 数据库连接池	130
6.8.1 Servlet 监听器工作原理	101	7.8.1 数据库连接池概述	130
6.8.2 Servlet 监听器类型	101	7.8.2 连接池的实现原理	130
6.8.3 Servlet 监听器典型应用	103	7.8.3 Tomcat 连接池的实现	131
小结	104	7.8.4 Proxool 连接池的实现	134
习题	104	7.8.5 其他连接池	137
第 7 章 JSP 操作数据库核心技术	105	小结	138
7.1 JDBC 技术概述	105	习题	138
7.2 JDBC 的结构	106	第 8 章 JSP 核心表达式与标签	139
7.2.1 JDBC 类型	106	8.1 JSP 表达式	139
7.2.2 数据库驱动程序	106	8.1.1 JSP 表达式概述	139
7.3 JDBC 核心编程接口	107	8.1.2 JSP 表达式使用	139
7.3.1 驱动器接口: Driver	107	8.1.3 访问作用域变量	141
7.3.2 驱动管理类: DriverManager	108	8.1.4 JSP 表达式隐藏对象	142
7.3.3 数据库连接接口: Connection	109	8.2 JSTL 标准标签库	144
7.3.4 执行静态 SQL 语句接口: Statement	109	8.2.1 JSTL 标签	144
7.3.5 执行预编译的 SQL 语句接口: PreparedStatement	110	8.2.2 JSTL 获取	144
7.3.6 处理存储过程语句接口: CallableStatement	112	8.2.3 JSTL 安装与配置	145
7.3.7 返回查询结果集接口: ResultSet	113	8.3 JSTL 核心标签	146
7.4 JDBC 操作数据库的步骤	115	8.3.1 输出结果标签	146
7.4.1 加载 JDBC 驱动程序	115	8.3.2 对象属性设置标签	146
7.4.2 取得数据库连接	116	8.3.3 对象值删除设置标签	147
7.4.3 执行各种 SQL 语句	116	8.3.4 捕捉异常标签	147
7.4.4 获取查询结果	118	8.3.5 if 条件判断标签	149
7.4.5 关闭数据库连接	119	8.3.6 choose 条件判断标签	149
7.6 JDBC 对事务的操作	119	8.3.7 条件分支标签	150
7.6.1 数据库事务的特性	119	8.3.8 其他条件分支标签	150
7.6.2 JDBC 事务的流程	119	8.3.9 迭代标签	151
7.6.3 JDBC 对事务的管理级别	120	8.3.10 导入 URL 资源标签	153
7.6.4 JDBC 对事务的设置	120	8.3.11 构造 URL 标签	154
7.7 JDBC 对数据库操作实例	121	8.3.12 重定向 URL 标签	154
		8.3.13 URL 参数传递标签	155
		8.4 JSTL 的 XML 标签	156
		8.5 JSTL 的格式化标签	159

8.6 JSTL 的其他标签.....	161	10.2.1 邮件传输协议.....	198
8.6.1 数据库标签.....	161	10.2.2 Java Mail 组件.....	198
8.6.2 函数标签.....	163	10.2.3 获取 Java Mail 组件.....	199
8.7 自定义标签.....	165	10.2.4 应用 Java Mail 组件完成电子 邮件的发送.....	199
8.7.1 自定义标签的格式.....	165	10.2.5 应用 Java Mail 组件完成电子 邮件的接收.....	202
8.7.2 自定义标签的构成.....	165	10.3 动态图表组件.....	204
8.7.3 自定义标签的实例.....	166	10.3.1 JFreeChart 组件.....	204
小结.....	167	10.3.2 获取 JFreeChart 组件.....	204
习题.....	168	10.3.3 使用 JFreeChart 绘制柱形图.....	205
第 9 章 Web 网页模板技术.....	169	10.3.4 使用 JFreeChart 绘制饼图.....	208
9.1 Web 模板概述.....	169	小结.....	210
9.2 Velocity 模板.....	169	习题.....	210
9.2.1 Velocity 的下载与安装.....	170	第 11 章 MVC 设计模式.....	211
9.2.2 初识 Velocity.....	170	11.1 表示层的两种架构模式.....	211
9.2.3 Velocity 的注释.....	171	11.1.1 Model1 架构模式.....	211
9.2.4 Velocity 的引用.....	171	11.1.2 Model2 架构模式.....	212
9.2.5 Velocity 的指令.....	173	11.2 MVC 的基础知识.....	212
9.2.6 Velocity 的其他特性.....	176	11.2.1 MVC 的发展史.....	212
9.2.7 在 Web 应用程序中使用 Velocity.....	177	11.2.2 MVC 的基本构成.....	213
9.3 FreeMarker 模板.....	178	11.2.3 MVC 的优缺点.....	214
9.3.1 FreeMarker 的下载与安装.....	179	11.3 Struts2 框架的 MVC 实现机制.....	214
9.3.2 初识 FreeMarker.....	179	11.3.1 Struts2 框架的基本工作流程.....	214
9.3.3 FreeMarker 的注释.....	180	11.3.2 Struts2 MVC 的实现方式.....	215
9.3.4 FreeMarker 的指令.....	180	11.3.3 Struts2 MVC 的实际应用.....	216
9.3.5 FreeMarker 的 Interpolation.....	186	11.4 Spring 框架的 MVC 实现机制.....	218
9.3.6 FreeMarker 的表达式.....	187	11.4.1 Spring MVC 的基本工作流程.....	218
9.3.7 在 Web 应用程序中使用 FreeMarker.....	190	11.4.2 Spring MVC 的实际应用.....	219
小结.....	191	小结.....	221
习题.....	192	习题.....	221
第 10 章 JSP 实用组件技术.....	193	第 12 章 JSP 实例开发 1—— 论坛.....	222
10.1 上传与下载组件.....	193	12.1 实例开发实质.....	222
10.1.1 Commons-FileUpload 组件概述.....	193	12.2 系统业务流程.....	222
10.1.2 获取 Commons-FileUpload 组件.....	193	12.3 数据表设计.....	223
10.1.3 应用 Commons-FileUpload 组件 完成文件上传.....	194	12.4 文件结构设计.....	224
10.1.4 文件的下载.....	196	12.5 公共模块设计.....	225
10.2 发送 E-mail 组件.....	198		

12.5.1 数据库连接类	225	13.3 数据表设计	251
12.5.2 分页生成器类	226	13.4 文件结构设计	252
12.5.3 验证码生成器类	228	13.5 公共模块设计	252
12.5.4 系统配置	229	13.5.1 数据库连接类	253
12.6 用户登录与安全退出	230	13.5.2 编码转换类	253
12.6.1 用户登录与退出功能概述	230	13.5.3 交易流水号生成类	254
12.6.2 用户登录与退出功能技术分析	231	13.5.4 系统配置	254
12.6.3 用户登录与退出功能实现过程	231	13.6 添加至购物车	255
12.7 查看帖子	234	13.6.1 添加至购物车模块概述	255
12.7.1 查看帖子功能概述	234	13.6.2 添加至购物车模块技术分析	256
12.7.2 查看帖子功能技术分析	234	13.6.3 添加至购物车模块实现过程	256
12.7.3 查看帖子功能实现过程	234	13.7 查看购物车	257
12.8 发布帖子	237	13.7.1 查看购物车模块概述	257
12.8.1 发布帖子功能概述	237	13.7.2 查看购物车模块技术分析	257
12.8.2 发布帖子功能技术分析	237	13.7.3 查看购物车模块实现过程	258
12.8.3 发布帖子功能实现过程	238	13.8 修改商品购买数量	259
12.9 回复帖子	239	13.8.1 修改商品购买数量模块概述	259
12.9.1 回复帖子功能概述	239	13.8.2 修改商品购买数量模块技术 分析	259
12.9.2 回复帖子功能技术分析	240	13.8.3 修改商品购买数量模块实现 过程	260
12.9.3 回复帖子功能实现过程	240	13.9 在购物车中移除指定商品	260
12.10 删除帖子	244	13.9.1 移除商品模块概述	260
12.10.1 删除帖子功能概述	245	13.9.2 移除商品模块技术分析	261
12.10.2 删除帖子功能技术分析	245	13.9.3 移除商品模块实现过程	261
12.10.3 删除帖子功能实现过程	245	13.10 收银台结账	261
小结	248	13.10.1 结账模块概述	261
第 13 章 JSP 实例开发 2—— 购物车	249	13.10.2 结账模块技术分析	262
13.1 实例开发实质	249	13.10.3 结账模块实现过程	262
13.2 系统业务流程	249	小结	264

第 1 章

JSP 初步

随着因特网和电子商务的普遍应用,各种动态网页语言陆续诞生。其中, JSP (Java Server Page) 自从发布以来, 在一些主流的动态网页技术中一直受到密切的关注。JSP 是由 Sun Microsystems 公司倡导的, 并与许多公司一起参与建立的一种动态网页技术标准, 该技术是在 Servlet 技术基础上发展而来的。如今, JSP 已经成为 Java 服务器编程的重要组成部分。

通过本章的学习, 读者可以对 JSP 的技术特点、JSP 运行原理及 JSP 开发环境有一定的了解, 并可以通过 Eclipse 工具开发一个简单的 Web 应用程序。

1.1 认识 JSP

JSP (Java Server Page) 是运行在服务器的一种脚本语言。熟悉 HTML 或者其他动态页面技术的读者, 在第 1 次看到 JSP 页面时可能会有一种似曾相识的感觉。这是因为, 从本质上说, 各种动态页面技术都是通过向 HTML 中添加其他语言脚本的方式来实现的, 而支持这些脚本的服务器可以执行这些脚本, 然后生成 HTML 页面。

为了让读者直观认识 JSP 技术, 先来看一个简单的 JSP 页面代码, 该 JSP 页面名称为 sanyang.jsp, 实现向页面输入一句话。具体代码如下:

```
<%@ page language="java" pageEncoding="GBK"%>
<html>
  <head>
    <title>第一个 JSP 程序</title>
  </head>
  <body>
    <%out.print("您好, 三扬科技");%>
  </body>
</html>
```

上述代码的代码风格和普通的 HTML 页面的代码非常相似, 不同的就是在 “<%” 和 “%>” 之间加入 Java 代码。将页面部署到 Tomcat 服务器上, 将该 Web 应用命名为 01, 启动 Tomcat 服务器, 在 IE 浏览器的地址栏中输入 URL 地址 “http://localhost:8080/01/sanyang.jsp”, 运行结果如图 1-1 所示。

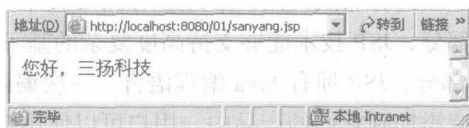


图 1-1 sanyang.jsp 页面的运行结果

1.2 JSP 技术特性

本节将介绍 JSP 的一些特性，如跨平台、分离静态内容和动态内容、强调可重用的组件等。

1. 跨平台

JSP 技术以 Java 为基础，所以不仅可以沿用 Java 强大的 API 功能，而且不管是在何种平台下，只要服务器支持 JSP，就可以运行使用 JSP 开发的 Web 应用程序，这体现了它的跨平台、跨服务器的特点。例如，在 Windows NT 下的 IIS 通过添加 JRUN 或 ServletExec 插件就能支持 JSP。如今最流行的 Web 服务器 Apache 同样能够支持 JSP，而且 Apache 支持多种平台，从而使得 JSP 可以跨平台运行。

在数据库操作中，因为 JDBC 同样是独立于平台的，所以在 JSP 中使用的 Java API 中提供的 JDBC 来连接数据库，就不用担心平台变更时的代码移植问题。

2. 将内容的生成和显示分离

使用 JSP 技术，Web 页面开发人员可以使用 HTML 或 XML 标识来设计和格式化最终页面，通过使用 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，并且捆绑在小脚本中，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，那么其开发人员，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（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。

3. 强调可重用的组件

绝大多数 JSP 页面依赖于可重用的、跨平台的组件（JavaBean 或者企业级 JavaBean）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者客户团体所使用。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。

4. 采用标识简化页面开发

JSP 采用标识简化页面开发，具有以下 4 个特点。

(1) Web 页面开发人员并不需要都是熟悉脚本语言的编程人员，因为 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所需要的。

(2) 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，以及执行用其他方法更难于编码和耗时的功能。通过开发定制化标识库，可以扩展 JSP 技术。第 3 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的执行特定功能的构件来工作。

(3) JSP 技术很容易整合到多种应用体系结构中，这样可以更好地利用现存的工具和技巧，并且可以扩展到能够支持企业级的分布式应用。作为采用 Java 技术家族的一部分，以及 J2EE（企业版体系结构）的一个组成部分，JSP 技术能够支持高度复杂的基于 Web 的应用。

(4) 作为 Java 平台的一部分，JSP 拥有 Java 编程语言“一次编译，到处运行”的特点。随着越来越多的供应商将 JSP 支持添加到它们的产品中，用户可以使用自己所选择的服务器和工具，更改工具或服务器而并不影响当前的应用。

5. 存储的健壮性与安全性

由于 JSP 页面的内置脚本语言是基于 Java 编程语言的，且都编译成 Java Servlet，它具有 Java 技术的所有好处，包括健壮的存储管理和安全性。

1.3 JSP 工作原理

从本质上说，JSP 是结合 markup（HTML 或 XML）和 Java 代码来处理的一种动态页面。每一页第 1 次被调用时，通过 JSP 引擎自动被编译成 Servlet 程序，然后被执行。例如，在 1.1 节中介绍的 sanyang.jsp 页面在 Tomcat 服务器运行时，该页面将会被编译成一个 Servlet 程序。该页面在 Tomcat 服务器上编译成的 Servlet 源代码如下：

```

package org.apache.jsp;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
public final class sanyang_jsp extends org.apache.jasper.runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent {
    private static final JspFactory _jspxFactory = JspFactory.getDefaultFactory();
    private static java.util.List _jspx_dependants;
    private javax.el.ExpressionFactory _el_expressionfactory;
    private org.apache.AnnotationProcessor _jsp_annotationprocessor;
    public Object getDependants() {
        return _jspx_dependants;
    }
    public void _jspInit() {
        _el_expressionfactory = _jspxFactory.getJspApplicationContext(getServletConfig().
getServletContext()).getExpressionFactory();
        _jsp_annotationprocessor = (org.apache.AnnotationProcessor) getServletConfig().
getServletContext().getAttribute(org.apache.AnnotationProcessor.class.getName());
    }
    public void _jspDestroy() {
    }
    public void _jspService(HttpServletRequest request, HttpServletResponse response)
        throws java.io.IOException, ServletException {
        PageContext pageContext = null;
        HttpSession session = null;
        ServletContext application = null;
        ServletConfig config = null;
        JspWriter out = null;
        Object page = this;
        JspWriter _jspx_out = null;
        PageContext _jspx_page_context = null;
        try {
            response.setContentType("text/html;charset=GBK");
            pageContext = _jspxFactory.getPageContext(this, request, response,
                null, true, 8192, true);
            _jspx_page_context = pageContext;
            application = pageContext.getServletContext();
            config = pageContext.getServletConfig();
            session = pageContext.getSession();
            out = pageContext.getOut();
            _jspx_out = out;

```

```

out.write("\r\n");
out.write("<html>\r\n");
out.write(" <head> \r\n");
out.write(" <title>第一个JSP程序</title>\t\r\n");
out.write(" </head> \r\n");
out.write(" <body>\r\n");
out.write(" ");
out.print("您好, 三扬科技");
out.write("\r\n");
out.write(" </body>\r\n");
out.write("</html>\r\n");
} catch (Throwable t) {
    if (!(t instanceof SkipPageException)){
        out = _jspx_out;
        if (out != null && out.getBufferSize() != 0)
            try { out.clearBuffer(); } catch (java.io.IOException e) {}
        if (_jspx_page_context != null) _jspx_page_context.handlePageException(t);
    }
} finally {
    _jspxFactory.releasePageContext(_jspx_page_context);
}
}
}

```



JSP 转换成 Servlet 的代码存放在 Tomcat 服务器的安装文件夹下, 一般情况下, 转换成 Servlet 的代码具体路径是: Tomcat 安装文件夹\work\Catalina\localhost\, 可以通过该文件夹下的工程名去寻找。

在一个 JSP 文件第 1 次被请求时, JSP 引擎先把该 JSP 文件转换成一个 Java 源文件, 在转换时, 如果发现 JSP 文件有任何语法错误, 转换过程将中断, 并向服务器端和客户端输出错误信息; 如果转换成功, JSP 引擎调用 Java 虚拟机的 javac 程序把该 Java 文件的源文件编译成相应的 class 文件, 该 class 文件也就是一个 Servlet 程序, 然后创建一个该 Servlet 的实例, 提供服务响应用户的请求。

JSP 转换成 Servlet 的流程如图 1-2 所示。

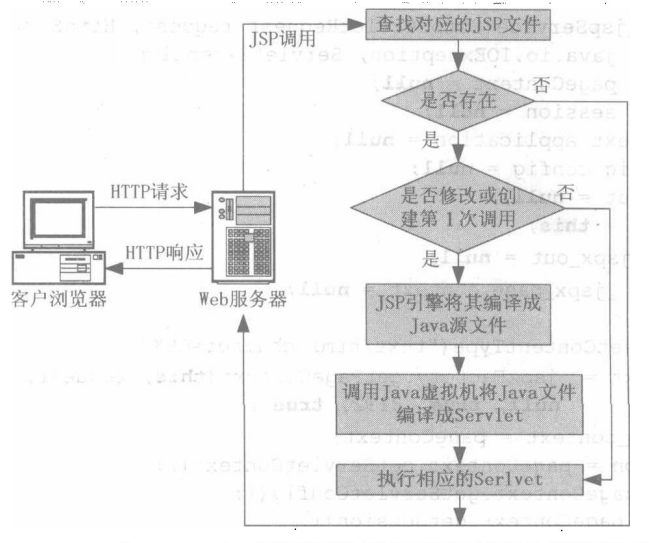


图 1-2 JSP 响应流程图

1.4 搭建 JSP 的运行环境

使用方便、高效快捷的 JSP 开发环境，对于开发学习 JSP 的读者来说会事半功倍。目前，比较流行的 JSP 开发平台和工具主要包括 JDK、Tomcat、Eclipse 等。其中，JDK 是 Java 语言的开发环境；Tomcat 是 Web 服务器；Eclipse 是一套简化的 Java EE 开发工具，实际上就是 JSP 应用开发的可视化 IDE。

1.4.1 JDK 的安装与配置

JDK 是 Java 语言的开发环境，由于 JSP 本身执行的计算机语言就是 Java，因此，想要开发与运行 JSP 程序也需要 JDK 的开发环境。JDK 的安装文件可以通过 <http://java.sun.com> 网站进行下载，目前的最新版本是 1.6。

下面介绍 JDK 的安装与配置。

1. JDK 的安装

从 Sun 公司官方网站中下载的安装文件的名称是 `jdk-6u10-windows-i586-p.exe`，下载完毕后，就可以在需要编译和运行 Java 程序的计算机安装 JDK 类，具体步骤如下。

(1) 双击 `jdk-6u10-windows-i586-p.exe` 文件开始安装。安装向导会要求接受 Sun 公司如图 1-3 所示的许可协议。

(2) 单击“接受”按钮接受许可协议后，打开设置 JDK 的安装路径及选择安装组件的对话框，如图 1-4 所示。

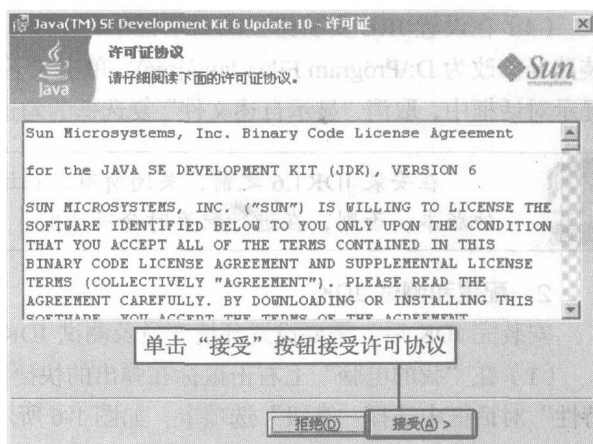


图 1-3 JDK1.6 的许可协议

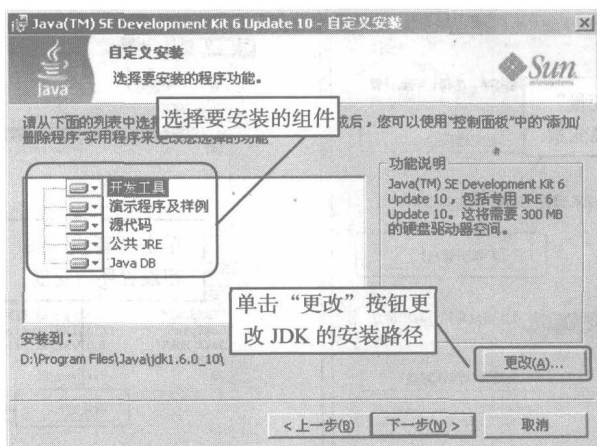


图 1.4 设置 JDK 的安装路径及选择安装组件对话框

(3) 在图 1-4 所示的对话框中单击“更改”按钮，如更改安装路径为 `D:\Program Files\Java\jdk1.6.0_10`，其他采用默认设置，单击“下一步”按钮将打开安装进度对话框安装 JDK。在安装过程中将打开图 1-5 所示的设置 JRE 安装路径的对话框。

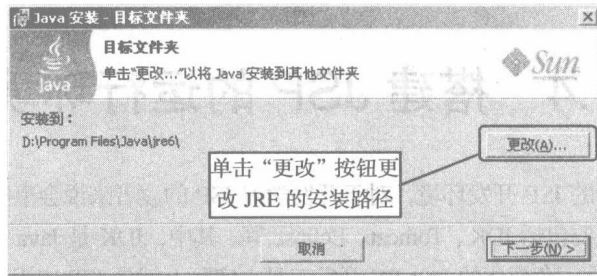


图 1-5 设置 JRE 的安装路径及选择安装组件对话框



由于 JDK 只是 Java 程序的开发环境，所以 JDK 的安装文件中还包含了一个 JRE (J2SE Runtime Environment) Java 运行环境，在默认情况下同 JDK 一起安装。

(4) 在设置 JRE 安装路径的对话框中，单击“更改”按钮，在打开的对话框中将 JRE 的安装路径修改为 D:\Program Files\Java\jre6\，单击“下一步”按钮继续安装 JRE。在弹出安装完成的提示对话框中，取消“显示自述文件”复选框的勾选，单击“完成”按钮，即可完成 JDK 的安装。



在安装 JDK1.6 之前，关闭所有正在运行的程序，并确认系统中没有安装 JDK 的其他版本，否则，在进行配置时会有冲突。

2. 配置和测试 JDK

安装完 JDK 后，需要设置环境变量及测试 JDK 配置是否成功，具体步骤如下。

(1) 在“我的电脑”上右击鼠标在弹出的快捷菜单中，选择“属性”选项。在打开的“系统特性”对话框中选择“高级”选项卡，如图 1-6 所示。

(2) 单击“环境变量”按钮，打开“环境变量”对话框。在这里可以添加针对单个用户的“用户变量”和针对所有用户“系统变量”，如图 1-7 所示。

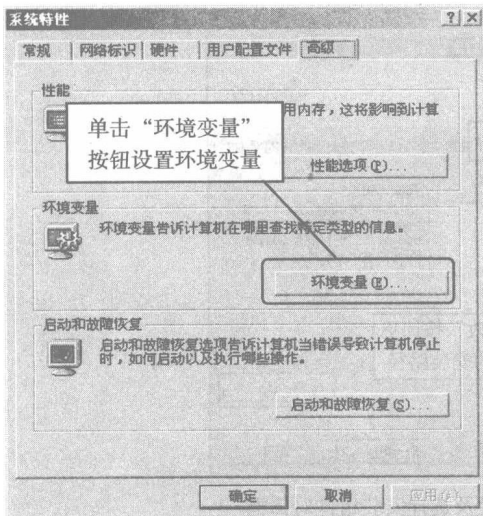


图 1-6 系统特性设置

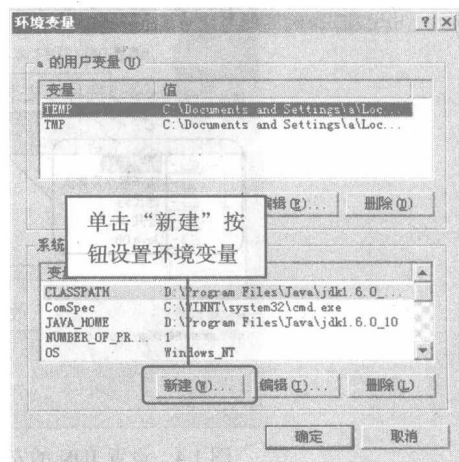


图 1-7 环境变量设置

(3) 单击“系统变量”区域中的“新建”按钮，弹出“编辑系统变量”对话框。该对话框中，在“变量名”文本框中输入“JAVA_HOME”，在“变量值”文本框中输入 JDK 的安装路径 D:\Program