

普.通.高.等.学.校
计算机教育“十二五”规划教材

JSP 程序设计教程

(第2版)

PROGRAMMING IN JAVASERVER PAGES
(2nd edition)

郭珍 王国辉 ◆ 主编
张敏 张永 ◆ 副主编



人民邮电出版社
POSTS & TELECOM PRESS

普.通.高.等.学.校
计算机教育“十二五”规划教材

JSP 程序设计教程

(第2版)

PROGRAMMING IN JAVASERVER PAGES
(2nd edition)

郭珍 王国辉 ◆ 主编
张敏 张永 ◆ 副主编

人 民 邮 电 出 版 社
北 京

图书在版编目 (C I P) 数据

JSP程序设计教程 / 郭珍, 王国辉主编. — 2版. —
北京: 人民邮电出版社, 2012. 10
普通高等学校计算机教育“十二五”规划教材
ISBN 978-7-115-29469-2

I. ①J… II. ①郭… ②王… III. ①
JAVA语言—网页制作工具—高等学校—教材 IV. ①
TP312②TP393. 092

中国版本图书馆CIP数据核字(2012)第225277号

内 容 提 要

本书作为 JSP 相关课程的教材, 系统地介绍了有关 JSP 开发所涉及的各类知识。全书共分 12 章, 内容包括 JSP 概述、JSP 开发基础、JSP 语法、JSP 内置对象、JavaBean 技术、Servlet 技术、JSP 实用组件、JSP 数据库应用开发和 JSP 高级程序设计, 并通过 JSP 综合开发实例——个人博客, 介绍 JSP 应用的开发流程和相关技术的综合应用。全书最后提供了两个课程设计方案, 即在线投票系统和 Ajax 聊天室, 供学生综合实践使用。

本书所讲知识都结合具体实例进行介绍, 力求详略得当, 使读者快速掌握应用 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”、“网络程序设计”、“Web 应用开发”、“动态网站制作”、“JSP 程序设计”等课程的教材, 同时也适合 JSP 初学者及网站开发人员参考使用。

普通高等学校计算机教育“十二五”规划教材

JSP 程序设计教程 (第 2 版)

-
- ◆ 编 著 郭 珍 王国辉
副 主 编 张 敏 张 永
责任编辑 李海涛
 - ◆ 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 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京昌平百善印刷厂印刷
 - ◆ 开本: 787×1092 1/16
印张: 17.5 2012 年 10 月第 2 版
字数: 457 千字 2012 年 10 月北京第 1 次印刷

ISBN 978-7-115-29469-2

定价: 36.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

第 2 版前言

2008 年 4 月《JSP 程序设计教程》一书完成并于当年出版,至今第 1 版已印刷近 2 万册。

虽然第 1 版发行至今仅四年,但近年来,教学改革步伐很大,在各专业的课程体系、课程设置和课时等方面都有较大的调整,专业课的调整尤为显著。作为计算机科学技术专业的专业课,JSP 编程的课时数逐年增加,而且技术更新很快,那么教材就要与之相适应了,为此我们决定对第 1 版做一次修订。

现就修订的具体情况作如下说明。

(1) 使用最新技术。在第 2 版中将所有的程序都使用最新的 Eclipse 开发环境+JDK 7+Tomcat 7 进行开发,以便适应读者的需要。

(2) 改进了以下内容,使读者能够快速掌握新技术的使用,并能应用于实践。

① 将原来使用的开发工具 Eclipse+MyEclipse 插件修改为 Eclipse IDE for Java EE,可以在不需要安装其他插件的情况下创建动态 Web 项目。

② 增加了 JDK 7 的新特性。例如,switch 语句支持 String(字符串)类型的表达式。

③ 增加了 Servlet 3.0 的新特性。例如,通过注解配置 Servlet。

④ 将原版本中过时的文件上传组件替换为 Commons-FileUpload 组件。

⑤ 将第 1 版中使用的 Jar 包,全部替换为当前最新版本。

(3) 本次修订更正了第 1 版中出现的一些错误及疏漏。

(4) 本次修订,保持第 1 版中的体系和内容基本不变。这样,任课教师自制的第 1 版教学课件,可稍加改动即可继续使用。

(5) 为方便教学和自学,本书免费提供教学课件,并提供全书案例代码的下载。

实用、易读是本书的编写宗旨。限于编者水平,书中难免存在疏漏之处,敬请读者批评指正。

编 者

2012 年 8 月

JSP (Java Server Page) 是由 Sun 公司在 Java 语言上开发出来的一种动态网页制作技术, 它是 Java 开发阵营中最具代表性的解决方案。JSP 不仅拥有与 Java 一样的面向对象、便利、跨平台等优点和特性, 还拥有 Java Servlet 的稳定性, 并且可以使用 Servlet 提供的 API、Java Bean 及 Web 开发框架技术, 使页面代码与后台处理代码分离, 提高工作效率。在目前流行的 Web 程序开发技术中, JSP 是比较热门的一种, 它依靠 Java 语言的稳定、安全、可移植性好的优点, 成为大、中型网站开发的首选。

本书通过通俗易懂的语言和实用生动的例子, 系统地介绍了 JSP 的基本常识、开发环境与开发工具、Java 和 JavaScript 语言基础、JSP 基本语法、内置对象、JavaBean 技术、Servlet 技术、实用组件、数据库应用开发和高级程序设计等技术, 并且在每一章的后面还提供了习题及上机指导, 方便读者及时验证自己的学习效果。最后, 通过一个综合实例及两个课程设计来使读者快速掌握 JSP 程序的开发过程。

本书适合作为普通高等院校计算机及相关专业“Web 程序设计”、“网络程序设计”、“Web 应用开发”、“动态网站制作”、“JSP 程序设计”等课程的教材, 同时也适合 JSP 初学者及网站开发人员参考使用。学习本书时, 读者最好具备 Java 程序设计、数据库原理、计算机网络等方面的基础知识。如果在前期已经开设了 Java 程序设计和 JavaScript 程序设计等相关课程, 则在教学过程中可以略讲或不讲第 2 章内容。对于第 9 章, 由于内容较深, 老师可以根据实际的教学情况选择是否讲解或掌握讲解的深度和难度。

本书作为教材使用时, 课堂教学建议 32~42 学时, 实验教学 16~32 学时。各章主要内容和学时建议分配如下, 老师可以根据实际教学情况进行调整。

章	主 要 内 容	课堂 学时	实验 学时
第 1 章	JSP 概述, 包括 JSP 技术概述、JSP 技术特征、JSP 的处理过程、JSP 开发环境的搭建、JSP 开发工具介绍和 JSP 程序开发模式	2	1
第 2 章	Java 语言基础和 JavaScript 脚本语言 (选讲)	6	4
第 3 章	JSP 技术的基本语法, 主要包括 JSP 中的指令标识、脚本标识、JSP 注释和动作标识	3	3
第 4 章	JSP 的内置对象, 包括 request 对象、response 对象、session 对象、application 对象、out 对象及其他内置对象的基本应用	4	2
第 5 章	JSP 程序开发中的 JavaBean 技术, 主要包括 JavaBean 的相关概念、JavaBean 中的属性、JavaBean 的创建以及 JavaBean 的应用	4	2
第 6 章	Servlet 的基础知识及 Servlet 的应用, 主要包括 Servlet 基础、Servlet API 编程常用接口和类以及 Servlet 的开发	4	2

续表

章	主 要 内 容	课堂学时	实验学时
第 7 章	JSP 的实用组件, 主要包括对文件进行操作的 Commons-FileUpload 组件、发送 E-mail 的 Java Mail 组件、生成动态图表的 JFreeChart 组件和生成 JSP 报表的 iText 组件	3	2
第 8 章	在 JSP 中应用数据库开发技术, 包括常用数据库管理系统、JDBC 技术概述及常用接口、连接数据库、数据库操作技术及连接池技术等	4	2
第 9 章	JSP 高级程序设计的相关技术, 主要包括 Ajax 技术在 JSP 中的应用、EL 表达式及 JSTL 标准标签库的应用, 以及 JSP 框架技术	4	
第 10 章	一个基于 JSP+Servlet+JavaBean 的个人博客网站	4	2
第 11 章	课程设计——在线投票系统	2	可选
第 12 章	课程设计——无刷新的聊天室		

本书所有例题和相关代码都已经调试通过, 同时制作了相关的多媒体课件。对于每一章的上机指导, 都提供了参考代码。对于本书中的代码, 为了避免版面的浪费, 我们在编写过程中一般只给出了核心代码或关键代码, 读者以此为基础可以很方便、快速地编写出实际运行代码。

由于作者水平有限, 书中难免存在错误、疏漏之处, 敬请广大读者批评指正。

编 者

2008 年 4 月

目 录

第 1 章 JSP 概述..... 1

1.1 JSP 技术概述.....	1
1.1.1 Java 语言.....	1
1.1.2 Servlet 技术.....	1
1.1.3 JavaBean 技术.....	2
1.1.4 JSP 技术.....	2
1.2 JSP 技术特征.....	3
1.3 JSP 的处理过程.....	4
1.4 JSP 与其他服务器端脚本语言的比较.....	4
1.4.1 CGI.....	4
1.4.2 ASP.....	5
1.4.3 PHP.....	5
1.4.4 ASP.NET.....	5
1.5 JSP 开发环境搭建.....	5
1.5.1 JSP 的运行环境.....	5
1.5.2 JDK 的安装与配置.....	6
1.5.3 Tomcat 的安装与启动.....	8
1.6 JSP 开发工具.....	10
1.6.1 Eclipse 的安装与启动.....	10
1.6.2 Eclipse 的使用.....	11
1.7 JSP 程序开发模式.....	16
1.7.1 单纯的 JSP 页面编程.....	16
1.7.2 JSP+JavaBean 编程.....	16
1.7.3 JSP+Servlet+JavaBean 编程.....	17
1.7.4 MVC 模式.....	18
1.8 第一个 JSP 应用.....	18
小结.....	20
习题.....	20
上机指导.....	20

第 2 章 JSP 开发基础.....21

2.1 Java 语言基础.....	21
2.1.1 面向对象程序设计.....	21

2.1.2 基本数据类型及基本数据类型间的转换.....	26
2.1.3 常量与变量.....	27
2.1.4 运算符的应用.....	29
2.1.5 流程控制语句.....	31
2.1.6 字符串处理.....	36
2.1.7 数组的创建与应用.....	39
2.1.8 集合类的应用.....	40
2.1.9 异常处理语句.....	42

2.2 JavaScript 脚本语言.....	43
2.2.1 JavaScript 脚本语言概述.....	43
2.2.2 在 JSP 中引入 JavaScript.....	43
2.2.3 JavaScript 的数据类型与运算符.....	43
2.2.4 JavaScript 的流程控制语句.....	46
2.2.5 函数的定义和调用.....	48
2.2.6 事件.....	48
2.2.7 JavaScript 常用对象的应用.....	49

小结.....	52
---------	----

习题.....	52
---------	----

上机指导.....	53
-----------	----

第 3 章 JSP 语法..... 54

3.1 了解 JSP 的基本构成.....	54
3.1.1 JSP 中的指令标识.....	54
3.1.2 HTML 标记语言.....	55
3.1.3 嵌入的 Java 代码片段.....	55
3.1.4 JSP 表达式.....	55
3.2 JSP 的指令标识.....	55
3.2.1 使用 page 指令.....	55
3.2.2 使用 include 指令.....	58
3.2.3 使用 taglib 指令.....	59
3.3 JSP 的脚本标识.....	59
3.3.1 JSP 表达式 (Expression).....	59
3.3.2 声明标识 (Declaration).....	60
3.3.3 脚本程序 (Scriptlet).....	61

3.4 JSP 的注释.....	61	4.6.2 向客户端输出数据.....	93
3.4.1 HTML 中的注释.....	62	4.7 其他内置对象.....	93
3.4.2 带有 JSP 表达式的注释.....	62	4.7.1 获取会话范围的 pageContext 对象.....	93
3.4.3 隐藏注释.....	62	4.7.2 读取 web.xml 配置信息的 config 对象.....	94
3.4.4 脚本程序 (Scriptlet) 中的注释.....	63	4.7.3 应答或请求的 page 对象.....	94
3.5 动作标识.....	65	4.7.4 获取异常信息的 exception 对象.....	95
3.5.1 <jsp:include>.....	66	小结.....	96
3.5.2 <jsp:forward>.....	67	习题.....	96
3.5.3 <jsp:useBean>.....	68	上机指导.....	96
3.5.4 <jsp:setProperty>.....	71	第 5 章 JavaBean 技术.....	97
3.5.5 <jsp:getProperty>.....	74	5.1 JavaBean 概述.....	97
3.5.6 <jsp:fallback>.....	74	5.1.1 JavaBean 技术介绍.....	97
3.5.7 <jsp:plugin>.....	75	5.1.2 JavaBean 的种类.....	98
小结.....	78	5.1.3 JavaBean 规范.....	98
习题.....	78	5.2 JavaBean 中的属性.....	99
上机指导.....	79	5.3 JavaBean 的应用.....	101
第 4 章 JSP 内置对象.....	80	5.3.1 创建 JavaBean.....	101
4.1 JSP 内置对象概述.....	80	5.3.2 在 JSP 页面中应用 JavaBean.....	103
4.2 request 对象.....	81	5.4 JavaBean 的应用实例.....	108
4.2.1 访问请求参数.....	81	5.4.1 应用 JavaBean 解决中文乱码.....	108
4.2.2 在作用域中管理属性.....	82	5.4.2 应用 JavaBean 实现购物车.....	111
4.2.3 获取 Cookie.....	83	小结.....	118
4.2.4 获取客户信息.....	84	习题.....	118
4.2.5 访问安全信息.....	85	上机指导.....	118
4.2.6 访问国际化信息.....	85	第 6 章 Servlet 技术.....	119
4.3 response 对象.....	85	6.1 Servlet 基础.....	119
4.3.1 重定向网页.....	85	6.1.1 Servlet 技术简介.....	119
4.3.2 设置 HTTP 响应报头.....	86	6.1.2 Servlet 技术功能.....	119
4.3.3 缓冲区配置.....	88	6.1.3 Servlet 技术特点.....	120
4.4 session 对象.....	88	6.1.4 Servlet 的生命周期.....	120
4.4.1 创建及获取客户的会话.....	89	6.1.5 Servlet 与 JSP 的区别.....	121
4.4.2 从会话中移除指定的对象.....	89	6.1.6 Servlet 的代码结构.....	122
4.4.3 销毁 session.....	90	6.1.7 开发简单的 Servlet 程序.....	122
4.4.4 会话超时的管理.....	90	6.2 Servlet API 编程常用接口和类.....	123
4.5 application 对象.....	91	6.2.1 Servlet 接口.....	123
4.5.1 访问应用程序初始化参数.....	91	6.2.2 HttpServlet 类.....	124
4.5.2 管理应用程序环境属性.....	92		
4.6 out 对象.....	93		
4.6.1 管理响应缓冲.....	93		

6.2.3 ServletConfig 接口	124	8.1.1 SQL Server 2008 数据库	167
6.2.4 HttpServletRequest 接口	125	8.1.2 MySQL 数据库	169
6.2.5 HttpServletResponse 接口	126	8.1.3 Oracle 数据库	169
6.2.6 GenericServlet 类	126	8.1.4 Access 数据库	170
6.3 Servlet 开发	127	8.2 JDBC 概述	170
6.3.1 Servlet 的创建	127	8.2.1 JDBC 技术介绍	170
6.3.2 Servlet 的配置	128	8.2.2 JDBC 驱动程序	171
6.4 Servlet 的应用实例	130	8.3 JDBC 中的常用接口	171
6.4.1 应用 Servlet 实现留言板	130	8.3.1 驱动程序接口 Driver	171
6.4.2 应用 Servlet 实现购物车	134	8.3.2 驱动程序管理器 DriverManager	172
小结	141	8.3.3 数据库连接接口 Connection	172
习题	141	8.3.4 执行 SQL 语句接口 Statement	173
上机指导	142	8.3.5 执行动态 SQL 语句接口 PreparedStatement	173
第 7 章 JSP 实用组件	143	8.3.6 执行存储过程接口 CallableStatement	174
7.1 JSP 文件操作	143	8.3.7 访问结果集接口 ResultSet	174
7.1.1 添加表单及表单元素	143	8.4 连接数据库	175
7.1.2 创建上传对象	144	8.4.1 加载 JDBC 驱动程序	175
7.1.3 解析上传请求	144	8.4.2 创建数据库连接	176
7.2 发送 E-mail	146	8.4.3 执行 SQL 语句	176
7.2.1 Java Mail 组件简介	147	8.4.4 获得查询结果	176
7.2.2 Java Mail 核心类简介	147	8.4.5 关闭连接	176
7.2.3 搭建 Java Mail 的开发环境	151	8.5 典型 JSP 数据库连接	177
7.2.4 在 JSP 中应用 Java Mail 组件 发送 E-mail	151	8.5.1 SQL Server 2008 数据库的连接	177
7.3 JSP 动态图表	153	8.5.2 Access 数据库的连接	178
7.3.1 JFreeChart 的下载与使用	154	8.5.3 MySQL 数据库的连接	178
7.3.2 JFreeChart 的核心类	155	8.6 数据库操作技术	179
7.3.3 利用 JFreeChart 生成动态图表	155	8.6.1 查询操作	179
7.4 JSP 报表	157	8.6.2 添加操作	180
7.4.1 iText 组件简介	157	8.6.3 修改操作	181
7.4.2 iText 组件的下载与配置	158	8.6.4 删除操作	182
7.4.3 应用 iText 组件生成 JSP 报表	158	8.7 连接池技术	182
小结	166	8.7.1 连接池简介	182
习题	166	8.7.2 在 Tomcat 中配置连接池	183
上机指导	166	8.7.3 使用连接池技术访问数据库	184
第 8 章 JSP 数据库应用开发	167	小结	185
8.1 数据库管理系统	167	习题	185
		上机指导	186

第9章 JSP 高级程序设计187

9.1 JSP 与 Ajax 技术	187
9.1.1 Ajax 的开发模式	187
9.1.2 Ajax 使用的技术	188
9.1.3 搭建 Ajax 开发框架	190
9.1.4 Ajax 开发需要注意的几个问题	192
9.2 EL 表达式及标签	193
9.2.1 表达式语言	193
9.2.2 JSTL 标准标签库	196
9.2.3 自定义标签库的开发	205
9.3 JSP 框架技术	209
9.3.1 Struts 2 框架	209
9.3.2 Spring 框架	210
9.3.3 Hibernate 技术	211
小结	212
习题	212
上机指导	212

**第10章 JSP 综合开发实例——
个人博客213**

10.1 网站总体设计	213
10.2 数据库设计	214
10.2.1 ER 图	214
10.2.2 数据表概要说明	215
10.2.3 主要数据表的结构	215
10.3 文件架构设计	216
10.4 公共类的编写	217
10.4.1 数据库连接及操作类	217
10.4.2 字符串处理类	219
10.5 前台主页面设计	220
10.5.1 前台主页面概述	220
10.5.2 前台主页面的布局	220
10.5.3 前台主页面的实现	221
10.6 文章显示模块设计	222
10.6.1 文章显示模块概述	222
10.6.2 首页面文章的列表显示 实现过程	223

10.6.3 文章详细显示的实现过程	226
10.7 博主登录模块设计	230
10.7.1 博主登录功能概述	230
10.7.2 博主登录实现过程	231
10.8 相册管理模块设计	234
10.8.1 相册管理模块概述	234
10.8.2 相册管理模块实现过程	235
小结	241

**第11章 课程设计一——
在线投票系统 243**

11.1 课程设计的目的	243
11.2 设计思路	244
11.3 设计过程	246
11.3.1 数据表的设计	246
11.3.2 值 JavaBean 的设计	247
11.3.3 数据库操作类的编写	248
11.3.4 工具类的编写	252
11.3.5 显示投票选项的设计	253
11.3.6 参与投票的设计	254
11.3.7 查看结果的设计	256
小结	258

**第12章 课程设计二——
Ajax 聊天室 259**

12.1 课程设计的目的	259
12.2 设计思路	259
12.3 设计过程	260
12.3.1 用户 JavaBean 的编写	260
12.3.2 登录页面的设计	260
12.3.3 聊天室主页面设计	261
12.3.4 在线人员列表的设计	262
12.3.5 用户发言的设计	264
12.3.6 显示聊天内容的设计	265
12.3.7 退出聊天室的设计	266
小结	267

参考文献 269

第 1 章

JSP 概述

本章介绍 JSP 技术的相关概念以及如何开发 JSP 程序，主要包括 JSP 技术的概述、JSP 技术特征、JSP 的处理过程、JSP 开发环境的搭建、JSP 开发工具介绍、JSP 程序开发模式等。通过本章的学习，读者应了解什么是 JSP、JSP 技术的特征和处理过程，并掌握 JDK 的安装与配置、Tomcat 的安装与配置、Eclipse 开发工具的使用等。尤其要深刻理解：在 JSP 程序开发模式中，JSP+Servlet+JavaBean 开发模式的设计及模式中各技术所扮演的角色。

1.1 JSP 技术概述

在了解 JSP 技术之前，先向读者介绍与 JSP 技术相关的一些概念，这样有助于读者阅读后面的内容。

1.1.1 Java 语言

Java 语言是由 Sun 公司于 1995 年推出的编程语言，一经推出，就赢得了业界的一致好评，并受到了广泛关注。Java 语言适用于 Internet 环境，目前已成为开发 Internet 应用的主要语言之一。它具有简单、面向对象、可移植性、分布性、解释器通用性、稳健、多线程、安全、高性能等优点。其中最重要的就是实现了跨平台运行，这使得应用 Java 开发的程序可以方便地移植到不同的操作系统中运行。

Java 是完全面向对象的编程语言，它的语法规则和 C++ 类似，但 Java 语言对 C++ 进行了简化和提高。例如，C 语言中的指针和多重继承通常会使得程序变得复杂，而 Java 通过接口取代了多重继承，并取消了指针、内存的申请和释放等影响系统安全的部分。

在 Java 语言中，最小的单位是类，不允许在类外面定义变量和方法，所以就不存在所谓的“全局变量”这一概念。在 Java 类中定义的变量和方法分别称为成员变量和成员方法，其中成员变量也叫做类的属性。在定义这些类的成员时，需要通过权限修饰符来声明它们的使用范围。

Java 语言编写的程序应被保存为后缀名为 .java 的文件，然后编译成后缀名为 .class 的字节码文件，最终通过执行该字节码文件执行 Java 程序。

1.1.2 Servlet 技术

Servlet 是在 JSP 之前就存在的运行在服务端的一种 Java 技术，它是用 Java 语言编写的服务

器端程序, Java 语言能够实现的功能, Servlet 基本上都可以实现(除图形界面外)。Servlet 主要用于处理 Http 请求, 并将处理的结果传递给浏览器生成动态 Web 页面。Servlet 具有可移植(可在多种系统平台和服务器平台下运行)、功能强大、安全、可扩展、灵活等优点。

在 JSP 中用到的 Servlet 通常都继承自 `javax.servlet.http.HttpServlet` 类, 在该类中实现了用来处理 Http 请求的大部分功能。

JSP 是在 Servlet 的基础上开发的一种新的技术, 所以 JSP 与 Servlet 有着密不可分的关系。JSP 页面在执行过程中会被转换为 Servlet, 然后由服务器执行该 Servlet。

关于 Servlet 技术的相关介绍请参看本书的第 6 章。

1.1.3 JavaBean 技术

JavaBean 是根据特殊的规范编写的普通的 Java 类, 可称它们为“独立的组件”。每一个 JavaBean 实现一个特定的功能, 通过合理地组织具有不同功能的 JavaBean, 可以快速地生成一个全新的应用程序。如果将这个应用程序比作一辆汽车, 那么程序中的 JavaBean 就好比组成这辆汽车的不同零件。对于程序开发人员来说, JavaBean 的最大优点就是充分提高了代码的可重用性, 并且对程序的后期维护和扩展起到了积极的作用。

JavaBean 可按功能划分为可视化和不可可视化两种。可视化 JavaBean 主要应用在图形界面编程的领域中, 在 JSP 中通常应用的是不可可视化 JavaBean, 应用该种 JavaBean 可用来封装各种业务逻辑, 如连接数据库、获取当前时间等。这样, 当在开发程序的过程中需要连接数据库或实现其他功能时, 就可直接在 JSP 页面或 Servlet 中调用实现该功能的 JavaBean 来实现。

通过应用 JavaBean, 可以很好地将业务逻辑和前台显示代码分离, 这大大提高了代码的可读性和易维护性。

关于 JavaBean 技术的相关介绍请参看本书的第 5 章。

1.1.4 JSP 技术

Java Server Pages 简称 JSP, 是由 Sun 公司倡导, 与多个公司共同建立的一种技术标准, 它建立在 Servlet 之上。应用 JSP, 程序员或非程序员可以高效率地创建 Web 应用程序 并使得开发的 Web 应用程序具有安全性高、跨平台等优点。

JSP 是运行在服务器端的脚本语言之一, 与其他的服务器端脚本语言一样, 是用来开发动态网页的一种技术。

JSP 页面由传统的 HTML 代码和嵌入到其中的 Java 代码组成。当用户请求一个 JSP 页面时, 服务器会执行这些 Java 代码, 然后将结果与页面中的静态部分相结合返回给客户端浏览器。JSP 页面中还包含了各种特殊的 JSP 元素, 通过这些元素可以访问其他的动态内容并将它们嵌入到页面中, 如访问 JavaBean 组件的 `<jsp:useBean>` 动作元素。程序员还可以通过编写自己的元素来实现特定的功能, 开发出更为强大的 Web 应用程序。

JSP 是在 Servlet 的基础上开发的技术, 它继承了 Java Servlet 的各项优秀功能。而 Java Servlet 是作为 Java 的一种解决方案, 在制作网页的过程中, 它继承了 Java 的所有特性。因此, JSP 同样继承了 Java 技术的简单、便利、面向对象、跨平台、安全可靠等优点, 比起其他服务器脚本语言, JSP 更加简单、迅速和有力。在 JSP 中利用 JavaBean 和 JSP 元素, 可以有效地将静态的 HTML 代码和动态数据区分开来, 给程序的修改和扩展带来了很大方便。

1.2 JSP 技术特征

本节将介绍 JSP 在开发 Web 应用程序时的一些特点，如跨平台、分离静态内容和动态内容、可重复使用的组件、沿用了 Java Servlet 的所有功能、能预编译等。

1. 跨平台

JSP 是以 Java 为基础开发的，所以它不仅可以用 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，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 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 可以在多个平台上运行。

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 Java API 提供的 JDBC 来连接数据库时，就不用担心平台变更时的代码移植问题。正是因为 Java 的这种特征，使得应用 JSP 开发的 Web 应用程序能够很简单地运用到不同的平台上。

2. 分离静态内容和动态内容

在前面提到的 Java Servlet，对于开发 Web 应用程序而言是一种很好的技术。但同时面临着一个问题：所有的内容必须在 Java 代码中来完成，包括 HTML 代码同样要嵌入到程序代码中来生成静态的内容。这使得即使因 HTML 代码出现的小问题，也需要有熟悉 Java Servlet 的程序员来解决。

JSP 弥补了 Java Servlet 在工作中的不足。使用 JSP，程序员可以使用 HTML 或 XML 标记来设计和格式化静态的内容部分，使用 JSP 标记及 JavaBean 组件或者小脚本程序来制作动态内容部分。服务器将执行 JSP 标记和小脚本程序，并将结果与页面中的静态部分结合后以 HTML 页面的形式发送给客户端浏览器。程序员可以将一些业务逻辑封装到 JavaBean 组件中，Web 页面的设计人员可以利用程序员开发的 JavaBean 组件和 JSP 标记来制作出动态页面，而且不会影响到内容的生成。

将静态内容与动态内容的明确分离，是以 Java Servlet 开发 Web 应用发展为以 JSP 开发 Web 应用的重要因素之一。

3. 可重复使用的组件

JavaBean 组件是 JSP 中不可缺少的重要组成部分之一，程序通过 JavaBean 组件来执行所要求的更为复杂的运算。JavaBean 组件不仅可以应用于 JSP 中，同样适用于其他的 Java 应用程序中。这种特性使得开发人员之间可以共享 JavaBean 组件，加快了应用程序的总体开发进程。

同样，JSP 的标准标签和自定义标签与 JavaBean 组件一样可以一次生成重复使用。这些标签都是通过编写的程序代码来实现特定功能的，在使用它们时与通常在页面中用到的 HTML 标记用法相同。这样可以将一个复杂而且需要出现多次的操作简单化，大大提高了工作效率。

4. 沿用了 Java Servlet 的所有功能

相对于 Java Servlet 来说，使用从 Java Servlet 发展而来的 JSP 技术开发 Web 应用更加简单易学，并且 JSP 同样提供了 Java Servlet 所有的特性。实际上服务器在执行 JSP 文件时先将其转换为 Servlet 代码，然后再对其进行编译，可以说 JSP 就是 Servlet，创建一个 JSP 文件其实就是创建一个 Servlet 文件的简化操作。理所当然，Servlet 中的所有特性在 JSP 中同样可以使用。

5. 预编译

预编译是 JSP 的另一个重要的特性。JSP 页面在被服务器执行前，都是已经被编译好的，并且通常只进行一次编译，即在 JSP 页面被第一次请求时进行编译，在后续的请求中如果 JSP 页面

没有被修改过, 服务器只需要直接调用这些已经被编译好的代码, 这大大提高了访问速度。

1.3 JSP 的处理过程

当客户端浏览器向服务器发出请求访问一个 JSP 页面时, 服务器根据该请求加载相应的 JSP 页面, 并对该页面进行编译, 然后执行。JSP 的处理过程如图 1-1 所示。

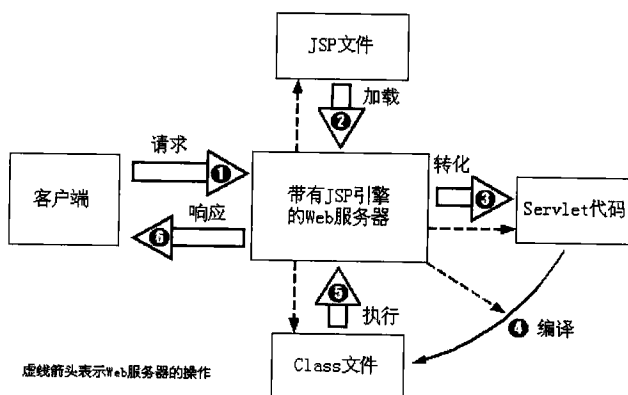


图 1-1 JSP 的处理过程

① 客户端通过浏览器向服务器发出请求, 在该请求中包含了请求资源的路径, 这样当服务器接收到该请求后就可以知道被请求的资源。

② 服务器根据接收到的客户端的请求来加载被请求的 JSP 文件。

③ Web 服务器中的 JSP 引擎会将被加载的 JSP 文件转化为 Servlet。

④ JSP 引擎将生成的 Servlet 代码编译成 Class 文件。

⑤ 服务器执行这个 Class 文件。

⑥ 最后服务器将执行结果发送给浏览器进行显示。

从上面的介绍中可以看到, JSP 文件被 JSP 引擎进行转换后, 又被编译成了 Class 文件, 最终由服务器通过执行这个 Class 文件来对客户端的请求进行响应。其中第③步与第④步构成了 JSP 处理过程中的翻译阶段, 而第⑤步为请求处理阶段。

但并不是每次请求都需要重复进行这样的处理。当服务器第一次接收到对某个页面的请求时, JSP 引擎就开始进行上述的处理过程, 将被请求的 JSP 文件编译成 Class 文件。在后续对该页面再次进行请求时, 若页面没有进行任何改动, 服务器只需直接调用 Class 文件执行即可。所以当某个 JSP 页面第一次被请求时, 会有一些延迟, 而再次访问时会感觉快了很多。如果被请求的页面经过修改, 服务器将会重新编译这个文件, 然后执行。

1.4 JSP 与其他服务器端脚本语言的比较

1.4.1 CGI

CGI 是最早用来创建动态网页的一种技术, 它可以使浏览器与服务器之间产生互动关系。

Common Gateway Interface 是 CGI 的全称,即通用网关接口。它允许使用不同的语言来编写适合的 CGI 程序,该程序被放在 Web 服务器上运行。当客户端发出请求给服务器时,服务器根据客户请求建立一个新的进程来执行指定的 CGI 程序,并将执行结果以网页的类型传输到客户端的浏览器上进行显示。CGI 可以说是当前应用程序的基础技术,但这种技术编制方式比较困难而且效率低下,因为每次页面被请求的时候,都要求服务器重新将 CGI 程序编译成可执行的代码。在 CGI 中使用最为常见的语言为 C/C++、Java 和 Perl (Practical Extraction and Report Language, 文件分析报告语言)。

1.4.2 ASP

ASP (Active Server Page) 是一种使用很广泛的开发动态网站的技术。它通过在页面代码中嵌入 VBScript 或 JavaScript 脚本语言来生成动态的内容,在服务器端必须安装了适当的解释器后,才可以通过调用此解释器来执行脚本程序,然后将执行结果与静态内容部分结合并传送到客户端浏览器上。对于一些复杂的操作,ASP 可以调用存在于后台的 COM 组件来完成,所以说 COM 组件无限地扩充了 ASP 的能力;正因如此依赖本地的 COM 组件,使得 ASP 主要用于 Windows 平台中。ASP 存在很多优点,简单易学,并且 ASP 是与微软公司的 IIS 捆绑在一起的,在安装 Windows 2000、Windows XP 的同时安装上 IIS,就可以运行 ASP 应用程序了。

1.4.3 PHP

PHP 来自于 Personal Home Page 一词,但现在的 PHP 已经不再表示名词的缩写,而是一种开发动态网页技术的名称。PHP 语法类似于 C,并且混合了 Perl、C++ 和 Java 的一些特性。它是一种开源的 Web 服务器脚本语言,与 ASP 和 JSP 一样可以在页面中加入脚本代码来生成动态内容。对于一些复杂的操作可以封装到函数或类中,在 PHP 中提供了许多已经定义好的函数,如提供的标准的数据库接口,使得数据库连接方便,扩展性强。PHP 可以被多个平台支持,主要被广泛应用于 UNIX/Linux 平台。由于 PHP 本身的代码对外开放,经过许多软件工程师的检测,因此到目前为止该技术具有公认的安全性能。

1.4.4 ASP.NET

ASP.NET 也是一种建立动态 Web 应用程序的技术,它是 .NET 框架的一部分,可以使用任何 .NET 兼容的语言,如 Visual Basic.NET, C#, J# 等来编写 ASP.NET 应用程序。这种 ASP.NET 页面 (Web Forms) 编译后可以提供比脚本语言更出色的性能表现。Web Forms 允许在网页基础上建立强大的窗体。当建立页面时,可以使用 ASP.NET 服务端控件来建立常用的 UI 元素,并对它们编程来完成一般的任务。这些控件允许开发者使用内建可重用的组件和自定义组件来快速建立 Web Form,使代码简单化。

1.5 JSP 开发环境搭建

1.5.1 JSP 的运行环境

使用 JSP 进行开发,需要具备以下对应的运行环境:Web 浏览器、Web 服务器、JDK 开发工

具包以及数据库。下面分别介绍这些环境。

1. Web 浏览器

浏览器主要是客户端用户访问 Web 应用的工具,与开发 JSP 应用不存在很大的关系,所以开发 JSP 对浏览器的要求并不是很高,任何支持 HTML 的浏览器都可以。

2. Web 服务器

Web 服务器是运行及发布 Web 应用的大容器,只有将开发的 Web 项目放置到该容器中,才能使网络中的所有用户通过浏览器进行访问。开发 JSP 应用所采用的服务器主要是 Servlet 兼容的 Web 服务器,比较常用的有 BEA WebLogic、IBM WebSphere、Apache Tomcat 等。

Weblogic 是 BEA 公司的产品,它又分为 WebLogic Server、WebLogic Enterprise 和 WebLogic Portal 系列,其中 WebLogic Server 的功能特别强大,它支持企业级的、多层次的和完全分布式的 Web 应用,并且服务器的配置简单、界面友好,对于那些正在寻求能够提供 Java 平台所拥有的一切的应用服务器的用户来说,WebLogic 是一个十分理想的选择。

Tomcat 服务器最为流行,它是 Apache-Jakarta 开源项目中的一个子项目,是一个小型的、轻量级的、支持 JSP 和 Servlet 技术的 Web 服务器,它已经成为学习开发 JSP 应用的首选,本书中的所有例子都使用了 Tomcat 作为 Web 服务器,所以对该服务器的安装与配置在后续章节中进行了解。目前 Tomcat 的最新版本为 apache-tomcat-7.0.29。

3. JDK

JDK (Java Develop Kit, Java 开发工具包)包括运行 Java 程序所必须的 JRE 环境及开发过程中常用的库文件。在使用 JSP 开发网站之前,首先必须安装 JDK,目前 JDK 的最新版本为 JDK 7 Update 5。

4. 数据库

任何项目的开发几乎都需要使用数据库,数据库用来存储项目中需要的信息。根据项目的规模,应采用合适的数据库。例如,大型项目可采用 Oracle 数据库,中型项目可采用 Microsoft SQL Server 或 MySQL 数据库,小型项目可采用 Microsoft Access 数据库。Microsoft Access 数据库的功能远比不上 Microsoft SQL Server 和 MySQL 强大,但它具有方便、灵活的特点,对于一些小型项目来说是比较理想的选择。

1.5.2 JDK 的安装与配置

JDK 由 Sun 公司提供,其中包括运行 Java 程序所必须的 JRE (Java Runtime Environment, Java 运行环境)及开发过程中常用的库文件。在使用 JSP 开发网站之前,首先必须安装 JDK 组件。

1. JDK 的安装

由于推出 JDK 的 Sun 公司已经被 Oracle 公司收购了,所以 JDK 可以到 Oracle 官方网站 (<http://www.oracle.com/index.html>) 中下载。目前,最新的版本是 JDK 7 Update 5,如果是 32 位的 Windows 操作系统,下载后得到的安装文件是 jdk-7u5-windows-i586.exe。具体安装步骤如下。

① 双击 jdk-7u5-windows-i586.exe 文件,在弹出的欢迎对话框中,单击“下一步”按钮,将弹出“自定义安装”对话框,在该对话框中选择 JDK 的安装路径。单击“更改”按钮更改安装路径,其他保留默认选项,如图 1-2 所示。

② 单击图 1-2 中的“下一步”按钮,开始安装。

③ 在安装的过程中,会弹出另一个“自定义安装”对话框提示用户选择 Java 运行时环境的

安装路径。单击“更改”按钮更改安装路径，其他保留默认选项，如图 1-3 所示。

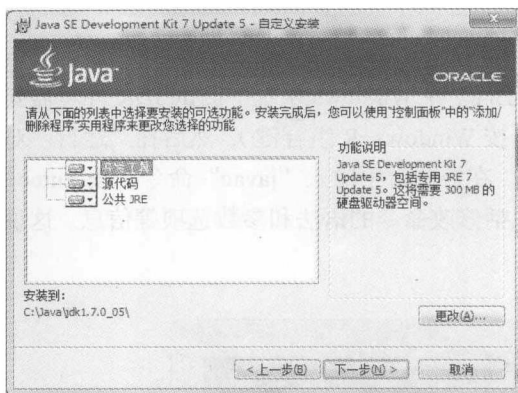


图 1-2 选择 JDK 安装路径

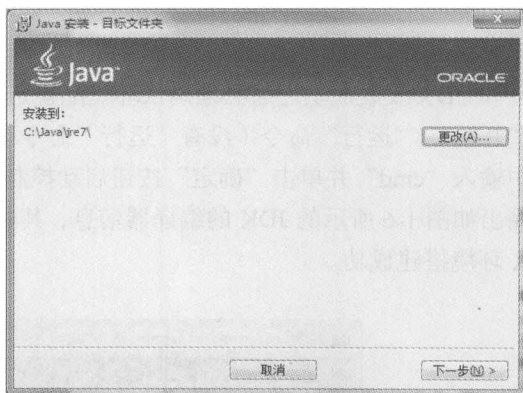


图 1-3 选择 JRE 安装路径

④ 单击图 1-3 中的“下一步”按钮继续安装。在 JDK 安装完毕后，单击“继续”按钮，将安装 JavaFX SDK，这里可以选择取消安装，也可以单击“下一步”按钮继续安装，直到出现“JavaFX SDK 安装程序 - 完成”对话框，单击“关闭”按钮。



JavaFX 2.0 是甲骨文宣布的，一款为企业业务应用提供的先进 Java 用户界面（UI）平台，它能帮助开发人员无缝地实现与本地 Java 功能及 Web 技术动态能力的混合与匹配。

2. JDK 的配置与测试

JDK 安装完成后，还需要在系统的环境变量中进行配置，下面将以在 Windows 7 系统中配置环境变量为例来介绍。JDK 的配置和测试。具体步骤如下。

① 在“开始”菜单的“计算机”图标上单击鼠标右键，在弹出的快捷菜单中选择“属性”命令，在弹出的“属性”对话框左侧单击“高级系统设置”超链接，将出现“系统属性”对话框。

② 在“系统属性”对话框中，单击“环境变量”按钮，将弹出“环境变量”对话框，单击“系统变量”栏中的“新建”按钮，创建新的系统变量。

③ 在弹出的“新建系统变量”对话框中，分别输入变量名“JAVA_HOME”和变量值（即 JDK 的安装路径），这里为 C:\Java\jdk1.7.0_05，如图 1-4 所示，读者需要根据自己的计算机环境进行修改。单击“确定”按钮，关闭“新建系统变量”对话框。

④ 在“环境变量”对话框中双击 Path 变量对其进行修改，在原变量值最前端添加“.;%JAVA_HOME%\bin;”变量值（注意：最后的“;”不要丢掉，它用于分割不同的变量值），如图 1-5 所示。单击“确定”按钮完成环境变量的设置。

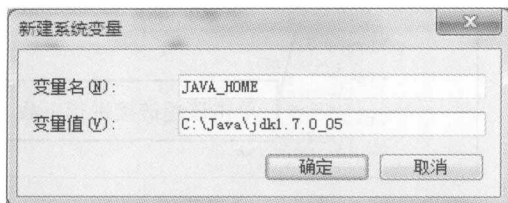


图 1-4 “新建系统变量”对话框

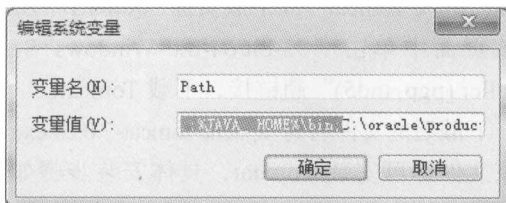


图 1-5 设置 Path 环境变量值