

教育部 - 浪潮集团产学研合作协同育人项目成果

普通高等学校计算机教育“十三五”规划教材

inspur 浪潮

JSP

程序设计与案例实战

慕课版

浪潮优派◎出品

刘何秀 郭建磊 姬忠红◎主编



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

教育部 - 浪潮集团产学研合作协同育人项目成果

普通高等学校计算机教育“十三五”规划教材

inspur 浪潮

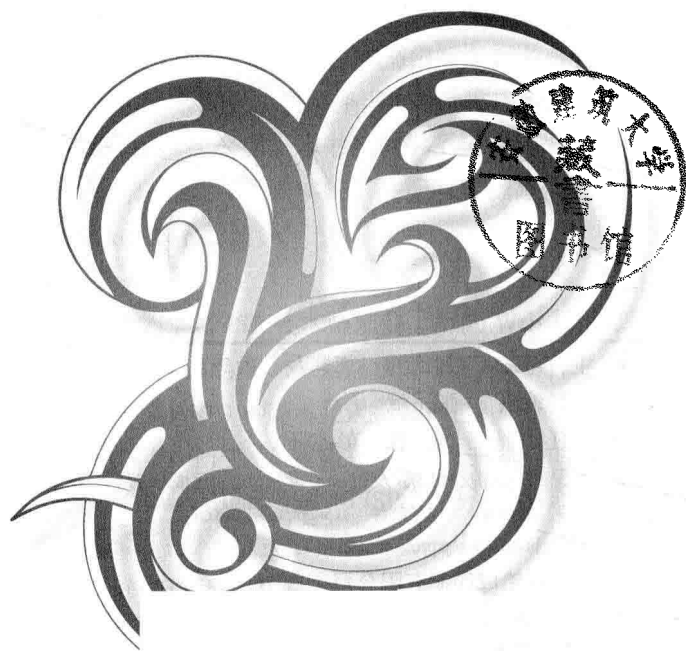
JSP

程序设计与案例实战

慕课版

浪潮优派◎出品

刘何秀 郭建磊 姬忠红◎主编



人民邮电出版社

北京

图书在版编目 (C I P) 数据

JSP程序设计与案例实战 : 慕课版 / 刘何秀, 郭建磊, 姬忠红主编. — 北京 : 人民邮电出版社, 2018. 5(2018. 8重印)
普通高等学校计算机教育“十三五”规划教材
ISBN 978-7-115-48013-2

I. ①J… II. ①刘… ②郭… ③姬… III. ①JAVA语言—网页制作工具—高等学校—教材 IV. ①TP312. 8
②TP393. 092. 2

中国版本图书馆CIP数据核字(2018)第040153号

内 容 提 要

JSP是目前企业中常用的一种动态网页开发技术,是Java EE企业级开发体系中非常重要的基础技术。

本书由浅入深地讲解了整个JSP知识体系。全书共13章,第1~12章是JSP基础知识,主要包括Web基础、JSP开发环境的搭建、Web服务器的使用、JSP脚本元素、JSP指令、JSP动作、JSP隐式对象、EL表达式、JDBC、Servlet、过滤器、MVC开发模式等;第13章是浪潮集团真实企业级项目案例——订单管理系统项目实战。

本书既可作为高等院校计算机相关专业的教材和辅导书,也可作为JSP初学者的入门读物。

-
- ◆ 主 编 刘何秀 郭建磊 姬忠红
责任编辑 张 斌
责任印制 沈 蓉 彭志环
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
固安县铭成印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 16.75 2018年5月第1版
字数: 451千字 2018年8月河北第2次印刷
-

定价: 49.80 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

JSP (Java Server Pages) 是一种动态网页技术标准, 该技术为创建显示动态生成内容的 Web 页面提供了一个简捷而快速的方法。JSP 继承了 Java 语言的优点, 具有跨平台的特性, 而 Java 技术也为 JSP 实现强大功能提供了技术支持。

JSP 技术的设计目的是使得构造基于 Web 的应用程序更加容易和快捷, 而这些应用程序能够与各种 Web 服务器、应用服务器、浏览器和开发工具共同工作。JSP 开发动态 Web 应用具有很多优势, 例如可以实现动态页面与静态页面的分离。JSP 开发脱离了硬件平台的束缚, 拥有和 Java 语言同样的“一次编写, 各处运行”的良好移植性。JSP 还强调应用可重用的组件 (JavaBeans 或 Enterprise JavaBeans 组件), 基于组件的方法可以提高企业级应用的可扩展性、提高开发效率, 并且使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。

浪潮集团是中国本土综合实力强大的大型 IT 企业之一, 是国内领先的云计算领导厂商, 是先进的信息科技产品与解决方案服务商。浪潮集团的很多大型企业级项目都采用了以 JSP 技术为基础的 Java EE 开发体系, 应用行业包括金融、医疗、电子政务、粮食储备等。对于开发人员来说, 掌握 JSP 技术, 是进行 Java EE 企业级项目开发的重要基础和必要条件。

浪潮优派科技教育有限公司 (以下简称浪潮优派) 是浪潮集团下属子公司, 结合浪潮集团的技术优势和丰富的项目案例, 致力于 IT 人才的培养。本书由浪潮优派具有多年开发经验和实训经验的 Java EE 培训讲师撰写, 全书各章节知识点讲解条理清晰、循序渐进, 每个知识点有丰富的案例演示, 并有企业级案例的应用演示贯穿全书。本书还提供了丰富的配套案例和微课视频, 读者可扫描二维码直接观看。每章有配套习题和上机实验, 并配有案例源代码和电子课件, 读者可登录人邮教育社区 (www.ryjiaoyu.com) 下载。

本书共 13 章, 各章内容如下。

第 1 章 Java Web 开发基础, 介绍学习 JSP 技术必须要了解的 Web 基础知识。

第 2 章 JSP 概述, 介绍 JSP 的概念、组成、开发环境的配置和运行。

第 3 章 JSP 脚本元素、指令, 对 JSP 中 3 种脚本元素和 JSP 的 3 个指令进行了详细讲解和演示。

第 4 章 JSP 隐式对象, 对 JSP 的 9 个内置对象进行了详细的讲解和案例演示。

第 5 章 JSP 标准动作, 详细演示 JavaBean 组件和 JSP 7 个标准动作的使用。

第 6 章 JSP 表达式语言, 讲解了 EL 表达式语言的语法, 并演示了各种场景下的应用方法。

第 7 章 JSP 中使用数据库, 详细讲解了 JDBC 技术及常用 API, 以 Oracle 为例, 演示了 JSP 连接和操作数据库的方法。

第 8 章 JSTL 概述, 讲解并演示了 JSP 标准标签库和核心标签的使用。

第 9 章 Servlet 概述, 讲解并演示了 Servlet 的概念、生命周期、创建和访问方法。

第 10 章 Servlet API, 讲解 Servlet 处理 HTTP 请求的流程, 并演示主要 API 的使用。

第 11 章 Servlet 过滤器, 讲解过滤器的原理并演示创建方法和典型应用案例。

第 12 章 MVC 开发模式，讲解两种开发模式及应用案例。

第 13 章综合案例：订单管理系统，采用浪潮集团真实的项目案例，详细讲解并演示基于 JSP 的 MVC 开发模式企业级的应用。

本书由刘何秀、郭建磊、姬忠红担任主编，并进行了全书审核和统稿。参与本书编写的还有崔瑞娟、周业勤、李海斌、李然。本书的综合案例订单管理系统来自浪潮世科，感谢他们在本书撰写过程中提供的帮助和支持。

由于时间仓促和编者水平有限，本书难免存在不足之处，欢迎读者朋友批评指正。

第1章 Java Web 开发基础..... 1

1.1 Web 基础知识	1
1.2 JSP 基础知识	3
1.3 本章小结	6
习题	6
上机指导	6

第2章 JSP 概述.....7

2.1 初识 JSP	7
2.1.1 JSP 概念	7
2.1.2 JSP 页面组成	7
2.2 安装配置 JSP 运行环境	9
2.2.1 JDK 安装及配置	9
2.2.2 Tomcat 下载安装与启动	9
2.2.3 Web 服务目录	12
2.3 JSP 执行过程	14
2.3.1 JSP 执行过程分析	14
2.3.2 JSP 执行过程转译文件	15
2.4 使用 MyEclipse 开发 JSP	16
2.5 本章小结	20
习题	20
上机指导	21

第3章 JSP 脚本元素、指令.....22

3.1 JSP 脚本元素	22
3.1.1 JSP 脚本元素概念及组成	22
3.1.2 表达式	22
3.1.3 Scriptlet	24
3.1.4 声明	27
3.2 JSP 指令	29
3.2.1 JSP 指令概念与分类	29
3.2.2 page 指令	30

3.2.3 include 指令	32
------------------------	----

3.2.4 taglib 指令	33
-----------------------	----

3.3 本章小结	35
----------------	----

习题	35
----------	----

上机指导	36
------------	----

第4章 JSP 隐式对象..... 37

4.1 JSP 隐式对象概述	37
4.2 输入/输出对象	38
4.2.1 out 对象	38
4.2.2 request 对象	40
4.2.3 response 对象	45
4.3 作用域通信对象	51
4.3.1 session 对象	51
4.3.2 application 对象	54
4.3.3 pageContext 对象	57
4.4 Servlet 对象	60
4.4.1 page 对象	60
4.4.2 config 对象	61
4.5 错误对象	62
4.6 本章小结	64
习题	64
上机指导	65

第5章 JSP 标准动作..... 66

5.1 JavaBean 组件	66
5.2 常用的 JSP 动作	68
5.2.1 <jsp:forward>动作	69
5.2.2 <jsp:param>动作	71
5.2.3 <jsp:include>动作	74
5.2.4 <jsp:useBean>动作	81
5.2.5 <jsp:getProperty>动作	84
5.2.6 <jsp:setProperty>动作	85

5.2.7 <jsp:plugin>动作	89	8.2.2 流程控制	142
5.3 本章小结	90	8.2.3 迭代操作	145
习题	90	8.2.4 URL 操作	150
上机指导	90	8.3 本章小结	155
第 6 章 JSP 表达式语言	91	习题	156
6.1 EL 简介和基本语法	91	上机指导	156
6.2 EL 常见应用	93	第 9 章 Servlet 概述	157
6.2.1 EL 获取数据	93	9.1 Servlet 简介	157
6.2.2 EL 执行运算	96	9.1.1 认识 Servlet	157
6.2.3 EL 获得 Web 开发常用对象	100	9.1.2 实现 Servlet	157
6.2.4 使用 EL 调用 Java 方法	102	9.1.3 Servlet 的生命周期	159
6.3 综合案例	104	9.2 使用 MyEclipse 演示 Servlet	162
6.4 本章小结	105	9.3 本章小结	164
习题	106	习题	164
上机指导	106	上机指导	165
第 7 章 JSP 中使用数据库	107	第 10 章 Servlet API	166
7.1 JDBC 概述	107	10.1 Servlet 规范和 HTTP Servlet 基础知识	166
7.2 JDBC 常用 API	108	10.2 Servlet API	168
7.2.1 DriverManager 类	108	10.3 ServletConfig 接口	169
7.2.2 Connection 接口	110	10.4 ServletContext 接口	172
7.2.3 Statement 接口	110	10.5 ServletRequest 接口	176
7.2.4 ResultSet 接口	116	10.6 ServletResponse 接口	176
7.2.5 ResultSetMetaData 接口	117	10.7 Servlet 异常	177
7.2.6 PreparedStatement 接口	117	10.8 HttpServletRequest 接口	177
7.2.7 CallableStatement 接口	120	10.9 HttpServletResponse 接口	187
7.3 使用 JDBC 进行事务处理	130	10.10 Web 资源重定向	189
7.4 本章小结	132	10.11 cookie 技术	191
习题	132	10.12 本章小结	196
上机指导	132	习题	196
第 8 章 JSTL 概述	134	上机指导	198
8.1 JSTL 简介	134	第 11 章 Servlet 过滤器	199
8.1.1 JSTL 概念和标签库	134	11.1 Servlet 过滤器简介	199
8.1.2 JSTL 配置的方式	135	11.2 Servlet 过滤器的实现和 生命周期	200
8.2 核心标签库	136		
8.2.1 表达式操作	136		

11.2.1 实现 Servlet 过滤器的 Filter 组件介绍及实现	200	12.3 MVC 模式的案例演示	222
11.2.2 实现 Filter 接口的方法	204	12.4 本章小结	224
11.2.3 Filter 过滤器的生命周期和 拦截流程	207	习题	225
11.3 Servlet 过滤器的功能	210	上机指导	225
11.4 本章小结	214	第 13 章 综合案例：订单管理 系统	226
习题	214	13.1 项目背景及项目结构	226
上机指导	215	13.2 数据库的设计	231
第 12 章 MVC 开发模式	216	13.3 环境搭建	236
12.1 MVC 的模式简介	216	13.4 系统管理	237
12.2 JSP 开发的两种模型	216	13.5 实现用户登录	246
12.2.1 Model1	217	13.6 实现货币管理	253
12.2.2 Model2	220	13.7 本章小结	260

学习目标

- 了解 Web 相关基础知识
- 了解 JSP 相关基础知识
- 理解 C/S 体系结构和 B/S 体系结构

Web 基础
知识

1.1 Web 基础知识

JSP (Java Server Pages) 是目前企业中常用的一种动态网页开发技术, 是 Java Web 开发体系中非常重要的一门技术。在学习 JSP 之前, 首先需要了解与 Web 相关的一些知识, 本节将从 Web 基础、HTTP 协议、Web 客户端应用技术、Web 服务器端应用技术等方面进行介绍。

1. Web 基础

Web 的本意是蜘蛛网和网, 在网页设计中称为网页。Web 出现于 1989 年 3 月, 由欧洲粒子物理研究所 (CERN) 的科学家蒂姆·伯纳斯·李 (Tim Berners Lee) 发明。1990 年 11 月, 第一个 Web 服务器正式运行, 通过 Web 浏览器可以看到最早的 Web 页面。1991 年, Web 技术标准正式发布。1993 年, 第一个图形界面的浏览器 Mosaic 开发成功。1995 年, 著名的 Netscape Navigator 浏览器问世。随后, 微软 (Microsoft) 公司推出了著名的 IE 浏览器 (Windows 操作系统默认安装 IE 浏览器)。目前, 与 Web 相关的各种技术标准都由万维网联盟 (World Wide Web Consortium, W3C) 管理和维护。

Web 是一个分布式的超媒体信息系统, 它将大量的信息分布在网上, 为用户提供更多的多媒体网络信息服务。从技术层面上看, Web 技术的核心有以下 3 点。

- 超文本传输协议 (HTTP), 实现网络的信息传输。
- 统一资源定位符 (URL), 实现互联网信息定位的统一标识 (如 <http://www.inspuruptec.com> 中的 “www.inspuruptec.com”)。
- 超文本标记语言 (HTML), 实现信息的表示与存储。

2. HTTP 协议简介

超文本传输协议 (HyperText Transfer Protocol, HTTP) 是专门为 Web 设计的一种应用层协议。在 Web 应用中, 服务器把网页传给浏览器, 实际上就是把网页的 HTML 代码发送给浏览器, 让浏览器显示出来。而浏览器和服务器之间的传输协议

就是 HTTP。也就是说, HTTP 协议是 Web 浏览器与 Web 服务器之间的一问一答的交互过程中必须遵循的规则。

HTTP 是 TCP/IP 协议集中的一个应用层协议,用于定义 Web 浏览器与 Web 服务器之间交换数据的过程以及数据本身的格式。HTTP 协议的版本有 HTTP/1.0、HTTP/1.1、HTTP-NG。深入理解 HTTP 协议,对管理和维护复杂的 Web 站点、开发具有特殊用途的 Web 服务器程序具有直接影响。

3. Web 客户端应用技术

Web 是开发互联网应用的技术总称,它是一种典型的分布式应用架构。Web 应用中的每一次信息交换都要涉及客户端和服务端两个层面。因此,Web 开发技术大体上也可以被分为客户端技术和服务端技术两类。本节主要介绍 Web 客户端技术。

Web 客户端的主要任务是展现信息内容。Web 客户端设计技术主要包括:HTML 语言、Java Applets、脚本程序、CSS、DHTML、插件技术及 VRML 技术。

(1) HTML 语言的诞生

Web 客户端的主要任务是展现信息内容,HTML 语言是信息展现的最有效载体之一。作为一种实用的超文本语言,HTML 的历史最早可以追溯到 20 世纪 40 年代。1969 年,IBM 公司的查尔斯·戈德法布(Charles Goldfarb)发明了可用于描述超文本信息的 GML 语言。1978—1986 年,在美国国家标准学会(American National Standards Institute, ANSI)等组织的努力下,GML 语言进一步发展成为著名的 SGML 语言标准。当蒂姆·伯纳斯·李在 1989 年试图创建一个基于超文本的分布式应用系统时,意识到 SGML 过于复杂,不利于信息的传递和解析。于是他对 SGML 语言做了大刀阔斧的简化和完善。1990 年,第一个图形化的 Web 浏览器“World Wide Web”终于可以使用一种为 Web 量身定制的语言——HTML 来展现超文本信息了。

(2) 从静态信息到动态信息

最初的 HTML 语言只能在浏览器中展现静态的文本或图像信息,随后由静态技术逐步向动态技术转变。Web 出现后,GIF 动画第一次为 HTML 页面引入了动感元素。1995 年,Java 语言的问世带来了更大的变革。Java 语言天生就具备的与平台无关的特点,让人们找到了在浏览器中开发动态应用的捷径。CSS 和 DHTML 技术真正让 HTML 页面又酷又炫、动感无限起来。1997 年,Microsoft 发布了 IE 4.0,并将动态 HTML 标记、CSS 和动态对象模型发展成了一套完整、实用、高效的客户端开发技术体系,Microsoft 称其为 DHTML。同样是实现 HTML 页面的动态效果,DHTML 技术无需启动 Java 虚拟机或其他脚本环境,可以在浏览器的支持下,获得更好的展现效果和更高的执行效率。

为了在 HTML 页面中实现音频、视频等更为复杂的多媒体应用,HTML 引入了对 QuickTime 插件的支持,插件这种开发方式也迅速风靡了浏览器的世界。20 世纪 90 年代中期间世的 COM 和 ActiveX 也一度很流行。Real Player 插件、Microsoft 自己的媒体播放插件 Media Player 也被预装到了各种 Windows 版本之中。随后,Flash 插件横空出世,被广泛应用于网页动画设计,成为当前网页动画设计最流行的插件之一。

4. Web 服务器端应用技术

与 Web 客户端技术从静态向动态的演进过程类似,Web 服务器端的开发技术也是由静态逐渐向动态发展、完善起来的,其技术也在不断变化。

最早的 Web 服务器只是简单地响应浏览器发来的 HTTP 请求,并将存储在服务器上的 HTML 文件返回给浏览器。

第一种真正使服务器能根据运行时的具体情况动态生成 HTML 页面的技术是大名鼎鼎的 CGI 技术。CGI 技术允许服务端的应用程序根据客户端的请求,动态生成 HTML 页面,这使客户端和服务端的动态信息交换成为可能。

早期的 CGI 程序大多是编译后的可执行程序,其编程语言可以是 C、C++、Pascal 等任何通用的程序设计语言。为了简化 CGI 程序的修改、编译和发布过程,人们开始探寻用脚本语言实现 CGI 应用的可行方式。

1994 年,出现了专用于 Web 服务器端编程的 PHP 语言。PHP 语言将 HTML 代码和 PHP 指令合成为完整的服务端动态页面,可以用一种更加简便、快捷的方式实现动态 Web 功能。

1996 年,Microsoft 在其 Web 服务器 IIS 3.0 中引入了 ASP 技术。ASP 使用的脚本语言是大众熟悉的 VB Script 和 JavaScript。

1998 年,JSP 技术诞生,JSP 页面使用的脚本语言是 Java 语言,JSP 开发服务器端的动态网页具有很多优势,因此成为很多开发者选择使用的主流 Web 服务器端开发技术,JSP 的优点见下节的介绍。

随后,XML 语言及相关技术又成为主流。XML 语言对信息的格式和表达方法做了最大程度的规范,应用软件可以按照统一的方式处理所有 XML 信息,信息在整个 Web 世界里的共享和交换就有了技术上的保障。HTML 语言关心的是信息的表现形式,而 XML 语言关心的是信息本身的格式和数据内容。



1.2 JSP 基础知识

JSP 作为 Java Web 开发体系中的核心技术,学习者除了要了解以上 Web 相关的基础知识,还应该对 JSP 依赖的先行知识有所了解,如 Java 语言、Servlet 技术 (Servlet 在后面章节中有详细讲解)、JSP 所开发的应用程序体系架构。下面针对这些内容进行介绍。

1. Java

Java 是 Sun 公司 (已被 Oracle 公司收购) 于 1995 年推出的面向对象的编程语言,一经推出,便吸引了全世界的目光,得到了业界的广泛应用和一致好评。Java 语言适用于 Internet 环境,Java Web 开发体系已经成为开发 Internet 应用的主要技术。Java 具有简单、面向对象、分布式、健壮、安全、平台独立与可移植性强、支持多线程、动态性好等特点。Java 除了用于开发 Web 应用程序,还可以编写桌面应用程序、分布式系统和嵌入式系统应用程序等。

Java 是一门完全面向对象的编程语言,不仅吸收了 C++ 语言的各种优点,还摒弃了 C++ 中难以理解的多继承、指针等概念,因此,Java 语言具有功能强大的特点,同时还具有简单易用的特征。Java 语言作为面向对象编程语言的代表,极好地实现了面向对象理论,允许程序员以优雅的思维方式进行复杂的编程。

Java 的开发和运行依赖开发环境 JDK (Java Development Kit),JDK 也称为 Java 开发包或 Java 开发工具。JDK 是整个 Java 的核心,包括了 Java 运行环境 (Java Runtime Environment, JRE)、一些 Java 工具和 Java 的核心类库 (Java API)。

2. Servlet

Servlet 是先于 JSP 出现的 Java Web 开发技术。Servlet 是一种服务器端的 Java 应用程序,具有独

立于平台和协议的特性,可以生成动态的 Web 页面。它担当客户请求(Web 浏览器或其他 HTTP 客户程序)与服务器响应(HTTP 服务器上的数据库或应用程序)的中间层。Servlet 是位于 Web 服务器内部的服务器端的 Java 应用程序,与传统的从命令行启动的 Java 应用程序不同,Servlet 由 Web 服务器进行加载,该 Web 服务器必须包含支持 Servlet 的 Java 虚拟机。

Servlet 与传统的 CGI 比较,具有使用方便、功能强大、可移植性好、架构设计先进等优点。但是 Servlet 在页面展现方面输出 HTML 语句还是采用了传统的 CGI 方式,需要在 Java 代码中一句句输出,编写和修改 HTML 非常不方便。后来便出现了基于 Java 语言的服务器页面 JSP,大大简化了页面的编写和维护的过程。

3. JSP

JSP 全称为 Java Server Pages,即 Java 服务器页面,是一种实现普通静态 HTML 和动态 HTML 混合编码的技术,JSP 并没有增加任何本质上不能用 Servlet 实现的功能。但是,在 JSP 中编写静态 HTML 更加方便,不必再用 `println` 语句来输出每一行 HTML 代码。更重要的是,借助内容和外观的分离,页面制作中不同性质的任务可以方便地分开。例如,由页面设计者进行 HTML 设计,同时留出供 Servlet 程序员插入动态内容的空间。

JSP 是基于 Java 的技术,用于创建可支持跨平台及 Web 服务器的动态网页。JSP 页面代码一般由普通的 HTML 语句和特殊的基于 Java 语言的嵌入标记组成,所以它具有 Web 和 Java 功能的双重特性。

JSP 1.0 规范是 1999 年 9 月推出的,同年 12 月又推出了 1.1 规范。此后,JSP 又经历了几个版本,本书介绍的技术是基于 JSP 2.0 规范的。

JSP 是一种动态网页技术标准,可以分离网页中的动态部分和静态的 HTML。用户可以使用平常得心应手的工具按照平常的方式来书写 HTML 语句,然后将动态部分用特殊的标记嵌入即可,这些标记常常以“<%”开始并以“%>”结束。

同 HTML 以及 ASP 等语言相比,JSP 虽然在表现形式上同它们的差别并不大,但是它却提供了一种更为简便、有效的动态网页编写手段,而且,JSP 程序同 Java 语言有着天然的联系,因此在众多基于 Web 的架构中,都可以看到 JSP 程序。

JSP 程序增强了 Web 页面程序的独立性、兼容性和可重用性,与传统的 ASP、PHP 网络编程语言相比,它具有以下特点。

- JSP 的执行效率比较高。由于每个基于 JSP 的页面都被 Java 虚拟机解析成一个 Servlet,服务器通过网络接收到来自客户端 HTTP 的请求后,Java 虚拟机解析产生的 Servlet 将开启一个“线程(Thread)”来提供服务,并在服务处理结束后自动销毁这个线程。这样的处理方式将大大提高系统的利用率,并能有效地降低系统的负载。

- 编写简单。JSP 是基于 Java 语言和 HTML 元素的一项技术,因此,只要熟悉 Java 和 HTML 的程序员都可以开发 JSP。

- 跨平台。JSP 运行在 Java 虚拟机之上,因此,它可以借助于 Java 本身的跨平台能力,在任何支持 Java 的平台和操作系统上运行。

- JSP 可以嵌套在 HTML 或 XML 网页中。这样不仅可以降低程序员开发页面显示逻辑效果的工作量,更能提供一种比较轻便的方式来同其他 Web 程序交互。

4. 应用程序体系结构

目前, 在应用开发领域中主要分为两种应用程序体系结构: 一种是 C/S (客户端/服务器) 体系结构, 另一种是 B/S (浏览器/服务器) 体系结构。基于 Web 的动态网站开发技术 (例如 JSP) 开发的应用程序都是采用 B/S 体系结构。下面对这两种体系结构进行介绍。

(1) C/S (客户端/服务器) 体系结构

C/S 结构把数据库内容放在远程的服务器上, 而在客户机上安装相应软件。C/S 软件一般采用两层结构, 由两部分构成: 前端是客户机, 即用户界面结合了表示与业务逻辑, 接受用户的请求, 并向数据库服务提出请求, 通常是一台个人计算机; 后端是服务器, 即数据管理将数据提交给客户端, 客户端将数据进行计算并将结果呈现给用户。

C/S 体系结构具有强大的数据操作和事务处理能力, 模型思想简单, 易于人们理解和接受, 随着企业规模的日益扩大, 软件的复杂程度不断提高, 传统的二层 C/S 结构存在着很多局限, 因此, 三层 C/S 体系结构应运而生, 其结构如图 1-1 所示。在三层 C/S 体系结构中, 增加了一个应用服务器, 可以将整个应用逻辑驻留在应用服务器上, 只有表示层存在于客户机上。这种结构被称为“瘦客户机”。三层 C/S 体系结构将应用功能分成表示层、功能层和数据层。

表示层是应用的用户接口部分, 担负着用户与应用的对话功能, 用于检查用户从键盘等输入的数据, 显示应用输出的数据。功能层相当于应用的本体, 是将具体的业务处理逻辑编入程序中。而数据层就是数据库管理系统, 负责管理对数据库数据的读写。在三层 C/S 体系结构中, 中间件是最重要的构件。所谓中间件, 就是一个用户 API 定义的软件层, 是具有强大通信能力和良好可扩展性的分布式软件管理框架。其功能是在客户机和服务器或服务器和服务器之间传送数据, 实现客户机群和服务器群之间的通信。

(2) B/S (浏览器/服务器) 体系结构

B/S 结构, 就是只安装维护一个服务器, 而客户端采用浏览器运行软件。该结构是随着 Internet 技术的兴起, 对 C/S 结构的一种变化和改进。主要利用了不断成熟的 WWW 浏览器技术, 结合多种 Script 语言和 ActiveX 技术, 是一种全新的软件系统构造技术。JSP、Servlet 技术开发的应用程序都是 B/S 结构。采用 B/S 结构的计算机应用系统的基本框架如图 1-2 所示。

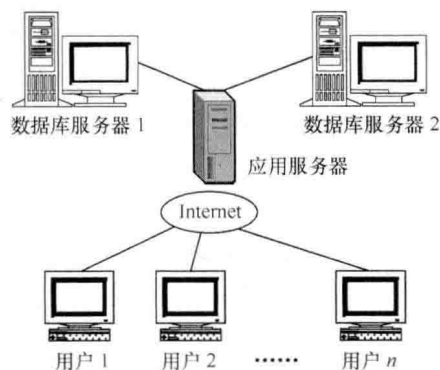


图 1-1 三层 C/S 结构示意图

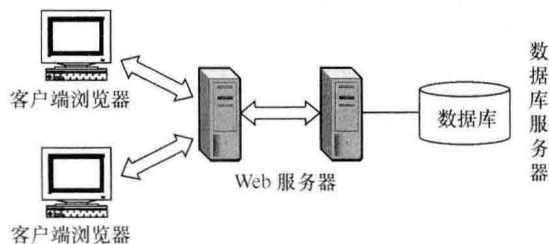


图 1-2 B/S 模式结构

B/S 三层体系结构采用三层客户/服务器结构, 在数据管理层和用户界面层增加了一层结构 (即中间件), 使整个体系结构成为三层。三层结构是伴随着中间件技术的成熟而兴起的, 核心概念是利

用中间层将应用分为表示层、业务逻辑层和数据存储层三个不同的处理层次。三个层次是从逻辑上划分的，具体的物理分法可以有多种组合。中间件作为构造三层结构应用系统的基础平台，提供了以下主要功能：负责客户机与服务器、服务器与服务器间的连接和通信；实现应用与数据库的高效连接；提供一个三层结构应用的开发、运行、部署和管理的平台。这种三层结构在层与层之间相互独立，任何一层的改变不会影响其他层的功能。

1.3 本章小结

本章介绍了 Web 基础知识，包括：Web 介绍、HTTP 协议介绍、Web 客户端应用技术和 Web 服务器端应用技术介绍。此外，本章还对 JSP 的基础知识进行了介绍，包括：Java、Servlet、应用程序体系结构。读者需要了解这些基础知识，为学习 JSP 技术打下基础。

习 题

1. 什么是 Web 技术？
2. 什么是 HTTP 协议？
3. 什么是 Web 客户端应用技术？列出几种典型的 Web 客户端技术。
4. 什么是 Web 服务器端应用技术？列出几种典型的 Web 服务器端技术。

上 机 指 导

创建一个静态网页，在浏览器正中间显示“Hello, JSP”。

学习目标

- 理解 JSP 的概念
- 理解 JSP 页面的各种构成元素
- 掌握安装配置 JSP 运行环境
- 掌握 JSP 页面的执行过程
- 会使用 MyEclipse 创建 JSP 页面

2.1 初识 JSP

2.1.1 JSP 概念



JSP 全名为 Java Server Pages (Java 服务器页面), 其根本是一个简化的 Servlet 设计, 它是由 Sun 公司倡导、多家公司参与一起建立的一种动态网页技术标准。

JSP 技术有些类似 ASP 技术, 它是在传统的网页 HTML (标准通用标记语言的字集) 文件 (*.htm, *.html) 中插入 Java 程序段 (Scriptlet) 和 JSP 标记 (tag)。形成的 JSP 文件, 后缀名为 “.jsp”。用 JSP 开发的 Web 应用是跨平台的, 既能在 Linux 下运行, 又能在其他操作系统上运行。

JSP 将网页逻辑与网页设计的显示分离, 支持可重用的基于组件的设计, 使基于 Web 的应用程序的开发变得迅速和容易。JSP 是一种动态页面技术, 它的主要目的是将表示逻辑从 Servlet 中分离出来。

2.1.2 JSP 页面组成



1. 组成

一个 JSP 页面由两部分组成: 静态部分 (如 HTML、CSS 标记等, 用来完成数据显示和样式) 和动态部分 (如 JSP 指令、JSP 脚本元素和变量等, 用来完成数据处理), 如图 2-1 所示。

2. 实例: 第一个 JSP 案例

新建一个 welcome.jsp 页面, 创建 JSP 页面时会自动创建带有例如 `<%@ page..%>` 的 JSP 指令元素、`<%=basePath%>` 的 JSP 表达式等页面元素 (代码详见 `/jspdemopro/WebRoot/ch2/welcome.jsp`)。

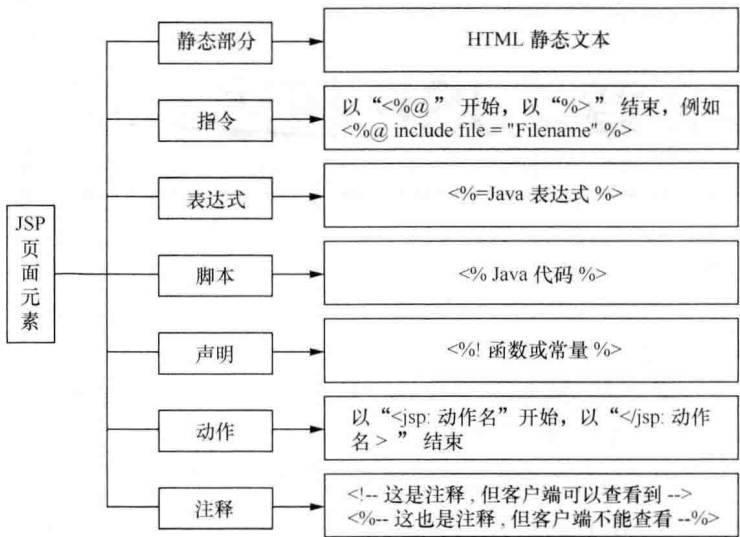


图 2-1 JSP 页面元素

```
<%@ page language="java" import="java.util.*" pageEncoding="utf-8"%>
<%
String path = request.getContextPath();
String basePath = request.getScheme()
+"://"+request.getServerName()
+": "+request.getServerPort()+path+"/";
%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>
<head>
<title>欢迎页面</title>
<meta http-equiv="pragma" content="no-cache">
</head>
<body>
<%!
String getHello(String name){
    return "Hello "+name+"!";
}
%>
<h1>你好，这就是一个 JSP 页面! <%=getHello("Jack") %>
</h1>
</body>
</html>
```

Diagram annotations:

- JSP 指令: points to the page directive.
- Java 代码: points to the scriptlet.
- HTML 静态文本: points to the HTML structure (head, body, title, meta, h1).
- JSP 声明: points to the declaration.
- JSP 表达式: points to the expression.

该页面的运行结果如图 2-2 所示。

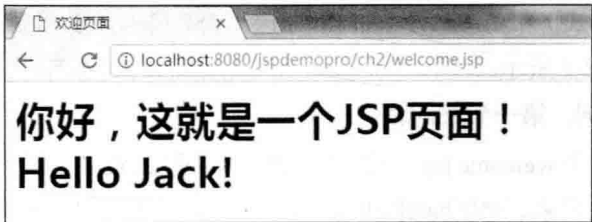


图 2-2 欢迎页面

2.2 安装配置 JSP 运行环境

自从 JSP 发布以后,出现了各式各样的 JSP 引擎。而 JSP 引擎其实就是一种统一管理和运行 Web 应用程序的软件。

1999 年 10 月, Sun 公司将 Java Server Page 1.1 代码交给 Apache 组织, Apache 组织对 JSP 进行了实用研究,并将这个服务器项目称为 Tomcat,从此,著名的 Web 服务器 Apache 开始支持 JSP,于是 Tomcat 就诞生了。目前, Tomcat 能和大部分主流服务器一起高效率的工作。

Tomcat 是一个免费的开源的 JSP 引擎,也称作 Tomcat 服务器。读者可以登录 Tomcat 的官方网址并找到相应的版本进行下载。在安装或运行 Tomcat 之前,必须首先安装 JDK。



JDK 安装及
配置

2.2.1 JDK 安装及配置

运行 Web 项目之前,需要先安装 Java 开发环境 JDK。而 JDK 的安装软件,读者可以登录 Oracle 官网进行下载,下载完后进行安装即可。

安装 JDK 之后,需要设计环境变量。对应 Windows 10 和 Windows XP 操作系统,用鼠标右键单击“计算机”|“我的电脑”,在弹出的快捷菜单中选择“属性”命令,弹出“系统特性”对话框,单击该对话框中的“高级系统设置”|“高级选项”,在弹出的对话框中单击“环境变量”按钮,分别添加如下的系统环境变量。

变量名: JAVA_HOME, 变量值: C:\Program Files\Java\jdk1.7.0_60。

变量名: Path, 变量值: %JAVA_HOME%\bin。

变量名: CLASSPATH, 变量值: .;%JAVA_HOME%\lib;
%JAVA_HOME%\lib\tools.jar。

如果曾经设置过环境变量 JAVA_HOME 和 Path,可单击该变量进行编辑操作,将环境变量的值加入即可,如图 2-3 ~ 图 2-5 所示。注意:各个环境变量值之间必须用分号分隔。

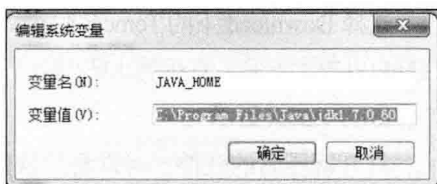


图 2-3 设置 JAVA_HOME

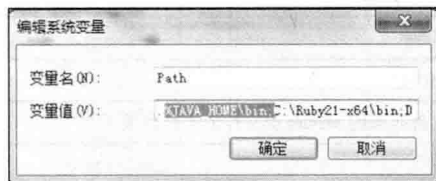


图 2-4 编辑 Path

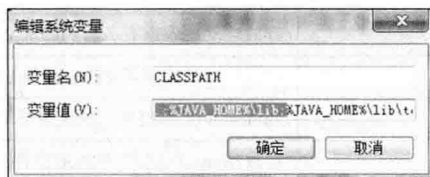


图 2-5 设置 CLASSPATH

环境变量解析: JAVA_HOME 配置的是 JDK 的安装路径; CLASSPATH 配置 Java 加载类路径,只有类在 CLASSPATH 中 Java 命令才能识别,在路径前加“.”表示当前路径。PATH 则配置的是系统在任何路径下都可以识别的 Java、Javac 命令。

2.2.2 Tomcat 下载安装与启动

1. Tomcat 下载

在 Tomcat 官网可以直接找到各个版本的 Tomcat 进行下载。图 2-6 所示为 Tomcat



Tomcat 下载
安装与启动