



实例源代码 · 程序软件包 · 行业规范

JSP

应用与开发技术

马建红 李占波 主 编
韩 颖 王瑞娟 姬莉霞 副主编



清华大学出版社

JSP 应用与开发技术

马建红 李占波 主 编

· 韩 颖 王瑞娟 姬莉霞 副主编



清华大学出版社

北 京

内 容 简 介

JSP(Java Server Pages)是由 Sun Microsystems 公司倡导的、许多公司参与一起建立的一种动态网页技术标准。Java 技术的强大功能也赋予了 JSP, 能够为用户提供功能强大的技术支持, JSP 继承了 Java 的优势从而可以建立安全的、跨平台的动态网站。

本书详细讲解了 JSP 的语法和基本的 Web 程序设计方法。全书共 16 章, 可分为 7 个部分。第 1~3 章是第 1 部分, 介绍了 JSP 应用技术的前导知识和开发环境的搭建; 第 4~6 章是第 2 部分, 详细讲解了 JSP 技术的基本语法; 第 7 章和第 8 章是第 3 部分, 是 JSP 应用开发的进阶, 讲述了 JavaBean、表单处理以及文件的操作; 第 9~11 章是第 4 部分, 以 MySQL 数据库为例详细讲解了 JSP 中使用数据库的操作; 第 12 章和第 13 章是第 5 部分, 详细讲述了 Servlet 技术; 第 14 章和第 15 章是第 6 部分, 介绍了标准标签库(JSTL)和自定义标签库; 第 16 章是第 7 部分, 以一个实例讲述了 Web 开发的实际应用。

本书光盘中收录了相关实例运行的开源程序、实例源代码, 读者可参照光盘说明进行调试运行。此外, 读者还可以通过 www.tupwk.com.cn 下载本书的实例源代码和电子课件。

本书语言简洁, 内容丰富, 既可作为 JSP 初学者的入门教材, 也可作为高等院校相关专业的教材和辅导用书, 而且对 JSP 开发人员的自学也具有较高的参考价值。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

JSP应用与开发技术/马建红, 李占波 主编. —北京: 清华大学出版社, 2011.2

ISBN 978-7-302-24762-3

I. ①J… II. ①马… ②李… III. ①JAVA语言—主页制作—程序设计 IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2011)第 012248 号

责任编辑: 王 定

封面设计: 李小凤

版式设计: 孔祥丰

责任校对: 成凤进

责任印制: 王秀菊

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京市世界知识印刷厂

装 订 者: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 185×260 印 张: 32 字 数: 799 千字

附光盘 1 张

版 次: 2011 年 2 月第 1 版

印 次: 2011 年 2 月第 1 次印刷

印 数: 1~5000

定 价: 58.00 元

前言

P R E F A C E

JSP(Java Server Pages)是目前动态网站开发技术中最典型的一种,它继承了Java语言的优势,是一种与平台无关的开发技术,而Java技术也赋予了JSP为用户提供强大功能的技术支持。JSP实现了动态页面与静态页面的分离,脱离了硬件平台的束缚,提高了执行效率而成为因特网上的主流开发技术,已经越来越受到编程者的关注和喜爱。

JSP虽然综合性地包括了Java和HTML这两类语法,但不能通过简单地使用JSP,让它集显示、业务逻辑和流程控制于一身,因为用这种方式开发出来的Web应用程序是非常难以维护的。所以对JSP使用观念的建立,以及JavaBean、数据库、Servlet等技术的了解运用是利用JSP开发复杂的商业级网站的重点。为了让读者在学习的过程中能够彻底掌握相关概念,除了基本语法介绍外,本书也同时将重点集中在面向对象的观点和JSP程序架构方面。

本书结合JSP和Servlet的最新规范,从基本的语法和规范入手,以经验为后盾,以实用为目标,以实例为导向,以实践为指导,深入浅出地讲解了JSP开发中的种种问题。以精简的内容介绍了JSP的语法、Servlet技术、JDBC技术、标签库技术、MVC等技术,本书每一章节的实例读者都可以直接使用。且实例讲解过程条理清晰、循序渐进,符合程序设计的自然思路,读者学完一个章节,也就相应地掌握了相关的JSP开发思想和技术。并且通过几个较完整的综合实例,让读者对JSP+Servlet+JavaBean+JDBC这样的Web架构有个整体认识。

本书共16章,可以分成7个部分。

第1部分介绍了JSP编程基础与环境配置,包括3章:第1章JSP概述,第2章JSP动态网页设计基础,第3章JSP环境配置。

第2部分介绍了JSP应用开发基础,包括3章:第4章JSP基本语法,第5章JSP内置对象,第6章使用Cookie记录信息。

第3部分介绍了JSP应用开发进阶,包括2章:第7章JavaBean和表单处理,第8章JSP中文件的操作。这部分主要介绍了JavaBean分离表示与实现、使用JSP处理HTML表单、使用Java类库里的I/O类,开发具备文件存取功能的网页程序。

第4部分介绍了JSP数据库编程基础,包括3章:第9章数据库操作基础;第10章应用JDBC进行数据库开发;第11章JSP+JavaBean的应用实例,以MySQL数据库为例详细讲解了JSP中使用数据库的操作,包括JDBC、连接池、分页处理等技术。

第5部分讲述了Servlet技术,包括2章:第12章Servlet基础,第13章使用Servlet过滤器和监听器,通过实例讲述了Servlet的开发和部署。

第6部分讲述了标签库的应用,包括2章:第14章标准标签库(JSTL),第15章创建自

定义标签库。这部分是使用 JSP 进行程序开发的扩展知识。

第 7 部分即第 16 章，通过一个网上书店实例讲解了 Web 开发的具体应用。

本书采用由浅入深、循序渐进的方法，全面系统地介绍了 JSP 程序设计的原理、方法和技术，还提供了大量的 JSP 应用开发实例，给出了相应的实用技巧、操作步骤及优化思路使读者可以很快地进行实际开发。在本书每章的最后还提供了习题，让读者能够检验自己对各章内容的学习、消化程度，并巩固所学到的知识。

本书由马建红、李占波主编，马建红、韩颖、王瑞娟、姬莉霞执笔编写，参与编写的人员还有卫权岗、尹辉、程凤娟、秦兴桥、赵玉娟、王永皎等。此外，林楠、李翠霞、王海玲、史苇航、张晗在整理材料方面给予了编者很大的帮助，在此，对他们表示衷心的感谢。

由于时间仓促，加之水平有限，书中不足之处在所难免，敬请读者批评指正。

编 者

目录

CONTENTS

第1章 JSP 概述	1
1.1 软件编程体系简介	2
1.1.1 C/S 结构编程体系	2
1.1.2 B/S 结构编程体系	3
1.2 企业应用开发架构	3
1.2.1 两层架构	3
1.2.2 三层架构	3
1.2.3 N 层架构	4
1.2.4 开发架构比较	4
1.3 JSP 概述	4
1.3.1 什么是 JSP	5
1.3.2 JSP 技术原理	6
1.3.3 JSP 和其他动态网站开发技术	7
1.4 JSP 知识体系及学习之路	8
1.4.1 JSP 知识体系	8
1.4.2 JSP 程序员学习路径	9
1.5 小结	10
1.6 习题	10
第2章 JSP 动态网页设计基础	11
2.1 HTML 快速入门	12
2.1.1 HTML 基本结构	12
2.1.2 HTML 常用标签	13
2.1.3 表单	26
2.1.4 XML 与 XHTML	31
2.2 JavaScript 技术	34
2.2.1 JavaScript 语法	34
2.2.2 JavaScript 使用方式	35
2.2.3 JavaScript 应用实例	36
2.3 小结	42

2.4 习题	42
第3章 JSP 的开发和运行环境	43
3.1 JSP 的开发和应用平台的介绍	44
3.1.1 Caucho 公司的 Resin 平台	44
3.1.2 Apache 公司的 Tomcat 平台	44
3.1.3 BEA 公司的 WebLogic 平台	45
3.1.4 IBM WebSphere Application Server 平台	45
3.2 Eclipse 集成开发环境的配置	46
3.2.1 安装和配置 J2SDK	46
3.2.2 安装配置 Eclipse 集成 开发环境	47
3.2.3 安装和配置 Tomcat 环境	50
3.2.4 在 Eclipse 中开发 JSP 代码	52
3.3 其他 JSP 开发工具的介绍	57
3.3.1 安装配置 MyEclipse 环境	57
3.3.2 JBuilder 开发环境	60
3.3.3 Sun ONE Studio 开发环境	64
3.4 小结	67
3.5 习题	68
第4章 JSP 基本语法	69
4.1 JSP 文件的结构	70
4.1.1 创建第一个 JSP 文件	70
4.1.2 分析 JSP 文件的组成元素	71
4.2 JSP 的脚本元素	72
4.2.1 隐藏注释(Hidden Comment)	72
4.2.2 声明语句(Declaration)	75
4.4.3 脚本段(Scriptlets)	78
4.4.4 表达式(Expression)	80

4.3	JSP 指令元素	80
4.3.1	page 指令	81
4.3.2	include 指令	85
4.3.3	taglib 指令	87
4.4	JSP 动作元素	88
4.4.1	<jsp:include>	88
4.4.2	<jsp:forward>	92
4.4.3	<jsp:param>	95
4.4.4	<jsp:useBean>、 <jsp:setProperty>和 <jsp:getProperty>动作	96
4.4.5	<jsp:plugin>	103
4.5	小结	105
4.6	习题	105
第5章	JSP 内置对象	107
5.1	JSP 内置对象概述	108
5.2	request 对象	109
5.2.1	request 对象常用方法	109
5.2.1	request 对象应用实例	110
5.3	response 对象	117
5.3.1	response 对象的常用方法	118
5.3.2	response 对象应用实例	118
5.4	out 对象	127
5.4.1	out 对象方法成员与 数据输出	127
5.4.2	缓冲区操作	128
5.4.3	out 对象应用实例	129
5.5	session 对象	132
5.5.1	session 的概念	132
5.5.2	session 对象的 Id	132
5.5.3	session 的有效期限	133
5.5.4	访问 session 中的数据	133
5.5.5	其他 session 对象的 常用方法	134
5.5.6	session 对象应用实例	135
5.6	application 内置对象	141

5.6.1	存取 application 中的数据	142
5.6.2	使用 application 对象 取得信息	143
5.6.3	application 对象应用实例	143
5.7	其他 JSP 内置对象	145
5.7.1	pageContext 对象	145
5.7.2	config 对象	148
5.7.3	page 对象	150
5.7.4	exception 对象	150
5.8	小结	151
5.9	习题	151

第6章 使用 Cookie 记录信息 153

6.1	Cookie 的概念和特性	154
6.1.1	什么是 Cookie	154
6.1.2	Cookie 的常见用途	155
6.1.3	对 Cookie 进行适当设置	156
6.2	在 JSP 中使用 Cookie	157
6.2.1	创建 Cookie	157
6.2.2	读写 Cookie	157
6.2.3	Cookie 中的主要方法	158
6.2.4	几个操作 Cookie 的 常用方法	159
6.3	Cookie 对象的应用实例	161
6.4	Cookie 的安全问题	166
6.5	小结	168
6.6	习题	168

第7章 JavaBean 和表单处理 169

7.1	非 MVC 模式(Model1)	170
7.1.1	Model1 的特点	170
7.1.2	Model1 的应用范围	171
7.2	MVC 编程模式(Model2)	171
7.2.1	什么是 MVC 模式	171
7.2.2	MVC 模式在 Web 编程 中的应用	172
7.3	剖析 JavaBean	173
7.3.1	什么是 JavaBean	174

7.3.2	JavaBean 的特征	175	9.3.7	备份数据库	219
7.3.3	创建一个 JavaBean	176	9.3.8	恢复数据库	220
7.4	在 JSP 中使用 JavaBean	178	9.3.9	备份表	220
7.4.1	调用 JavaBean	178	9.3.10	恢复表	220
7.4.2	访问 JavaBean 属性	178	9.3.11	查看、创建、删除和 选择数据库命令	221
7.4.3	设置 JavaBean 属性	178	9.3.12	导入命令	222
7.4.4	JavaBean 的生命周期	179	9.4	常用查询的例子	222
7.4.5	类型自动转换规则	183	9.4.1	查询时间	222
7.5	使用 JavaBean 处理表单数据	183	9.4.2	查询当前用户	222
7.5.1	JSP 处理与 form 相关的 常用标签简单实例	184	9.4.3	查询数据库版本	223
7.5.2	设置中文编码	190	9.4.4	查询当前使用的数据库	223
7.5.3	Post 与 Get 的差异	190	9.4.5	使用 AUTO_INCREMENT	223
7.6	小结	191	9.4.6	列的最大值	224
7.7	习题	191	9.4.7	列的最大值	225
第 8 章	JSP 中的文件操作	193	9.4.8	拥有某个字段的组间 最大值的行	225
8.1	数据流和 File 类	194	9.4.9	使用用户变量	225
8.1.1	数据流	194	9.5	小结	226
8.1.2	File 类	194	9.6	习题	226
8.2	读写文本文件	197	第 10 章	应用 JDBC 进行 数据库开发	227
8.3	文件的浏览	199	10.1	JDBC 概述	228
8.4	创建和删除目录	200	10.1.1	JDBC 的用途	228
8.5	文件的上传和下载	201	10.1.2	JDBC 的典型用法	228
8.6	使用 O'Reilly 上传包	206	10.1.3	JDBC 体系结构	229
8.7	小结	209	10.1.4	驱动器类型	229
8.8	习题	209	10.1.5	安装驱动器	231
第 9 章	数据库操作基础	211	10.2	JDBC 连接数据库的方法	231
9.1	关系数据库及 SQL	212	10.3	使用 JDBC 操作数据库	232
9.2	在 Windows 上安装 MySQL	213	10.3.1	使用 JDBC 访问数据库 的过程	233
9.3	MySQL 的常用操作	217	10.3.2	使用 Statement 执行 SQL 语句	236
9.3.1	设置环境变量	217	10.3.3	PreparedStatement 接口	245
9.3.2	启动 MySQL 数据库	218	10.3.4	CallableStatement 对象	249
9.3.3	连接 MySQL	218			
9.3.4	退出 MySQL	218			
9.3.5	增加用户	219			
9.3.6	删除授权	219			

10.3.5	使用 ResultSet 处理 结果集	251
10.4	Java 与 SQL 的数据 类型转换	256
10.5	使用 JDBC 连接不同 的数据库	258
10.5.1	连接 Oracle 数据库	258
10.5.2	连接 DB2 数据库	258
10.5.3	连接 SQL Server 数据库	258
10.5.4	连接 Sybase 数据库	259
10.5.5	连接 Access 数据库	259
10.6	连接池	259
10.6.1	连接池的实现原理	259
10.6.2	在 Tomcat 上配置 数据源与连接池	260
10.6.3	配置连接池时需要 注意的问题	263
10.7	存取二进制文件	264
10.7.1	图像文件存取到 数据库的过程	264
10.7.2	声音文件存取到 数据库的过程	269
10.7.3	视频文件存取到 数据库的过程	273
10.8	实现分页显示	278
10.8.1	分页显示技术的 优劣比较	278
10.8.2	分页显示的 JavaBean 实现	279
10.9	小结	287
10.10	习题	287
第 11 章	JSP 与 JavaBean 应用实例	289
11.1	留言板模块需求和设计	290
11.1.1	功能介绍	290
11.1.2	文件结构	290
11.1.3	数据库设计	291
11.2	使用 JavaBean 封装 数据库的访问	292
11.3	留言板模块页面实现	301
11.3.1	首页面	301
11.3.2	用户登录功能	302
11.3.3	用户注册功能	305
11.3.4	查看留言信息列表页面	307
11.3.5	查看留言信息内容页面	309
11.3.6	登录用户功能选择页面	310
11.3.7	发表留言页面	311
11.3.8	留言信息保存页面	312
11.3.9	用户退出页面	314
11.4	小结	314
11.5	习题	314
第 12 章	Servlet 基础	315
12.1	Servlet 介绍	316
12.1.1	什么是 Servlet	316
12.1.2	Servlet 技术特点	317
12.1.3	JSP 与 Servlet 的关系	317
12.1.4	Servlet 的工作原理	318
12.1.5	Servlet 常用接口和类	319
12.2	开发部署一个简单的 Servlet	320
12.2.1	创建 Servlet 文件	322
12.2.2	Servlet 的配置文件	323
12.3	Servlet 实现相关的 接口和类	325
12.3.1	GenericServlet	326
12.3.2	HttpServlet	327
12.3.3	Servlet 实现相关实例	327
12.4	Servlet 请求和响应相关	331
12.4.1	HttpServletRequest 接口	331
12.4.2	HttpServletResponse 接口	333
12.4.3	Servlet 请求和响应 相关实例	334

12.5	Servlet 配置相关	337	13.2	Servlet 监听器	390
12.5.1	ServletConfig 接口	337	13.2.1	监听器接口	391
12.5.2	获取 Servlet 配置信息的例子	337	13.2.2	ServletContextListener 接口	392
12.6	Servlet 中的会话追踪	342	13.2.3	ServletContextAttribute Listener 接口	393
12.6.1	HttpSession 接口	342	13.2.4	HttpSessionBindingListener 接口	396
12.6.2	HttpSession 应用实例	344	13.2.5	在线人数统计程序	397
12.7	Servlet 上下文	346	13.3	小结	402
12.7.1	ServletContext 接口	346	13.4	习题	402
12.7.2	ServletContext 接口的应用实例	347	第 14 章	JSTL 标准标签库	403
12.8	Servlet 协作	349	14.1	JSTL 标记库简介	404
12.8.1	RequestDispatcher	349	14.2	设置 JSTL 运行环境	404
12.8.2	forward()控制页面跳转	350	14.2.1	JSTL 的安装	404
12.8.3	include()控制页面包含	351	14.2.2	JSTL 应用示例	405
12.9	Servlet 异常相关	353	14.3	使用核心标签	406
12.9.1	声明式异常处理	353	14.3.1	表达式操作	407
12.9.2	程序式异常处理	355	14.3.2	建立 URL	411
12.10	Servlet 应用实例	358	14.3.3	条件控制	415
12.11	小结	369	14.3.4	迭代-运行循环	417
12.12	习题	369	14.4	使用 JSTL 的数据库标签	420
第 13 章	使用 Servlet 过滤器和监听器	371	14.4.1	指定数据源	420
13.1	过滤器在 Web 开发中的应用	372	14.4.2	进行查询或更新操作	421
13.1.1	过滤器概述	372	14.4.3	对返回的结果进行处理	424
13.1.2	Filter API	373	14.4.4	其他 SQL 标签库的标签	425
13.1.3	Filter 接口	373	14.5	i18n 与国际化	426
13.1.4	FilterConfig 接口	374	14.5.1	国际化设置标签	426
13.1.5	FilterChain 接口	374	14.5.2	消息标记库	427
13.1.6	编写过滤器类	375	14.5.3	数字、日期格式化	429
13.1.7	过滤器的部署	376	14.6	函数标签	433
13.1.8	对请求数据进行处理 的过滤器	380	14.7	小结	433
13.1.9	对响应内容进行压缩 的过滤器	386	14.8	习题	434
第 15 章	自定义标签库	435	15.1	自定义标签体系介绍	436
15.1.1	标签的形式	436			

15.1.2 标签类相关接口和类.....	437	第 16 章 Web 应用开发实践	463
15.1.3 标签库描述文件.....	442	16.1 快乐购书网介绍.....	464
15.1.4 在 Web 部署描述符中 引入标签库文件.....	443	16.2 系统需求和设计.....	464
15.1.5 在页面中使用标签.....	444	16.3 数据库表设计.....	465
15.1.6 标签在 Web 页面中 的作用.....	445	16.4 JSP 页面开发.....	466
15.2 传统标签的开发.....	445	16.5 JavaBean 开发.....	469
15.2.1 带属性标签的开发.....	445	16.5.1 使用 Java Bean 封装 数据库的访问.....	469
15.2.2 带 Body 标签的开发.....	449	16.5.2 购物车 JavaBean.....	475
15.2.3 嵌套标签的开发.....	452	16.6 Servlet 开发.....	478
15.2.4 迭代标签的开发.....	456	16.6.1 Servlet 开发.....	478
15.3 Simple 标签的开发.....	459	16.6.2 Listener 开发.....	490
15.3.1 SimpleTag 接口.....	459	16.6.3 Filter 开发.....	491
15.3.2 Simple 标签的开发示例.....	460	16.7 其他 Bean 类——Util.....	497
15.4 小结.....	462	16.8 部署描述符.....	498
15.5 习题.....	462	16.9 小结.....	501
		16.10 习题.....	502

第 1 章

JSP 概述

本章主要对 JSP 技术进行概要介绍，并给出了一些学习 JSP 的建议。为了让读者在开始学习之前能对 JSP 技术有一个清晰与完整的概念，本章首先介绍了软件编程体系、企业应用开发架构，然后通过编写一个简单的 JSP 页面实例让读者对 JSP 技术有一个直观的感性认识。接着通过介绍 JSP 的技术原理以及与其他主流动态网页技术的比较，进一步了解 JSP 技术是一种功能强大、可以实现跨平台操作的动态网页开发技术。本章对 JSP 知识体系的剖析有助于读者学习和掌握 JSP 知识体系中的各个模块，对 JSP 技术有一个总体性的了解。

本章学习目标：

- ◎ 了解软件编程体系
- ◎ 了解企业应用开发架构
- ◎ 掌握 JSP 的基本概念
- ◎ 掌握 JSP 的知识体系
- ◎ 了解 JSP 的学习之路

1.1 软件编程体系简介

目前,在应用开发领域中主要分为两大编程体系,一种是基于浏览器的 B/S(Browser/Server)结构,另一种是 C/S(Client/Server)结构。应用程序开发体系如图 1-1 所示。

开发基于 C/S 结构项目,传统的开发环境有 VB, VC 以及 Delphi 等,随着 Java 体系以及 .NET 体系的普及,目前更流行 .NET 编程体系和 Java 编程体系。

开发基于 B/S 结构项目,目前主要采用三种服务器端语言: JSP(Java Server Pages)、PHP(Personal Home Page)和 ASP.NET。这三种语言构成三种常用应用开发组合: JSP+Oracle 体系、PHP+MySQL 体系以及 ASP.NET+SQL Server 体系。

软件开发涉及的语言很多,学习起来也是有规律可寻的。图 1-1 最下面的方框将目前常用的开发语言分成两大语系: Basic 语系和 C 语系,语系中的语言所有的流程控制语句都是一样的,常用的函数也大同小异。所以只要精通其中任何一门语言,该语系中的其他语言也就比较容易掌握了。

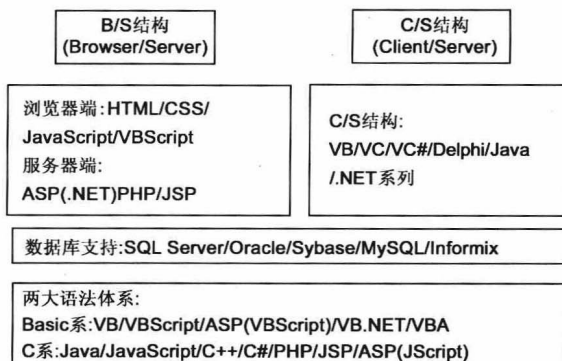


图 1-1 应用程序开发体系

1.1.1 C/S 结构编程体系

2000 年以前, C/S 结构占据开发领域的主流, 随着 B/S 结构的发展, C/S 结构主流地位已经逐步被 B/S 结构取代, 目前在整个开发领域中, C/S 结构的应用大概能占到 40% 的份额。C/S 结构应用程序最大特点是在每个用户端需要安装程序, 所有用户端程序和中心服务器进行信息交互, 这种结构优点是用户端程序一致, 比较方便控制, 服务器端和用户本地的数据很容易进行交互, 通信速度比较快, 缺点是每个用户都需要安装客户端, 比较繁琐, 而且不能很好地跨操作系统平台。

C/S 结构通常适用于具有固定的用户端或者少量的用户端, 并且是对安全性要求比较高的应用。比如银行信息管理系统、邮局信息管理系统和飞机票火车票售票系统等。

传统的 C/S 结构通常使用 PowerBuilder、Delphi、Visual Basic、Visual C++、JBuilder 作为开发环境, 使用 SQL Server、Oracle 或者 DB2 作为数据库支持。随着时间的发展, 版本的更新, 主流的 C/S 开发环境开始向 .NET 和 Java 两大主流体系转变, 目前大部分 C/S 结构应用都使用 VB.NET、VC#.NET 以及 Java 开发, 其中, VB.NET 和 VC#.NET 只是描述的语言不一样, 设计思想和开发环境全部一样, 因此只要掌握其中一个, 就可以满足开发要求了。

VB.NET 是从 Visual Basic 发展而来的, Visual Basic 曾经具有开发领域世界第一的程序员数量, 因此非常多的 C/S 应用采用 VB.NET 开发环境。

1.1.2 B/S 结构编程体系

B/S 结构编程语言分成浏览器端编程语言和服务端编程语言。浏览器端包括 HTML (Hypertext Markup Language, 超文本标记语言)、CSS(Cascading Style Sheets, 层叠样式表单)、JavaScript 语言和 VBScript 语言。

所谓浏览器端编程语言就是这些语言都是被浏览器解释执行的。HTML 和 CSS 是由浏览器解释的, JavaScript 语言和 VBScript 语言也是在浏览器上执行的。

为了实现一些复杂的操作, 比如连接数据库、操作文件等, 需要使用服务器端编程语言。目前主要是 3P(ASP.NET、JSP 和 PHP)技术。ASP.NET 是微软公司推出的, 在这三种语言中是用得最为广泛的一种。JSP 是 SUN 公司推出的 J2EE(Java 2 Enterprise Edition, Java2 企业版)核心技术中重要的一种。PHP 在 1999 年的下半年和 2000 年用得非常广泛, 因为 Linux+PHP+MySQL(一种中小型数据库管理系统)构成全免费的而且非常稳定的应用平台, 这三种语言是目前应用开发体系的主流。

数据库支持是必须的, 目前应用领域的数据库系统全部采用关系型数据库(Relation Database Management System, RDBMS)。在企业级的开发领域中, 主要采用三大厂商的关系数据库系统: 微软公司的 SQL Server、Oracle 公司的 Oracle 和 IBM 公司的 DB2。

1.2 企业应用开发架构

在构建企业级应用时, 通常需要大量的代码, 而且这些代码一般分布在不同的计算机上, 划分代码运行在不同计算机上的理论就是多层设计理论。

企业级应用系统通常分为两层、三层和 N 层架构。

1.2.1 两层架构

传统的两层应用包括用户接口和后台程序, 后台程序通常是一个数据库, 用户接口直接同数据库进行对话。实现上, 通常使用 JSP、ASP 或者 VB 等技术编写这类软件, 结构如图 1-2 所示。

两层应用架构显示逻辑层一般由 HTML、JSP、ASP 实现, 通过 JSP 和 ASP 直接和数据库相连。

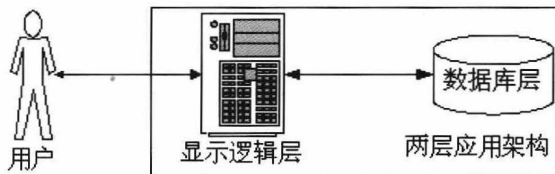


图 1-2 两层应用架构

1.2.2 三层架构

在两层应用中, 应用程序直接同数据库进行对话。三层结构在用户接口代码和数据库中间加入了一个附加的逻辑层, 通常这个层叫做“商务逻辑层”, 如图 1-3 所示。

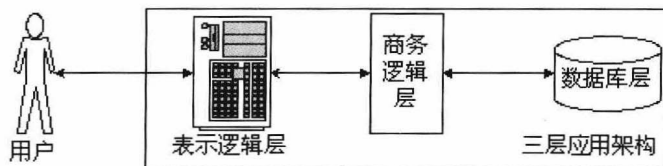


图 1-3 三层应用架构

1.2.3 N 层架构

如果某个应用超过 3 个独立的代码层，那么这个应用叫做 N 层应用，就不再叫四层或者五层等名称，而是统称为 N 层，如图 1-4 所示。

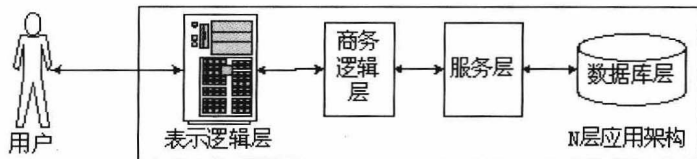


图 1-4 N 层应用架构

1.2.4 开发架构比较

两层架构的优点是开发过程比较简单，利用服务器端的程序直接访问数据库，部署起来比较方便。缺点是程序代码维护起来比较困难，程序执行的效率比较低，用户容量比较少。

三层架构基本解决了两层架构的缺点，将显示部分和逻辑流程控制分开，利用服务器应用程序实现显示部分，利用商务逻辑层实现程序的流程控制，分层使维护变的方便一些，而且执行效率也会有所提高，但是相对部署起来就比较困难一些。

根据实际需要，会进一步细化每一层，或者添加一些层，就形成了 N 层架构，和三层架构一样，组件化的设计使维护相对容易，但是部署相对困难。

1.3 JSP 概述

JSP(Java Server Pages)是由 Sun Microsystems 公司倡导的、许多公司参与一起建立的一种动态网页技术标准。JSP 技术是用 Java 语言作为脚本语言，JSP 网页为整个服务器端的 Java 库单元提供了一个接口来服务于 HTTP 的应用程序。

在传统的网页 HTML 文件(*.htm, *.html)中加入 Java 程序片段(Scriptlet)和 JSP 标记(Tag)，就构成了 JSP 网页(*.jsp)。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。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。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，可以实现无 Plugin，无 ActiveX，无 Java Applet，甚至无 Frame。

1.3.1 什么是 JSP

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。从构成情况上来看,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,所以它具有了 Web 和 Java 功能的双重特性。

JSP 1.0 规范是 1999 年 9 月推出的,12 月又推出了 1.1 规范。此后 JSP 又经历了几个版本,最新版本是 2003 年发布的 JSP 2.0。本书介绍的技术都基于 JSP 2.0 规范。

为了让读者对 JSP 技术有一个直观的认识,先来看一个非常简单的 JSP 页面及其运行效果。以下是 helloWorld.jsp 的源代码。

```
<%@ page language="java" contentType="text/html; charset=gbk"%>
<html>
  <head>
    <title>Hello World!</title>
  </head>
  <body bgcolor="#FFFFFF">
    <h3>
      <%
        out.println("JSP Hello World!");
      %>
    </h3>
  </body>
</html>
```

JSP 是一种动态网页技术标准。可以将网页中的动态部分和静态的 HTML 相分离。用户可以使用平常得心应手的工具并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。

程序运行效果如图 1-5 所示。

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,由于 JSP 程序同 Java 语言有着天然的联系,所以在众多基于 Web 的架构中,都可以看到 JSP 程序。

由于 JSP 程序增强了 Web 页面程序的独立性、兼容性和可重用性,所以,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 虚拟机事先解析成一个 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,如图 1-6 所示,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。由于 JSP 是基于 Java 语言和 HTML 元素的一项技术,所以只要熟悉 Java 和 HTML 的程序员都可以开发 JSP。
- 跨平台。由于 JSP 运行在 Java 虚拟机之上,所以它可以借助于 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。



图 1-5 helloWorld.jsp 运行效果

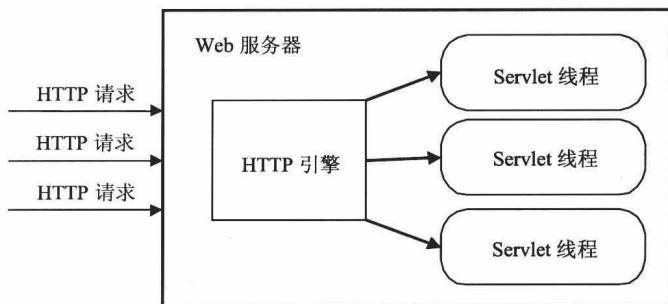


图 1-6 Web 服务器使用 Servlet 提供服务的示意图

- 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。

1.3.2 JSP 技术原理

JSP 文件的执行方式是“编译式”，而不是“解释式”，即在执行 JSP 页面时，是把 JSP 文件先翻译为 Servlet 形式的 Java 类型的字节码文件，然后通过 Java 虚拟机来运行。所以从本质上来讲，运行 JSP 文件最终还是要通过 Java 虚拟机，不过根据 JSP 技术的相关规范，JSP 语言必须在某个构建于 Java 虚拟机之上的特殊环境中运行，这个特殊环境就是 Servlet Container(通常被译为 Servlet 容器)，而且，每个 JSP 页面在被系统调用之前，必须先被 Servlet 容器解析成一个 Servlet 文件。

图 1-7 显示了整个 JSP 的运行流程。每次 Servlet 容器接受到一个 JSP 请求时，都会遵循以下步骤：

(1) Servlet 容器查询所需要加载的 JSP 文件是否已经被解析成 Servlet 文件，如果没有在 Servlet 容器里找到对应的 Servlet 文件，容器将根据 JSP 文件新建一个 Servlet 文件。反之，如果在容器里有此 Servlet 文件，容器则比较两者的时间，如果 JSP 文件的时间要晚于 Servlet 文件，则说明此 JSP 文件已被重新修改过，需要容器重新生成 Servlet 文件，反之容器将使用原先的 Servlet 文件。

(2) 容器编译好的 Servlet 文件被加载到 Servlet 容器中，执行定义在该 JSP 文件里的各项操作。

(3) Servlet 容器生成响应结果，并返回给客户端。

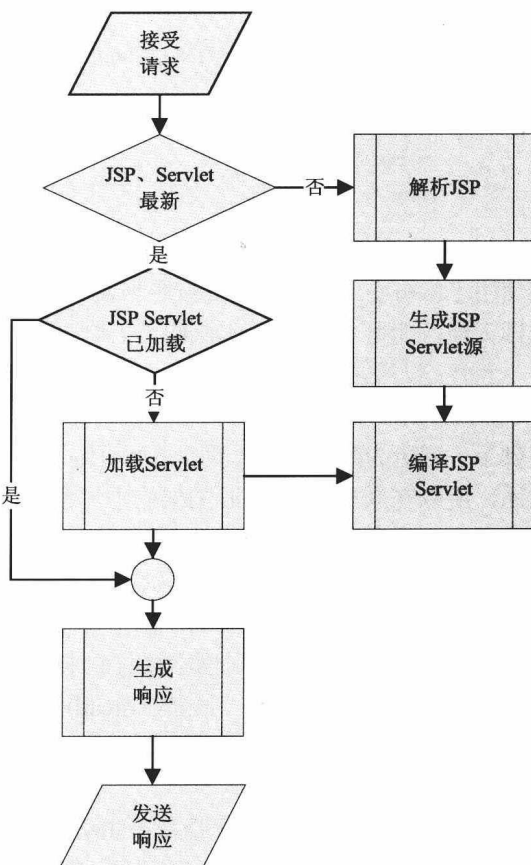


图 1-7 JSP 运行原理