

软件工程师培养丛书

# JSP动态网站开发

武汉厚溥教育科技有限公司 编著



“理论→总结→上机→习题”四阶段教学模式

- ★ 理论结合实践，注重动手能力培养
- ★ 任务驱动讲解，有效激发学习兴趣
- ★ 典型项目案例，扎实培养专业素质
- ★ 教学做一体化，极大提高教学效率

清华大学出版社

软件工程师培养丛书

# JSP 动态网站开发

武汉厚溥教育科技有限公司 编著

清华大学出版社

北 京

## 内 容 简 介

本书按照高等学校、高职高专计算机课程基本要求,以案例驱动的形式来组织内容,突出计算机课程的实践性特点。本书共包括 14 章:JavaWeb 简介、Servlet 入门、JSP 简介、JSP 隐式对象、JavaBean 和 JSP 动作、会话管理和使用、会话和 Servlet 综合应用、表达式语言、JSP 标准标签库、过滤器、MVC 模式、MVC 介绍与 Struts 1 框架、Struts 控制器组件、Struts 视图组件。

本书内容安排合理,层次清楚,通俗易懂,实例丰富,突出理论和实践的结合,可作为各类高等学校、高职高专及培训机构的教材,也可供广大程序设计人员参考。

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 动态网站开发 / 武汉厚溥教育科技有限公司 编著. —北京:清华大学出版社, 2014  
(软件工程师培养丛书)

ISBN 978-7-302-36472-6

I. ①J… II. ①武… III. ①JAVA 语言—网页制作工具 IV. ①TP312②TP393.092

中国版本图书馆 CIP 数据核字(2014)第 099298 号

责任编辑:刘金喜 蔡 娟

封面设计:崔东方

版式设计:妙思品位

责任校对:成凤进

责任印制:

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 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](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课 件 下 载: <http://www.tup.com.cn>, 010-62794504

印 刷 者:

装 订 者:

经 销:全国新华书店

开 本:185mm×260mm

印 张:25.5 字 数:589千字

版 次:2014年7月第1版

印 次:2014年7月第1次印刷

印 数:1~3000

定 价:39.80元

---

产品编号:

# 编委会

## 主任:

翁高飞

## 副主任:

王 鹏 余晓刚 刘 伟 曹 静  
方风波 邹治伟 李建利 管胜波

## 委员:

罗 炜 谢日星 吴金秀 胡智方  
夏超群 陈 琴 夏 晶 彭 莉  
徐 霞 明素华 王 敏 严 滔

# 前 言

JSP 是由 Sun Microsystems 公司倡导、许多公司参与一起建立的一种动态技术标准。在传统的网页 HTML 文件(\*.htm, \*.html)中加入 Java 程序片段(Scriptlet)和 JSP 标签, 就构成了 JSP 网页。Java 程序片段可以操纵数据库、重新定向网页以及发送 E-mail 等, 实现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, 这样大大降低了对客户端浏览器的要求, 即使客户端浏览器不支持 Java, 也可以访问 JSP 网页。

JSP 全名为 Java Server Pages, 其根本是一个简化的 Servlet 设计, 它实现了 HTML 语法中的 Java 扩展(以<%, %>形式)。JSP 与 Servlet 一样, 是在服务器端执行的。通常返回给客户端的就是一个 HTML 文本, 因此客户端只要有浏览器就能浏览。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段, 然后将执行结果连同 JSP 文件中的 HTML 代码一起返回给客户端。插入的 Java 程序段可以操作数据库、重新定向网页等, 以实现建立动态网页所需要的功能。

本书是“软件工程师培养丛书”中的一本专业教材, 该丛书是由武汉厚溥教育科技有限公司开发, 以培养符合企业需求的软件工程师应用开发、实施为目标的 IT 职业教育丛书。在开发该丛书之前, 我们对 IT 行业的岗位序列做了充分的调研, 包括研究从业人员技术方向、项目经验和职业素质等方面的需求, 通过对所面向的学生的特点、行业需求的现状以及实施等方面的详细分析, 结合“厚溥”对软件人才培养模式的认知, 按照软件专业总体定位要求, 进行软件专业产品课程体系设计。该丛书集应用软件知识和多领域的实践项目于一体, 着重培养学生的熟练度和规范性、集成和项目能力, 从而达到预定的培养目标。

本书共包括 14 章: JavaWeb 简介、Servlet 入门、JSP 简介、JSP 隐式对象、JavaBean



和 JSP 动作、会话管理和使用、会话和 Servlet 综合应用、表达式语言、JSP 标准标签库、过滤器、MVC 模式、MVC 介绍与 Struts1 框架、Struts 控制器组件、Struts 视图组件。

我们对本书的编写体系做了精心的设计，按照“理论学习—知识总结—上机操作—课后习题”这一思路进行编排。“理论学习”部分描述了通过本案例要达到的学习目的与涉及的相关知识点，使学习目标更加明确；“知识总结”部分概括了案例所涉及的知识点，使知识点完整系统地呈现；“上机操作”部分对案例进行了详尽分析，通过完整的步骤帮助读者快速掌握该案例的操作方法；“课后习题”部分帮助读者理解章节的知识点。本书在内容编写方面，力求细致全面；在文字叙述方面，注意言简意赅、重点突出；在案例选取方面，强调案例的针对性和实用性。

本书凝聚了编者多年来的教学经验和成果，可作为各类高等学校、高职高专及培训机构的教材，也可供广大程序设计人员参考。

本书 PPT 教学课件可通过 <http://www.tupwk.com.cn/downpage> 下载。

本书由武汉厚溥教育科技有限公司编著，由翁高飞、余晓刚等多名企业实战项目经理编写。本书编者长期从事项目开发和教学实施，并且对当前高校的教学情况非常熟悉，在编写过程中充分考虑到不同学生的特点和需求，加强了项目实战方面的教学。本书编写过程中，得到了武汉厚溥教育科技有限公司各级领导的大力支持，在此向他们表示衷心的感谢。

参与本书编写的人员还有：武汉商学院曹静、荆州职业技术学院方风波、武汉工程职业技术学院邹治伟、湖北三峡职业技术学院李建利和严滔、武汉软件工程职业学院罗炜、黄冈职业技术学院夏晶、湖北国土资源职业学院管胜波和王敏等。

限于编写时间和编者的水平，书中难免存在不足之处，希望广大读者批评指正。

服务邮箱：[wkservice@vip.163.com](mailto:wkservice@vip.163.com)。

编者  
2014 年 2 月

# 目 录

第 1 章 Java Web 简介.....	1	2.3.1 Servlet 初始化时期.....	19
1.1 Java 开发平台介绍.....	2	2.3.2 Servlet 响应客户请求时期.....	19
1.2 Java 开发模式.....	2	2.3.3 Servlet 结束时期.....	19
1.2.1 C/S 和 B/S 的概念.....	3	2.4 HTTP 协议和 HttpServlet.....	20
1.2.2 C/S 和 B/S 的区别.....	3	2.4.1 HTTP 请求和 HTTP 响应.....	20
1.3 三层结构.....	5	2.4.2 HttpServletRequest 和 HttpServletResponse 接口.....	21
1.4 Web 服务器简介.....	7	2.5 Servlet Web 应用开发.....	22
1.4.1 Web 服务器.....	7	2.5.1 使用 Eclipse 开发第一个 Servlet.....	22
1.4.2 Tomcat 简介.....	7	2.5.2 Servlet 应用实例.....	26
1.5 使用 Eclipse 开发第一个 Web 应用程序.....	9	【小结】.....	29
【小结】.....	11	【自测题】.....	29
【自测题】.....	11	上机部分.....	29
第 2 章 Servlet 入门.....	13	课后作业.....	43
2.1 Servlet API.....	14	第 3 章 JSP 简介.....	45
2.2 扩展 Servlet.....	15	3.1 JSP 请求模型.....	46
2.2.1 继承 GenericServlet.....	15	3.2 JSP 引擎.....	46
2.2.2 继承 HttpServlet.....	16	3.3 JSP 页面元素.....	47
2.2.3 ServletRequest 和 ServletResponse 接口.....	17	3.3.1 JSP 静态内容.....	47
2.3 Servlet 的生命周期.....	18		



3.3.2 JSP 注释	47	5.1.1 JavaBean 的概念	96
3.3.3 脚本元素	48	5.1.2 JavaBean 的属性、方法和事件	97
3.3.4 指令	54	5.1.3 JavaBean 中的 setter 和 getter 方法	98
3.3.5 JSP 动作	60	5.2 JSP 标准动作	99
【小结】	60	5.2.1 <jsp:useBean>动作	99
【自测题】	60	5.2.2 <jsp:getProperty>动作	100
上机部分	61	5.2.3 <jsp:setProperty>动作	101
课后作业	64	5.2.4 <jsp:forward>动作	107
<b>第 4 章 JSP 隐式对象</b>	<b>65</b>	5.2.5 <jsp:include>动作	109
4.1 输入/输出对象	66	【小结】	111
4.1.1 request 对象	66	【自测题】	111
4.1.2 response 对象	70	上机部分	112
4.1.3 out 对象	74	课后作业	122
4.2 作用域通信和控制对象	74	<b>第 6 章 会话管理和使用</b>	<b>123</b>
4.2.1 session 对象	74	6.1 HTTP 协议的无状态	124
4.2.2 application 对象	78	6.1.1 什么是会话	124
4.2.3 pageContext 对象	84	6.1.2 状态和会话管理	125
4.3 Servlet 相关对象	86	6.2 Servlet API 对会话的支持	125
4.3.1 page 对象	86	6.2.1 HttpSession 接口	125
4.3.2 config 对象	86	6.2.2 会话对象的创建	126
4.4 错误处理对象	88	6.2.3 会话管理	126
【小结】	89	6.3 会话跟踪	129
【自测题】	90	6.3.1 使用 Session 的会话跟踪	129
上机部分	90	6.3.2 使用 Cookie	133
课后作业	94		
<b>第 5 章 JavaBean 和 JSP 动作</b>	<b>95</b>		
5.1 JavaBean	96		



6.3.3 URL 重写.....	137	8.6 点号记法与数组记法	
6.3.4 隐藏表单域.....	141	的等同性.....	198
【小结】.....	141	8.7 EL 隐式对象.....	198
【自测题】.....	142	8.8 EL 中的运算符.....	200
上机部分.....	142	【小结】.....	201
课后作业.....	155	【自测题】.....	201
第 7 章 会话和 Servlet 综合应用.....	157	上机部分.....	202
7.1 使用 Session 实现 Servlet		课后作业.....	206
之间的通信.....	158	第 9 章 JSP 标准标签库.....	207
7.1.1 验证用户是否登录.....	158	9.1 基本概念.....	208
7.1.2 Servlet 间的数据共享.....	161	9.2 在 Web 应用中使用 JSTL.....	209
7.2 使用 ServletContext 实现		9.3 核心标签库.....	209
Servlet 之间的通信.....	165	9.3.1 通用标签.....	210
7.3 Servlet 控制器的作用.....	170	9.3.2 条件标签.....	213
7.3.1 RequestDispatcher 接口.....	170	9.3.3 迭代标签.....	214
7.3.2 使用 sendRedirect()方法.....	174	9.4 国际化和格式化标签库.....	219
【小结】.....	176	9.4.1 支持国际化的 Java 类.....	220
【自测题】.....	176	9.4.2 国际化标签.....	221
上机部分.....	177	9.4.3 支持格式化的 Java 类.....	225
课后作业.....	186	9.4.4 格式化标签.....	227
第 8 章 表达式语言.....	187	【小结】.....	231
8.1 应用表达式语言的动力.....	188	【自测题】.....	232
8.2 JSP 页面使用 EL.....	189	上机部分.....	232
8.3 JSP 页面禁用 EL.....	190	课后作业.....	238
8.4 访问作用域变量.....	191	第 10 章 过滤器.....	239
8.5 访问 JavaBean 的属性.....	194	10.1 Servlet 过滤器的机制	
		和特点.....	240



10.2 过滤器的生命周期.....	240	12.4 Web 框架的诞生.....	305
10.3 过滤器的 API.....	241	12.5 Apache 组织和 Struts 框架介绍.....	307
10.4 实现过滤器.....	241	12.6 Struts 1 下载与 Struts 1 安装目录.....	308
10.4.1 创建 Servlet 过滤器.....	242	12.7 Hello World 程序.....	309
10.4.2 部署 Servlet 过滤器.....	243	12.7.1 开发步骤.....	309
10.4.3 测试 Servlet 过滤器.....	244	12.7.2 分析 web.xml.....	316
10.5 使用 Servlet 过滤器链.....	246	【小结】.....	318
【小结】.....	251	【自测题】.....	318
【自测题】.....	251	上机部分.....	318
上机部分.....	252	课后作业.....	324
课后作业.....	265		
<b>第 11 章 MVC 模式.....</b>	<b>266</b>	<b>第 13 章 Struts 控制器组件.....</b>	<b>325</b>
11.1 MVC 模式在 Web 中 的应用.....	267	13.1 Struts 框架的控制器组件.....	326
11.2 JSP Model 1 体系结构.....	268	13.2 控制器组件的控制机制.....	326
11.3 JSP Model 2 体系结构.....	269	13.2.1 ActionServlet 类和 其初始化过程.....	326
11.3.1 实现 MVC 框架.....	270	13.2.2 Action 类.....	329
11.3.2 使用 MVC 框架.....	277	13.2.3 ActionForward 类.....	331
【小结】.....	285	13.2.4 制作简单登录实例.....	334
【自测题】.....	286	13.3 使用内置的 Struts Action 类.....	340
上机部分.....	286	13.3.1 DispatchAction 类.....	340
课后作业.....	299	13.3.2 MappingDispatch Action 类.....	349
<b>第 12 章 MVC 介绍与 Struts 1 框架.....</b>	<b>301</b>	【小结】.....	353
12.1 什么是 Web 框架.....	302		
12.2 模型 1.....	303		
12.3 模型 2 与 MVC.....	303		



【自测题】 .....	353	14.3.1 配置动态 ActionForm .....	374
上机部分 .....	354	14.3.2 访问动态 ActionForm 属性 .....	375
课后作业 .....	358	14.3.3 动态 ActionForm 的 reset() 方法 .....	376
<b>第 14 章 Struts 视图组件</b> .....	<b>359</b>	14.3.4 基于动态 ActionForm 的登录 .....	376
14.1 视图概述 .....	360	<b>14.4 Struts 常用的标签</b> .....	<b>378</b>
14.2 ActionForm Bean .....	361	14.4.1 标签的公共属性 .....	378
14.2.1 DTO 数据传输对象 .....	361	14.4.2 HTML 标签 .....	378
14.2.2 Struts 框架提供的 DTO: ActionForm Bean .....	362	14.4.3 Bean 标签 .....	381
14.2.3 ActionForm 的配置 与访问 .....	364	14.4.4 Logic 标签 .....	384
14.2.4 ActionForm 中的方法 .....	365	<b>【小结】</b> .....	<b>387</b>
14.2.5 ActionForm 的生命周期 ..	367	<b>【自测题】</b> .....	<b>387</b>
14.2.6 基于 ActionForm 的 登录案例 .....	368	上机部分 .....	388
14.3 动态 ActionForm 的使用 .....	374	课后作业 .....	394

# 第1章

# Java Web简介



## 课程目标

- ▶ 理解 Web 应用程序的概念
- ▶ 理解 Web 服务器的作用
- ▶ 掌握 Java Web 应用程序的目录结构



## 简介

前面课程中我们已经学习了面向对象的 Java 语法、JDBC 等相关技术。实际上, Java 语言更多的应用领域是在 Java Web 开发中。本章通过介绍基于 Java 语言的三大开发平台、C/S 和 B/S 两种开发模式、三层体系结构等概念, 让大家对 Java Web 技术有一定的了解, 最后学习 Tomcat 服务器的使用。

### 1.1 Java 开发平台介绍

随着 Java 技术的不断发展, 针对不同市场需要, 在 1999 年, Sun 公司把 Java 技术细分为 3 个不同的开发平台, 分别是针对普通 PC 应用的 J2SE(Java 2 Standard Edition)、针对嵌入式设备及消费类电器的 J2ME(Java 2 Micro Edition)、针对企业应用的 J2EE(Java 2 Enterprise Edition)。

J2SE 包含那些构成 Java 语言核心的类, 主要用于桌面应用软件的开发。J2ME 主要应用于手机、PDA、智能冰箱、智能空调、机顶盒等硬件设备, 后面我们有相关的课程将深入学习。在前面我们学习的 Java 技术属于 J2SE, 从本章开始, 将进入真正的企业级平台 J2EE 的学习, IO、JDBC 等技术将在后面的学习中继续使用。

2004 年 9 月, Sun 公司发布 J2SE 1.5, 这是 Java 语言的发展史上的又一里程碑事件, 为了表示这个版本的重要性, J2SE 1.5 更名为 J2SE 5.0。2005 年 6 月, JavaOne 大会召开, Sun 公司发布 Java SE 6。此时, Sun 公司为 Java 的各种版本更名并取消其中的数字“2”。J2EE 更名为 Java EE, J2SE 更名为 Java SE, J2ME 更名为 Java ME。

本书和以后的 Java 课程中, 我们将深入地学习 Java EE。Java EE 分为组件和服务两个部分, 其中组件包括 Servlet、JSP、EJB; 服务包括 JNDI、RMI-IIOP、JMS、JavaMAIL、Java Connector、JIDL、JAAS、JTA 等。

到底什么是 Java Web 呢? 其实就是前面提到的 Servlet 和 JSP 以及相关技术。Servlet 是服务器端的 Java 程序, JSP(Java Server Page)是服务器端的页面。如前所述, Servlet 和 JSP 是大部分开发人员必须掌握的, 主要是针对基于 Java 语言的 Web 开发, 所以我们将接下来的学习中, 专门来学习这两项技术。

### 1.2 Java 开发模式

前面学习过 Java SE 技术, 可以开发桌面应用程序, 也学习了 HTML 网页技术和 Java



Script 技术，可以开发网页，这些技术和我们将要接触到的 Java Web 有关系吗？在了解这些之前，首先让我们了解一下不同的开发模式。

### 1.2.1 C/S 和 B/S 的概念

在目前的企业级开发中，常用的开发模式有两种：C/S 和 B/S。

C/S，即 Client / Server。虽然从定义上，CS 模式是指任何将事务处理分开进行的网络系统，但绝大多数的 CS 应用系统都是 CS 数据库系统，充当服务器的是大型数据库系统，如 SQL Server、Oracle、MySql 等。

比如说使用 Java Swing 开发的数据库管理系统，Java 程序安装在各个不同的客户端，数据库则只存在于一台服务器主机上。多个用户在各自的主机上，使用该程序操作同一个数据库，如图 1-1 所示。

B/S，即 Browser/Server。BS 模式是指软件的主体部分都在服务器端，用户只需使用浏览器发出请求，由服务器执行对请求动作的运算，并将最终运算结果发回到客户端。

比如我们经常上的网站、论坛，客户机上只需要有浏览器，不需要安装任何客户端程序，所有的程序都在 Web 服务器上。用户通过浏览器向服务器发送请求，浏览器根据不同用户发出的不同请求，执行不同的程序，将计算出的结果显示到各个用户的浏览器上。在此要说明一下，服务器包括数据库服务器和 Web 服务器，用户请求的是 Web 服务器，通过 Web 服务器上的程序再去对数据库进行操作，如图 1-2 所示。

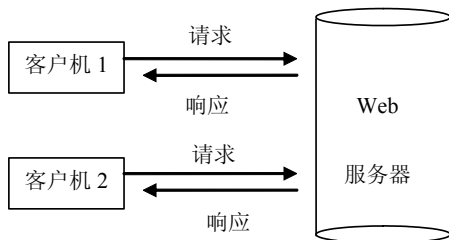


图 1-1

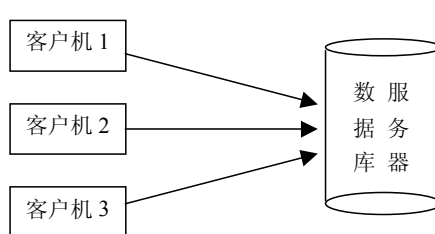


图 1-2

### 1.2.2 C/S 和 B/S 的区别

我们将从以下几个方面分别阐述这两种开发模式的区别。

#### 1. 数据安全性

由于 C/S 结构软件的数据分布特性，客户端所发生的火灾、盗抢、地震、病毒、黑客等都成了可怕的数据杀手。另外，对于集团级的异地软件应用，C/S 结构的软件必须在各地安装多个服务器，并在多个服务器之间进行数据同步。如此一来，每个数据点上的数



据安全都影响了整个应用的数据安全。所以，对于集团级的大型应用来讲，C/S 结构软件的安全性是令人无法接受的。

对于 B/S 结构的软件来讲，由于其数据集中存放于总部的数据库服务器，客户端不保存任何业务数据和数据库连接信息，也无需进行什么数据同步，所以这些安全问题也就自然不存在了。

### 2. 数据一致性

在 C/S 结构软件的解决方案里，对于异地经营的大型集团都采用各地安装区域级服务器，然后再进行数据同步的模式。这些服务器每天必须同步完毕之后，总部才可得到最终的数据。由于局部网络故障造成个别数据库不能同步不说，即使同步上来，各服务器也不是一个时点上的数据，数据永远无法一致，不能用于决策。

对于 B/S 结构的软件来讲，其数据是集中存放的，客户端发生的每一笔业务单据都直接进入中央数据库，不存在数据一致性的问题。

### 3. 数据实时性

在集团级应用里，C/S 结构不可能随时随地看到当前业务的发生情况，看到的都是事后数据。B/S 结构则不同，它可以实时看到当前发生的所有业务，方便了快速决策，有效地避免了企业损失。

### 4. 可维护性

企业的业务流程、业务模式不是一成不变的，随着企业不断发展，必然会不断调整。软件供应商提供的软件也不是完美无缺的，所以，对已经部署的软件产品进行维护、升级是正常的。

C/S 结构软件，由于其应用是分布的，需要对每一个客户端进行程序安装，所以，即使非常小的程序缺陷都需要很长的重新部署时间，重新部署时，为了保证各程序版本的一致性，必须暂停一切业务进行更新。

B/S 结构的软件不同，其应用都集中于总部服务器上，各应用节点并没有任何程序，一个地方更新则全部应用程序更新，可以做到快速服务响应。

### 5. 响应速度

C/S 结构中，由于客户端实现与数据库服务器的直接相连，所以响应速度非常快。

B/S 结构中，Web 应用程序动态刷新，所以响应速度明显变慢。



## 6. 界面设计

C/S 结构中，客户操作界面个性化，具有简单、直观、方便的特点，可以根据客户要求订制，充分满足客户个性化需求。

B/S 结构中，个性化特点明显降低，以鼠标为基本操作方式，无法实现快速操作的要求。

## 7. 服务器负载

C/S 结构中，客户端和服务端都能处理任务，虽然对客户机要求较高，但是可以大大减少服务器的压力。

B/S 结构中，绝大部分工作由服务器承担，使得服务器负担很重。

以上是从几个方面对 C/S、B/S 结构进行比较，可以看出它们各有优缺点，不同的需求适合不同的开发模式。但是我们能看出，随着网络越来越普及，而 PDA、智能手机、智能家电等计算机以外的上网方式发展迅速，也加速了 B/S 模式的发展。

# 1.3 三层结构

三层结构，通常是指表示层、业务逻辑层、数据访问层。在这种体系结构中，层中的代码相互间保持独立，业务层充当数据访问层和表示层之间的接口，表示层不直接与数据访问层进行通信。一个设计良好的 Java Web 应用通常是基于三层体系结构的。表示层主要指用户界面，它要求尽可能的简单，使最终用户不需要进行任何培训就能方便地访问信息；业务层就是应用服务器，也就是常说的中间件，所有的应用系统、应用逻辑、控制都在这一层，系统的复杂性也主要体现在应用层；最后的数据库服务器存储大量的数据信息和数据逻辑，所有与数据有关的安全、完整性控制、数据的一致性、并发操作等都是在第三层完成的。三层体系结构相较传统的开发方式，有如下特点。

### 1. 能有效降低建设和维护成本，简化管理

多层应用结构在各层次上的组件能单独更新、替换或增加、拆除。因此，系统维护更方便，代价相对较低。而且，因各组件互相独立，更换组件就好比更换组合音响的一个部件，对系统其他部分并无影响，所以更新维护更加安全可靠。

客户端采用瘦客户机，因为客户机不必进行大量的计算或数据处理，它的硬件配置就不需要太高。

通过将业务逻辑集中到中间层，系统获得了对业务逻辑的独立性，即当用户的需求改变时，开发人员可以迅速地在中间层(应用服务器)上更新业务逻辑，而无需将更新后



的应用提交到众多的 PC 终端系统上去, 即客户端无需任何改动(改动众多的客户端并不是件轻松的事)。

### 2. 适应大规模和复杂的应用需求

如果说结构化方法使软件开发从一门手工艺术走向科学的工程方法, 组件技术则使软件工程从个体作坊走向大规模工业。虽然结构化方法对中小型系统开发能够行之有效, 但对大型系统, 结构化分析的结果往往是错综复杂的网状结构, 而不是结构清晰的层次结构。这也正是面向对象方法学诞生的原因。组件技术能使复杂系统的设计变得简单可行, 具有良好的伸缩性。

三层或多层结构, 可以将数据处理从客户端转移到应用服务器和数据库服务器上。这样, 尽管客户端与应用服务器之间可能存在着多个甚至数百个的连接, 但是应用服务器与数据库服务器之间的连接却只有少数几个, 从而达到减少通信线路上传递的数据量的目标。这样的功能分配提供了很强的系统可伸缩性, 使得在用户数量急剧增加时还能保持系统性能的稳定。使用传统的客户机/服务器模式根本无法胜任上千个客户机同时运行同时需要访问数据库的工作。即使在用户数量很大的情况下, 数据库仍能保持良好的工作负载, 保持系统快速的响应速度。

### 3. 可适应不断的变化和新的业务需求

任何应用系统实施的重点不在于需求确定以后能否实现这些需求, 而是在系统实施后如何适应变化的需求。Java EE 系统结构和组件式系统的开发和维护过程中, 技术人员可以按照新的需求, 通过在不同系统层次上调度更新的组件或新加入的组件来调整旧的系统, 以适应新的与不断变化的要求。以往的系统只能靠专业维护人员或系统开发商的再次开发或修改原有系统, 才能满足新的需求, 代价往往很大, 无法保证时间上的要求。

### 4. 访问异构数据库

多层结构的中间层即应用服务器能够提供广泛的异构数据库访问和复制能力。传统的客户机/服务器结构则需要在客户端安装许多访问异构数据库的驱动程序, 而三层/多层结构只要在中间层有相应的驱动程序就可以访问异构数据源。

### 5. 能有效提高系统安全性

多层体系结构将数据与程序、数据控制与应用逻辑分层独立管理, 能更严格地控制信息访问; 信息传递中采用数据加密技术, 可进一步降低信息失密的风险。应用服务器内建安全控制数据库, 实现应用服务器与数据服务器的双重权限控制, 对权限的划分更准确、灵活、严格。新系统在信息访问、传递和存储 3 个环节上均有严格的安全措施。



## 1.4 Web 服务器

前面我们反复提到一个重要概念——Web 服务器，那到底什么是 Web 服务器呢？它到底是如何工作呢？既然是服务器，都有哪些种类呢？如何使用呢？

### 1.4.1 Web 服务器简介

Web 服务器，即在网络中为实现信息发布、资料查询、数据处理等诸多应用而搭建基本平台的服务器。有时，我们也常常称 Web 服务器为 Web 容器。常用的服务器有 Tomcat、Resin、Weblogic、Websphere 等。在今后的学习中，我们将使用 Tomcat 服务器。

Web 服务器工作原理：在 Web 应用中的处理可分为三个步骤，第一步，Web 浏览器向一个特定的 Web 服务器发出 Web 页面请求；第二步，Web 服务器接收到 Web 页面请求后，寻找所请求的 Web 页面，执行相应的功能；第三步，将用户请求的最后结果以 Web 页面形式发送到客户的 Web 浏览器，原理如图 1-3 所示。

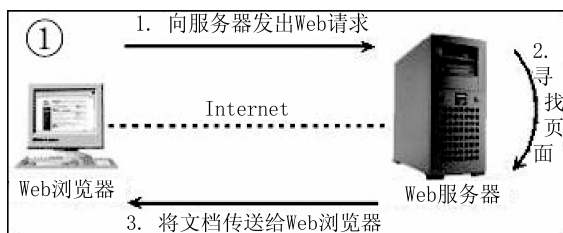


图 1-3

### 1.4.2 Tomcat 简介

Tomcat 是一个免费的开源的 Web 服务器，提供对 Servlet 和 JSP 的支持。它是 Apache 基金会的 Jakarta 项目中的一个核心项目，由 Apache、Sun 和其他一些公司及个人共同开发。由于有了 Sun 的参与和支持，最新的 Servlet 和 JSP 规范总能在 Tomcat 中得到体现。本书介绍 Tomcat 6.0，支持 Servlet 2.5，JSP 2.1，下载网站是 <http://tomcat.apache.org/>。Tomcat 服务器非常适合学习 Java Web 开发技术的初学者。

#### 1. Tomcat 安装

在安装 Tomcat 之前，必须首先安装 JDK，因为 Tomcat 本身是纯 Java 程序，需要 JVM 才能运行，此外 JSP 页面需要 javac 来编译运行，因此必须安装 JDK。本书中使用的 JDK 版本是 1.6.0\_06。



## 2. Tomcat 目录结构

Apache 的官方网站提供基于 Windows 的安装版本，很容易安装。成功安装 Tomcat 后，会产生如图 1-4 所示的目录和文件。

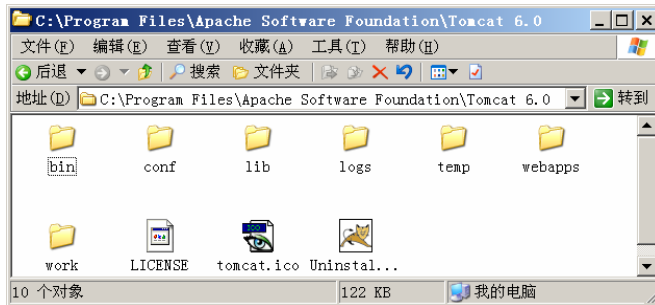


图 1-4

表 1-1 列出了每个目录的作用。

表 1-1

目 录	作 用
bin	存放启动 Tomcat 的可执行文件
conf	存放 Tomcat 服务器的各种配置文件，最重要的配置文件是 server.xml
lib	存放服务器需要的各种 jar 文件
logs	存放服务器的日志文件
temp	存放临时文件
webapps	存放发布 Web 应用的目录
work	Tomcat 把 JSP 生成的 Servlet 放在该目录下

这里我们需要先了解 webapps 目录，这个目录是用来部署所有 Web 应用程序的。

## 3. Tomcat 服务的启动和关闭

将编写好的 Web 应用程序放到 webapps 目录中，启动 Tomcat 服务器，用户就可以通过浏览器访问该程序了。如何启动 Tomcat 呢？首先进入 bin 目录，双击 tomcat6w.exe 文件，会看到 Tomcat 的启动界面，如图 1-5 所示。或者从“程序”中选择“Apache Tomcat 6.0”，再选择“Configure Tomcat”，一样会出现图 1-5 所示的启动界面。此时，单击“Start”按钮，就启动了服务，若单击“Stop”按钮，就停止了服务。

服务启动后，打开浏览器，在地址栏输入 <http://127.0.0.1:8080>，将显示如图 1-6 所示的界面，表明服务器成功启动。

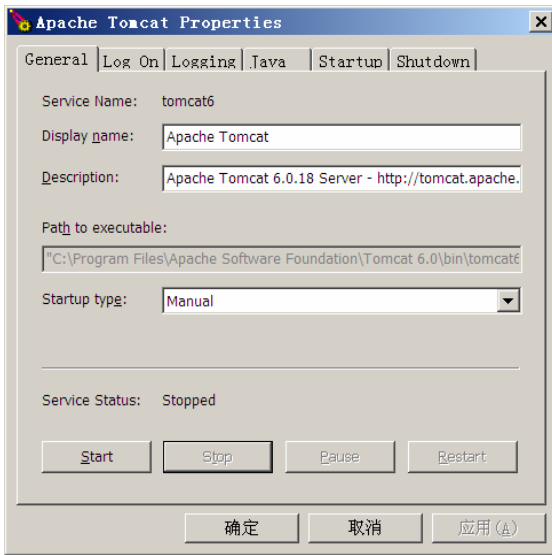


图 1-5

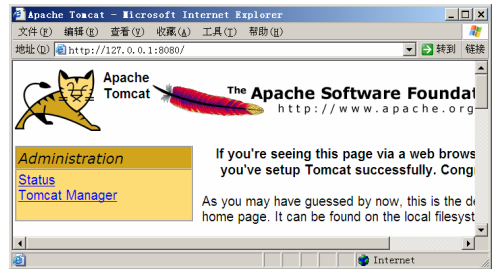


图 1-6

## 1.5 使用 Eclipse 开发第一个 Web 应用程序

Java Web 的开发，最好的集成开发工具是 Eclipse+MyEclipse，本书使用 Eclipse 3.3 版本和 MyEclipse 6.5 版本。下面我们来开发第一个 Web 应用程序。启动 Eclipse，依次在菜单上单击 File→New→Web Project，进入 New Web Project 的界面，如图 1-7 所示。

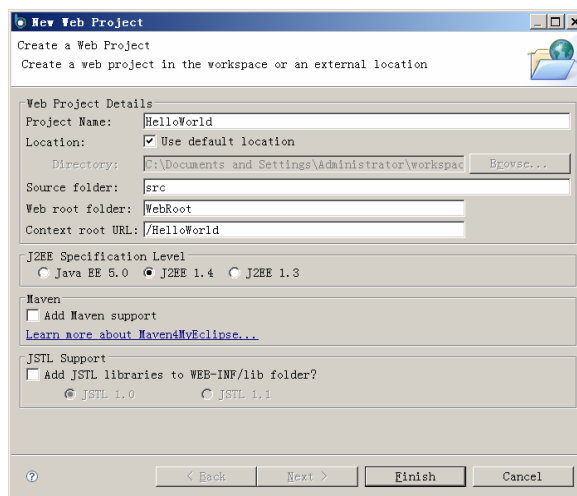


图 1-7

输入项目名称，选择 J2EE 的版本 Java EE 5.0，单击“Finish”按钮，界面如图 1-8 所示。

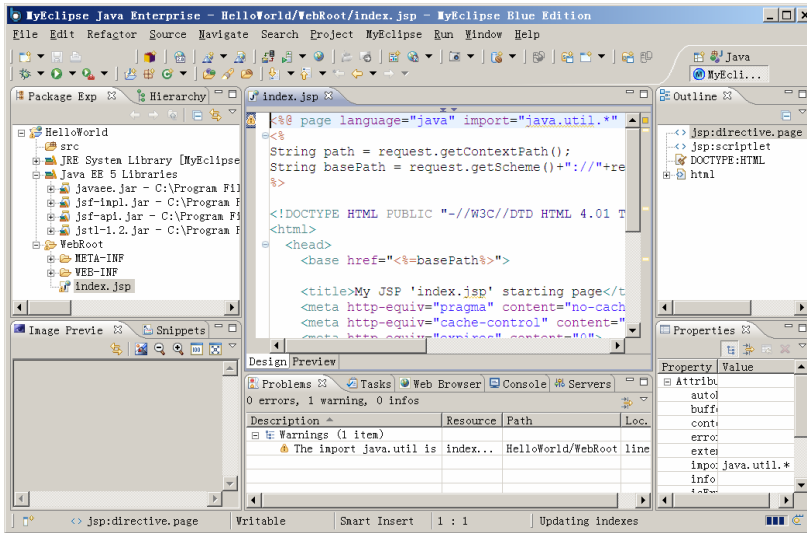



图 1-8

可以看到开发环境已经为我们生成好了目录结构，并生成了一个叫 `index.jsp` 的页面文件。修改 `index.jsp` 页面，去掉自动生成的代码，修改后的代码如示例 1-1 所示。

### 示例 1-1:

```
<%@ page language="java" import="java.util.*" pageEncoding="GBK"%>
<html>
  <head>
    <title>第一个 Web 应用程序</title>
  </head>
  <body>
    欢迎进入 Java Web 世界! <br>
  </body>
</html>
```

接下来，部署该 Web 应用，单击工具栏中的  按钮，显示如图 1-9 所示的界面。

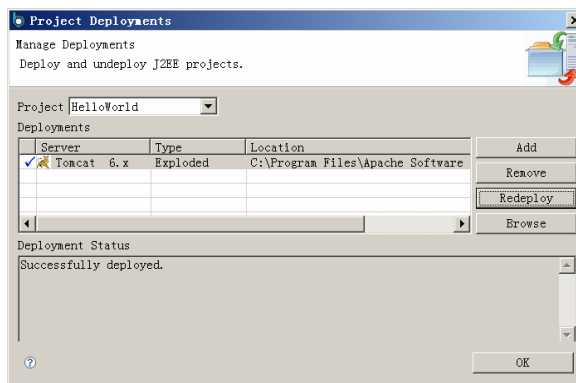



图 1-9



选择要部署的 Web 应用和要部署到的 Web 服务器，图中使用 Tomcat 6.x 服务器，完成部署后会显示“Successfully deployed”信息，表明 Web 应用成功部署到服务器。接下来，在开发环境中启动 Tomcat 服务器，在工具栏上单击  按钮，选择要启动的服务器。注意，需要在 MyEclipse 环境中配置该服务器，也可以按照上面讲的方式独立启动服务器。打开 IE 浏览器，输入“http://127.0.0.1:8080/HelloWorld”，显示结果如图 1-10 所示。

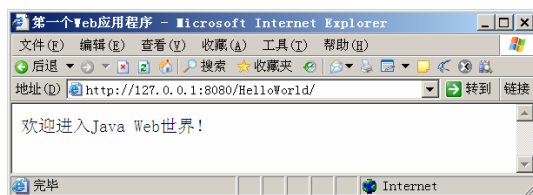


图 1-10

让我们来看看部署在 webapps 目录下的第一个 Web 程序的目录结构，如表 1-2 所示。

表 1-2

目 录	作 用
/HelloWorld	Web 应用的根目录，所有 JSP 和 HTML 文件存放于此目录下
/HelloWorld/WEB-INF	存放 Web 应用的发布描述文件 web.xml
/HelloWorld/WEB-INF/classes	存放各种编译好的 class 文件
/HelloWorld/WEB-INF/lib	存放 Web 应用需要的各种 jar 文件

## 【 小结 】

- Java 的三大开发平台
- C/S、B/S 的区别
- 了解三层体系结构的概念
- Web 服务器简介

## 【 自测题 】

1. 以下哪些属于 Java 的开发平台？(     )
 

A. Java SE	B. Java EE	C. JDK	D. MyEclipse
------------	------------	--------	--------------
2. 以下哪些是 Web 服务器？(     )
 

A. MyEclipse	B. Tomcat	C. Weblogic	D. Eclipse
--------------	-----------	-------------	------------



3. 三层体系结构, 包括以下哪些项? ( )  
A. 数据访问层    B. 数据库服务器    C. 业务逻辑层    D. 表示层
4. 启动 Tomcat 的命令是( )。  
A. shutdown.bat    B. javac.exe    C. startup.bat    D. java.exe
5. 以下关于 B/S 和 C/S, 哪些说法是不正确的? ( )  
A. 在数据安全性方面, C/S 比 B/S 更有优势  
B. 在数据一致性方面, B/S 比 C/S 更有优势  
C. B/S 比 C/S 程序更加容易维护