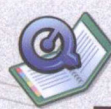


 高等教育规划教材

# Java Web 程序设计

夏辉 李晋 毕婧 白萍 编著



提供电子教案、源代码  
下载网址 <http://www.cmpedu.com>



机械工业出版社  
CHINA MACHINE PRESS

高等教育规划教材

# Java Web 程序设计

夏辉 李晋 毕婧 白萍 编著



机械工业出版社

本书以 Web 开发流程为主线, 让读者了解真实的 Web 开发过程, 体会规范化、标准化、专业化的软件开发流程和管理规范。本书通过大量的案例让学生在体验中掌握 Java Web 的相关知识, 同时获得编程能力、排错能力和学习能力。本书在章节安排上循序渐进, 几乎覆盖了 Web 开发的所有基本知识点, 还涉及 Web 容器、HTTP 协议和 JDBC 开发等与 Web 开发密切相关的知识点, 帮助读者深刻理解所学知识, 掌握实现 Web 编程的不同技术特点。

本书详细介绍了 Java Web 程序设计的前端技术、开发和部署, 以及一些衍生技术变化。在内容上, 本着通过使用不同技术尽可能实现相同功能的原则, 让读者能够充分体会并认识每种技术的优缺点。每章均配有实践与练习, 以指导读者深入地进行学习。

本书既可作为高等学校计算机软件技术课程的教材, 也可作为管理信息系统开发人员的技术参考书。

本书配套授课电子课件, 需要的教师可登录 [www.cmpedu.com](http://www.cmpedu.com) 免费注册, 审核通过后下载, 或联系编辑索取 (QQ: 2850823885, 电话: 010 - 88379739)。

## 图书在版编目 (CIP) 数据

Java Web 程序设计/夏辉等编著. —北京: 机械工业出版社, 2016. 9

高等教育规划教材

ISBN 978-7-111-55559-9

I. ①J… II. ①夏… III. ①JAVA 语言 - 程序设计 - 高等学校 - 教材  
IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2016) 第 294825 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 郝建伟

责任校对: 张艳霞 责任印制: 李 洋

北京京师印务有限公司印刷

2017 年 1 月第 1 版 · 第 1 次印刷

184mm × 260mm · 20.25 印张 · 493 千字

0001 - 3000 册

标准书号: ISBN 978-7-111-55559-9

定价: 49.90 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

电话服务

服务咨询热线: (010) 88379833

读者购书热线: (010) 88379649

封面无防伪标均为盗版

网络服务

机工官网: [www.cmpbook.com](http://www.cmpbook.com)

机工官博: [weibo.com/cmp1952](http://weibo.com/cmp1952)

教育服务网: [www.cmpedu.com](http://www.cmpedu.com)

金书网: [www.golden-book.com](http://www.golden-book.com)

# 出版说明

当前，我国正处在加快转变经济发展方式、推动产业转型升级的关键时期。为经济转型升级提供高层次人才，是高等院校最重要的历史使命和战略任务之一。高等教育要培养基础性、学术型人才，但更重要的是加大力度培养多规格、多样化的应用型、复合型人才。

为顺应高等教育迅猛发展的趋势，配合高等院校的教学改革，满足高质量高校教材的迫切需求，机械工业出版社邀请了全国多所高等院校的专家、一线教师及教务部门，通过充分的调研和讨论，针对相关课程的特点，总结教学中的实践经验，组织出版了这套“高等教育规划教材”。

本套教材具有以下特点：

1) 符合高等院校各专业人才的培养目标及课程体系的设置，注重培养学生的应用能力，加大案例篇幅或实训内容，强调知识、能力与素质的综合训练。

2) 针对多数学生的学习特点，采用通俗易懂的方法讲解知识，逻辑性强、层次分明、叙述准确而精炼、图文并茂，使学生可以快速掌握，学以致用。

3) 凝结一线骨干教师的课程改革和教学研究成果，融合先进的教学理念，在教学内容和方法上做出创新。

4) 为了体现建设“立体化”精品教材的宗旨，本套教材为主干课程配备了电子教案、学习与上机指导、习题解答、源代码或源程序、教学大纲、课程设计和毕业设计指导等资源。

5) 注重教材的实用性、通用性，适合各类高等院校、高等职业学校及相关院校的教学，也可作为各类培训班教材和自学用书。

欢迎教育界的专家和老师提出宝贵的意见和建议。衷心感谢广大教育工作者和读者的支持与帮助！

机械工业出版社

# 前 言

随着因特网的迅猛发展,以及“互联网+”横空出世,Java Web 开发语言已经成为全球最流行、使用最广泛的开发语言之一。大多数企业的系统开发、网站开发和 OA 开发等 B/S 系统都采用 Java Web 开发技术,由于 Java 语言的简单性、可执行性、稳定与安全性,以及多线程性等优良特性,使得它成为基于因特网应用技术和 Web 开发的首选编程语言。学习和掌握这样一种技术语言已经成为计算机相关专业学生的迫切需求。

本书以 Java Web 开发环境为背景,主要介绍 Web 开发的前端技术和服务器技术,在内容的编排上力争体现新的教学思想和方法。本书内容编写遵循“从简单到复杂”“从抽象到具体”的原则。书中通过各个章节所穿插的很多示例,介绍 Java Web 开发技术及 Web 开发的基本步骤和基本方法。程序设计既是一门理论课,也是一门实践课。学生除了要在课堂上学习程序设计的理论方法,掌握编程语言的语法知识和编程技巧外,还要进行大量的课外练习和实践操作。为此本书各章都配有练习题,并且每章都有一个综合案例。除此之外,每章还安排了实验题目,以供教师实验教学使用。

本书共分 10 章。第 1 章介绍 Java Web 开发概述,主要介绍 Web 开发的基本概念和开发环境安装。第 2 章和第 3 章介绍 Java Web 前端技术,分别介绍 Web 开发的前端静态和动态技术。第 4 章介绍 Tomcat 服务器,主要讲解服务器知识。第 5 章介绍 HTTP 协议知识,主要讲解 HTTP 请求的响应字段。第 6 章介绍 JSP 语法基础。第 7 章介绍 Servlet 技术。第 8 章介绍 JSP 会话技术。第 9 章介绍 JSP 内置对象和作用域。第 10 章介绍 JDBC 数据库技术。

在设计应用方面,以当前最流行的用户机/服务器应用程序前端开发工具 MyEclipse IDE 为背景,介绍应用程序开发的实施步骤。此外,还介绍了许多实际开发过程中所使用的编程技巧,通过这些技巧,可以大大提升应用程序的开发速度。

本书内容全面,包括 Web 开发的前端技术和服务器技术,案例新颖,针对性强。本书中所介绍的实例都是在 MyEclipse 6.6、Tomcat 7.0 和 JDK 1.7 环境下调试运行通过的。每一章都有与本章知识点相关的案例和实验,以帮助读者顺利完成开发任务。从应用程序的设计到应用程序的发布,读者都可以按照书中所讲述的内容进行实施。

本书由夏辉负责书的整体策划,夏辉、李晋、毕婧和白萍负责全书编写,并且最终完成书稿的修订、完善、统稿和定稿工作,参加调试和审稿工作的还有王晓薇教授、吴鹏教授等。刘杰教授和李航教授为本书的策划和编写提供了有益的帮助和支持,并且对本书初稿在教学过程中存在的问题提出了宝贵的意见。本书也借鉴了中外参考文献中的原理知识和资料,在此一并表示感谢。

本书配有电子课件、课后习题答案,以及各个章节的案例代码和实验代码,以方便教学和自学参考使用,如有需要,请到 <http://www.cmpedu.com> 网站中下载。

由于时间仓促,书中难免存在不妥之处,请读者谅解,并提出宝贵意见。

编 者

# 目 录

出版说明	
前言	
<b>第 1 章 Java Web 开发概述</b> .....	1
1.1 背景知识简介 .....	1
1.2 Web 开发基础知识 .....	1
1.2.1 Web 程序运行原理 .....	2
1.2.2 Web 服务器介绍 .....	2
1.2.3 Web 开发流程 .....	3
1.3 Java Web 开发平台的搭建 .....	4
1.3.1 Java 环境变量配置 .....	4
1.3.2 集成开发环境 MyEclipse 的下载和配置 .....	6
1.3.3 Tomcat 服务器配置 .....	8
1.3.4 部署项目 .....	9
1.3.5 创建第一个 Java Web 项目 .....	11
本章小结 .....	13
实践与练习 .....	14
实验指导 .....	14
<b>第 2 章 Java Web 开发前端静态技术</b> .....	16
2.1 Java Web 开发前端技术简介 .....	16
2.2 HTML 5 标签技术 .....	16
2.2.1 HTML 5 声明文件 .....	17
2.2.2 HTML 5 文本和段落 .....	18
2.2.3 HTML 5 链接 .....	20
2.2.4 HTML 5 图像 .....	22
2.2.5 HTML 5 表格和列表 .....	23
2.2.6 HTML 5 表单 .....	26
2.3 CSS 技术 .....	29
2.3.1 CSS 样式 .....	30
2.3.2 CSS 框模型 .....	35
2.3.3 CSS 定位 .....	36
2.3.4 CSS 选择器 .....	40
2.3.5 CSS 高级特性 .....	42
2.4 HTML 5/CSS 设计页面案例 .....	43
2.5 案例: DIV + CSS 精美窗口设计 .....	45
本章小结 .....	49
实践与练习 .....	49
实验指导 .....	50
<b>第 3 章 Java Web 开发前端动态技术</b> .....	53
3.1 JavaScript 基本语法 .....	53
3.1.1 JavaScript 的输出 .....	53
3.1.2 JavaScript 的注释 .....	55
3.1.3 JavaScript 标识符与关键字 .....	55
3.1.4 JavaScript 基本数据类型 .....	56
3.1.5 JavaScript 变量 .....	60
3.1.6 JavaScript 运算符 .....	64
3.2 JavaScript 中的流程控制 .....	69
3.2.1 分支结构 .....	69
3.2.2 循环结构 .....	72
3.2.3 控制循环结构 .....	74
3.3 JavaScript 函数 .....	75
3.3.1 函数定义 .....	76
3.3.2 函数调用 .....	76
3.3.3 函数的参数 .....	77
3.4 JavaScript 对象 .....	78
3.4.1 对象编程基础 .....	78
3.4.2 String 对象 .....	84
3.4.3 Boolean 对象 .....	86
3.4.4 Math 对象 .....	87
3.4.5 Array 对象 .....	88
3.5 案例: 用户注册信息的验证 .....	89
本章小结 .....	92
实践与练习 .....	92
实验指导 .....	93
<b>第 4 章 Tomcat 服务器</b> .....	97

4.1 Tomcat 服务器简介 .....	97	6.3.1 JSP 模板元素 .....	167
4.2 部署 Tomcat 服务器 .....	100	6.3.2 JSP 表达式 .....	167
4.2.1 Tomcat 服务器的安装与启动 .....	101	6.3.3 JSP 脚本片段 .....	168
4.2.2 Tomcat 服务器安装目录介绍 .....	107	6.3.4 JSP 声明 .....	169
4.2.3 在 MyEclipse 中配置 Tomcat 服务器 .....	108	6.3.5 JSP 注释 .....	170
4.2.4 Tomcat 服务器异常纠错 .....	114	6.4 JSP 动作标签 .....	171
4.3 在 Tomcat 服务器中发布 Web 应用 .....	117	6.4.1 JSP 动作标签概述 .....	171
4.3.1 Web 应用的组成结构 .....	117	6.4.2 <jsp:include> 标签 .....	171
4.3.2 配置 Web 应用虚拟目录和默认 页面 .....	118	6.4.3 <jsp:forward> 标签 .....	173
4.3.3 发布 Web 应用 .....	122	6.4.4 <jsp:param> 标签 .....	175
4.4 配置虚拟主机 .....	126	6.5 案例：简单的用户登录功能 .....	175
4.5 案例：创建一个 Java Web 网站 .....	127	本章小结 .....	179
本章小结 .....	130	实践与练习 .....	179
实践与练习 .....	130	实验指导 .....	180
实验指导 .....	131	<b>第 7 章 Servlet 技术</b> .....	183
<b>第 5 章 HTTP 协议</b> .....	135	7.1 Servlet 概述 .....	183
5.1 HTTP 协议概述 .....	135	7.2 Servlet 接口 .....	184
5.1.1 HTTP 协议简介 .....	135	7.2.1 GenericServlet 和 HttpServlet .....	185
5.1.2 HTTPWatch 工具简介 .....	135	7.2.2 请求消息和响应消息 .....	186
5.2 HTTP 请求 .....	140	7.3 Servlet 生命周期 .....	187
5.2.1 HTTP 请求信息 .....	142	7.4 创建 Servlet .....	188
5.2.2 HTTP 请求常见消息头 .....	143	7.4.1 Servlet 的创建 .....	188
5.3 HTTP 响应 .....	148	7.4.2 Servlet 的配置 .....	191
5.3.1 HTTP 响应消息 .....	149	7.4.3 ServletConfig 接口 .....	192
5.3.2 HTTP 常见响应消息头 .....	151	7.4.4 ServletContext 接口 .....	193
5.4 HTTP 通用头字段 .....	154	7.5 Servlet 请求 HttpServletRequest 接口 .....	195
本章小结 .....	156	7.5.1 获取请求消息 .....	195
实践与练习 .....	156	7.5.2 获取表单数据 .....	197
实验指导 .....	157	7.5.3 请求转发 .....	200
<b>第 6 章 JSP 语法基础</b> .....	161	7.6 Servlet 响应 HttpServletResponse 接口 .....	204
6.1 JSP 页面概述 .....	161	7.6.1 设置状态码 .....	204
6.2 JSP 指令 .....	162	7.6.2 设置 HTTP 协议响应头 .....	204
6.2.1 page 指令 .....	163	7.6.3 请求重定向 .....	205
6.2.2 include 指令 .....	165	7.7 Servlet 会话 .....	205
6.3 JSP 语法 .....	166	7.7.1 Servlet 会话跟踪技术 .....	205
		7.7.2 HttpSession 接口 .....	206
		7.8 Servlet 监听器 .....	209

7.8.1 Servlet 上下文监听 .....	209	9.3 四大作用域 .....	271
7.8.2 HTTP 会话监听 .....	210	9.3.1 page 范围 .....	271
7.8.3 Servlet 请求监听 .....	210	9.3.2 request 范围 .....	272
7.8.4 在 Web 应用中使用事件 监听器 .....	211	9.3.3 session 范围 .....	273
7.9 案例：在多个请求页面中购书 信息的获取 .....	212	9.3.4 application 范围 .....	274
本章小结 .....	216	9.4 案例：带验证码的用户登录 模块 .....	275
实践与练习 .....	216	本章小结 .....	278
实验指导 .....	217	实践与练习 .....	278
<b>第 8 章 JSP 会话技术</b> .....	<b>220</b>	实验指导 .....	279
8.1 会话概述 .....	220	<b>第 10 章 JDBC 数据库技术</b> .....	<b>282</b>
8.2 Cookie 对象 .....	221	10.1 JDBC 概述 .....	282
8.2.1 Cookie 原理 .....	222	10.2 JDBC 常用 API .....	285
8.2.2 Cookie 对象的常用方法 .....	225	10.2.1 Driver 类 .....	285
8.3 Session 对象 .....	232	10.2.2 DriverManager 类 .....	286
8.3.1 Session 原理 .....	234	10.2.3 Connection 类 .....	287
8.3.2 Session 对象的常用方法 .....	237	10.2.4 Statement 类 .....	289
8.4 案例：用户订单管理 .....	242	10.2.5 PreparedStatement 类 .....	291
本章小结 .....	249	10.2.6 CallableStatement 接口 .....	293
实践与练习 .....	249	10.2.7 ResultSet 接口 .....	294
实验指导 .....	251	10.3 数据库连接过程 .....	295
<b>第 9 章 JSP 内置对象和作用域</b> .....	<b>256</b>	10.4 JDBC 对象数据库操作 .....	298
9.1 内置对象和作用域概述 .....	256	10.4.1 增加数据 .....	298
9.2 内置对象 .....	257	10.4.2 修改数据 .....	298
9.2.1 out 对象 .....	258	10.4.3 删除数据 .....	299
9.2.2 request 对象 .....	259	10.4.4 查询数据 .....	299
9.2.3 response 对象 .....	262	10.4.5 批处理 .....	300
9.2.4 session 对象 .....	265	10.4.6 存储过程调用 .....	303
9.2.5 application 对象 .....	267	10.5 案例：分页查询大型数据库 .....	305
9.2.6 exception 对象 .....	268	本章小结 .....	309
9.2.7 pageContext 对象 .....	268	实践与练习 .....	309
9.2.8 page 对象 .....	270	实验指导 .....	311
9.2.9 config 对象 .....	270	<b>参考文献</b> .....	<b>316</b>



# 第 1 章 Java Web 开发概述

Java Web 是用 Java 技术来解决相关 Web 领域的技术总和。本章首先介绍了 Java 语言的历史和现状，然后对 Web 开发的基础知识进行了简单介绍，最后介绍了 Java Web 开发环境的搭建过程，为后面的学习做好准备。

## 1.1 背景知识简介

Java 是 Sun 公司（已被甲骨文公司收购）于 1995 年正式推出的面向对象的程序设计语言，它具有简单、面向对象、跨平台、健壮和安全、可移植性、支持网络编程，以及支持多线程等许多优良特性，特别适用于 Internet 应用程序的开发。

1991 年，Sun 公司成立了由 James Gosling 领导的 Green Project 研发小组，其目的是研发一种分布式系统结构，使其能够在各种家用消费电子产品上运行。最终，Green 项目组开发出了一种新的编程语言——Oak（Java 的前身）。1994 年，随着 Internet 的快速发展，Sun 公司看到了 Oak 在计算机网络上的广阔应用前景，于是对 Oak 做了进一步的改进，1995 年 5 月，Oak 以 Java 的名称正式发布。目前，Java 已成为网络编程的首选语言。

Sun 公司于 1996 年 1 月 23 日发布了 JDK 1.0。1998 年 12 月 4 日，Sun 发布了 Java 历史上最重要的一个 JDK 版本——JDK 1.2。这个版本标志着 Java 已经进入 Java 2 时代，这个时期也是 Java 飞速发展的时期。目前 Java 2 平台有 3 个版本。

### 1. Java SE

Java SE（Java Platform, Standard Edition）称为 Java 平台标准版，主要用于开发 Java 桌面应用程序和低端的服务器应用程序，也可以开发 Java Applet 程序。

### 2. Java EE

Java EE（Java Platform, Enterprise Edition）称为 Java 平台企业版，主要用于构建企业级的服务应用。Java EE 是在 Java SE 的基础上构建的，增加了附加类库，可以支持目录管理、交易管理和企业级消息处理等功能。

### 3. Java ME

Java ME（Java Platform, Micro Edition）称为 Java 平台微型版，主要用于嵌入式的消费产品中，如移动电话、掌上电脑或其他无线设备等。

## 1.2 Web 开发基础知识

随着网络技术的不断发展，各种网络程序开发体系结构应运而生，其中基于客户端/服务器的 C/S 结构和基于浏览器/服务器的 B/S 结构成为两大主流。Java Web 开发是基于 B/S 结构的 Java 应用程序开发。

## 1.2.1 Web 程序运行原理

在 B/S 结构中，用户通过浏览器向 Web 服务器端发送请求，由 Web 服务器处理用户的请求并将处理结果返回客户端。Web 程序的工作流程如图 1-1 所示，具体操作步骤如下。

1) 用户通过浏览器向 Web 服务器发送请求（步骤 a），Web 服务器接收到该请求后，根据请求的文件名在 Web 服务器中查找对应的文件（步骤 b），并根据文件的扩展名（如 .html 或 .jsp）来判断当前所请求的是静态页面还是动态页面。

2) 当请求页面为静态页面时，则直接将静态页面返回（步骤 c），并将该页面作为响应发送给客户端浏览器（步骤 e）。

3) 当请求页面为动态页面时，Web 服务器首先执行页面中的代码部分；如果动态页面中存在数据库操作代码，Web 服务器还要负责完成与数据库服务器的信息交互（步骤 d），之后再讲动态页面解释为静态页面（步骤 c）；最后 Web 服务器将该静态页面作为响应发送给客户端浏览器（步骤 e）。

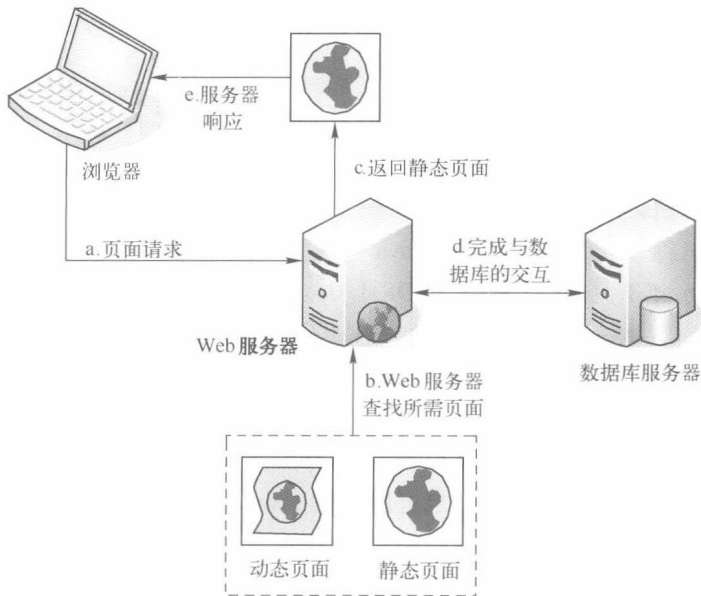


图 1-1 Web 程序的工作流程

## 1.2.2 Web 服务器介绍

市场上有很多种 Web 服务器，UNIX 和 Linux 平台下常用的 Web 服务器有 Apache 的 Tomcat、IBM 公司的 WebSpherer 和 BEA 公司的 WebLogic 等，而 Windows NT/2000/2003 平台下最常用的服务器则是微软公司的 IIS（Internet Information Server）。

下面介绍几种常用的 Web 服务器。

### 1. Apache Tomcat

Tomcat 是 Apache 软件基金会（Apache Software Foundation）的 Jakarta 项目中的一个核心项目，由 Apache、Sun 公司和其他一些公司及个人共同开发而成。由于有了 Sun 公司的参

与和支持,最新的 Servlet 和 JSP 规范总是能在 Tomcat 中得到体现。因为 Tomcat 技术先进,性能稳定,而且是免费的,因而深受 Java 爱好者的喜爱并得到了部分软件开发商的认可,成为目前比较流行的 Web 应用服务器。

## 2. IBM WebSphere

WebSphere 应用服务器是一种功能完善、开放的 Web 应用程序服务器,是 IBM 电子商务计划的核心部分,是基于 Java 和 Servlet 的 Web 应用程序运行环境,为 Web 站点提供了所需的一切服务,包括项目管理、连接数据库、Java Servlet 代码生成器、HTML 编辑器和网站发布等,为开发 Servlet 和 JavaBean 提供了多种向导。WebSphere 适应 Web 应用程序服务器的需要,范围从简单到高级直到企业级。

## 3. BEA WebLogic

WebLogic 最早由 WebLogic Inc. 开发,后并入 BEA 公司,最终 BEA 公司又并入甲骨文公司。WebLogic 是一个 Web 应用服务器,确切地说是一个基于 Java EE 架构的中间件,是用于开发、集成、部署与管理大型分布式 Web 应用、网络应用和数据库应用的 Java 应用服务器。由于 WebLogic 具有全面的功能、对开放标准的遵从性、多层架构,并支持基于组件的开发,基于 Internet 的企业都选择它来开发和部署最佳的应用。

## 4. Microsoft IIS

Microsoft 的 Web 服务器产品为 Internet Information Server (IIS),IIS 是允许在公共 Intranet 或 Internet 上发布信息的 Web 服务器。IIS 提供了一个图形界面的管理工具,称为 Internet 服务管理器,可用于监视配置和控制 Internet 服务。

IIS 是一种 Web 服务组件,其中包括 Web 服务器、FTP 服务器、NNTP 服务器和 SMTP 服务器,分别用于网页浏览、文件传输、新闻服务和邮件发送等方面,使得在网络(包括因特网和局域网)上发布信息成了一件很容易的事情。

### 1.2.3 Web 开发流程

在 Web 应用程序的开发中,有两种主流的开发体系结构:C/S 架构和 B/S 架构。

所谓 C/S 架构,又称 Client/Server 或客户端/服务器模式。服务器端通常采用高性能的 PC、工作站或小型机,并采用大型数据库系统,如 Oracle、Sybase、Informix 或 SQL Server;客户端则需要安装专用的客户端软件。在这种架构方式中,客户端应用程序负责用户业务逻辑的处理,服务器端仅对重要的过程和数据库进行处理和存储,如图 1-2 所示。

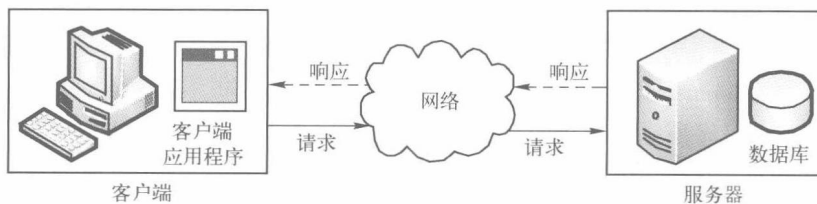


图 1-2 C/S 架构

所谓 B/S 架构,又称 Browser/Server 或浏览器/服务器模式。在这种架构方式中,客户端只需要有通用的浏览器软件,将开发的 Web 应用程序部署在 Web 服务器中即可,通过浏览器向 Web 服务器发送请求,就可以访问服务器端提供的所有服务,如图 1-3 所示。B/S

架构已经成为当今应用软件的首选体系结构。

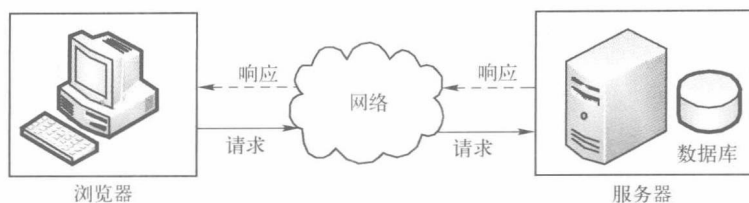


图 1-3 B/S 架构

无论采用哪种体系结构，Web 开发通常都遵循一个基本的流程：需求分析阶段、设计阶段、开发阶段、测试阶段、运行与维护阶段，如图 1-4 所示。

首先，通过对用户的调研，结合分析市场现状及发展趋势完成项目的需求分析。然后，根据需求分析结果进行系统的分析和设计，确定系统的规模和目标，规划开发方案，并完成数据库设计。在开发阶段中，由于 Web 标准是结构和表现相分离的，所以设计师和程序员可以并行开发，后期再完成整个网站的整合。在整个 Web 项目开发完成后，需要设计测试案例，测试项目功能的实现情况。最后，将整个 Web 项目部署到 Web 服务器上，运行并维护。

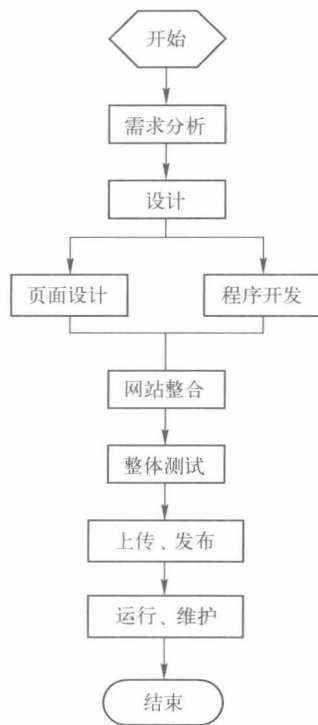


图 1-4 Web 开发流程

## 1.3 Java Web 开发平台的搭建

开发 Java Web 应用程序需要运行环境和开发工具，因此，搭建一个方便、快捷、高效的开发平台就显得尤为重要。目前，比较常用的开发平台和工具包括 JDK、Tomcat 和 MyEclipse 等。其中，JDK 是 Java 语言的开发环境，Tomcat 是 Web 服务器，MyEclipse 是开发 Java EE 的集成开发环境。

本书以 Windows 64 位操作系统为例，采用 JDK 1.7 + Tomcat 6.0 + MyEclipse 6.6 作为开发与运行环境。读者可根据自己系统的要求下载相应的版本。

### 1.3.1 Java 环境变量配置

JDK (Java Development Kit) 是 Java 语言的软件开发工具包，是整个 Java 的核心，它提供了编译和运行 Java 程序所需的各种工具和资源，包括 Java 编译器、Java 运行时环境 (Java Runtime Environment, JRE)，以及常用的 Java 类库等。由于 Java Web 使用 Java 语言作为开发语言，因此，要想开发与运行 Java Web 应用程序，就必须有 JDK 的开发环境。本书中采用的是 JDK 1.7 版本，读者可到甲骨文公司的官方网站进行下载，网址为 <http://www.oracle.com/technetwork/cn/java/javase/downloads/jdk7-downloads-1880260.html>。

#### 1. JDK 的安装

下载后的文件名称为 `jdk-7u79-windows-x64.exe`，双击该文件即可开始安装。具体

安装步骤如下。

1) 双击 `jdk-7u79-windows-x64.exe` 文件，在弹出的对话框中单击“下一步”按钮。

2) 在弹出的“定制安装”对话框中，设置JDK的安装路径并选择安装组件。单击“更改”按钮，将安装路径更改为 `D:\Java\jdk1.7.0_79`，其他采用默认设置，如图 1-5 所示，单击“下一步”按钮，开始安装。

3) 在安装的过程中，将会弹出设置JRE安装路径的对话框。单击“更改”按钮，将安装路径更改为 `D:\Java\jre7`，如图 1-6 所示，单击“下一步”按钮，继续安装JRE。

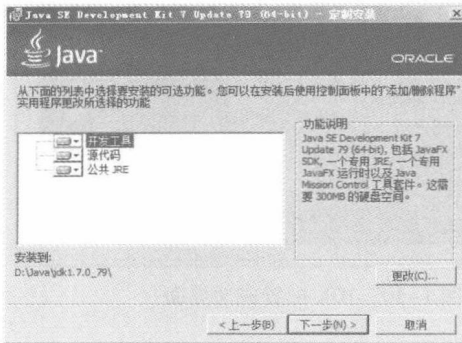


图 1-5 设置 JDK 的安装路径

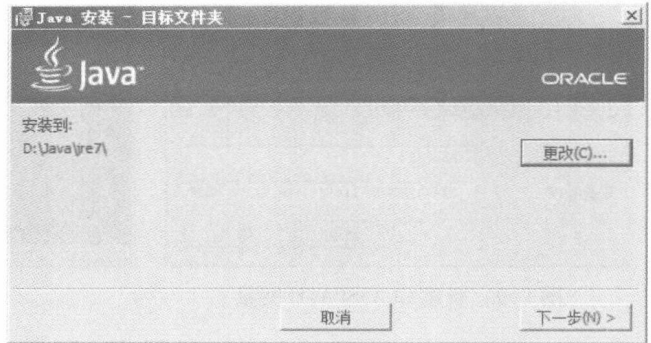


图 1-6 设置 JRE 的安装路径

4) 在弹出的安装完成对话框中，单击“关闭”按钮，即可完成JDK的安装。

## 2. JDK 的配置与测试

JDK 安装完成后，需要设置环境变量并测试JDK配置是否成功。具体操作步骤如下。

1) 在桌面上右击“计算机”图标，在弹出的快捷菜单中选择“属性”命令，弹出“系统”对话框，选择“高级系统设置”选择卡。在弹出的“系统属性”对话框中选择“高级”选项卡，然后单击“环境变量”按钮，弹出“环境变量”对话框。

2) 单击“系统变量”选项组中的“新建”按钮，弹出“新建系统变量”对话框。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:\Java\jdk1.7.0_79`，如图 1-7 所示。单击“确定”按钮，完成系统变量 `JAVA_HOME` 的创建。

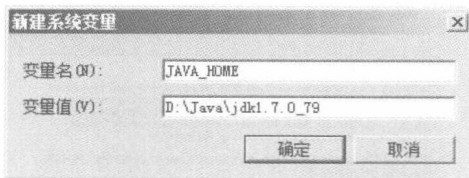


图 1-7 新建 `JAVA_HOME` 变量

3) 在系统变量列表框中查看是否存在 `PATH` 变量（注意：不区分大小写）。如果不存在，则新建系统变量 `PATH`；若存在，则选中该变量，单击“编辑”按钮，弹出“编辑系统变量”对话框。在“变量值”文本框中原有变量值的最前面增加变量值 `%JAVA_HOME%\bin;`，如图 1-8 所示。单击“确定”按钮，完成系统变量 `PATH` 的编辑。

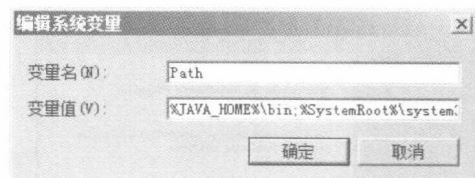


图 1-8 编辑 `PATH` 变量

4) 用同样的方法新建系统变量 `CLASSPATH`，设置变量值为 `.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar`，如图 1-9 所示。单击“确定”按钮，完成系统变量的

创建。

5) 环境变量设置完成后, 还要测试 JDK 是否配置成功。

单击“开始”按钮, 选择“运行”命令, 在弹出的“运行”对话框中输入 cmd 命令, 单击“确定”按钮, 进入 DOS 命令窗口。在命令提示符后输入 javac 命令, 按下 <Enter> 键, 如果系统输出 javac 命令的使用帮助信息, 如图 1-10 所示, 则表明 JDK 配置成功。

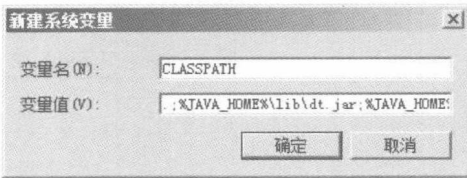


图 1-9 新建 CLASSPATH 变量



图 1-10 JDK 配置成功界面

### 1.3.2 集成开发环境 MyEclipse 的下载和配置

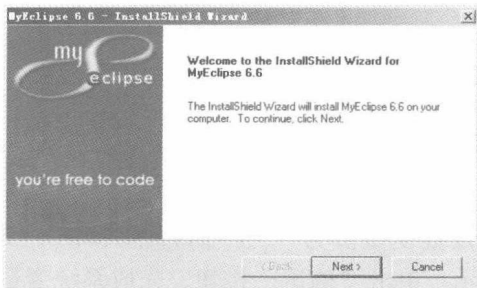
MyEclipse 是基于 Eclipse 开发的功能强大的企业级集成开发环境, 主要用于 Java、Java EE 和移动应用的开发, 支持代码编写、调试、测试及发布等功能, 提供了对 UML、Ajax、Web、Web Services、J2EE、JSP、XML、Struts、JSF、Java Persistence、EJB、扩展数据库, 以及各种应用服务器的集成支持。

#### 1. 下载 MyEclipse

MyEclipse 的官方网站地址是 <http://www.myeclipseide.com/>, 可根据需要购买或者下载试用版的 MyEclipse。本书使用的是 MyEclipse 6.6 版本。下载后的文件名称为 MyEclipse\_6.6.0\_E3.3.2\_Installer.exe, 双击该文件即可开始安装。具体安装步骤如下。

1) 双击 MyEclipse\_6.6.0\_E3.3.2\_Installer.exe 文件, 在弹出的对话框中单击 Next 按钮, 如图 1-11 所示。

2) 在弹出的“许可协议”对话框中, 选择 I accept the terms of the license agreement (接受许可协议) 单选按钮, 如图 1-12 所示。单击 Next 按钮, 进入选择安装路径对话框, 单击 Change 按钮, 将安装路径更改为 D:\MyEclipse 6.6, 如图 1-13 所示, 也可以使用默认路径。



6 图 1-11 MyEclipse 安装启动界面

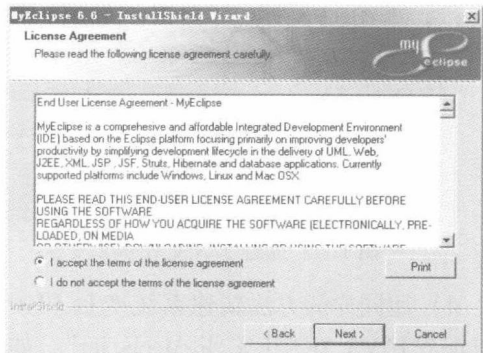


图 1-12 安装许可协议

3) 设置完安装路径后, 单击 Next 按钮, 将显示准备安装的提示界面, 单击 Install 按钮, 开始进行安装。安装完成后, 将弹出如图 1-14 所示的对话框, 提示安装已完成。单击 Finish 按钮, 即可完成 MyEclipse 的安装。

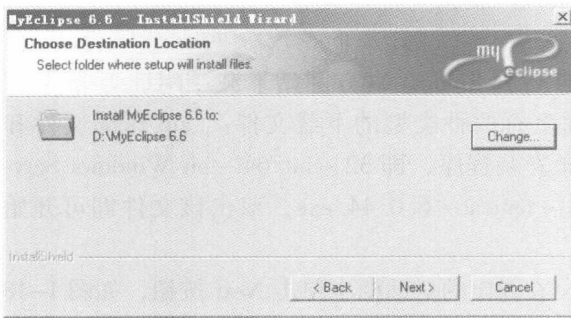


图 1-13 选择安装路径

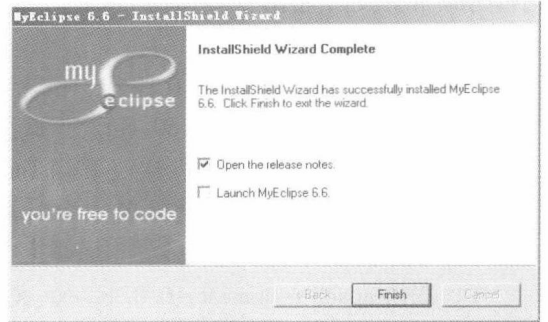


图 1-14 安装完成

## 2. 配置 MyEclipse

MyEclipse 安装完成后, 单击“开始”按钮, 选择“所有程序”→MyEclipse 6.6→MyEclipse 6.6 命令, 启动 MyEclipse。启动时, 首先会提示选择工作空间, 如图 1-15 所示。如果要改变工作空间默认值, 可以单击 Browse 按钮, 在弹出的对话框中选择路径。设置完成后, 单击 OK 按钮, 进入 MyEclipse 主界面。

如果是第一次启动 MyEclipse, 会显示欢迎界面, 如图 1-16 所示, 关闭 Welcome 页面将进入工作窗口, 如图 1-17 所示。

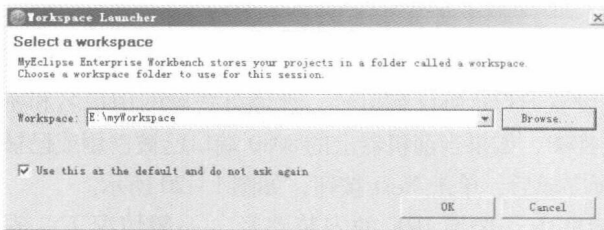


图 1-15 选择工作空间

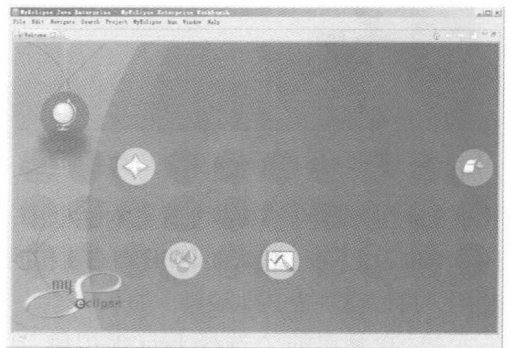


图 1-16 MyEclipse 主界面

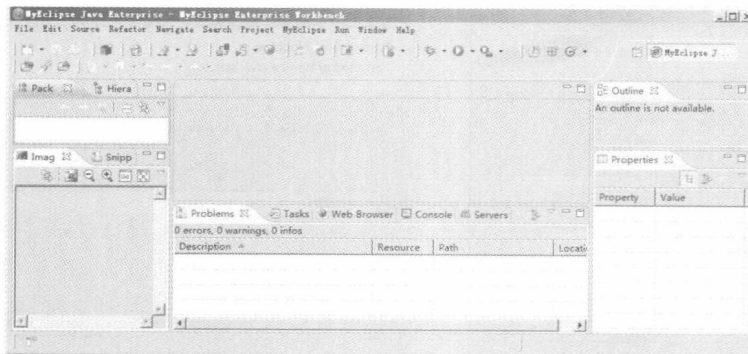


图 1-17 MyEclipse 工作窗口

### 1.3.3 Tomcat 服务器配置

Tomcat 服务器是一个免费开源的 Web 应用服务器，属于轻量级应用服务器，通常应用于中小型系统和并发访问用户不是很多的场合，是开发和调试 Java Web 应用程序的首选。

#### 1. 安装 Tomcat

本书使用的是 Tomcat 6.0 版本，读者可自行到 Tomcat 官方网站下载，网址为 <http://tomcat.apache.org/download-60.cgi>。下载页面上有两种类型的下载文件：.zip 压缩文件和 .exe 安装程序，对于初学者，建议下载 Tomcat 安装程序，即 32-bit/64-bit Windows Service Installer 文件。下载后的文件名称为 apache-tomcat-6.0.44.exe，双击该文件即可开始安装。具体安装步骤如下。

1) 双击 apache-tomcat-6.0.44.exe 文件，在弹出的对话框中单击 Next 按钮，如图 1-18 所示。

2) 在弹出的“许可协议”对话框中，单击 I Agree 按钮，弹出“选择组件”对话框，可选择安装的组件，如图 1-19 所示，这里保持默认设置，单击 Next 按钮。



图 1-18 Tomcat 安装启动界面

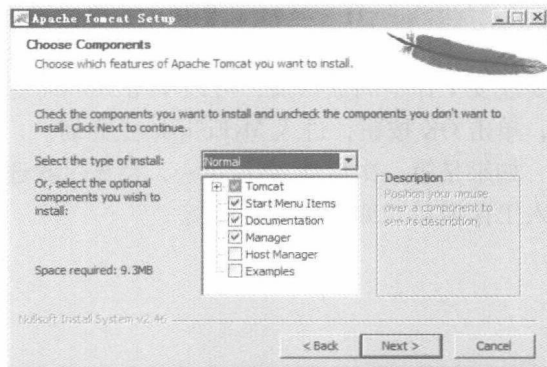


图 1-19 选择组件

3) 在弹出的“基本配置”对话框中，可设置 HTTP 协议端口号、管理员登录的用户名和密码。Tomcat 服务器默认的 HTTP/1.1 端口号是 8080，如果当前机器上的 8080 端口已被占用或已被禁用，可以设置为其他端口，如 9080 等。设置完成后，单击 Next 按钮，如图 1-20 所示。

4) 在弹出的“选择 Java 虚拟机”对话框中，选择 JRE 的安装路径，一般情况下，安装程序会自动找到 Java 虚拟机安装路径，如图 1-21 所示，设置完成后单击 Next 按钮。

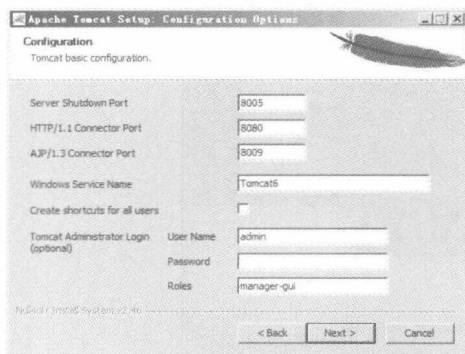


图 1-20 基本配置

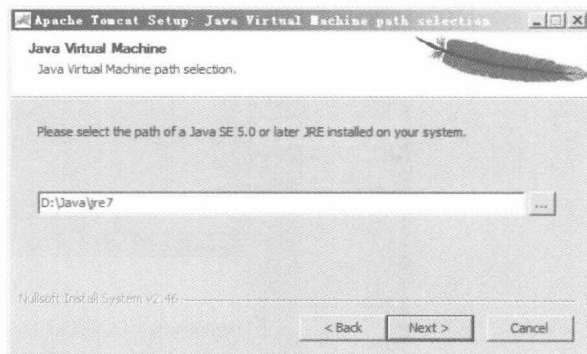


图 1-21 选择 Java 虚拟机



5) 这时将弹出“选择安装路径”对话框, 设置安装路径为 D:\Tomcat 6.0, 如图 1-22 所示, 然后单击 Install 按钮, 开始进行安装。安装完成后, 会弹出如图 1-23 所示的对话框, 单击 Finish 按钮, 完成 Tomcat 的安装。

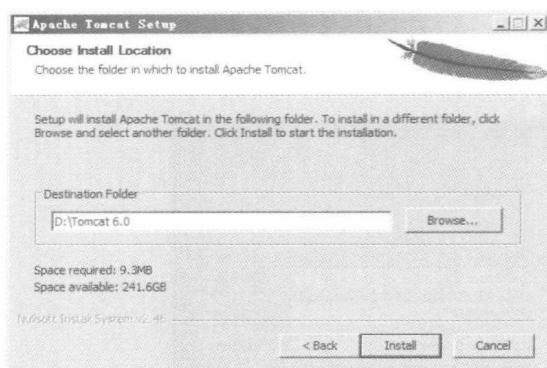


图 1-22 选择安装路径

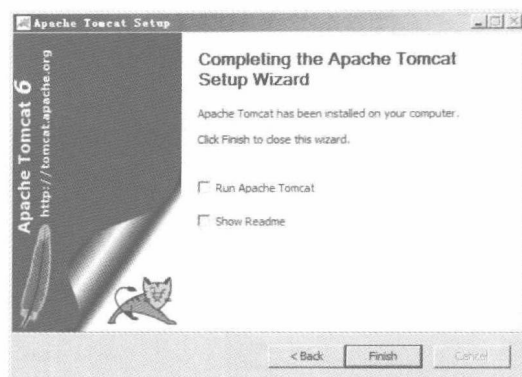


图 1-23 安装完成

## 2. 配置 Tomcat

Tomcat 安装完成后, 单击“开始”按钮, 选择“所有程序” → Apache Tomcat 6.0 → Configure Tomcat 命令, 将弹出 Tomcat 属性对话框, 如图 1-24 所示。单击 Start 按钮, 将启动 Tomcat 服务。单击 Stop 按钮, 将关闭 Tomcat 服务。如果需要自动启动 Tomcat, 可将 Startup type (启动类型) 设置为 Automatic。

启动 Tomcat 服务器后, 在浏览器地址栏中输入 <http://localhost:8080/> 或 <http://127.0.0.1:8080/>, 如果出现的是如图 1-25 所示的测试界面, 则表示 Tomcat 已安装成功并正常启动。

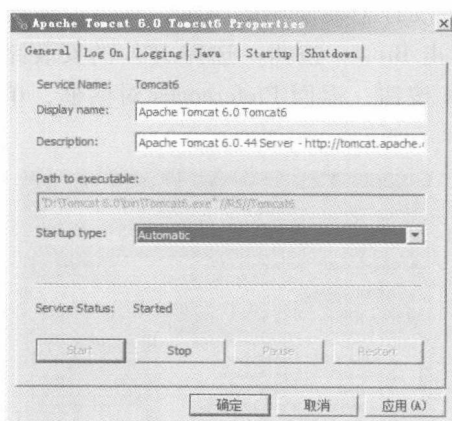


图 1-24 Tomcat 属性对话框

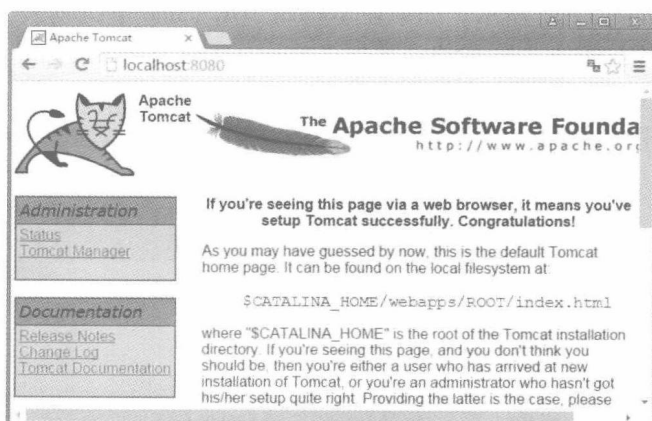


图 1-25 Tomcat 测试界面

### 1.3.4 部署项目

虽然 MyEclipse 6 已经内置了一个简易的 Tomcat, 但在实际开发过程中, 最常用到的还是用单独安装的 Tomcat 作为服务器进行 Web 应用程序的开发。