


21世纪重点大学规划教材

张继军 董卫 编著

# Java Web

## 应用开发技术与案例教程



 机械工业出版社  
CHINA MACHINE PRESS

 配电子教案

21 世纪重点大学规划教材

# Java Web 应用开发技术与案例教程

张继军 董卫 编著



机械工业出版社

本书从实用的角度出发,为 Java Web 开发人员提供了一套实用的开发技术,通过案例由浅入深地介绍这些技术的基本原理和应用,以及它们的整合应用。全书共 12 章,第 1~7 章是基础篇,介绍了 Java Web 开发所必需的基础知识,主要包括:Java Web 开发环境的搭建、静态网页设计技术(HTML、JavaScript、CSS)、JSP 技术、Servlet 技术、JavaBean 技术、JDBC 技术,并基于 Java Web 常用的开发模式介绍了这些技术之间的关系与整合方法;第 8~12 章为提高篇,介绍了 Java Web 应用程序开发的高级技术和常用框架技术,主要包括:EL、JSTL、Ajax 界面设计技术,过滤器、监听器技术,Web 开发中常用的实用技术,Struts2 框架技术、Hibernate 框架技术。

本书提供了丰富的案例程序,通过这些应用案例对开发、集成、部署及具体实现的过程和方法都给出了详尽阐释,使理论与实践紧密结合。力求让读者通过这些案例领会并掌握 Java Web 开发中的各种基本技巧和设计方法。

本书主要面向初学者,特别适合于高等院校和职业院校学生学习 Java Web 应用程序开发技术课程,也可作为 Java Web 开发人员的学习资料和参考书。

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:2399929378,电话:010-88379753)。

## 图书在版编目(CIP)数据

Java Web 应用开发技术与案例教程 / 张继军,董卫编著. —北京:机械工业出版社,2013.9

21 世纪重点大学规划教材

ISBN 978-7-111-44207-3

I. ①J… II. ①张… ②董… III. ①JAVA 语言-程序设计-高等学校-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 230514 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

责任印制:张楠

高教社(天津)印务有限公司印刷

2014 年 1 月第 1 版·第 1 次印刷

184mm×260mm·20.5 印张·509 千字

0001-3000 册

标准书号:ISBN 978-7-111-44207-3

定价:45.00 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

网络服务

社服务中心:(010) 88361066

教材网:<http://www.cmpedu.com>

销售一部:(010) 68326294

机工官网:<http://www.cmpbook.com>

销售二部:(010) 88379649

机工官博:<http://weibo.com/cmp1952>

读者购书热线:(010) 88379203

封面无防伪标均为盗版

# 前 言

Java Web 应用开发技术是目前最主流的 Web 应用开发技术之一。无论是高校的计算机专业还是计算机相关的专业、IT 培训机构都将 Java Web 应用技术作为教学的内容之一。但目前有关 Java Web 应用的书籍多为技术参考书，不适合作为教材；而教材类书籍大多都以 JSP 为主，缺乏多种 Web 技术的整合应用，不适应社会对 Web 技术人才的需求，更不能满足学生学习的需要。

目前，Java Web 应用的开发，都是多种 Web 技术的结合或整合的应用。一个 Web 应用系统是由多种组件构成的，在开发、设计时，要根据不同的组件功能特点，选取不同的 Web 技术给予实现，并将这些技术整合，从而完成应用系统的开发。为此，笔者以培养和提高学生解决实际问题的应用能力，并能适应社会对 Web 应用开发的需求为目标编写了本书。

本书从实用的角度出发，介绍了 Java Web 应用开发的编程技术，从最基本的网页技术到 Struts2 MVC 框架技术和 Hibernate 框架技术，都给出了较详细的介绍和应用案例。

本书的编写特别突出了以下两点：

(1) 突出“系统观点和系统设计”的思想。Java Web 应用的开发实际上就是一个应用系统的开发，需要读者有一个整体的系统观念来组织、理解各部分的功能及其所使用的技术，在内容组织上围绕着提高“系统能力”，提高读者的“系统设计能力”为目标。

(2) 贯穿“项目驱动、设计主导、案例教学”的思想。通过典型的案例，将知识要点融入案例中，在求解案例时，进一步加深对有关技术方法、知识的理解和应用；同时，每个案例都是一个 Web 应用系统，在设计中需要采用工程、系统的思想和方法。

书中的每个案例都按照软件工程的思想和方法给出了详细的设计思想、设计方法、实现步骤的分析和描述，使读者在阅读学习中逐渐培养应用系统的开发方法和技能，提高读者的设计能力，这也是本书较突出的特点。

本书的编写按 Web 技术设置章节，每种开发技术都与其相关的开发案例相结合。对每种技术，采用“技术的基本知识”→“技术的应用案例”→“使用该技术所遇到的问题及其解决方法”的线路组织内容，在应用中提出问题，解决问题，引导读者探讨解决方法，提高读者的学习兴趣和积极性。

本书的第 7 章是第 1~6 章技术的整合应用，基于 Web 开发模式，实现 Web 技术之间的融合，将 Web 开发模式集中介绍并形成对比，同时开发方法由简单模式到 MVC 模式逐步加深扩展，是培养读者提高系统认知能力和系统设计能力的特色内容；第 12 章的应用案例整合了 Struts2+Hibernate 及其相关的技术，便于读者理解和掌握各种开发方法的使用及其特点，加深学生对 Web 技术的理解和掌握。

本书中所介绍的案例和例题都是在 Win7 和 MyEclipse、MySQL 环境下调试运行通过的。每个案例都按软件工程的思想和方法，给出了完整的设计思想和设计步骤，以帮助读者顺利地完成任务。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内容实施。作为教材，每章后附有习题。

本书主要面向初学者，特别适合作为高等院校和职业院校学生学习 Java Web 应用程序开发技术课程的教材，也可作为 Java Web 应用开发人员的学习资料和参考书。

本书由张继军、董卫编著。其中，第 1~4 章、第 7 章、第 11 章由张继军编写，第 5 章、第 6 章、第 8 章、第 9 章由张继军、董卫共同编写，第 10 章、第 12 章由董卫编写。另外，特别感谢费玉奎教授对本书的编写提出了很多宝贵的建议。

为了方便教师备课，本书还配有电子教案（PPT 文件）和案例的源代码。如有需要可在机械工业出版社网站下载。

感谢读者选择使用本书，由于时间仓促，书中难免存在不妥之处，欢迎广大读者对本书内容提出意见和建议，我们将不胜感激。

编 者

# 目 录

出版说明

前言

## 第 1 章 Java Web 应用开发技术概述 ··· 1

### 1.1 Java Web 应用开发技术简介 ··· 1

1.1.1 Java Web 应用 ··· 1

1.1.2 Java Web 应用开发技术 ··· 3

### 1.2 Java Web 开发环境及开发工具 ··· 4

1.2.1 JDK 的下载与安装 ··· 5

1.2.2 Tomcat 服务器的安装和配置 ··· 5

1.2.3 MyEclipse 集成开发工具的  
安装与操作 ··· 7

### 1.3 Java Web 应用程序的 开发与部署 ··· 8

1.3.1 Java Web 应用程序的开发  
过程示例 ··· 8

1.3.2 Java Web 应用程序的目录结构 ··· 14

1.3.3 Java Web 应用程序的打包与部署  
以及导入与导出 ··· 14

1.3.4 配置虚目录 ··· 15

本章小结 ··· 16

习题 ··· 17

## 第 2 章 静态网页开发技术 ··· 18

### 2.1 HTML 网页设计 ··· 18

2.1.1 HTML 文档结构与基本语法 ··· 18

2.1.2 HTML 基本标记与使用 ··· 20

2.1.3 HTML 表单标签与表单设计 ··· 25

2.1.4 表单设计案例——学生入校  
注册页面设计 ··· 27

2.1.5 HTML 框架标签与框架设计 ··· 28

2.1.6 框架设计案例——多媒体播放  
系统设计 ··· 30

### 2.2 CSS 样式表 ··· 32

2.2.1 CSS 样式表的定义与使用 ··· 32

2.2.2 CSS 常用属性 ··· 35

2.2.3 案例——利用 CSS 对注册页面  
实现修饰 ··· 36

## 2.3 JavaScript 脚本语言 ··· 38

2.3.1 JavaScript 的基本语法 ··· 39

2.3.2 JavaScript 的事件 ··· 40

2.3.3 JavaScript 的对象 ··· 42

2.3.4 案例——JavaScript 实现  
输入验证 ··· 44

## 2.4 基于 HTML+JavaScript+CSS 的 开发案例 ··· 46

2.4.1 JavaScript+CSS+DIV 实现  
下拉菜单 ··· 46

2.4.2 JavaScript+CSS+DIV 实现  
表格变色 ··· 48

本章小结 ··· 49

习题 ··· 50

## 第 3 章 JSP 技术 ··· 52

### 3.1 JSP 技术简介 ··· 52

3.1.1 JSP 页面的结构 ··· 52

3.1.2 JSP 程序的运行机制 ··· 53

### 3.2 JSP 语法 ··· 54

3.2.1 JSP 基本元素 ··· 54

3.2.2 JSP 指令元素 ··· 56

3.2.3 JSP 动作元素 ··· 59

### 3.3 JSP 内置对象概述 ··· 62

### 3.4 request 对象 ··· 63

3.4.1 request 对象的常用方法 ··· 63

3.4.2 访问(获取)请求参数 ··· 64

3.4.3 新属性的设置和获取 ··· 66

3.4.4 获取客户端信息 ··· 68

### 3.5 response 对象 ··· 68

3.5.1 response 对象的常用方法 ··· 69

|                    |                                    |     |                    |                                    |     |
|--------------------|------------------------------------|-----|--------------------|------------------------------------|-----|
| 3.5.2              | 重定向网页                              | 69  | 4.3.6              | 删除记录模块的设计与实现                       | 104 |
| 3.5.3              | 页面定时刷新或自动跳转                        | 70  | 4.3.7              | 数据库操作的模板                           | 107 |
| 3.6                | session 对象                         | 70  | 4.3.8              | 整合各设计模块形成完整的应用系统                   | 107 |
| 3.6.1              | session 对象主要方法                     | 71  | 4.3.9              | 问题与思考                              | 109 |
| 3.6.2              | 创建及获取客户的会话信息                       | 71  | 4.4                | 数据源与连接池技术                          | 109 |
| 3.7                | application 对象                     | 72  | 4.4.1              | 配置数据源                              | 110 |
| 3.7.1              | application 对象的主要方法                | 72  | 4.4.2              | 使用连接池技术访问数据库的处理步骤                  | 111 |
| 3.7.2              | 案例——统计网站访问人数                       | 72  | 4.4.3              | 连接池应用——学生身体体质信息显示模块的设计与实现          | 111 |
| 3.8                | out 对象                             | 73  | 4.4.4              | 问题与思考                              | 113 |
| 3.9                | JSP 应用程序设计综合示例                     | 74  | 本章小结               |                                    | 113 |
| 3.9.1              | 网上答题及其自动评测系统                       | 74  | 习题                 |                                    | 113 |
| 3.9.2              | 设计简单的购物车应用案例                       | 76  | 第 4 章 JDBC 数据库访问技术 |                                    | 81  |
| 本章小结               |                                    | 79  | 4.1                | JDBC 技术简介                          | 81  |
| 习题                 |                                    | 79  | 4.1.1              | 驱动程序接口 Driver                      | 82  |
| 第 4 章 JDBC 数据库访问技术 |                                    | 81  | 4.1.2              | 驱动程序管理器 DriverManager              | 82  |
| 4.1                | JDBC 技术简介                          | 81  | 4.1.3              | 数据库连接接口 Connection                 | 82  |
| 4.1.1              | 驱动程序接口 Driver                      | 82  | 4.1.4              | 执行 SQL 语句接口 Statement              | 82  |
| 4.1.2              | 驱动程序管理器 DriverManager              | 82  | 4.1.5              | 执行动态 SQL 语句接口 PreparedStatement    | 83  |
| 4.1.3              | 数据库连接接口 Connection                 | 82  | 4.1.6              | 访问结果集接口 ResultSet                  | 83  |
| 4.1.4              | 执行 SQL 语句接口 Statement              | 82  | 4.2                | JDBC 访问数据库                         | 83  |
| 4.1.5              | 执行动态 SQL 语句接口 PreparedStatement    | 83  | 4.2.1              | 注册驱动 MySQL 的驱动程序                   | 84  |
| 4.1.6              | 访问结果集接口 ResultSet                  | 83  | 4.2.2              | JDBC 连接数据库创建连接对象                   | 84  |
| 4.2                | JDBC 访问数据库                         | 83  | 4.2.3              | 创建数据库的操作对象                         | 86  |
| 4.2.1              | 注册驱动 MySQL 的驱动程序                   | 84  | 4.2.4              | 执行 SQL                             | 87  |
| 4.2.2              | JDBC 连接数据库创建连接对象                   | 84  | 4.2.5              | 获得查询结果并进行处理                        | 88  |
| 4.2.3              | 创建数据库的操作对象                         | 86  | 4.2.6              | 释放资源                               | 89  |
| 4.2.4              | 执行 SQL                             | 87  | 4.2.7              | 数据库乱码解决方案                          | 90  |
| 4.2.5              | 获得查询结果并进行处理                        | 88  | 4.3                | 综合案例——学生身体体质信息管理系统的开发              | 91  |
| 4.2.6              | 释放资源                               | 89  | 4.3.1              | 数据库和数据表的建立                         | 91  |
| 4.2.7              | 数据库乱码解决方案                          | 90  | 4.3.2              | 注册驱动并建立数据库的连接                      | 92  |
| 4.3                | 综合案例——学生身体体质信息管理系统的开发              | 91  | 4.3.3              | 添加记录模块的设计与实现                       | 92  |
| 4.3.1              | 数据库和数据表的建立                         | 91  | 4.3.4              | 查询记录模块的设计与实现                       | 95  |
| 4.3.2              | 注册驱动并建立数据库的连接                      | 92  | 4.3.5              | 修改记录模块的设计与实现                       | 99  |
| 4.3.3              | 添加记录模块的设计与实现                       | 92  | 4.3.6              | 删除记录模块的设计与实现                       | 104 |
| 4.3.4              | 查询记录模块的设计与实现                       | 95  | 4.3.7              | 数据库操作的模板                           | 107 |
| 4.3.5              | 修改记录模块的设计与实现                       | 99  | 4.3.8              | 整合各设计模块形成完整的应用系统                   | 107 |
| 4.3.6              | 删除记录模块的设计与实现                       | 104 | 4.3.9              | 问题与思考                              | 109 |
| 4.3.7              | 数据库操作的模板                           | 107 | 4.4                | 数据源与连接池技术                          | 109 |
| 4.3.8              | 整合各设计模块形成完整的应用系统                   | 107 | 4.4.1              | 配置数据源                              | 110 |
| 4.3.9              | 问题与思考                              | 109 | 4.4.2              | 使用连接池技术访问数据库的处理步骤                  | 111 |
| 4.4                | 数据源与连接池技术                          | 109 | 4.4.3              | 连接池应用——学生身体体质信息显示模块的设计与实现          | 111 |
| 4.4.1              | 配置数据源                              | 110 | 4.4.4              | 问题与思考                              | 113 |
| 4.4.2              | 使用连接池技术访问数据库的处理步骤                  | 111 | 本章小结               |                                    | 113 |
| 4.4.3              | 连接池应用——学生身体体质信息显示模块的设计与实现          | 111 | 习题                 |                                    | 113 |
| 4.4.4              | 问题与思考                              | 113 | 第 5 章 JavaBean 技术  |                                    | 115 |
| 本章小结               |                                    | 113 | 5.1                | JavaBean 技术                        | 115 |
| 习题                 |                                    | 113 | 5.1.1              | JavaBean 的设计                       | 115 |
| 第 5 章 JavaBean 技术  |                                    | 115 | 5.1.2              | JavaBean 的安装部署                     | 116 |
| 5.1                | JavaBean 技术                        | 115 | 5.2                | 在 JSP 中使用 JavaBean                 | 116 |
| 5.1.1              | JavaBean 的设计                       | 115 | 5.2.1              | 声明 JavaBean 对象                     | 118 |
| 5.1.2              | JavaBean 的安装部署                     | 116 | 5.2.2              | 访问 JavaBean 属性——设置 JavaBean 属性值    | 118 |
| 5.2                | 在 JSP 中使用 JavaBean                 | 116 | 5.2.3              | 访问 JavaBean 属性——获取 JavaBean 属性值并显示 | 120 |
| 5.2.1              | 声明 JavaBean 对象                     | 118 | 5.2.4              | 访问 JavaBean 方法——调用 JavaBean 业务处理方法 | 120 |
| 5.2.2              | 访问 JavaBean 属性——设置 JavaBean 属性值    | 118 | 5.2.5              | 案例——基于 JavaBean+JSP 求任意两数代数和       | 121 |
| 5.2.3              | 访问 JavaBean 属性——获取 JavaBean 属性值并显示 | 120 | 5.3                | 多个 JSP 页面共享 JavaBean               | 122 |
| 5.2.4              | 访问 JavaBean 方法——调用 JavaBean 业务处理方法 | 120 | 5.3.1              | 共享 JavaBean 的创建                    | 122 |
| 5.2.5              | 案例——基于 JavaBean+JSP 求任意两数代数和       | 121 | 5.3.2              | 案例——网页计数器 JavaBean 的设计与使用          | 123 |
| 5.3                | 多个 JSP 页面共享 JavaBean               | 122 | 5.4                | 综合案例——数据库访问 JavaBean 的设计           | 125 |
| 5.3.1              | 共享 JavaBean 的创建                    | 122 | 本章小结               |                                    | 129 |
| 5.3.2              | 案例——网页计数器 JavaBean 的设计与使用          | 123 | 习题                 |                                    | 129 |
| 5.4                | 综合案例——数据库访问 JavaBean 的设计           | 125 | 第 6 章 Servlet 技术   |                                    | 130 |
| 本章小结               |                                    | 129 | 6.1                | Servlet 技术                         | 130 |
| 习题                 |                                    | 129 | 6.1.1              | Servlet 编程接口                       | 130 |
| 第 6 章 Servlet 技术   |                                    | 130 |                    |                                    |     |
| 6.1                | Servlet 技术                         | 130 |                    |                                    |     |
| 6.1.1              | Servlet 编程接口                       | 130 |                    |                                    |     |

|  |     |   |     |
|--|-----|---|-----|
| 6.1.2 设计 Servlet .....                       | 130 | 7.3.2 JSP+Servlet 开发案例——求和运算 .....                | 160 |
| 6.2 Servlet 常用对象及其方法 .....                   | 136 | 7.3.3 JSP+Servlet+JDBC 开发案例——基于数据库的登录验证 .....     | 162 |
| 6.3 综合案例——基于 JSP+Servlet 的用户登录验证 .....       | 137 | 7.3.4 JSP+Servlet 开发模式的优点与缺点 .....                | 164 |
| 6.4 在 Servlet 中使用 JavaBean .....             | 139 | 7.4 JSP+Servlet+JavaBean 开发模式 .....               | 164 |
| 6.5 JSP 与 Servlet 的数据共享 .....                | 139 | 7.4.1 基于 JSP+Servlet+JavaBean 的 MVC 的实现 .....     | 164 |
| 6.5.1 基于请求的数据共享 .....                        | 139 | 7.4.2 JSP+Servlet+JavaBean 开发案例——求和运算 .....       | 165 |
| 6.5.2 基于会话的数据共享 .....                        | 140 | 7.4.3 JSP+Servlet+JavaBean 案例——基于数据库的登录验证 .....   | 166 |
| 6.5.3 基于应用的数据共享 .....                        | 141 | 7.4.4 JSP+Servlet+JavaBean 案例——学生体质信息管理系统 .....   | 168 |
| 6.6 JSP 与 Servlet 的关联关系 .....                | 141 | 7.5 JSP+Servlet+JavaBean+DAO 开发模式 .....           | 176 |
| 6.7 基于 JSP+Servlet+JavaBean 实现复数运算 .....     | 143 | 7.5.1 DAO 模式与数据库访问架构 .....                        | 176 |
| 6.8 Cookie 管理 .....                          | 146 | 7.5.2 JSP+Servlet+JavaBean+DAO 案例——学生体质信息管理 ..... | 176 |
| 6.8.1 Cookie 的基本用法 .....                     | 146 | 本章小结 .....  | 181 |
| 6.8.2 Cookie 的相关方法 .....                     | 147 | 习题 .....  | 181 |
| 6.8.3 案例——利用 Cookie 实现自动登录 .....             | 148 | 第 8 章 EL、JSTL 和 Ajax 技术 .....                     | 183 |
| 本章小结 .....                                   | 150 | 8.1 表达式语言 EL .....                                | 183 |
| 习题 .....                                     | 150 | 8.1.1 EL 语法 .....                                 | 183 |
| 第 7 章 Java Web 常用开发模式与案例 .....               | 151 | 8.1.2 EL 内部对象 .....                               | 186 |
| 7.1 单纯的 JSP 页面开发模式 .....                     | 151 | 8.1.3 EL 对 JavaBean 的访问 .....                     | 190 |
| 7.1.1 单纯的 JSP 页面开发模式简介 .....                 | 151 | 8.2 JSTL 标签库 .....                                | 191 |
| 7.1.2 JSP 页面开发模式案例——求和运算 .....               | 151 | 8.2.1 JSTL 简介 .....                               | 191 |
| 7.1.3 JSP+JDBC 开发模式案例——实现基于数据库的登录验证 .....    | 153 | 8.2.2 常用 JSTL 标签 .....                            | 192 |
| 7.1.4 单纯的 JSP 页面开发模式存在的问题与缺点 .....           | 155 | 8.3 综合案例——使用 EL 和 JSTL 显示查询结果 .....               | 195 |
| 7.2 JSP+JavaBean 开发模式 .....                  | 155 | 8.4 Ajax 技术 .....                                 | 197 |
| 7.2.1 JSP+JavaBean 开发模式简介 .....              | 155 | 8.4.1 Ajax 技术简介 .....                             | 197 |
| 7.2.2 JSP+JavaBean 开发案例——求和运算 .....          | 156 | 8.4.2 XMLHttpRequest 对象 .....                     | 197 |
| 7.2.3 JSP+JavaBean+JDBC 案例——基于数据库的登录验证 ..... | 157 | 8.5 Ajax 应用案例 .....                               | 200 |
| 7.2.4 JSP+JavaBean 开发模式的优点与缺点 .....          | 159 | 8.5.1 案例——异步表单验证 .....                            | 201 |
| 7.3 JSP+Servlet 开发模式 .....                   | 159 | 8.5.2 案例——实现级联列表 .....                            | 203 |
| 7.3.1 JSP+Servlet 开发模式简介 .....               | 159 |   |     |



|   |            |  |            |
|---|------------|--|------------|
| 8.5.3 案例——输入提示和<br>自动完成 .....           | 205        | 10.4.4 案例——使用 Cos 组件实现<br>作业上传 .....               | 230        |
| 本章小结 .....                              | 208        | 10.5 Java Mail 编程 .....                            | 232        |
| 习题 .....                                | 208        | 10.5.1 Java Mail 简介 .....                          | 232        |
| <b>第 9 章 过滤器和监听器技术</b> .....            | <b>209</b> | 10.5.2 使用 Java Mail 发送邮件 .....                     | 232        |
| 9.1 过滤器技术 .....                         | 209        | 10.5.3 案例——使用 Java Mail 实现<br>邮件发送 .....           | 233        |
| 9.1.1 过滤器编程接口 .....                     | 209        | 10.6 页面分页技术 .....                                  | 234        |
| 9.1.2 设计过滤器 .....                       | 210        | 10.6.1 分页技术的设计思想 .....                             | 234        |
| 9.1.3 案例——基于过滤器的用户<br>权限控制 .....        | 213        | 10.6.2 分页具体实现 .....                                | 235        |
| 9.1.4 案例——基于过滤器的中文<br>乱码解决 .....        | 214        | 本章小结 .....   | 237        |
| 9.1.5 案例——禁止未授权的 IP 访问<br>站点过滤器 .....   | 215        | 习题 .....   | 237        |
| 9.2 监听器技术 .....                         | 216        | <b>第 11 章 Struts2 框架技术</b> .....                   | <b>238</b> |
| 9.2.1 监听器编程接口 .....                     | 216        | 11.1 Struts2 简介 .....                              | 238        |
| 9.2.2 设计监听器 .....                       | 218        | 11.1.1 Struts2 的组成与工作原理 .....                      | 238        |
| 9.2.3 案例——会话计数监听器的<br>设计 .....          | 219        | 11.1.2 搭建 Struts2 开发环境 .....                       | 239        |
| 本章小结 .....                              | 220        | 11.1.3 Struts2 入门案例——基于 Struts2<br>任意两数据的代数和 ..... | 241        |
| 习题 .....                                | 220        | 11.1.4 Struts 2 的中文乱码问题处理 .....                    | 244        |
| <b>第 10 章 Java Web 实用开发技术</b> .....     | <b>221</b> | 11.2 Struts2 的配置文件 .....                           | 244        |
| 10.1 图形验证码 .....                        | 221        | 11.3 Struts2 的业务控制器——<br>Action 类设计 .....          | 248        |
| 10.1.1 图形验证码简介 .....                    | 221        | 11.3.1 Action 实现类 .....                            | 248        |
| 10.1.2 图形验证码的实现 .....                   | 221        | 11.3.2 Action 访问 Web 对象 .....                      | 250        |
| 10.1.3 案例——带图形验证码的<br>登录模块 .....        | 223        | 11.3.3 多方法的 Action .....                           | 255        |
| 10.2 MD5 加密 .....                       | 224        | 11.4 Struts2 的 OGNL 表达式、<br>标签库、国际化 .....          | 259        |
| 10.2.1 MD5 加密算法简介 .....                 | 224        | 11.4.1 Struts2 的 OGNL 表达式 .....                    | 259        |
| 10.2.2 MD5 算法的实现 .....                  | 225        | 11.4.2 Struts2 的标签库 .....                          | 260        |
| 10.3 在线编辑器 .....                        | 225        | 11.4.3 Struts2 的国际化 .....                          | 266        |
| 10.3.1 在线编辑器简介 .....                    | 225        | 11.4.4 Struts2 的国际化应用案例 .....                      | 268        |
| 10.3.2 CKEditor 的使用 .....               | 226        | 11.5 Struts2 的拦截器 .....                            | 270        |
| 10.3.3 案例——使用 CKEditor 编辑<br>公告内容 ..... | 227        | 11.5.1 Struts2 的内建拦截器 .....                        | 270        |
| 10.4 文件的上传与下载 .....                     | 228        | 11.5.2 Struts2 拦截器的自定义实现 .....                     | 271        |
| 10.4.1 常见文件上传下载组件 .....                 | 228        | 11.5.3 案例——文字过滤器的<br>设计与应用 .....                   | 273        |
| 10.4.2 文件上传的实现 .....                    | 229        | 11.6 Struts2 的文件上传和下载 .....                        | 275        |
| 10.4.3 文件下载的实现 .....                    | 230        | 11.6.1 文件上传 .....                                  | 276        |
|   |            | 11.6.2 文件下载 .....                                  | 278        |

|                                      |     |  |     |
|--------------------------------------|-----|--|-----|
| 11.7 Struts2 的输入验证 .....             | 280 | 12.4 Hibernate 运行过程与<br>编程步骤 .....                 | 298 |
| 11.7.1 使用 validate()方法实现<br>验证 ..... | 280 | 12.4.1 Hibernate 运行过程 .....                        | 298 |
| 11.7.2 使用验证文件实现验证 .....              | 280 | 12.4.2 使用 Hibernate 编程步骤 .....                     | 299 |
| 11.7.3 案例——实现客户注册<br>输入验证 .....      | 283 | 12.4.3 Hibernate 编程入门案例 .....                      | 300 |
| 本章小结 .....                           | 289 | 12.5 Hibernate 的实体映射 .....                         | 303 |
| 习题 .....                             | 289 | 12.5.1 实体映射基础 .....                                | 303 |
| <b>第 12 章 Hibernate 持久化技术</b> .....  | 291 | 12.5.2 实体关系映射 .....                                | 304 |
| 12.1 Hibernate 技术简介 .....            | 291 | 12.6 Hibernate 的实体操作与<br>数据查询 .....                | 307 |
| 12.1.1 Hibernate 简介 .....            | 291 | 12.6.1 实体操作 .....                                  | 307 |
| 12.1.2 Hibernate 的体系结构 .....         | 291 | 12.6.2 数据查询 .....                                  | 308 |
| 12.2 Hibernate 软件包的<br>下载与配置 .....   | 292 | 12.6.3 案例——使用 Hibernate 实现<br>UserDao .....        | 310 |
| 12.3 Hibernate 核心组件 .....            | 293 | 12.7 综合案例——基于 Struts2+Hibernate<br>的学生信息管理系统 ..... | 311 |
| 12.3.1 Hibernate 核心类 .....           | 294 | 本章小结 .....   | 317 |
| 12.3.2 Hibernate 的 PO 对象 .....       | 296 | 习题 .....   | 317 |
| 12.3.3 Hibernate 配置文件 .....          | 296 | 参考文献 .....   | 318 |
| 12.3.4 Hibernate 映射文件 .....          | 297 |  |     |

# 第 1 章 Java Web 应用开发技术概述

Java Web 应用程序会生成各种类型的标记语言（HTML、XML 等）和动态内容的交互 Web 网页。它通常由 Web 组件（如 JSP、Servlet、JavaBean 等）组成，可用来修改和临时存储数据，与数据库和 Web 服务器交互，以及根据客户端的请求呈现内容。开发 Web 应用程序需要有关的技术和工具，本章简单介绍 Java Web 开发所需要的主流技术和常用框架技术，以及开发 Java Web 应用所需要的开发环境、运行环境和开发工具。

## 1.1 Java Web 应用开发技术简介

Java Web 应用开发是基于 JavaEE（Java Enterprise Edition）框架的，而 JavaEE 是建立在 Java 平台上的企业级应用的解决方案。JavaEE 框架提供的 Web 开发技术主要支持两类软件的开发和应用，一类是做高级信息系统框架的 Web 应用服务器（Web Application Server），另一类是在 Web 应用服务器上运行的 Web 应用（Web Application）。本书所介绍的 Java Web 应用开发就是这里的第 2 类，即在 Web 应用服务器上运行的 Web 应用开发。

JavaEE 框架是由 J2EE 更名的。Sun 公司在 1998 年发布 JDK1.2 版本的时候，开始使用名称 Java 2 Platform，即 Java 2 平台，修改后的 JDK 称为 Java 2 Platform Software Developing Kit，即 J2SDK，并分为标准版（Standard Edition，J2SE）、企业版（Enterprise Edition，J2EE）和微型版（Micro Edition，J2ME）。2006 年 5 月，Sun 公司推出 JavaSE5，此时，Java 的各种版本又更名，J2EE 更名为 JavaEE，J2SE 更名为 JavaSE，J2ME 更名为 JavaME。

### 1.1.1 Java Web 应用

“Java Web 应用”一般定义为：一个由 HTML/XML 文档、Java Servlet、JSP（Java Server Pages）、JSTL（Java Server Pages Standard Tag Library）、类以及其他任何种类文件可以捆绑起来，并在来自多个厂商的多个 Web 容器上运行的 Web 资源构成的集合。可以将 Web 应用从一个服务器移到另外一个服务器，或者移动到同一服务器的不同位置，而不需要对组成 Web 应用的任何种类的文件作任何改动。而 Java Web 应用开发是基于 JavaEE 框架的，需要在该框架的容器和组件支持下完成。

#### 1. 容器

“容器（Container）”指的是提供特定程序组件服务的标准化运行环境，通过这些组件可以在 JavaEE 平台上得到所期望的服务。容器的作用是为用户提供与部署、执行、生命周期管理、安全和其他组件需求相关的服务。此外，不同类型的容器明确地为它们管理的各种类型的组件提供附加服务。例如，Web 容器都提供响应客户请求、执行请求时间的处理，以及将结果返回到客户端的运行环境支持；Web 容器还负责管理某些基本服务，像诸如组件的生命周期、数据库连接资源的共享、数据持久性等。

一般来说，软件开发人员只要开发出满足 JavaEE 应用需要的组件并能安装在容器内就可以了。程序组件的安装过程包括设置各个组件在 JavaEE 应用服务器中的参数，以及设置 JavaEE 应用服务器本身，这些设置决定了在底层由 JavaEE 服务器提供的多种服务（例如安全、交易管理、JNDI 查寻和远程方法调用等）。

JavaEE 平台对每一种主要的组件类型都定义了相应的容器类型。JavaEE 平台由 Applet 容器、应用客户端容器（Application Client Container）、Web 容器（Servlet、JSP 容器）和 EJB 容器（Enterprise JavaBeans Container）4 种类型的程序容器组成。

(1) EJB 容器——为 Enterprise JavaBean 组件提供运行时环境，它对应于业务层和数据访问层，主要负责数据处理以及和数据库或其他 Java 程序的通信。

(2) Web 容器——管理 JSP 和 Servlet 等 Web 组件的运行，主要负责 Web 应用和浏览器的通信，它对应于表示层。Web 容器是本书所使用的容器。

(3) 应用客户端容器——负责 Web 应用在客户端组件的运行，对应于用户界面层。

(4) Applet 容器——负责在 Web 浏览器和 Java 插件（Java Plug-in）上运行 Java Applet 程序，对应于用户界面层。

每种容器内都使用相关的 Java Web 编程技术。这些技术包括应用组件技术（如 Servlet、JSP、EJB 等技术构成了应用的主体）、应用服务技术（如 JDBC、JNDI 等服务保证组件具有稳定的运行时环境）、通信技术（如 RMI、JavaMail 等技术在平台底层实现机器和应用程序之间的信息传递）等 3 类。

## 2. 组件

为了降低软件开发成本，适应企业快速发展的需求，JavaEE 平台提供了基于组件的方式设计、开发、组装和部署企业应用系统。按照这种方式开发出来的 JavaEE 组件，不依赖于某个特定厂商提供的产品或者 API，不管是开发商还是最终用户，都有最大的自由去选择那些能更好地满足业务或技术需求的产品或组件。

组件（Component）是指在应用程序中能发挥特定功能的软件单位，实质上是几种特定的 Java 程序，只不过这些程序被规定了固定的格式和编写方法，它们的功能和使用方式在一定程度上被标准化了。例如，在 Java 2 标准版中提供的 JavaBean 组件，就是按照特定格式编写的 Java 类文件，JavaBean 可以通过 get/set 方法访问对象中的属性数据。

JavaEE 平台主要提供了以下 3 类 JavaEE 组件：

(1) 客户端组件——客户端的 Applet 和客户端应用程序。

(2) Web 组件——Web 容器内的 JSP、Servlet、Web 过滤器、Web 事件监听器等。

(3) EJB 组件——EJB 容器内的 EJB 组件。

## 3. 组件与容器的关系

组件是组装到 JavaEE 平台中独立的软件功能单元，每一个 JavaEE 组件在容器中执行，容器为组件提供标准服务和 API，容器充当通向底层 JavaEE 平台的接口。“连接器（Connector）”在概念上驻留在 JavaEE 平台的下方，连接器提供了可移植服务的 API，JavaEE 应用使用这些 API 来插入到现有的企业应用中。连接器也称为资源适配器，它为 JavaEE 体系结构增加了另一种灵活性。

## 4. Java Web 应用的定义

基于“组件”和“容器”的视角，在 JavaEE 平台下，Web 应用是满足下列要求的软件

体系。

(1) Java Web 应用由软件组件构成，这些组件根据其各自所属的层进行分类。

(2) 组成 Java Web 应用的各种组件在对应容器中执行，容器为组件提供底层 JavaEE API 的统一视图。

(3) 容器管理组件并为组件提供多种系统级服务。例如，生命周期管理、事务管理、数据缓存、异常处理实例池、线程以及安全性。Java Web 应用以分布式组件集合的形式存在，而各分布式软件组件在其各自的容器中运行。

(4) Java Web 应用客户为应用提供用户界面，客户端向最终用户提供了一个窗口，最终用户可以通过该窗口使用 Java Web 应用提供的各种服务。

### 1.1.2 Java Web 应用开发技术

Java Web 应用程序供用户通过浏览器（如 IE）发送请求，程序通过执行产生 Web 页面，并将页面传递给客户机器上的浏览器，将得到的 Web 页面呈现给用户。

一个完整的 Java Web 应用程序通常由多种组件构成的，一般由表示层组件、控制层组件、业务逻辑层组件及其数据访问层（或持久层）组件组成。

- 表示层组件一般由 HTML 和 JSP 页面组成。
- 控制层组件一般由 Servlet 组成。
- 业务逻辑层一般是 JavaBean 或 EJB。
- 持久层组件一般是 JDBC、Hibernate。
- 此外，Java Web 应用的各个组件需要在 XML 格式的配置文件中进行声明，然后打包，部署到 Java Web 服务器（如 Tomcat）中运行。

下面简单介绍 HTML、CSS、JavaScript JSP、Servlet、JavaBean、JDBC、XML、Tomcat 技术以及 Struts2、Hibernate 等框架技术。对于它们的具体内容，将在以后各章中将详细介绍。

#### 1. HTML

HTML（Hypertext Markup Language）即超文本链接标示语言，使用它可以设计静态网页。

#### 2. CSS

CSS（Cascading Style Sheets）即层叠样式表，简称“样式表”，是一种美化网页的技术，主要完成字体、颜色、布局等方面的各种设置。

在 HTML 基础上，使用 CSS 不仅能够统一、高效地组织页面上的元素，还可以使页面具有多样的外观。

#### 3. JavaScript

JavaScript 是一种简单的脚本语言，在浏览器中直接运行，无须服务器端的支持。这种脚本语言可以直接嵌套在 HTML 代码中，它响应一系列的事件。当一个 JavaScript 函数响应的动作发生时，浏览器就会执行对应的 JavaScript 代码，从而在浏览器端实现与客户的交互。

JavaScript 增加了 HTML 网页的互动性，它可以在浏览器端实现一系列动态的功能，仅仅依靠浏览器就可以完成一些与用户的互动。

#### 4. JSP

JSP 页面由 HTML 代码和嵌入其中的 Java 代码组成。在页面被客户端请求后，Web 服务器对 Java 代码进行处理，然后将生成的 HTML 页面返回客户端的浏览器。JSP 页面一般包含 JSP 指令、JSP 脚本元素、JSP 标准动作以及 JSP 内置对象。

#### 5. Servlet

Servlet (Java 服务器小程序) 是用 Java 语言编写的服务器端程序，是由服务器端调用和执行的。它可以处理客户端传来的 HTTP 请求，并返回一个响应。它是按照 Servlet 自身规范设计的一个 Java 类，具有可移植性、功能强大、安全、继承、模块化和可扩展性好等特点。

#### 6. JavaBean

JavaBean 用 Java 语言编写并遵循一定规范的类，该类的一个实例称为 JavaBean，简称 Bean。JavaBean 可以被 JSP 引用，也可以被 Servlet 引用。

#### 7. JDBC

JDBC (Java Database Connectivity, 数据库访问接口) 是 Java Web 应用程序开发中最主要的 API 之一，任何应用程序总是需要访问数据库。它使数据库开发人员能够用标准的 Java API 编写数据库应用程序。JDBC API 主要用来连接数据库和直接调用 SQL 命令执行各种 SQL 语句。

#### 8. XML

XML (eXtensible Markup Language, 可扩展的标记语言)，在 Java Web 应用程序中，XML 主要用于描述配置信息。Servlet、Struts2 以及 Hibernate 框架都需要配置文件，它们的配置文件都是 XML 格式的。

#### 9. Struts2

Struts2 框架，提供了一种基于 MVC 体系结构的 Web 程序的开发方法，具有组件模块化、灵活性和重用性等优点，使基于 MVC 模式的程序结构更加清晰，同时也简化了 Web 应用程序的开发，是目前最常用的开发框架。

#### 10. Hibernate

Hibernate 是一个面向 Java 环境的对象/关系数据库映射工具，即 ORM (Object-Relation Mapping 对象——关系映射) 工具。它对 JDBC API 进行了封装，负责 Java 对象的持久化，在分层的软件架构中位于下持久化层，封装了所有数据访问细节，使业务逻辑层可以专注于实现业务逻辑。

另外，还有 Ajax、EL、JSTL、过滤器、监听器等技术。

## 1.2 Java Web 开发环境及开发工具

Java Web 应用开发，就是如何使用 Java 语言及其相关的开发技术，来完成 Web 应用程序的开发过程。开发 Java Web 应用程序，需要相应的开发环境和开发工具。本节主要介绍 Java Web 开发环境的搭建和开发工具的使用。

主要内容包括：下载并安装 Java 的 JDK，下载并安装 (Tomcat) 服务器，下载集成开发工具并配置开发环境，设计简单的 Java Web 应用程序并部署和测试，并给出如何创建定制的 Web 应用。

## 1.2.1 JDK 的下载与安装

JDK 即 Java Development Kit (Java 开发工具包) 的缩写。它是整个 Java 的核心, 其中不仅包含了 Java 运行环境 JRE (Java Runtime Environment), 还包括了众多的 Java 开发工具和 Java 基础类库 (\*.jar)。

目前主流的 JDK 是 Sun 公司发布的 JDK。在开发和运行 Java Web 程序前, 首先必须安装 JDK。本书所使用的 JDK 是 Sun 公司发布的 JDK6。

### 1. 下载 JDK 安装程序

Sun 公司提供免费的 JDK 供 Windows 以及 Linux 平台使用, 可从 <http://java.sun.com/javase/downloads/index.jsp> 网站下载最新的 JDK 版本。在本书中, 使用了基于 Windows 操作系统的“jdk-6u7-windows-i586-p.exe”文件。

### 2. 安装 JDK

双击安装文件“jdk-6u7-windows-i586-p.exe”, 系统自动进入安装进程, 按照向导指示即可完成安装。假设将 JDK 安装于“C:\Java\”目录下, 则在该目录下有 jdk1.6.0\_20 和 jre6 两个子目录, 分别存放 Java 程序的开发环境和运行环境。

另外, 一般还需要对安装后的 JDK 进行配置 (设置环境变量)。由于本书中使用的是 MyEclipse 开发工具, 将在 1.3 节中开发案例中给出, 此处将不再配置。

## 1.2.2 Tomcat 服务器的安装和配置

Tomcat 是一个免费的开源的 Servlet 容器, 可从 <http://tomcat.apache.org> 处下载最新版本。本书使用 tomcat-6.0.26 版本。对于 Windows 操作系统, tomcat-6.0.26 提供了两种安装文件, 一种是 apache-tomcat-6.0.26.exe, 一种是 apache-tomcat-6.0.26.zip。本书下载 apache-tomcat-6.0.26.exe。

### 1. 安装和配置 Tomcat

双击 Tomcat 安装文件 apache-tomcat-6.0.26.exe 将启动 Tomcat 安装程序, 如图 1-1 所示。按照向导一直单击 Next, 可自动完成 Tomcat 的安装。但要注意以下几点:

(1) 安装目录: 此处设置 Tomcat 的安装路径“C:\Java”。通常其默认路径是“C:\Program Files\Apache Software Foundation\Tomcat 6.0”。

(2) 安装到图 1-2 所示的安装界面时, 要选择端口号和配置管理员的用户名和密码。可按照默认值安装, 也可根据需要修改各项内容, 但一定要记住修改后的端口号和管理员的用户名及密码, 因为在以后使用 Tomcat 的过程中要用到这两项内容。一般按默认值安装 (端口号为: 8080, 用户名: admin, 密码: 空)。

(3) 在安装过程中安装程序会自动搜索 Java 虚拟机的安装路径, 然后提供给用户确认, 如图 1-3 所示。



图 1-1 Tomcat 安装向导首页

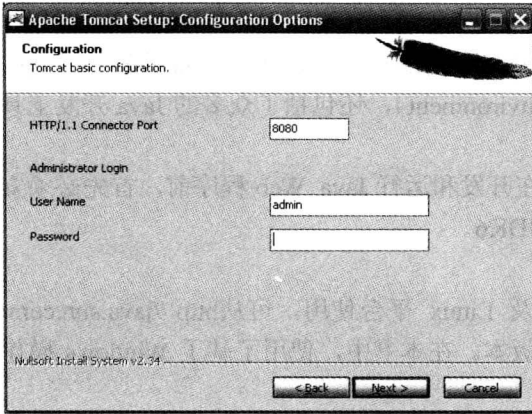


图 1-2 安装设置

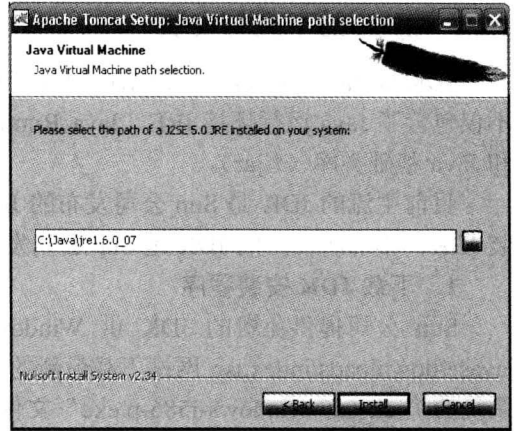


图 1-3 自动选择 JDK 安装路径

(4) 最后选择安装，则可完成 Tomcat 6.0 的安装。安装完成后，在 Windows 系统的“开始”->“程序”菜单下会添加 Apache Tomcat 6.0 菜单组。

## 2. 测试 Tomcat

打开 IE 浏览器，在地址栏中输入 `http://localhost:8080` 或 `http://127.0.0.1:8080` (localhost 和 127.0.0.1 均表示本地机器，8080 是 Tomcat 默认监听的端口号)，将会打开 Tomcat 的默认主页，如图 1-4 所示，表示 Tomcat 安装成功。

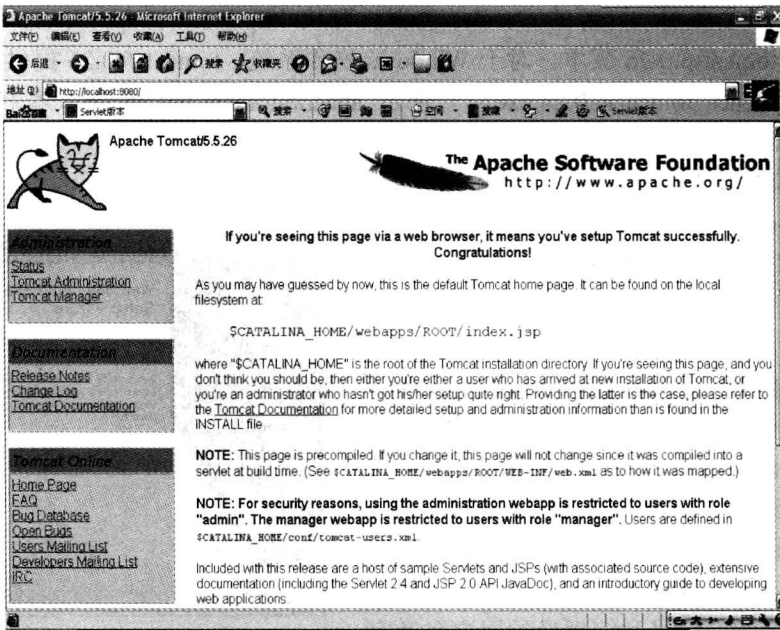


图 1-4 Tomcat 默认主页

## 3. Tomcat 的目录结构

Tomcat 6.0 安装目录下有 bin、conf、lib、logs、temp、webapps 和 work 等子目录，其目录结构及其用途，如表 1-1 所示。



表 1-1 Tomcat 的目录结构及用途

| Tomcat 目录 | 用 途                                       |
|-----------|---|
| /bin      | 存放启动和关闭 Tomcat 的命令文件                      |
| /lib      | 存放 Tomcat 服务器及所有 Web 应用程序都可以访问的 JAR 文件    |
| /conf     | 存放 Tomcat 的配置文件，如 server.xml，web.xml 等    |
| /logs     | 存放 Tomcat 的日志文件                           |
| /temp     | 存放 Tomcat 运行时产生的临时文件                      |
| /webapps  | 通常把 Web 应用程序的目录及文件放到这个目录下                 |
| /work     | Tomcat 将 JSP 生成的 Servlet 源文件和字节码文件放到这个目录下 |

注意：对于开发的 Java Web 应用程序，在部署后，其应用程序按照一定的目录结构放置在目录/webapps 下。

### 1.2.3 MyEclipse 集成开发工具的安装与操作

MyEclipse 是一个基于 Java 的开放源代码的可扩展的应用开发平台，它为编程人员提供了一流的 Java 集成开发环境。MyEclipse 更新比较快，目前最新的版本为 MyEclipse 11 版本。无论新版本还是以前的版本，其基本操作类似，只是在新版本中集成了更多的框架和第三方的 jar 包。本书使用 MyEclipse 6.0 版本。

#### 1. 安装、配置 MyEclipse

MyEclipse 是一款商业的基于 Eclipse 的 Java EE 集成开发工具，官方站点是 <http://www.myeclipseide.com/>。进入到 MyEclipse 的下载页面后，有几个不同版本可供下载，推荐下载 ALL in ONE 版本。双击下载的文件，然后一直单击 Next，直至结束。

#### 2. 运行 MyEclipse

安装完成后可以单击“开始”下的“所有程序”，找到 MyEclipse 6.0 程序组中的 MyEclipse 6.0，单击即可启动 MyEclipse 6.0。

第一次启动后主界面显示欢迎页面，关闭欢迎页面后就可以开发 Web 程序了。这时 MyEclipse 界面如图 1-5 所示。

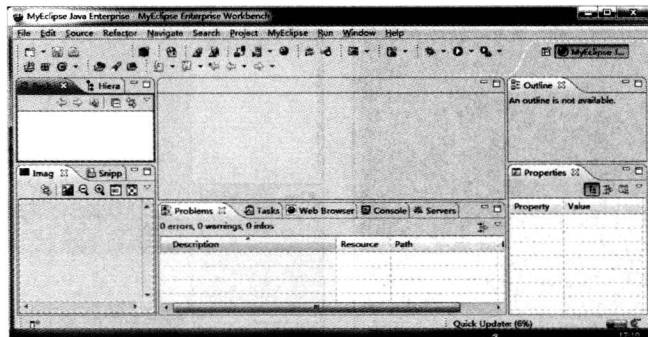


图 1-5 MyEclipse 的工作界面

#### 3. MyEclipse 的操作与快捷键

利用 MyEclipse，可以较容易地开发 Java Web 的各组件，并能部署和运行，具体操作过