

高等院校教学改革示范教材 软件开发系列

JavaEE

架构与程序设计（第2版）

◎ 王晓华 谢晓东 主 编

◎ 陈 永 孔德华 邹金萍 副主编

- ☑ 一个案例贯穿始终，循序渐进，边做边学，边学边做，快速提高实践能力
- ☑ 图文并茂，大量实际操作截图以及代码展示，配备视频课件，保证学习效果
- ☑ 教材内容围绕企业真实需求设计，重点学习企业实用技术，摒弃冗长理论铺垫
- ☑ 将PPT编辑在教材中，起到提纲挈领的作用，学习者能快速掌握章节重点

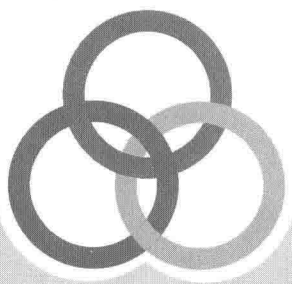
高等职业院校教学改革创新示范教材·软件开发系列
中软国际卓越人才培养系列丛书

JavaEE

架构与程序设计

第2版

王晓华 谢晓东 主 编
陈 永 孔德华 邹金萍 副主编



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

JavaEE 包含一系列的技术, 对于 Web 开发人员来说, 关键是掌握 Web 组件技术、JDBC 编程以及常用框架等。本书主要分为四部分: 第一部分快速学习 Servlet、JSP 组件基本知识, 并通过简单示例演示 MVC 模式的含义及使用; 第二部分详解 Servlet 组件开发, 对 Servlet 相关技术进行深入剖析; 第三部分深入学习 JSP 组件开发; 第四部分学习与 JavaEE 架构设计相关的几个高级主题, 包括 Log4j、Ajax、JSF 框架。本书在第一部分设计了一个案例, 并贯穿始终, 随着学习的深入不断完善案例, 将所学技能直接应用到案例开发中, 做到“学中做, 做中学”。

本书适合各层次Web开发人员阅读。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目(CIP)数据

JavaEE 架构与程序设计 / 王晓华, 谢晓东主编. —2 版. —北京: 电子工业出版社, 2014.12
高等职业院校教学改革创新示范教材·软件开发系列
ISBN 978-7-121-25136-8

I. ①J… II. ①王… ②谢… III. ①JAVA 语言—程序设计—高等职业教育—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2014)第294363号

策划编辑: 程超群

责任编辑: 程超群

印 刷: 三河市华成印务有限公司

装 订: 三河市华成印务有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 15.75 字数: 403 千字

版 次: 2011 年 10 月第 1 版

2014 年 12 月第 2 版

印 次: 2014 年 12 月第 1 次印刷

定 价: 35.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlls@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

序

当我翻阅了“中软国际卓越人才培养系列丛书”后，不禁为这套丛书的立意与创新之处感到欣喜。教育部“卓越工程师教育培养计划”有三个主要特征：一是行业企业深度参与培养过程；二是学校按通用标准和行业标准培养工程人才；三是强化培养学生的工程能力和创新能力。这套丛书紧紧围绕“卓越计划”的要求展开，以企业人才需求为前提，同时又充分考虑了高校教育的特点，能让企业有效参与高校培养过程，是一套为“卓越计划”量身打造的丛书。

丛书的设计理念紧扣中软国际 ETC 的“5R”理念，即真实的企业环境、真实的项目经理、真实的项目案例、真实的工作压力、真实的就业机会，切实地将企业真实需求展现给读者。丛书中的知识点力求精简、准确、实用，显然是编著者经过反复推敲并精心设计的成果。丛书对企业用之甚少的知识点，都进行了弱化，用较少篇幅讲解，而对于企业关注的知识点，都使用非常详尽的内容进行学习。这样的设计对初学者尤其是在校学生非常必要，能够节省很多学习时间，在有限的时间内学习到企业关注的技能，而不是花费很多精力去钻研并不实用的内容。

丛书非常强调“快速入门”这一法宝，能够对某门技术“快速入门”永远是激发学习兴趣的关键。丛书设计了很多“快速入门”章节，使用详尽丰富的图示以及代码示例，保证读者只要根据丛书的指导进行操作，就能够尽快构建出相关技术的实例。

丛书非常注重实际操作，很多知识点都是从提出问题引出，从而在解决这个问题过程中讲解相关的技能。丛书中没有大篇幅的理论描述，尽力用最通俗、最简练的语言讲解每一个问题，而不是“故作高深”地使用很多新名词。

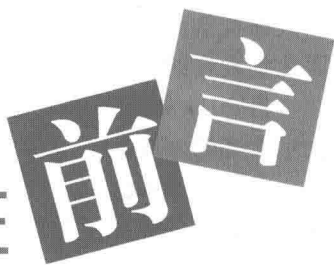
非常值得一提的是，丛书配备了对应的 PPT 讲义，并将 PPT 讲义显示到了相应章节，这种形式令人耳目一新。首先能起到提纲挈领的作用，帮助读者快速了解每个章节的主要内容，掌握完整的知识体系。另外，这种方式非常适合在高校教学中使用，能够完全与教材同步，方便学生课后复习及课前预习，可以有效提高教学效果。

这套丛书是中软国际多年行业经验的积累和沉淀，也是众多编者智慧与汗水的结晶，一定能在校企合作的道路上发挥积极长远的作用。

国家示范性软件学院建设工作办公室副主任
北京交通大学软件学院院长



PREFACE



如果您已经完全掌握 JavaSE (Java 语言标准版) 核心编程技术, 那么就可以胜任桌面应用开发。然而, 目前大多数的企业级 Java 应用都是基于 B/S 结构的。要想使用 Java 技术开发 B/S 结构的应用, 就必须掌握 JavaEE (Java 企业版) 相关技术, 本教材将主要学习与 JavaEE 架构设计相关的内容。

众所周知, JavaEE 包括一系列的技术, 本教材主要关注 JavaEE 开发人员必须掌握的组件和技术, 旨在通过对本教材的学习, 能够快速胜任 Web 应用开发。教材的内容分为四部分, 力求简单明了, 深入浅出地阐述 Web 应用开发。

第一部分: Servlet/JSP 入门。本部分将以最快速度使读者了解 Web 组件, 能够搭建一个简单的 Web 应用, 了解 Web 应用的开发、部署过程, 了解 Servlet/JSP 的基本概念, 掌握 JDBC 编程, 旨在带领读者进入使用 JavaEE 进行 Web 应用开发的大门。

第二部分: 详解 Servlet 组件开发。由于 JSP 的本质就是 Servlet, 所以本教材先从 Servlet 开始学习。第一部分使读者初步了解 Servlet 和 JSP 的基本概念和使用, 第二部分将详细学习 Servlet 组件。本部分将从请求、响应、会话、上下文、监听器、过滤器等各方面学习 Servlet 核心技术, 学习完本部分后, 读者将能够全面掌握 Servlet 组件的开发。

第三部分: 详解 JSP 组件开发。有了 Servlet 的扎实基础后, JSP 的学习就相对容易。本部分将从内置对象、指令、标准动作、JavaBean、标签、EL 等各方面学习 JSP 组件的开发。

第四部分: 高级主题。全面掌握 Servlet/JSP 技术后, 最后一部分学习几个经常使用的高级主题, 包括 Log4j、Ajax 技术以及 JSF 框架。

在本书的第一部分, 设计了一个“教材案例”, 这个案例没有复杂的业务逻辑, 但是在本书中一直使用, 贯穿始终。随着知识点的深入, 将不断完善这个案例, 给案例增加新的功能, 从而使得读者能够边学边做, 边做边学, 一定程度上保证了教材的实用性。书中有关实践的部分, 都有详细的截图以及代码示例, 使得读者能够快速上手, 而不仅仅停留在了解理论的层面。另外, 附录部分提供了企业关注的技能点, 并从企业的角度给予了解析, 能够帮助读者进一步整理书中内容, 掌握企业需要的技能。

本书配套的 PPT 也显示在对应的章节, 这样的设计能够帮助读者快速了解每个章节的主

要内容，起到提纲挈领的作用，也能够帮助读者建立一个完整的知识结构，而不仅仅是掌握了一些凌乱的知识。另外，这样的设计也使得本书非常适合作为高校中 JavaEE 架构与设计相关课程的教材，能够方便教师授课，保证授课思路及内容与教材完全匹配、完全同步，从而达到较好的教学效果。

自 2011 年本书第 1 版面世以来，得到了广大院校师生和培训机构学员的普遍认可，在此表示衷心的感谢！

在此次改版修订过程中，对第 1 版中的一些笔误进行了更正，对一些难以理解的表达进行了修改，并更新了部分软件的版本。为了能够帮助读者更好地理解、实践书中内容，在每章结尾提供了思考与练习题，通过这些题目，不仅能对本章的学习内容进行总结，更能够有效提升学习效果。

本书所有配套的讲义、源代码及视频均可到华信教育资源网（www.hxedu.com.cn）免费下载。

在编写本书的过程中，得到了很多领导、同事以及朋友的帮助。感谢中软国际的所有领导以及 CTO 办公室的所有同事，是他们的帮助、鼓励以及支持才有这本书的问世。感谢电子工业出版社的编辑们，如果没有他们的辛苦工作以及真诚建议，这本书的出版将不会这么顺利。

本书由王晓华和谢晓东担任主编，陈永、孔德华和邹金萍担任副主编，全书由王晓华统稿。

由于编者水平有限，也由于时间仓促，书中一定存在一些不尽如人意的地方，甚至会有一些错误。如果您发现了任何内容方面的问题，烦请一定通知编著（wangxh@chinasofti.com），我们会尽快进行勘误。

编 者

CONTENTS

目录

第一部分 Servlet/JSP 入门

第 1 章	JavaEE Web 开发概述	2	第 3 章	JSP 快速入门	29
1.1	JavaEE 技术概述	2	3.1	什么是 JSP	29
1.2	搭建运行环境——Tomcat	4	3.1.1	JSP 概述	29
1.3	搭建开发环境	6	3.1.2	JSP 运行步骤	30
1.4	MyEclipse 管理 Tomcat	7	3.2	JSP 脚本元素入门	32
1.5	使用 Eclipse 开发 Web 应用	10	3.3	JSP 内置对象入门	34
1.6	本章小结	14	3.4	本章小结	35
1.7	思考与练习	15	3.5	思考与练习	35
第 2 章	Servlet 快速入门	16	第 4 章	JDBC 编程	36
2.1	请求与响应	16	4.1	JDBC 概述	36
2.2	什么是 Servlet	17	4.2	JDBC API 中常用接口和类	37
2.3	第一个 Servlet 程序	18	4.3	使用 JDBC 进行增、删、改	38
2.4	如何访问 Servlet	19	4.4	使用 JDBC 进行查询	41
2.4.1	配置 Servlet 信息	19	4.5	JDBC 的语句对象	43
2.4.2	访问 Servlet 的三种方式	20	4.6	本章小结	44
2.5	web.xml 文件	22	4.7	思考与练习	44
2.6	使用 Eclipse 开发 Servlet	23	第 5 章	MVC 模式	45
2.7	请求与响应	26	5.1	一个简单 Demo	45
2.7.1	请求接口	26	5.2	Demo 中的 MVC 体现	48
2.7.2	响应接口	27	5.3	MVC 模式总结	48
2.8	本章小结	28	5.4	“教材案例”准备	50
2.9	思考与练习	28	5.5	本章小结	55
			5.6	思考与练习	56

第二部分 详解 Servlet 组件开发

第 6 章 Servlet 生命周期.....	58	9.6 会话超时.....	87
6.1 Servlet 实例的特征.....	58	9.7 本章小结.....	87
6.2 Servlet 的“家谱”.....	60	9.8 思考与练习.....	88
6.3 Servlet 的生命周期.....	62	第 10 章 Servlet 上下文.....	89
6.4 本章小结.....	63	10.1 上下文接口.....	89
6.5 思考与练习.....	63	10.2 上下文的使用实例.....	90
第 7 章 请求与响应.....	64	10.3 上下文参数.....	91
7.1 请求接口.....	64	10.4 请求、会话、上下文属性 比较.....	92
7.2 响应接口.....	66	10.5 本章小结.....	93
7.3 请求转发器.....	67	10.6 思考与练习.....	93
7.4 请求属性.....	69	第 11 章 监听器.....	94
7.5 本章小结.....	71	11.1 事件类与监听器接口.....	94
7.6 思考与练习.....	72	11.2 监听器使用实例.....	95
第 8 章 cookie 编程.....	73	11.3 监听器的开发步骤.....	98
8.1 cookie 的概念与使用.....	73	11.4 本章小结.....	99
8.2 cookie 开发实例.....	76	11.5 思考与练习.....	100
8.3 JSESSIONID cookie.....	78	第 12 章 过滤器.....	101
8.4 本章小结.....	79	12.1 过滤器概述.....	101
8.5 思考与练习.....	80	12.2 过滤器有关的 API.....	102
第 9 章 会话.....	81	12.3 过滤器开发实例.....	103
9.1 会话接口.....	81	12.4 过滤器的执行过程.....	105
9.2 会话使用实例.....	82	12.5 过滤器的配置.....	107
9.3 JSP 中的会话对象.....	84	12.6 本章小结.....	109
9.4 会话的实现机制.....	84	12.7 思考与练习.....	109
9.5 URL 重写.....	85		

第三部分 详解 JSP 组件开发

第 13 章 脚本元素与内置对象.....	111	第 14 章 JSP 指令与标准动作.....	117
13.1 JSP 中常用的脚本元素.....	111	14.1 JSP 指令.....	117
13.2 内置对象.....	113	14.2 JSP 标准动作.....	120
13.3 本章小结.....	116	14.3 本章小结.....	121
13.4 思考与练习.....	116	14.4 思考与练习.....	121

第 15 章	JavaBean 编程	122	实例	146	
15.1	JavaBean 概述	122	17.5	本章小结	149
15.2	JavaBean 的使用	123	17.6	思考与练习	149
15.3	JavaBean 的标准动作	125	第 18 章	JSTL	150
15.4	本章小结	128	18.1	JSTL 概述	150
15.5	思考与练习	128	18.2	使用 JSTL	151
第 16 章	EL 语言	129	18.3	常用的 JSTL 标记	153
16.1	EL 语言概述	129	18.4	JSTL 使用实例	154
16.2	EL 的内置对象	130	18.5	本章小结	155
16.3	EL 中的运算符	133	18.6	思考与练习	156
16.4	EL 的其他知识点	135	第 19 章	Web 应用中的异常处理	157
16.5	EL 的使用实例	137	19.1	Model 层抛出异常	157
16.6	本章小结	139	19.2	在 Servlet 中捕获异常	158
16.7	思考与练习	139	19.3	声明 <error-page>	159
第 17 章	JSP 自定义标记	140	19.4	JSP 中使用错误页面	160
17.1	自定义标记的概念	140	19.5	JSP 中捕获异常	161
17.2	如何开发自定义标记	141	19.6	本章小结	161
17.3	如何使用自定义标记	144	19.7	思考与练习	162
17.4	开发与使用自定义标记				

第四部分 高级主题

第 20 章	日志处理	164	21.4.2	XML DOM	195
20.1	Log4j 概述	164	21.5	使用 Ajax 进行异步校验	198
20.2	日志记录器 Logger	166	21.6	本章小结	201
20.3	输出目的地 Appender	169	21.7	思考与练习	202
20.4	格式化器 Layout	171	第 22 章	JSF 框架	203
20.5	Log4j 的配置	175	22.1	JSF 框架快速入门	203
20.6	在 Web 应用中使用 Log4j	181	22.2	UI 标准组件	208
20.7	本章小结	183	22.2.1	显示数据的组件	209
20.8	思考与练习	183	22.2.2	面板组件	210
第 21 章	Ajax 编程	184	22.2.3	输入组件	211
21.1	Ajax 概述	184	22.2.4	命令组件	214
21.2	JavaScript 语言	185	22.3	国际化	215
21.3	XMLHttpRequest 对象	189	22.4	输入校验	219
21.4	文档对象模型 DOM	192	22.4.1	标准校验器	219
21.4.1	HTML DOM	193	22.4.2	自定义校验器	221

22.5 本章小结	223	第一部分 Servlet/JSP 快速入门	227
22.6 思考与练习	223	第二部分 详解 Servlet 组件开发	230
附录 A web.xml 文件	224	第三部分 详解 JSP 组件开发	235
附录 B 企业关注的技能	227	第四部分 高级主题	239

Servlet/JSP 入门

万事开头难。任何技术学习，入门都是非常关键的一步，第一部分的关键在于“入门”。本部分从搭建 Web 应用的开发运行环境开始，逐步掌握 Web 应用的基本开发步骤。然后对 Servlet 和 JSP 进行快速入门，了解 Servlet 和 JSP 的概念、优点、开发过程等。构建 Web 应用，首先需要建立正确的架构思想，MVC 模式是 Web 应用中常用的一种架构模式。本章通过一个简单 Demo，学习 MVC 模式的概念，帮助读者尽早建立正确的 MVC 思想，理解 Servlet 和 JSP 在开发中的实际作用。另外，本部分设计了一个案例，并实现了案例的业务逻辑和页面部分。该案例将贯穿于教材的后续章节中，并不断完善，以辅助读者学习相关知识点。

JavaEE Web 开发概述

Servlet 和 JSP 是 JavaEE 技术的一部分，本章首先了解 JavaEE 中主要技术的基本含义，并介绍典型的 JavaEE 应用架构。工欲善其事，必先利其器。本章将逐步介绍如何搭建 Servlet/JSP 的运行及开发环境，并展示如何创建、部署、测试一个 Web 应用，使得读者能快速搭建必需的环境，为后续学习做好准备。

1.1 JavaEE 技术概述



Java 平台有三个版本，包括微型版 JavaME (Java Platform, Micro Edition)，用来开发适用于小型设备和智能卡的应用；标准版 JavaSE (Java Platform, Standard Edition)，用来开发桌面应用系统；企业版 JavaEE (Java Platform, Enterprise Edition)，用来创建企业级应用。JavaEE 版本的基础是 JavaSE 版本，也就是说，要想使用 Java 技术构建企业级应用，不仅仅需要掌握必要的 JavaEE 技术，精通 JavaSE 更是必要的前提。

JavaEE 是由一系列的技术和服务组成的，通常来说有 13 种。具体包括：

- (1) JDBC (Java Database Connectivity)：用来访问数据库的 API。
- (2) Java Servlet：是一种小型的 Java 程序，扩展了 Web 服务器的功能。
- (3) JSP (Java Server Pages)：JSP 页面由 HTML 代码和嵌入其中的 Java 代码组成，用来实现动态视图。
- (4) JNDI (Java Name and Directory Interface)：JNDI API 被用于访问名字和目录服务。
- (5) EJB (Enterprise JavaBean)：实现业务逻辑的组件，可以构建分布式系统。
- (6) RMI (Remote Method Invoke)：调用远程对象方法。

- (7) Java IDL/CORBA: 将 Java 和 CORBA 集成在一起。
- (8) XML (Extensible Markup Language): 可以用来定义其他标记语言的语言。
- (9) JMS (Java Message Service): 用于和消息中间件相互通信的 API。
- (10) JTA (Java Transaction Architecture): 一种标准的 API, 可以访问各种事务管理器。
- (11) JTS (Java Transaction Service): 是 CORBA OTS 事务监控的基本实现。
- (12) JavaMail: 用于存取邮件服务器的 API。
- (13) JAF (JavaBeans Activation Framework): JavaMail 利用 JAF 来处理 MIME 编码的邮件附件。

使用 JavaEE 技术构建企业应用, 往往会采用如图 1-1 所示的架构。

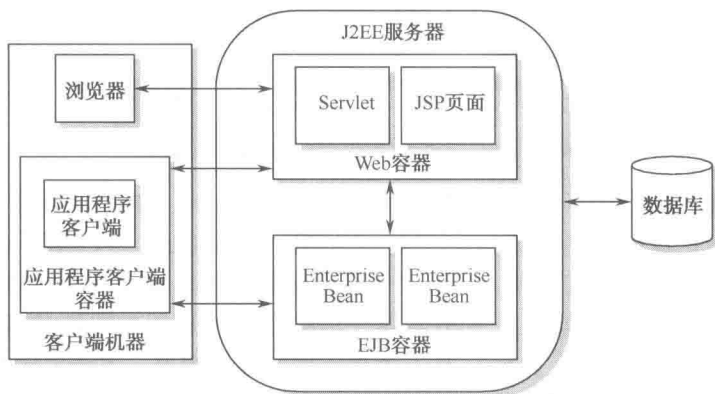


图 1-1 常用的 JavaEE 应用架构

常用的 JavaEE 应用架构中, 客户端可能是浏览器, 也可能是应用程序客户端。客户端的请求发送给 JavaEE 服务器, 由服务器调用客户端请求的组件, 如 Servlet、JSP、EJB 等。组件可以通过 JDBC 连接到数据库, 进行数据永久存储操作, 也可以通过其他服务访问不同的资源, 如通过 JMS 访问消息服务器, JNDI 访问名字和目录服务器等。

目前, 大多数的企业应用都是 B/S 结构, 即客户端是浏览器。从图 1-1 中可见, 浏览器只能直接访问 Web 容器中的组件, 如 Servlet 和 JSP。也就是说, 如果要使用 JavaEE 核心技术开发 B/S 结构的 Web 应用, 那么至少要使用 Servlet 和 JSP 技术, 因为浏览器无法直接访问 JavaEE 服务器中的其他组件。本教材将重点学习 Servlet 和 JSP 技术, 往往二者被统称为 JavaEE Web 组件开发技术。熟练掌握 Servlet 和 JSP 技术, 是学习 MVC 框架的必要基础。值得一提的是, 目前很多企业 Web 应用都会使用一些开源框架, 例如 Struts、Hibernate、Spring 等, 然而, 对于 JavaEE 初学者来说, 先了解最核心的 JavaEE 技术及架构是必要的学习途径。



J2EE 和 JavaEE 有什么区别? 在 JDK5.0 发布以前, Java 的企业版本称为 J2EE; JDK5.0 发布后, Java 的企业版本则称为 JavaEE, 对应其他版本为 JavaSE、JavaME。J2EE1.4 和 JavaEE5.0 是两个较流行的版本, 较新的版本是 JavaEE6.0。

1.2 搭建运行环境——Tomcat



Servlet 和 JSP 被称为 JavaEE Web 组件，使用 Servlet 和 JSP 可以开发 Web 应用。组件 (component) 的一个最明显特征就是必须运行在容器 (container) 中，容器可以理解为组件的运行环境，往往采用软件形式实现。Web 组件的运行环境称为 Web 容器 (Web container)。开发 Servlet 和 JSP 后，必须先安装好 Web 容器，然后将 Web 应用部署到容器中才能运行。

Web 容器有很多种，其中 Tomcat 是一个最常用的免费容器，本教材中将以 Tomcat 作为 Web 应用的运行容器。首先下载 Tomcat 安装文件 (<http://tomcat.apache.org/>)，Tomcat 安装文件往往有两种版本：一种是不需要安装的，直接解压后即可使用；另外一种是需要安装的本。本教材中使用第一种版本的 Tomcat。下载 Tomcat 安装文件后，解压到某目录下，该目录建议不要有中文，不要有空格。解压后的目录结构如图 1-2 所示。

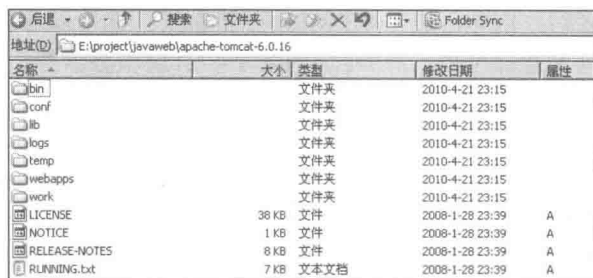


图 1-2 Tomcat 解压后的目录结构

安装 Tomcat 后，要成功启动 Tomcat 才能提供服务。启动 Tomcat 前，首先需要在环境变量中配置名字为 java_home 的环境变量，其值为 JDK 的安装目录，如图 1-3 所示。

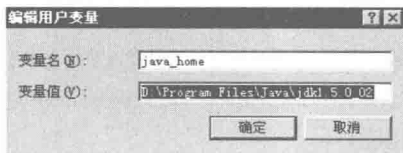


图 1-3 设置 java_home 的环境变量

接下来,运行 cmd 命令,转到 Tomcat 目录的 bin 目录下,运行 startup.bat 即可启动 Tomcat,如图 1-4 所示。

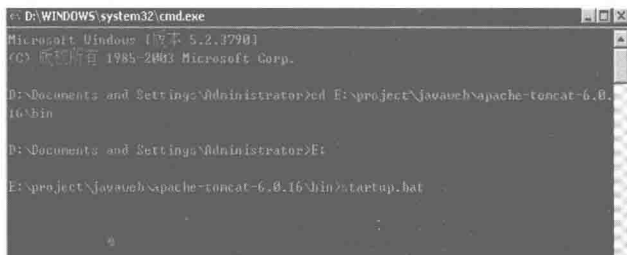


图 1-4 启动 Tomcat

启动 Tomcat 后,将在控制台打印输出启动信息,如果启动出错,通过查看启动信息可以获悉出错原因,如图 1-5 所示。

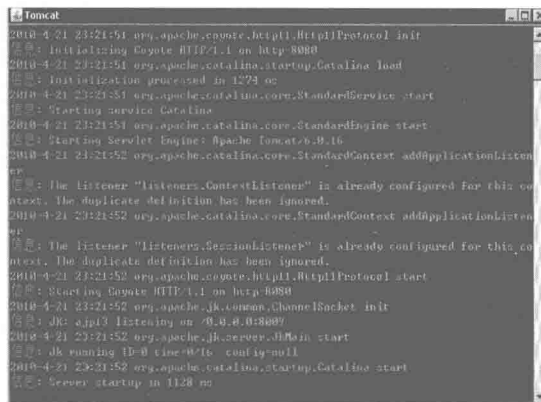


图 1-5 启动信息

当启动信息中提示成功启动后, Tomcat 就已经正常启动,默认端口为 8080。可以通过浏览器访问 Tomcat 的 8080 端口服务,显示界面如图 1-6 所示。

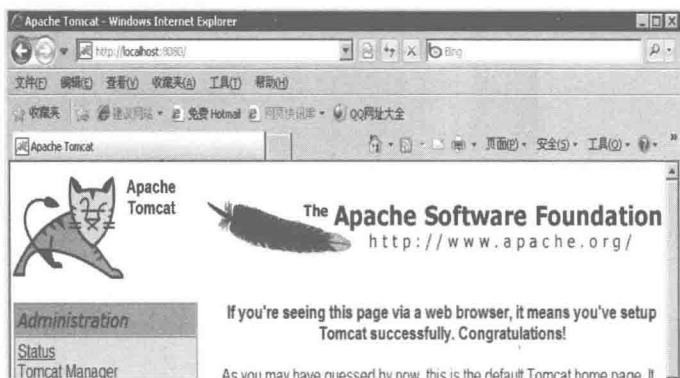
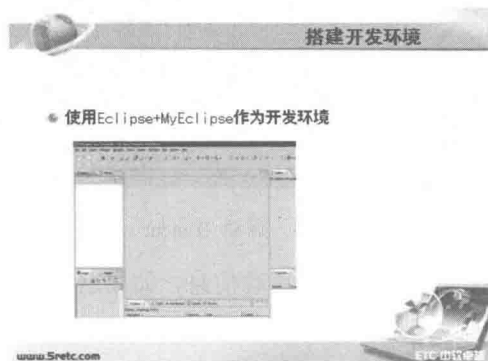


图 1-6 Tomcat 首页面

只要在浏览器中显示了如图 1-6 所示的 Tomcat 首页面,就证明 Tomcat 已经成功启动,即已经为 Web 组件提供了一个可用的运行环境。

1.3 搭建开发环境



通过上节学习，已经可以成功搭建 Servlet/JSP 的运行环境，本节将学习如何搭建开发环境。只要有文本编辑器，即使是 notepad（记事本）这样简单的编辑器，都可以进行 Web 应用开发。然而为了提高开发效率，真实的企业开发过程中，一定会使用某种集成开发环境（IDE）。

JavaEE 的集成开发环境有很多，本教材使用 IBM 公司提供的免费 Eclipse 平台。Eclipse 是目前较流行的免费 JavaEE 开发平台，使用 Eclipse 开发 JavaEE 应用前，往往需要下载或者自主开发相应插件。MyEclipse 是一款收费的 Eclipse 插件，集成了 JavaEE 开发所需的大部分插件。Eclipse 不需要安装，解压缩后即可使用，购买 MyEclipse 插件后，进行默认安装即可。本教材中使用 MyEclipse 作为 Eclipse 的 JavaEE 开发插件，进行 JavaEE 应用开发。

安装结束后，可以启动 Eclipse。启动过程中，会提示选择工作空间（work space）。工作空间是使用 Eclipse 开发的工程源文件存放的位置，可以使用默认位置，也可以自行创建。启动后界面如图 1-7 所示。

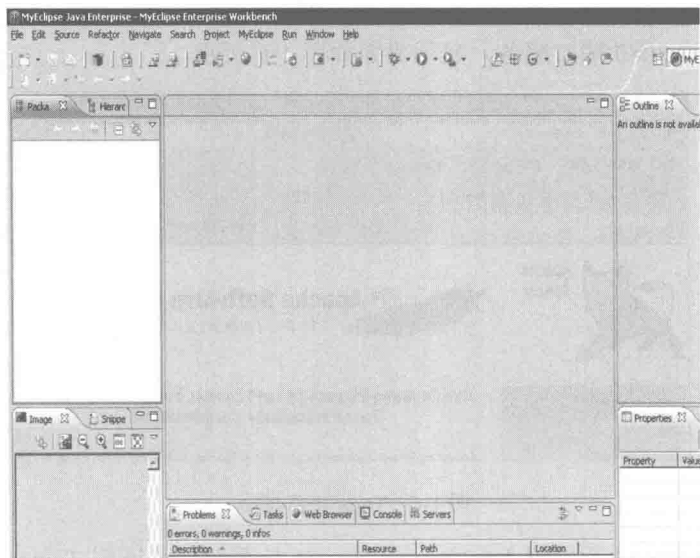


图 1-7 Eclipse+MyEclipse 工作界面

到此为止，已经成功搭建了运行环境 Tomcat，并使用字符界面成功启动，同时也成功安装了 Eclipse 平台以及 MyEclipse 插件，搭建了常用的开发环境。

1.4 MyEclipse 管理 Tomcat



在 1.2 节中，已经演示了如何使用字符界面自行启动 Tomcat。然而，安装好 MyEclipse 后，由于 MyEclipse 中集成了管理 Tomcat 的插件，所以可以使用 MyEclipse 启动或关闭 Tomcat。这种方式比起 1.2 节学习的字符方式更为简单直观，本节将学习如何使用 MyEclipse 管理 Tomcat。

(1) 单击服务器图标，选择 Configure Server 菜单项，如图 1-8 所示。

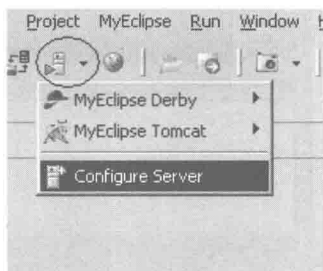


图 1-8 选择 configure server 菜单项

(2) 在弹出的窗口中配置 Tomcat 具体信息，如图 1-9 所示，主要配置 Tomcat 的安装根目录，并选择“Enable”选项。

(3) 选择 Tomcat 下的 JDK 选项，单击“Add”按钮，如图 1-10 所示。

(4) 在弹出的窗口中浏览选择 JDK 的根目录，如图 1-11 所示。

(5) 应用所有配置，选择服务器图标，启动 Tomcat，如图 1-12 所示。

(6) 在 Console 视图中，可以看到启动信息，如图 1-13 所示。