

从基础
到实践

基础知识全面覆盖
实践操作循序渐进

从必需
到拓展

理论讲解详尽具体
动手应用实操实练

从入门
到进阶

内容编排由浅入深
进阶案例综合拓展

重点
推荐

Java EE

编程技术 (第2版)

■ 温立辉◎主编



 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

Java EE 编程技术

(第2版)

主 编 温立辉

副主编 冯昭强 练敏灵 巫锦润

 **北京理工大学出版社**
BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

本教材共分 11 章，以 Java EE 技术体系为中心贯穿各章节，内容包括 Java EE 入门导论、Struts 框架应用、Spring 框架应用、Hibernate 框架应用、版本管理工具应用、日志组件应用、单元测试技术、Java Web 集成开发工具（IDE）、Web 服务器配置与应用、UML 统一建模语言、设计模式等。

本教材有配套的电子课件、章节源码、相关插件、工具包等，可在出版社网站下载使用。

本书适合计算机相关专业作为 Java Web 开发方向的专业课程教材，也可供技术人员阅读和参考。

版权专有 侵权必究

图书在版编目（CIP）数据

Java EE 编程技术 / 温立辉主编. —2 版. —北京：北京理工大学出版社，2021. 6
ISBN 978 - 7 - 5682 - 9446 - 1

I. ①J… II. ①温… III. ①JAVA 语言 - 程序设计 IV. ①TP312. 8

中国版本图书馆 CIP 数据核字（2021）第 005048 号

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775 (总编室)

(010) 82562903 (教材售后服务热线)

(010) 68948351 (其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 河北盛世彩捷印刷有限公司

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 15.5

字 数 / 358 千字

版 次 / 2021 年 6 月第 2 版 2021 年 6 月第 1 次印刷

定 价 / 65.00 元

责任编辑 / 王玲玲

文案编辑 / 王玲玲

责任校对 / 刘亚男

责任印制 / 李志强

图书出现印装质量问题，请拨打售后服务热线，本社负责调换

前 言

软件工程是一门应用性非常强的学科，在人才培养过程中，应侧重培养学生的实践技能，使其掌握主流的软件应用技术。实践教学内容要符合培养高素质、国际化、工程化、创新性人才的目标要求，突出工程型人才培养模式的特点，在实际工程项目中培养学生，加强科学研究和软件产品开发与应用，适应软件产业发展的需要。

本教材在第一版基础上进行修订，本次修订主要是针对企业、行业的最新岗位能力需求与教材中的知识、技能目标所存在的偏差，做相关的完善与补充。具体的修订内容主要体现在以下三大方面：①增加了重要技能模块：UML 建模技术、Web 服务器高级应用、IDE 集成开发工具企业级应用。②为每章增加了章节习题，增加课程趣味性，丰富学生的学习元素，增加教师授课的题材。③针对原教材使用过程中专业教师及学生反馈的问题做进一步的完善与处理。本次对原教材做了较大幅度的修订，使教材既体现岗位技能目标，又体现时代性。

本教材在修订过程中，着重突出职业教育的新理念，以就业为导向，以学生为主体，以能力为本位，积极倡导“做中学、学中做”的职业教育方式，注重学生的职业发展和职业素养的培养，教学内容积极与职业标准结合，反映产业升级和技术进步，以“技能必需和知识够用”为原则，体现信息技术产业特点，服务当代产业经济。

本教材修订后共 11 章，以 Java EE 技术体系为中心贯穿各章节。第 1 章为 Java EE 相关编程技术的入门导论，总述开发环境的搭建与 Java EE 领域的基础技术。第 2 章为 Struts 框架的应用，讲述 Struts2 框架的基本使用方法。第 3 章为 Spring 框架的应用，重点讲述 IoC 的依赖注入。第 4 章为 Hibernate 框架的应用，讲述通过 ORM 操纵数据库的过程。第 5 章为 CVS 版本管理工具的使用，讲述团队开发过程中版本管理的过程与步骤。第 6 章为日志组件的应用，介绍 Log4j 插件的使用方法。第 7 章为单元测试技术，介绍 JUnit 技术的使用。第 8 章为 Java Web 集成开发工具，介绍 Eclipse 如何集成插件，进行 Web 应用开发。第 9 章为 Web 服务器配置与应用，介绍 Tomcat 的核心配置及高级功能应用，如内存管理、多节点配置、远程项目部署等。第 10 章为 UML 统一建模语言，介绍常用的 UML 建模图形及在实际开发中的应用。第 11 章为设计模式，介绍常用的几种设计模式的用法。

针对高职学生动手能力较弱的问题，本教材在任务项目上设置较大的程序代码编写量，以求强化学生的编码能力；针对高职学生的知识体系及抽象能力较弱的特点，教材在表现形式上力求直观和新颖，图文并茂，绝大部分学生只要按照案例操作步骤，均可独立完成，易于学生的自我学习，以保证实操项目的质量与效果；同时，在语言文字上力求形象化、具体化、通俗易懂，以求达到与高职学生的抽象思维能力相匹配的目的。

本教材有配套的电子课件、章节源码、相关的插件、工具包等，所有资源均可在 <http://www.bitpress.com.cn> 下载。在本教材编写过程中，得到了编者所在学院领导和同事的

帮助，他们提出了许多宝贵意见和建议，在此表示衷心的感谢。

由于时间仓促和编者水平有限，书中难免存在不妥或疏漏之处，敬请广大读者和专家不吝赐教。作者 E-mail: wenlihui2004@163.com。

编者

目 录

第 1 章 Java EE 入门导论	1
1.1 Java EE 概述	1
1.1.1 Java EE 核心技术	1
1.1.2 Java EE 体系结构	2
1.2 Servlet 应用	3
1.2.1 Servlet 生命周期	3
1.2.2 自定义一个 Servlet	4
1.2.3 Servlet 案例开发	5
1.3 Filter 技术*	16
1.3.1 Filter 的结构	16
1.3.2 在 Web 应用中添加 Filter	16
本章练习	18
第 2 章 Struts 框架应用	21
2.1 Struts 框架概述	21
2.1.1 Struts 框架的起源	21
2.1.2 MVC 模式	22
2.2 Struts2 应用框架	23
2.2.1 Struts2 框架组件	23
2.2.2 Struts2 框架流程	25
2.2.3 Struts2 框架案例开发	25
本章练习	36
第 3 章 Spring 框架应用	38
3.1 Spring 框架概述	38
3.1.1 Spring 框架的作用	38
3.1.2 相关术语	38
3.2 IoC 模型	39
3.2.1 为什么要使用 IoC 模型	39
3.2.2 IoC 运行时加载及相关组件	41
3.2.3 运行时加载案例开发	43
3.2.4 IoC 容器启动加载及相关组件	48
3.2.5 容器启动时加载 <bean> 案例开发	48
3.3 AOP 模型*	56
3.3.1 AOP 五大装备	56

3.3.2 AOP 案例开发	57
本章练习	60
第4章 Hibernate 框架应用	62
4.1 Hibernate 概述	62
4.1.1 认识 Hibernate	62
4.1.2 对象/关系映射	62
4.1.3 ORM 技术规则	63
4.2 Hibernate 组件及其应用	64
4.2.1 为什么使用 Hibernate	64
4.2.2 Hibernate 框架组件	65
4.2.3 持久化过程	67
4.2.4 案例开发	67
4.3 ORM 工具的高级运用	77
本章练习	91
第5章 版本管理工具应用	93
5.1 版本管理工具概述	93
5.1.1 认识版本管理工具	93
5.1.2 常用的版本管理工具	93
5.2 CVS 的配置与使用	96
5.2.1 安装 CVS 服务器端	96
5.2.2 配置 CVS 管理工具	103
5.2.3 案例应用	109
本章练习	118
第6章 日志组件应用	120
6.1 Log4j 概述	120
6.2 Log4j 应用配置	121
6.2.1 配置文件	121
6.2.2 案例应用	123
本章练习	128
第7章 单元测试技术	130
7.1 单元测试概述	130
7.1.1 认识单元测试	130
7.1.2 为什么要使用单元测试	131
7.2 JUnit 技术	132
7.2.1 JUnit 测试框架	132
7.2.2 Testcase 案例应用	133
7.2.3 测试套件 TestSuite 类	141
7.2.4 TestSuite 案例应用	141
本章练习	145

第 8 章 Java Web 集成开发工具	147
8.1 集成开发工具概述	147
8.1.1 IDE 的作用	147
8.1.2 常见的 IDE 工具	148
8.2 Eclipse 开发工具的使用	149
8.2.1 Eclipse 搭建 Web 工程	149
8.2.2 Eclipse 集成 Tomcat	152
8.2.3 Eclipse 打包部署项目	157
本章练习	160
第 9 章 Web 服务器配置与应用	162
9.1 Web 服务器概述	162
9.1.1 Web 服务器原理	162
9.1.2 常见的 Web 服务器	163
9.2 Tomcat 开发环境集成	164
9.2.1 Tomcat 服务器 IDE 的集成	164
9.2.2 Tomcat 资源目录结构	171
9.3 Tomcat 基本服务配置	173
9.3.1 多节点配置	173
9.3.2 ROOT 应用配置	176
9.3.3 服务器路径以外的应用部署	177
9.4 Tomcat 远程项目管理	179
9.4.1 用户管理	179
9.4.2 远程项目部署与运维管理	181
9.5 Tomcat 内存管理	186
9.5.1 内存管理入门	186
9.5.2 配置管理操作	188
本章练习	190
第 10 章 UML 统一建模语言	194
10.1 统一建模语言入门	194
10.1.1 UML 模型概述	194
10.1.2 UML 模型事物	195
10.1.3 UML 模型关系	196
10.2 静态建模视图	199
10.2.1 用例图	199
10.2.2 类图	201
10.2.3 对象图	202
10.2.4 构件图	202
10.2.5 部署图	203
10.3 动态建模视图	204

10.3.1	时序图	204
10.3.2	协作图	206
10.3.3	状态图	206
10.3.4	活动图	207
10.4	Rational Rose 建模工具	209
10.4.1	建模视图介绍	210
10.4.2	各类图形建模设计	211
	本章练习	217
第11章	设计模式*	219
11.1	设计模式概述	219
11.1.1	设计模式的几大原则	219
11.1.2	设计模式和框架	220
11.2	单例模式	220
11.2.1	单例模式的基本要求	221
11.2.2	两种形式的单例模式	222
11.3	工厂模式	225
11.3.1	简单工厂	225
11.3.2	工厂方法	228
11.3.3	抽象工厂	229
11.4	外观模式	231
11.4.1	认识外观模式	231
11.4.2	外观模式结构	231
11.5	模板方法	234
11.5.1	认识模板方法	234
11.5.2	模板方法的结构	235
	本章练习	238
	参考文献	240

第 1 章

Java EE 入门导论

能力目标

- ◇ 全面认识、了解 Java EE 体系结构。
- ◇ 掌握开发环境搭建过程，并能熟练使用各种工具。
- ◇ 掌握 Tomcat 服务器的配置与使用。
- ◇ 掌握 Servlet 组件应用。
- ◇ 认识 Filter 技术。
- ◇ 认识 JNDI 技术。



Java EE 历史
背景概述

1.1 Java EE 概述

Java EE (Java Platform, Enterprise Edition) 是 Sun 公司推出的企业级应用程序版本。这个版本以前称为 J2EE，其能够帮助我们开发和部署可移植、健壮、可伸缩且安全的服务器端 Java 应用程序。Java EE 是在 Java SE 的基础上构建的，它提供 Web 服务、组件模型、管理和通信 API，可以用来实现企业级的面向服务体系结构 (Service - Oriented Architecture, SOA) 和 Web 2.0 应用程序。

1.1.1 Java EE 核心技术

Java EE 平台由一整套服务 (Service)、应用程序接口 (API) 和协议构成，对 Web 的多层应用提供支持，其包含的核心技术有十多种，其中较为基础的有如下几种：

① JDBC (Java Database Connectivity)，提供连接各种关系数据库的统一接口，可以为多种关系数据库提供统一访问，像 ODBC 一样，JDBC 对开发者屏蔽了一些实现细节，其对数据库的访问具有平台无关性。

② JavaBeans，一个开放的标准组件体系结构，它独立于平台，但使用 Java 语言。一个 JavaBean 是一个满足 JavaBeans 规范的 Java 类，通常定义了一个现实世界的事物或概念。

③ Java Servlet，一种小型的 Java 程序，它扩展了 Web 服务器的功能，作为一种服务器端的应用，重在逻辑控制。

④ JSP (Java Server Page)，服务器在页面被客户端被客户请求后，对 Java 代码进行处理，然后将生成的 HTML 页面返回给客户端的浏览器。

⑤ XML (Extensible Markup Language)，可扩展标记语言，一种用来定义其他标志语言的语言，它被用来在不同的商务过程中共享数据。

⑥ EJB (Enterprise JavaBeans)，使得开发者方便地创建、部署和管理跨平台的基于组件的企业应用。

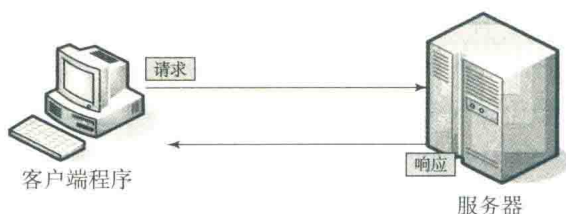
⑦ RMI (Remote Method Invocation), 远程方法调用, 用来开发分布式 Java 应用程序。一个 Java 对象的方法能被远程 Java 虚拟机调用。

⑧ JNDI (Java Naming and Directory Interface), 提供从 Java 平台到企业级资源的统一的无缝的连接。这个接口屏蔽了企业网络所使用的各种命名和目录服务。

⑨ JMS (Java Message Service), 提供企业消息服务, 如可靠的消息队列、发布和订阅通信, 以及有关推/拉 (Push/Pull) 技术的各个方面。

1.1.2 Java EE 体系结构

设计 Java EE 的初衷就是解决两层模式 (Client/Server) 的弊端。在传统的 C/S 模型 (图 1-1) 中, 客户端担当了过多的角色而显得臃肿。在这种模式下, 第一次部署的时候比较容易, 但难以升级或改进, 可伸展性也不理想, 这使重用业务逻辑和界面逻辑变得非常困难。



C/S 及 B/S 系统
应用模式

图 1-1 C/S 模型结构

与 C/S 模式相对应, B/S (Browser/Server) 模式 (图 1-2) 的企业级应用模型通常分为三层: 客户层、服务器层、数据层, 每个层能够扮演不同的角色、提供不同的服务。在这种结构下, 客户层是通过 IE 浏览器来实现的, 用户可以通过 WWW 浏览器去访问 Web 服务器层, 而每一个 Web 服务器又可以与数据库服务器层连接, 大量的数据实际存放在数据库服务器中。

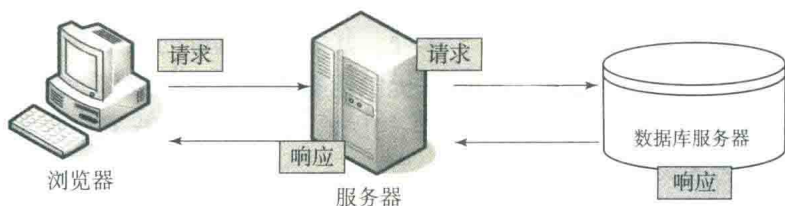


图 1-2 B/S 模型结构

与 C/S 结构相比, 在 B/S 模式下, 用户可以在任何有互联网的地方直接用浏览器与 Web 服务器交互, 使用非常方便; 客户端仅仅是浏览器, 非常简单, 非常有利于日后系统的升级与维护。但 B/S 模式架构的系统也在很大程度上加重了服务的负载, 客户端能实现的功能相对简单, 在某些必要的情况下还是要用 C/S 模式去开发应用程序, 如大型的网络游戏。

Java EE 使用多层的分布式应用模型 (图 1-3), 应用逻辑按功能划分为组件, 各个应用组件根据它们所在的层分布在不同的机器上。通常分为三层或四层, 其中, 客户层: 运行在客户计算机上的组件; 控制层与业务逻辑层: 运行在 Java EE 服务器上的组件; 企业信息系统层 (EIS): 运行在 EIS 服务器上的信息系统。

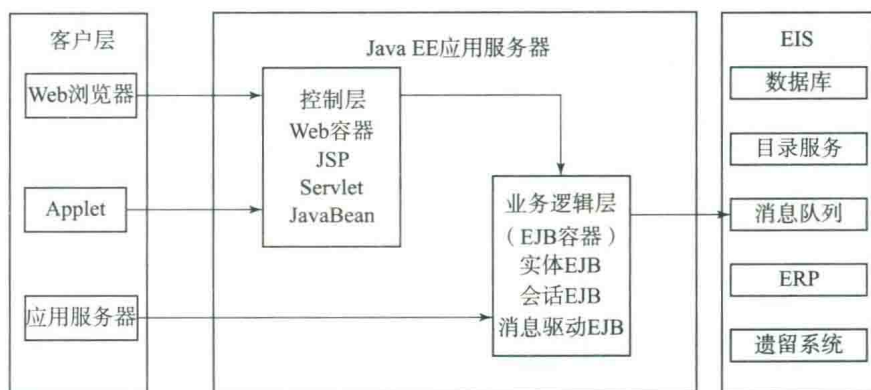


图 1-3 Java EE 体系结构

客户层的组件可以是基于 Web 方式的，也可以是基于传统方式（一个应用程序客户端）的。控制层也叫表示层，其组件可以是 Servlet、JavaBean 等元素；业务逻辑层组件由企业级 JavaBean 来担当，有三种类型，分别为会话（Session）Beans、实体（Entity）Beans 和消息驱动（Message-driven）Beans。企业信息系统层处理企业信息系统软件，包括企业基础建设系统，如企业资源计划（ERP）、大型机事务处理、数据库系统和其他遗留系统。业务层组件为了连接数据库，可能需要访问企业信息系统。

1.2 Servlet 应用

Servlet 是一种服务器端的 Java 应用程序，其主要功能是交互式地浏览和修改数据，生成动态 Web 内容。它担当客户请求层与数据响应层的中间层，即控制层，可担当控制器的角色。Servlet 位于 Web 服务器内部的服务器端，与传统的从命令行启动的 Java 应用程序不同，Servlet 由 Web 服务器进行加载，该 Web 服务器必须包含支持 Servlet 的 Java 虚拟机。

1.2.1 Servlet 生命周期

当客户机发送请求至服务器时，服务器可以将请求信息发送给 Servlet，并让 Servlet 根据客户端请求生成响应内容并将其传给服务器，服务器将响应返回客户端。

Servlet 的生命周期始于将它装入 Web 服务器的内存之时，并在服务器终止或重新装入 Servlet 时结束，其生命周期主要分为三个阶段：

(1) 初始化

在下列时刻装入 Servlet：

- ① 如果已配置自动装入选项，则在启动服务器时自动装入；
- ② 在服务器启动后，客户机首次向 Servlet 发出请求时；
- ③ 重新装入 Servlet 时，装入 Servlet 后，服务器创建一个 Servlet 实例并且调用 Servlet 的 `init()` 方法。

(2) 请求处理

对于到达服务器的客户机请求，服务器创建特定于请求的一个“请求”对象和一个“响应”对象。服务器调用 Servlet 的 `service()` 方法，该方法用于传递“请求”和“响应”

对象。service()方法从“请求”对象获得请求信息、处理该请求并用“响应”对象的方法以将响应传回客户机。service()方法可以调用其他方法来处理请求，例如 doGet()、doPost()或其他方法。

(3) 终止

当服务器不再需要 Servlet，或重新装入 Servlet 的新实例时，服务器会调用 Servlet 的 destroy()方法。

1.2.2 自定义一个 Servlet

如果自己需要定义一个 Servlet (图 1-4)，一般来说，必须要实现以下五步：

```

import java.io.IOException;

public class HelloServlet extends HttpServlet {

    public HelloServlet() {
        super();
    }

    //初始化时调用此方法
    public void init() throws ServletException {
    }

    //销毁时调用此方法
    public void destroy() {
        super.destroy();
    }

    //客户端以get方式发送请求时，调用此方法
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
    }

    //客户端以post方式发送请求时，调用此方法
    public void doPost(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
    }
}
    
```



JSP 运行原理

图 1-4 Servlet 类结构

- ① 自定义类必须继承 javax.servlet.http.HttpServlet 类。
- ② 自定义类中要覆盖父类中的 init()方法，此方法在初始化时调用。
- ③ 自定义类中要覆盖父类中的 destroy()方法，此方法在销毁时调用。
- ④ 自定义类中要覆盖父类中的 doGet()方法，客户端以 get 方式发送服务器请求时，调用此方法。此方法带有两个参数，分别为 HttpServletRequest 与 HttpServletResponse。
- ⑤ 自定义类中要覆盖父类中的 doPost()方法，客户端以 post 方式发送服务器请求时，调用此方法，此方法的参数与 doGet()方法的相同。

1.2.3 Servlet 案例开发

使用 MyEclipse 集成工具与 Tomcat 服务器开发一个 JSP + Servlet + JavaBean 的入门案例，以通过实践来加深对 Servlet 组件的认识与理解。

应用的题材为：假如今年是十二生肖中的猪年，输入你的年龄，则应用能算出你属于哪个生肖（即你所出生的年份所属生肖）。

实现过程如图 1-5 所示。

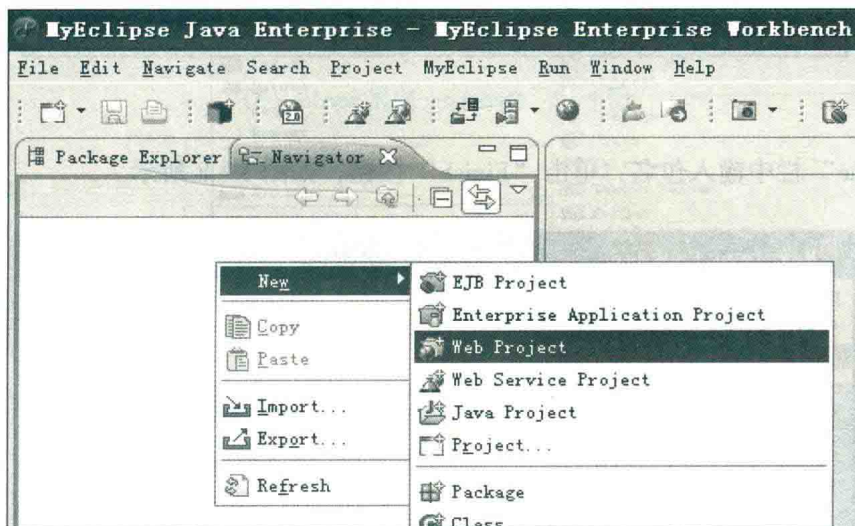


图 1-5 MyEclipse 构建 Servlet (1)

打开 MyEclipse 集成工具，创建一个 Web Project，如图 1-6 所示。

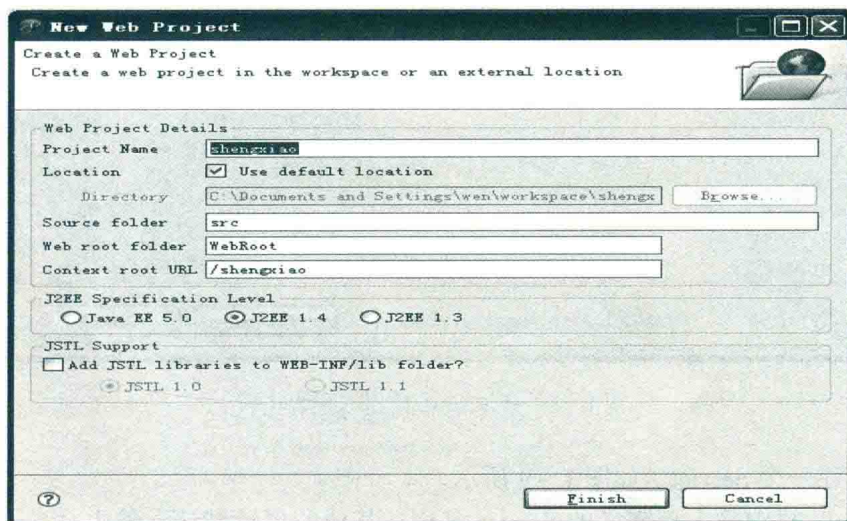


图 1-6 MyEclipse 构建 Servlet (2)

只需在“Project Name”中输入应用的名称即可，其他属性默认，单击“Finish”按钮。在应用中创建一个 Package 包，如图 1-7 所示。



Java EE 技术
体系



Java EE 核心
技术

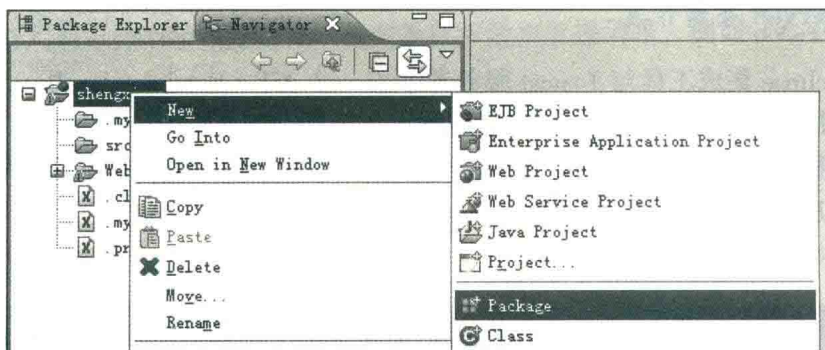


图 1-7 MyEclipse 构建 Servlet (3)

在“Name”栏中输入包名，单击“Finish”按钮，如图 1-8 所示。

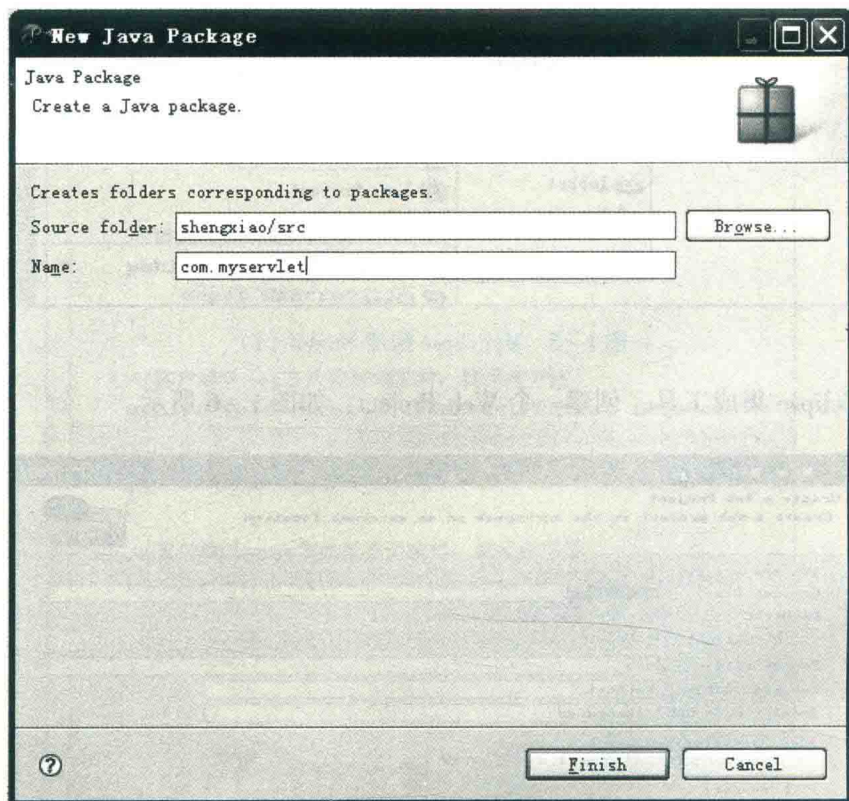


图 1-8 MyEclipse 构建 Servlet (4)

在包中创建一个 Servlet，如图 1-9 所示。

在“Name”栏中输入自定义的 Servlet 名称，其他的属性默认，单击“Next”按钮，如图 1-10 所示。

在“Servlet/JSP Mapping URL”栏中将值修改为“*.s”，其他属性默认，如图 1-11 所示。



EJB 组件

敏捷开发
框架

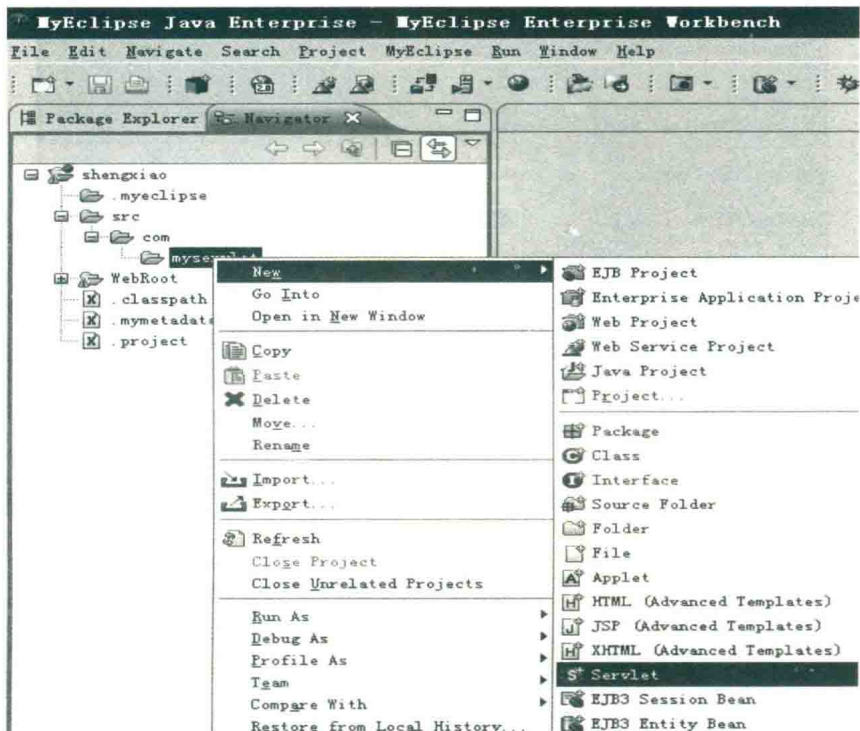


图 1-9 MyEclipse 构建 Servlet (5)

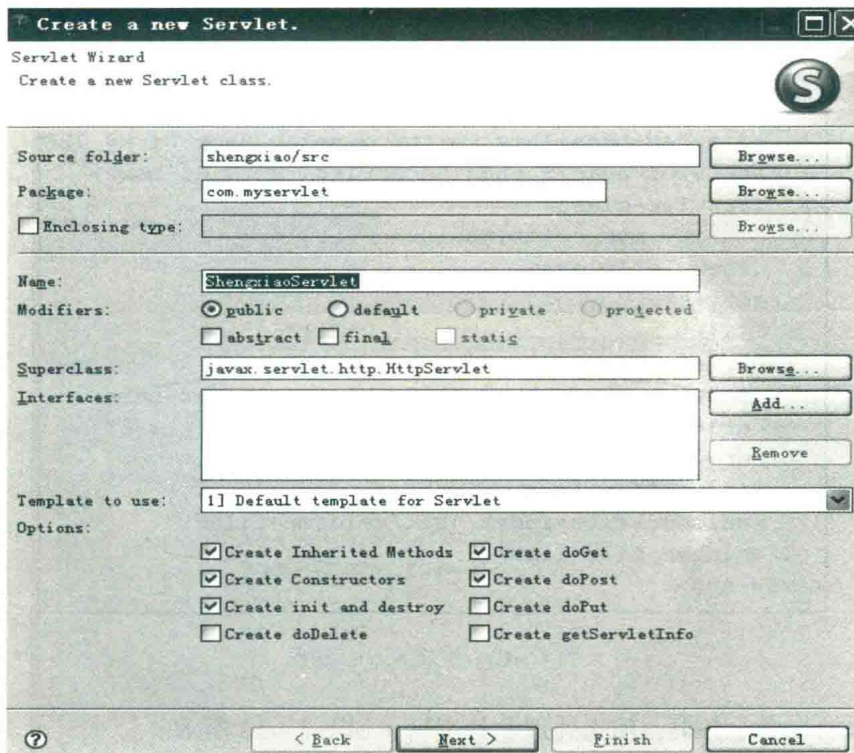


图 1-10 MyEclipse 构建 Servlet (6)

<input checked="" type="checkbox"/>	Generate/Map web.xml file
Servlet/JSP Class Name:	com.myServlet.ShengxiaoServlet
Servlet/JSP Name:	ShengxiaoServlet
Servlet/JSP Mapping URL:	*.s
File Path of web.xml:	/shengxiao/WebRoot/WEB-INF
Display Name:	This is the display name of my J2EE component
Description:	This is the description of my J2EE component

图 1-11 MyEclipse 构建 Servlet (7)

“*.s”表示在客户端向服务器发送请求时，只要是“.s”结尾的请求，都会被 Servlet 组件截获。

Servlet 组件添加到应用后，会在 web.xml 文件中添加相应的映射与配置项，如图 1-12 所示。

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4"
  xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
  http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <servlet>
    <description>This is the description of my J2EE component</description>
    <display-name>This is the display name of my J2EE component</display-name>
    <servlet-name>ShengxiaoServlet</servlet-name>
    <servlet-class>
      com.myServlet.ShengxiaoServlet
    </servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>ShengxiaoServlet</servlet-name>
    <url-pattern>*.s</url-pattern>
  </servlet-mapping>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
    
```

图 1-12 生成 Servlet 映射

在包中添加一个业务类，用于处理业务逻辑，如图 1-13 所示。

在“Name”栏中输入自定义的业务类的名字，其他属性默认，单击“Finish”按钮，如图 1-14 所示。