

21世纪高等学校计算机类课程创新规划教材 · 微课版



# Java Web 编程技术

## (第3版) 题解与实验指导

◎ 沈泽刚 编著

清华大学出版社

21世纪高等学校计算机类课程创新规划教材 · 微课版



# Java Web 编程技术

## (第3版) 题解与实验指导

◎ 沈泽刚 编著

清华大学出版社  
北京

## 内 容 简 介

本书是《Java Web 编程技术（第3版）》（沈译刚编著，清华大学出版社出版）的配套实验指导与习题解析，目的是帮助读者完成上机实训和课后习题。全书共13章，每章包含如下内容：① 本章知识点总结，该部分总结了本章讲授的主要知识点。② 实训任务，该部分以任务驱动方式给出实训题目，并给出操作的详细步骤，指导读者一步步完成任务。③ 思考与练习答案，该部分给出主教材的所有习题的参考答案，供读者学习参考。

本书适合作为“Java Web 编程技术”课程的教学辅助用书，也可供自学 Java Web 技术的人员参考。

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

Java Web 编程技术（第3版）题解与实验指导/沈译刚编著. —北京：清华大学出版社，2019

（21世纪高等学校计算机类课程创新规划教材·微课版）

ISBN 978-7-302-50340-8

I. ①J… II. ①沈… III. ①JAVA 语言—程序设计—高等学校—教学参考资料 IV. ①TP312.8

中国版本图书馆 CIP 数据核字（2018）第 114923 号

责任编辑：魏江江 薛 阳

封面设计：刘 键

责任校对：徐俊伟

责任印制：宋 林

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>, 010-62795954

印 装 者：三河市铭诚印务有限公司

经 销：全国新华书店

开 本：185mm×260mm 印 张：10.25 字 数：251千字

版 次：2019年4月第1版 印 次：2019年4月第1次印刷

印 数：1~1500

定 价：29.50元

产品编号：079449-01

# 前 言

---

本书是《Java Web 编程技术（第 3 版）》（沈译刚编著，清华大学出版社出版）一书的配套教学辅导教材，它是为帮助读者更好地学习主教材而编写的。本书与主教材的各章一致，共分为 13 章。每章包括如下三方面的内容。

## 1. 知识点总结

这部分内容总结了本章讲述的主要知识点，包括基本概念和基本方法，指出读者应该学习掌握的重点内容。读者可以将这部分内容作为阅读教材的提纲。

## 2. 实训任务

这部分内容以任务的方式给出操作题目，指导读者一步步完成任务。通过这些任务，读者可以掌握本章的知识点并提高操作能力。学习软件开发不上机实践是学不好的，学生通过上机实践可以巩固所学知识、发现问题，找到和学会解决这些问题的方法。上机是学习这门课程的重要环节，必须做好。

## 3. 思考与练习答案

这部分内容给出了教材中每章的思考与练习参考答案。除选择题答案外，还给出一些编程题参考程序。读者在完成主教材中的习题后，对照这里的答案，可以发现问题，从而有助于掌握所学知识。

本书是在《Java Web 编程技术（第 3 版）》一书的基础上编写的，是主教材的补充。希望本书能够对读者更好地掌握这门课程的基本要求，更好地掌握 Java Web 开发的基本技术和实际应用有所帮助。我们希望此教材能为广大教师在 Java Web 教学方面提供一些便利，为学生学习 Java Web 编程技术提供实用的帮助。

本教材在编著过程中得到了很多老师的大力支持和帮助，在此表示感谢！也由衷地希望广大读者多提宝贵意见。由于作者水平有限，书中难免存在错误和不足，欢迎读者和同行专家批评指正。

编 者

2018 年 3 月

# 目 录

---

<b>第 1 章 Java Web 技术概述</b> .....	1
1.1 知识点总结 .....	1
1.2 实训任务 .....	1
1.3 思考与练习答案 .....	9
<b>第 2 章 Servlet 核心技术</b> .....	11
2.1 知识点总结 .....	11
2.2 实训任务 .....	12
2.3 思考与练习答案 .....	17
<b>第 3 章 JSP 技术基础</b> .....	21
3.1 知识点总结 .....	21
3.2 实训任务 .....	22
3.3 思考与练习答案 .....	27
<b>第 4 章 会话与文件管理</b> .....	32
4.1 知识点总结 .....	32
4.2 实训任务 .....	33
4.3 思考与练习答案 .....	40
<b>第 5 章 JDBC 访问数据库</b> .....	43
5.1 知识点总结 .....	43
5.2 实训任务 .....	44
5.3 思考与练习答案 .....	52
<b>第 6 章 表达式语言</b> .....	64
6.1 知识点总结 .....	64
6.2 实训任务 .....	65
6.3 思考与练习答案 .....	70

<b>第7章 JSTL 与自定义标签</b> .....	73
7.1 知识点总结 .....	73
7.2 实训任务 .....	74
7.3 思考与练习答案 .....	79
<b>第8章 Java Web 高级应用</b> .....	85
8.1 知识点总结 .....	85
8.2 实训任务 .....	86
8.3 思考与练习答案 .....	93
<b>第9章 Web 安全性入门</b> .....	96
9.1 知识点总结 .....	96
9.2 实训任务 .....	97
9.3 思考与练习答案 .....	101
<b>第10章 AJAX 技术基础</b> .....	104
10.1 知识点总结 .....	104
10.2 实训任务 .....	105
10.3 思考与练习答案 .....	114
<b>第11章 Struts 2 框架基础</b> .....	116
11.1 知识点总结 .....	116
11.2 实训任务 .....	118
11.3 思考与练习答案 .....	126
<b>第12章 Hibernate 框架基础</b> .....	128
12.1 知识点总结 .....	128
12.2 实训任务 .....	130
12.3 思考与练习答案 .....	145
<b>第13章 Spring 框架基础</b> .....	147
13.1 知识点总结 .....	147
13.2 实训任务 .....	147
13.3 思考与练习答案 .....	157

本章主要学习 Java Web 开发的基本概念以及开发环境构建，包括 Web 客户/服务器体系结构、Web 开发前端技术、Tomcat 服务器与 Eclipse 的安装、配置等。

## 1.1 知识点总结

(1) Web 是基于客户/服务器 (C/S) 的一种体系结构，客户在计算机上使用浏览器向 Web 服务器发出请求，服务器响应客户请求，向客户回送所请求的网页，客户在浏览器窗口上显示网页的内容。

(2) 超文本传输协议 HTTP 是 Web 使用的协议。它是一个基于请求-响应的无状态协议。客户向服务器发送请求，服务器对请求进行处理。

(3) Web 客户端技术包括 HTML、CSS、JavaScript 等。HTML 用来编写 Web 文档，CSS 是一种样式设计语言，JavaScript 是一种脚本语言。

(4) Web 文档是一种重要的 Web 资源，它通常是使用某种语言（如 HTML、JSP 等）编写的页面文件，因此也称为 Web 页面。Web 文档又分为静态文档和动态文档。

(5) Servlet 是用 Servlet API 以及相关的类编写的 Java 程序，这种程序运行在 Web 容器中，主要用来扩展 Web 服务器的功能。

(6) JSP (JavaServer Pages) 页面是在 HTML 页面中嵌入 JSP 元素的页面，这些元素称为 JSP 标签。JSP 元素具有严格定义的语法并包含完成各种任务的语法元素，例如声明变量和方法、JSP 表达式、指令和动作等。

(7) Web 容器（或 Servlet 容器）是运行 Servlet 和 JSP 的软件。Tomcat 服务器是最常用的 Web 容器。

(8) Eclipse for Java EE Developer 是开发 Java Web 项目的开发工具。使用它可以创建动态 Web 项目。

## 1.2 实训任务

### 【实训目标】

学会 Java Web 开发环境构建，包括 Tomcat 服务器安装、Eclipse IDE 安装与配置；掌握用 Eclipse 创建动态 Web 项目、编写和运行 Servlet、编写和运行 JSP。

### 任务 1 学习安装 Tomcat 服务器

运行 Tomcat 需要 JDK 环境，因此首先需要安装 JDK，假设系统已经安装最新版本 JDK 9。

下面完成 Tomcat 的安装。

Tomcat 服务器可到 <http://tomcat.apache.org/> 网站下载各种版本的软件。假设下载的是 Windows 可执行的安装文件 apache-tomcat-9.0.0.M20.exe。

具体安装步骤如下：

(1) 双击下载的安装文件 apache-tomcat-9.0.0.M20.exe，在出现的如图 1-1 所示的界面中选择安装的类型。这里选择完全安装，在 Select the type of install 下拉框中选择 Full，然后单击 Next 按钮，出现如图 1-2 所示的界面。

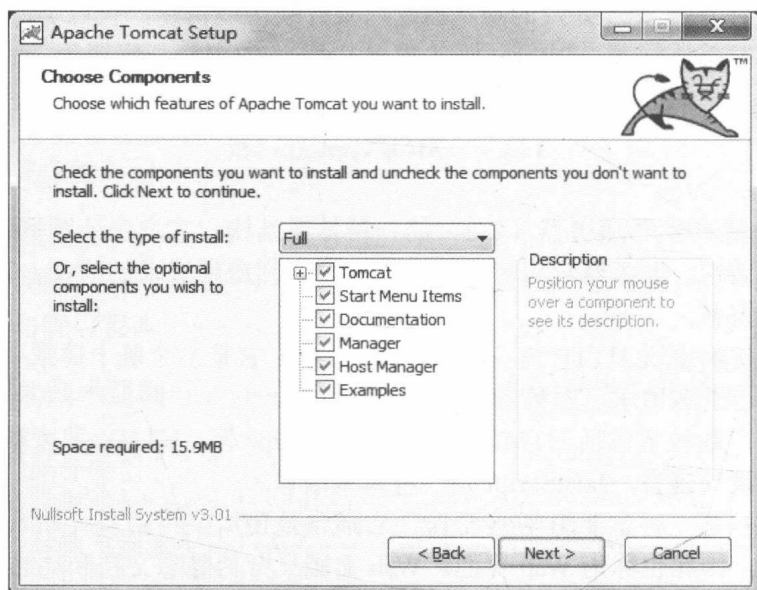


图 1-1 选择安装类型

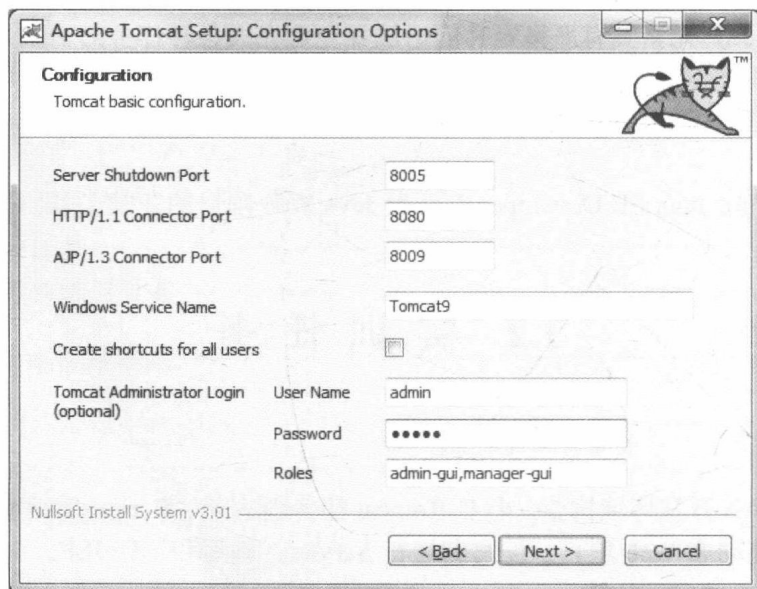


图 1-2 指定端口号、用户名和口令



这里要求用户输入服务器的端口号、管理员的用户名和口令。Tomcat 默认的端口号为 8080，管理员的用户名和口令都填为 admin。

(2) 单击 Next 按钮，出现如图 1-3 所示的对话框，这里需要指定 Java 虚拟机的运行环境的安装路径。接下来出现如图 1-4 所示的对话框，这里要求用户指定 Tomcat 软件的安装路径，默认路径是 C:\Program Files\Apache Software Foundation\Tomcat 9.0。该目录为 Tomcat 的安装目录，在下文中用<tomcat-install>表示。

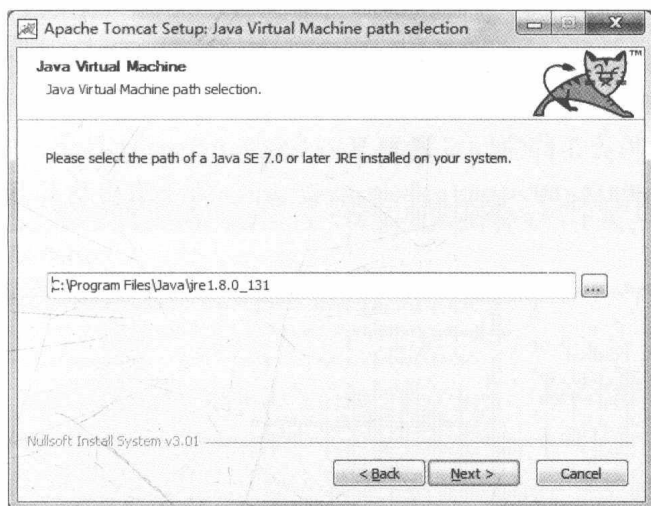


图 1-3 设置 Tomcat 安装路径

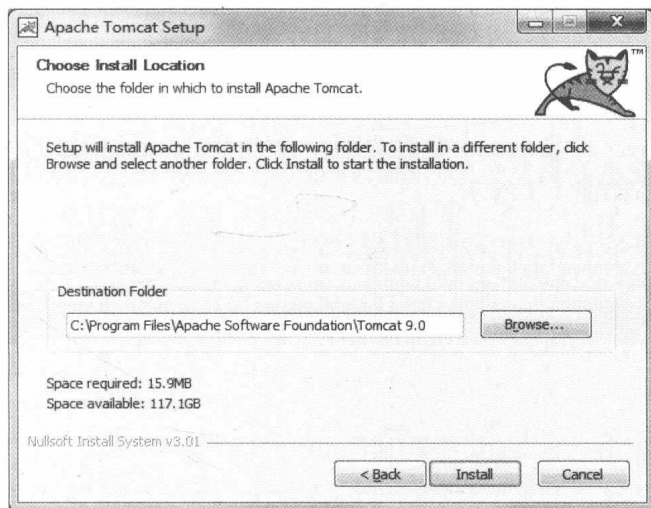


图 1-4 指定 Java 运行时环境的路径

单击 Install 按钮，系统开始安装。在最后出现的窗口中单击 Finish 按钮结束安装。

(3) 查看 Tomcat 安装目录。Tomcat 安装后的目录结构如图 1-5 所示。其中，bin 目录包含启动和停止服务器的程序 Tomcat9w.exe，lib 目录包含各种库文件，webapps 目录中包含所有的 Web 应用程序。

(4) 启动 bin 目录的 Tomcat9w.exe 程序，查看 Tomcat 的启动状态。

(5) 启动浏览器, 在地址栏中输入 “http://localhost:8080/”, 测试 Tomcat 是否正常运行。

## 任务 2 安装 Eclipse 开发工具

Eclipse 下载地址为 <http://www.eclipse.org/downloads/>, 注意应下载 Eclipse IDE for Java EE Developers。

(1) 假设下载的文件是 eclipse-jee-neon-3-win32-x86\_64.zip, 将其解压到 E:\eclipse 目录。

(2) 直接运行解压目录中的 eclipse.exe 程序即可启动 Eclipse。启动 Eclipse 时首先弹出 Eclipse Launcher 对话框, 要求用户选择一个工作空间以存放项目文档, 读者可自行设置自己的工作空间, 这里将工作空间设置为 E:\eclipse-workspace 目录, 如图 1-6 所示。如果选中 Use this as the default and do not ask again 复选框, 则下次启动 Eclipse 将不再显示设置工作空间对话框。

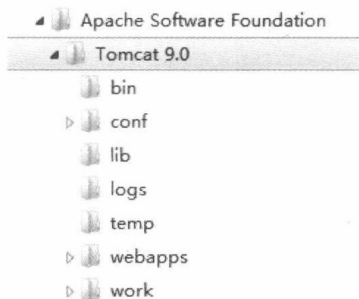


图 1-5 Tomcat 目录结构

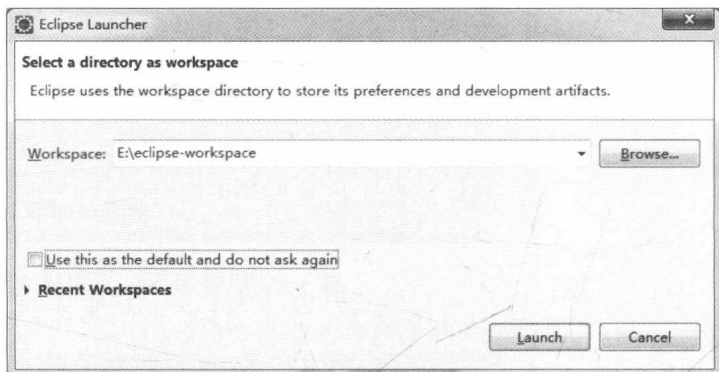


图 1-6 选择工作空间对话框

第一次运行 Eclipse 将显示一个欢迎界面, 单击 Welcome 标签的关闭按钮, 就可以进入 Eclipse 开发环境, 如图 1-7 所示。

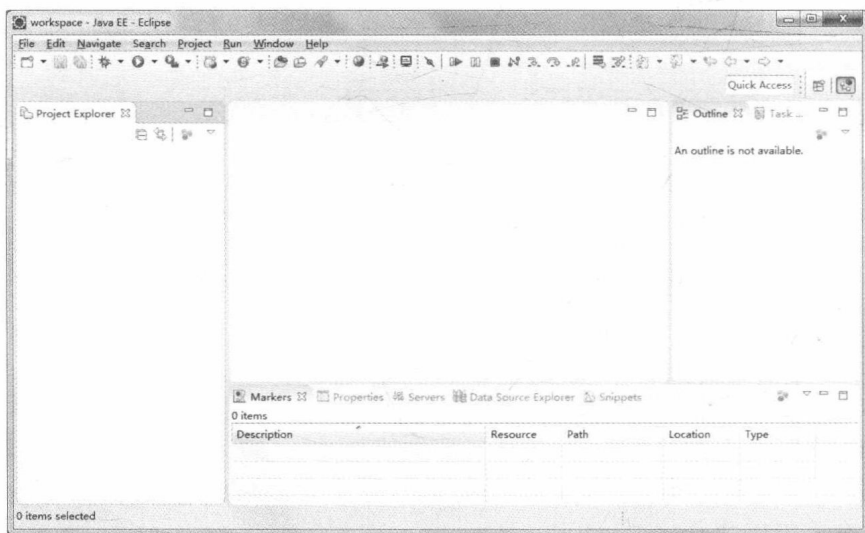


图 1-7 Eclipse 的开发界面

(3) 在 Eclipse 中配置 Tomcat 服务器。选择 Window→Preferences 命令，在打开的对话框左边列表框中选择 Server 节点中的 Runtime Environments。单击窗口右侧的 Add 按钮，打开 New Server Runtime Environmen 对话框，在该对话框中可选择服务器的类型和版本，这里使用的是 Apache Tomcat v 9.0。

### 任务 3 使用 Eclipse IDE 创建动态 Web 项目

创建一个名为 helloworld 的动态 Web 项目。

(1) 在 Eclipse 中选择 File→New→Dynamic Web Project，打开新建动态 Web 项目对话框。在 Project name 文本框中输入项目名 hello-demo，下面的选项采用默认值即可，如图 1-8 所示。

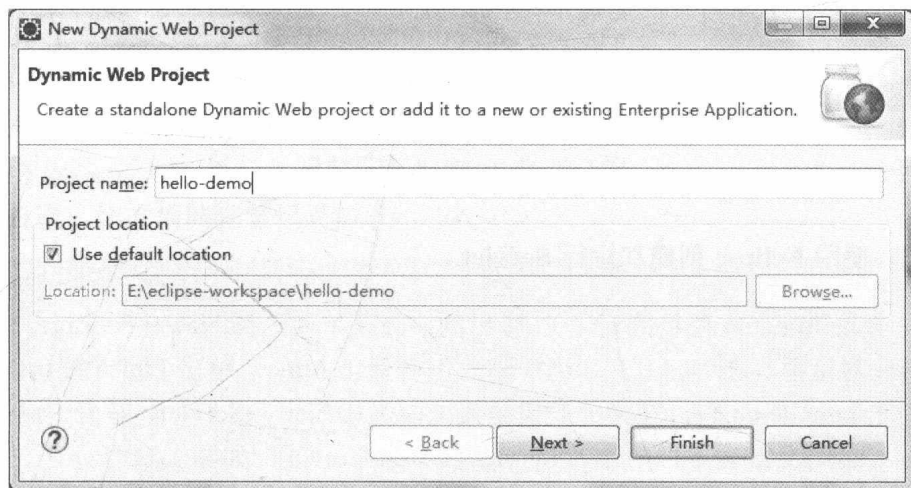


图 1-8 新建动态 Web 项目对话框

(2) 单击 Next 按钮，打开 Web Module 对话框，在这里需要指定 Web 应用程序上下文根目录名称和 Web 内容存放的目录，这里采用默认值，选中 Generate web.xml deployment descriptor 复选框，由 Eclipse 产生部署描述文件，如图 1-9 所示。

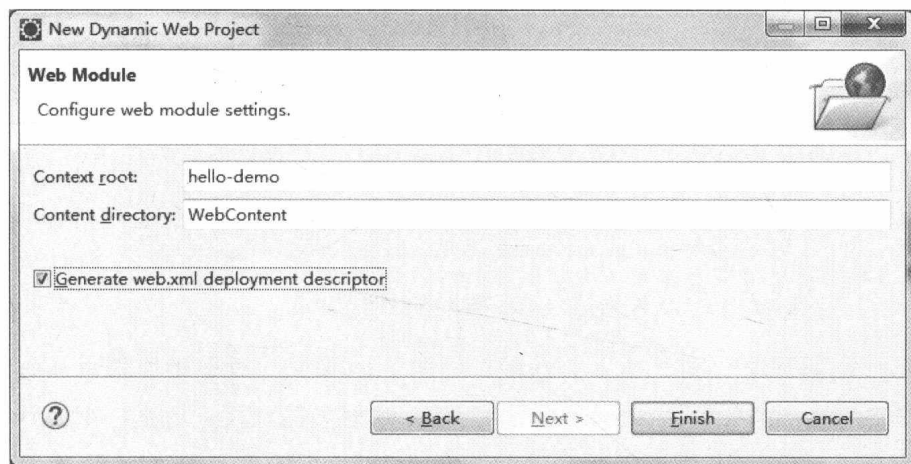


图 1-9 Web Module 对话框

最后单击 Finish 按钮, 结束项目的创建。项目创建后, 在 Eclipse 中显示项目结构, 如图 1-10 所示。这里, src 目录用于存放 Java 源程序, WebContent 目录用于存放其他资源文件, 如 JSP 页面等。



图 1-10 hello-demo 项目结构

#### 任务 4 使用 Eclipse 创建和运行 Servlet

(1) 右击 hello-demo 项目的 src, 在弹出的快捷菜单中选择 New→Package, 打开 New Java Package 对话框。在 Name 文本框中输入包名 com.demo, 单击 Finish 按钮创建包。

(2) 右击 com.demo 包, 在弹出的快捷菜单中选择 New→Servlet, 打开 Create Servlet 对话框。在 Class name 文本框中输入 Servlet 名 HelloServlet, 如图 1-11 所示。

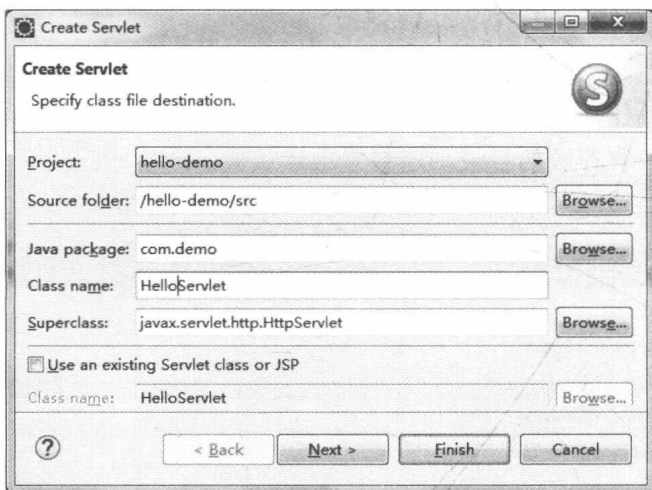


图 1-11 创建 HelloServlet 对话框

(3) 单击 Next 按钮, 进入下一对话框, 如图 1-12 所示。这里需要指定 Servlet 在部署描述文件中的信息, 主要包括 Servlet 名称和 URL 映射名的定义。这里, 将 Servlet 名称修改为 helloServlet, 将 URL 映射名称修改为/hello-servlet。

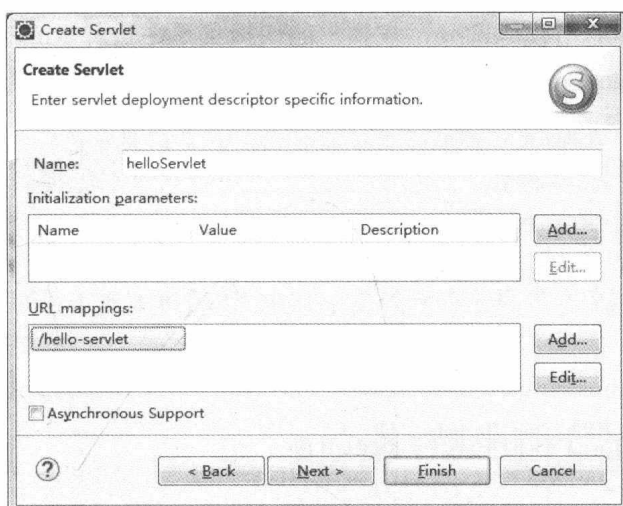


图 1-12 Servlet 映射配置对话框

(4) 单击 Next 按钮，在出现的对话框中指定 Servlet 实现的接口及自动生成的方法，这里只选中 doGet 方法。最后单击 Finish 按钮，Eclipse 将生成该 Servlet 的部分代码并在编辑窗口中打开，修改后的完整代码如下：

```
package com.demo;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.time.LocalDate;
import java.io.*;

@WebServlet(name = "helloServlet", urlPatterns = { "/hello-servlet" })
public class HelloServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<body><head><title>当前日期</title></head>");
        out.println("<h3>Hello, World!</h3>");
        out.println("今天的日期是："+ LocalDate.now() );
        out.println("</body>");
        out.println("</html>");
    }
}
```

(5) 在 Eclipse 中右击代码部分, 在弹出的快捷菜单中选择 Run As→Run on Server 即可运行该 Servlet。Eclipse 将打开内部浏览器显示运行结果, 如图 1-13 所示。

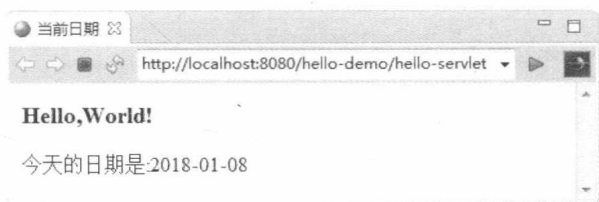


图 1-13 HelloServlet 运行结果

### 任务 5 使用 Eclipse 创建和运行 JSP 页面

创建一个名为 hello.jsp 的 JSP 页面, 实现与任务 4 的 Servlet 相同的功能。

(1) 右击 hello-demo 项目的 WebContent 节点, 在弹出的快捷菜单中选择 New→JSP File, 打开 New JSP File 对话框。选择 JSP 页面存放的目录, 这里为 WebContent。在 File name 文本框中输入文件名 hello.jsp。

(2) 单击 Next 按钮, 打开选择 JSP 模板对话框, 从模板列表中选择要使用的模板, 这里选择 New JSP File(html)模板, 然后单击 Finish 按钮。

(3) Eclipse 创建 hello.jsp 页面并在工作区中打开该文件, 在<body>标签中插入两行代码:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import="java.time.LocalDate" %>
<html>
<head><title>当前日期</title></head>
<body>
    <h3>Hello,World!</h3>
    今天的日期是: <%=LocalDate.now() %>
</body>
</html>
```

(4) 在 JSP 页面编辑区中右击, 在弹出的快捷菜单中选择 Run As→Run on Server 即可执行该 JSP 页面, 运行结果如图 1-14 所示。

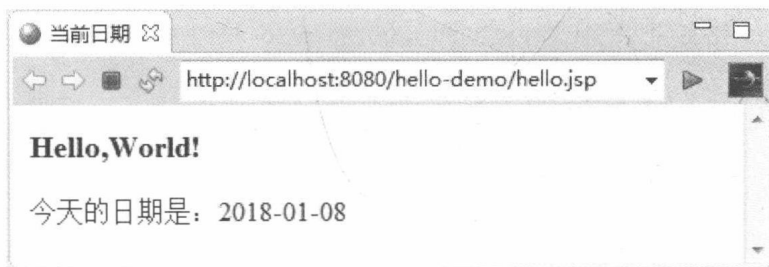


图 1-14 hello.jsp 页面运行结果

## 1.3 思考与练习答案

1. 主机名 localhost 对应的 IP 地址是 ( )。

- A. 192.168.0.1
- B. 127.0.0.1
- C. 0:0:0:0:0:0:1
- D. 1:0:0

【答】 B。127.0.0.1 是本地主机的 IP 地址，它对应的主机名是 localhost。

2. 下面哪个是 URL? ( )

- A. www.tsinghua.edu.cn
- B. http://www.baidu.com
- C. 121.52.160.5
- D. /localhost:8080/webcourse

【答】 B。A 是一个主机名，C 是一个 IP 地址，D 是一个 URI。

3. 要在页面中导入 css/layout.css 样式单文件，下面哪两个选项是正确的? ( )

- A. `<link type="text/css" href="css/layout.css" rel="stylesheet" />`
- B. `<script type="text/javascript" src="css/layout.css "></script>`
- C. `<style type="text/css">@import url(css/layout.css);</style>`
- D. `<meta http-equiv="Content-Type" content=" css/layout.css; charset=UTF-8">`

【答】 A, C。使用这两个标签都可以导入样式单。

4. 若访问的资源不存在，服务器向客户发送一个错误页面，该页面中显示的 HTTP 状态码是 ( )。

- A. 500
- B. 200
- C. 404
- D. 403

【答】 C。状态码 500 表示服务器内部错误，403 表示页面禁止访问，200 表示请求成功。

5. 下面哪个不是服务器页面技术? ( )

- A. JSP
- B. ASP
- C. PHP
- D. JavaScript

【答】 D。JavaScript 是一种脚本语言，可用来编写脚本代码实现客户端动态页面技术。JSP、ASP 和 PHP 都是服务器端动态页面技术。

6. Servlet 必须在什么环境下运行? ( )

- A. 操作系统
- B. Java 虚拟机
- C. Web 容器
- D. Web 服务器

【答】 C。Servlet 必须在 Web 容器或 Servlet 容器中运行，Tomcat 是最常用的 Web 容器。

7. 下面是 URL 的为 ( )，是 URI 的为 ( )，是 URN 的为 ( )。

- ① `http://www.myserver.com/hello`
- ② `files/sales/report.html`
- ③ `ISBN:1-930110-59-6`

【答】 ①是 URL，①和②都是 URI，③是 URN。

8. 在 Tomcat 服务器中，一个 Web 应用程序应该存放在 Tomcat 的哪个目录中? ( )

- A. bin 目录
- B. confs 目录
- C. webapps 目录
- D. work 目录

【答】 C。Tomcat 服务器的 webapps 目录中每个子目录是一个 Web 应用程序。

9. 什么是 URL, 什么是 URI, 它们都由哪几部分组成? URL 与 URI 有什么关系?

**【答】** URL 称为统一资源定位符, URL 通常由 4 部分组成: 协议名称、主机的 DNS 名、可选的端口号和资源的名称。URI 称为统一资源标识符, 是以特定语法标识一个资源的字符串。URI 由模式和模式特有的部分组成, 它们之间用冒号隔开, 一般格式如下:

```
schema: schema-specific-part
```

URI 是 URL 和 URN 的超集。

10. 动态 Web 文档技术有哪些? 服务器端动态文档技术和客户端动态文档技术有何不同?

**【答】** 动态 Web 文档技术包括服务器端动态文档技术和客户端动态文档技术, 前者包括 CGI 技术、服务器扩展技术和 HTML 页面中嵌入脚本技术。其中 HTML 页面中嵌入脚本技术包括 ASP、PHP 和 JSP 技术。

最流行的客户端动态文档技术是在 HTML 页面中嵌入 JavaScript 脚本代码。使用 JavaScript 可以设计交互式页面。与服务器端动态文档不同, JavaScript 脚本是在客户端执行的。

11. 什么是 Servlet? 什么是 JSP? 它的主要作用是什么?

**【答】** Servlet 是用 Servlet API 开发的 Java 程序, 它运行在 Web 容器中。Web 容器是运行 Servlet 的软件, 主要用来扩展 Web 服务器的功能。

12. 哪些资源应该存放在 Web 应用程序的 WEB-INF 目录中?

**【答】** 如果资源仅供服务器访问, 存放在 WEB-INF 目录中即可, 以避免用户直接用 URL 访问它。



本章学习 Servlet 核心编程技术，包括 `HttpServlet` 类、`HttpServletRequest` 请求对象、`HttpServletResponse` 响应对象、处理请求参数、请求转发、处理响应等。

## 2.1 知识点总结

(1) Servlet API 是 Java Web 开发的基础，它由 4 个包组成：`javax.servlet`、`javax.servlet.http`、`javax.servlet.annotation` 和 `javax.servlet.descriptor`。

(2) Servlet 接口是核心接口，每个 Servlet 必须直接或间接实现该接口，该接口定义了 `init()`、`service()` 和 `destroy()` 生命周期方法以及 `getServletInfo()` 与 `getServletConfig()`。

(3) `GenericServlet` 抽象类实现了 Servlet 接口和 `ServletConfig` 接口。

(4) `ServletConfig` 在 Servlet 初始化时，容器将调用 `init(ServletConfig)`，并为其传递一个 `ServletConfig` 对象，该对象称为 Servlet 配置对象，使用该对象可以获得 Servlet 初始化参数、Servlet 名称、`ServletContext` 对象等。

(5) `HttpServlet` 类扩展了 `GenericServlet` 类，在 `HttpServlet` 中针对不同的 HTTP 请求方法定义了不同的处理方法，如处理 GET 请求的 `doGet()`，该方法有两个参数：一个是请求对象，一个是响应对象。

(6) `HttpServletRequest` 接口对象是请求对象，使用它可以检索客户请求信息，如使用 `getParameter()` 可以获取请求参数，`getMethod()` 可以获取请求的 HTTP 方法（如 GET 或 POST），`getRequestURI()` 返回请求 URI 等。

(7) `HttpServletResponse` 接口对象是响应对象，通过它可向客户端发送响应消息，如 `getWriter()` 返回 `PrintWriter` 对象，它可以向客户发送文本数据；`setContentType()` 设置响应的内容类型；`setHeader()` 设置响应头；`sendRedirect()` 响应重定向等。

(8) 在客户端发生下面的事件，浏览器就向 Web 服务器发送一个 HTTP 请求。① 用户在浏览器的地址栏中输入 URL 并按 Enter 键。② 用户点击了 HTML 页面中的超链接。③ 用户在 HTML 页面中填写一个表单并提交。

(9) 要实现请求转发，调用请求对象的 `getRequestDispatcher()` 得到 `RequestDispatcher` 对象，然后调用它的 `forward()` 方法将请求转发其他资源（Servlet 或 JSP）。

(10) 请求对象是一个作用域对象，通过它的 `setAttribute()` 将一个对象作为属性存储到请求对象中，然后可以在请求作用域的其他资源中使用 `getAttribute()` 检索出属性。

(11) 每个 Web 应用程序在它的根目录中都必须有一个 `WEB-INF` 目录，其中 `classes` 目录存放类文件，`lib` 目录存放库文件，该目录下还应该有一个 `web.xml` 文件，称为部署描