

21世纪高等学校计算机教育实用规划教材

Java Web编程技术 (第2版) 题解与实验指导

沈泽刚 任敏贤 编著



清华大学出版社

21世纪高等学校计算机教育实用规划教材

Java Web编程技术 (第2版) 题解与实验指导

沈泽刚 任敏贤 编著

清华大学出版社

内 容 简 介

本书是《Java Web 编程技术(第2版)》(清华大学出版社出版)的配套习题解析和实验指导,目的是为学生课后学习和课堂上机实验提供支持。全书共分12章,每章包含如下内容:1.本章要点,该部分总结了本章讲授的主要内容;2.实验指导,该部分给出了实验目的和若干实验题目;3.习题解析,该部分给出了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2版)题解与实验指导/沈泽刚,任敏贤编著.--北京:清华大学出版社,2014

21世纪高等学校计算机教育实用规划教材

ISBN 978-7-302-35143-6

I. ①J… II. ①沈… ②任… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 012439 号

责任编辑:魏江江 薛 阳

封面设计:常雪影

责任校对:时翠兰

责任印制:李红英

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

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:河北新华第一印刷有限责任公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:9.75 字 数:187千字

版 次:2014年8月第1版 印 次:2014年8月第1次印刷

印 数:1~2000

定 价:25.00元

产品编号:053205-01

出版说明

随着我国高等教育规模的扩大以及产业结构调整的不断深入,社会对高层次应用型人才的需求将更加迫切。各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,合理调整和配置教育资源,在改革和改造传统学科专业的基础上,加强工程型和应用型学科专业建设,积极设置主要面向地方支柱产业、高新技术产业、服务业的工程型和应用型学科专业,积极为地方经济建设输送各类应用型人才。各高校加大了使用信息科学等现代科学技术提升、改造传统学科专业的力度,从而实现传统学科专业向工程型和应用型学科专业的发展与转变。在发挥传统学科专业师资力量强、办学经验丰富、教学资源充裕等优势的同时,不断更新教学内容、改革课程体系,使工程型和应用型学科专业教育与经济建设相适应。计算机课程教学在从传统学科向工程型和应用型学科转变中起着至关重要的作用,工程型和应用型学科专业中的计算机课程设置、内容体系和教学手段及方法等也具有不同于传统学科的鲜明特点。

为了配合高校工程型和应用型学科专业的建设和发展,急需出版一批内容新、体系新、方法新、手段新的高水平计算机课程教材。目前,工程型和应用型学科专业计算机课程教材的建设工作仍滞后于教学改革的实践,如现有的计算机教材中有不少内容陈旧(依然用传统专业计算机教材代替工程型和应用型学科专业教材),重理论、轻实践,不能满足新的教学计划、课程设置的需要;一些课程的教材可供选择的品种太少;一些基础课的教材虽然品种较多,但低水平重复严重;有些教材内容庞杂,书越编越厚;专业课教材、教学辅助教材及教学参考书短缺,等等,都不利于学生能力的提高和素质的培养。为此,在教育部相关教学指导委员会专家的指导和帮助下,清华大学出版社组织出版本系列教材,以满足工程型和应用型学科专业计算机课程教学的需要。本系列教材在规划过程中体现了如下一些基本原则和特点。

(1) 面向工程型与应用型学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映基本理论和原理的综合应用,强调实践和应用环节。

(2) 反映教学需要,促进教学发展。教材规划以新的工程型和应用型专业目录为

依据。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材建设仍然把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现工程型和应用型专业教学内容和课程体系改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材要配套,同一门课程可以有多种具有不同内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材,教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要真实实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机教育实用规划教材编委会

联系人:魏江江 weijj@tup.tsinghua.edu.cn

前 言

本书是作者编著的《Java Web 编程技术(第 2 版)》一书的配套教学辅导教材,它是为了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内容一致,共分 12 章。每章讲述了下述三方面的内容。

1. 本章要点

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. 实验指导

这部分内容包括本章的实验要求和实验题目,并根据这些要求给出了上机实验题目。学习程序设计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. 习题解答

这部分内容对教材中每章的习题给出了参考答案,不仅给出简答题答案,也给出编程题的参考程序。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 2 版)》一书的基础上编写的,是对主教材的补充和扩展。希望本书对读者更好地掌握这门课程的基本要求,掌握 Java Web 开发的基本技术和实际应用,能够有所帮助。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 编程技术提供一本实用的教材。

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之处,欢迎读者和同行专家批评指正。

编 者

2014 年 4 月

目 录

第 1 章 Java Web 技术概述	1
1.1 本章要点	1
1.2 实验指导	2
1.3 习题解析	5
第 2 章 Servlet 技术模型	7
2.1 本章要点	7
2.2 实验指导	8
2.3 习题解析	19
第 3 章 Servlet 容器模型	24
3.1 本章要点	24
3.2 实验指导	25
3.3 习题解析	36
第 4 章 JSP 技术模型	40
4.1 本章要点	40
4.2 实验指导	41
4.3 习题解析	45
第 5 章 表达式语言	50
5.1 本章要点	50
5.2 实验指导	52

5.3	习题解析	56
第6章	JSP 标签技术	59
6.1	本章要点	59
6.2	实验指导	60
6.3	习题解析	66
第7章	JDBC 数据库访问	72
7.1	本章要点	72
7.2	实验指导	74
7.3	习题解析	82
第8章	Servlet 高级应用	89
8.1	本章要点	89
8.2	实验指导	91
8.3	习题解析	94
第9章	Web 安全性入门	98
9.1	本章要点	98
9.2	实验指导	99
9.3	习题解析	102
第10章	AJAX 技术基础	106
10.1	本章要点	106
10.2	实验指导	108
10.3	习题解析	111
第11章	Hibernate 框架基础	113
11.1	本章要点	113
11.2	实验指导	115
11.3	习题解析	126

第 12 章 Struts 2 框架基础	128
12.1 本章要点	128
12.2 实验指导	130
12.3 习题解析	142

1.1 本章要点

Java Web 技术是用 Java 技术来解决 Web 相关领域问题的技术总和,通常包括 Web 服务器端技术和 Web 客户端技术两部分。

Web 是 Internet 的最重要应用,它是基于客户/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。

服务器端技术包括 HTTP、Web 服务器和 Web 容器、动态 Web 页面技术等。客户端技术包括 HTML 和 XML、CSS、JavaScript 以及 Ajax 等。

超文本传输协议(Hypertext Transfer Protocol,HTTP)是 Web 使用的协议。它是一个基于请求-响应的无状态协议,这种请求-响应的过程如图 1-1 所示。



图 1-1 HTTP 请求-响应示意图

客户首先通过浏览器程序建立到 Web 服务器的连接并向服务器发送 HTTP 请求消息。服务器接收到客户的请求后,对请求进行处理,然后向客户发回 HTTP 响应。客户接收服务器发送的响应消息,对消息进行处理并关闭连接。

HTML 称为超文本标记语言,它用来编写 Web 文档。文档是由一些标签(tag)组成的文本文件,标签标识了内容和类型,描述了如何格式化文档。Web 浏览器通过解析这些标签显示文档。

CSS 是能够真正做到网页表现与内容分离的一种样式设计语言。JavaScript 是一种广泛用于客户端 Web 开发的脚本语言,常被用来编写校验表单数据的代码等。

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。

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。JSP(Java Server 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,比如声明变量和方法、JSP 表达式、指令和动作等。

Servlet 和 JSP 需要运行在 Servlet 容器(或 Web 容器)中,它可以是 Web 服务器的一个模块,也可以是单独的模块。Tomcat 服务器是最常用的 Web 容器,最新版本 Tomcat 7.0.39 实现了 Servlet 3.0 和 JSP 2.2 的规范,另外它本身具有作为 Web 服务器运行的能力。

在 Web 应用开发中建议采用 MVC(Model-View-Controller)设计模式,该设计模式将 Web 组件分为模型(Model)、视图(View)和控制器(Controller)三种类型,其中控制器使用 Servlet 或 Filter 实现,模型使用 JavaBeans 或 JOPO 实现,视图使用 JSP 页面实现。使用 MVC 设计模式可以实现表示逻辑与业务逻辑分离,从而使 Web 应用程序的开发和维护变得更容易。

1.2 实验指导

【实验目的】

- (1) 掌握用 Eclipse 开发动态 Web 项目。
- (2) 掌握简单的 Servlet 和 JSP 页面的开发。

【实验内容】

实验题目 1: 使用 Eclipse IDE 创建动态 Web 项目。

(1) 在 Eclipse 中配置 Tomcat。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 Environment 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 7.0。

(2) 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,如 helloweb,下面的选项采用默认值即可。

(3) 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 1-2 所示。最后单击 Finish 按钮,结束项目的创建。

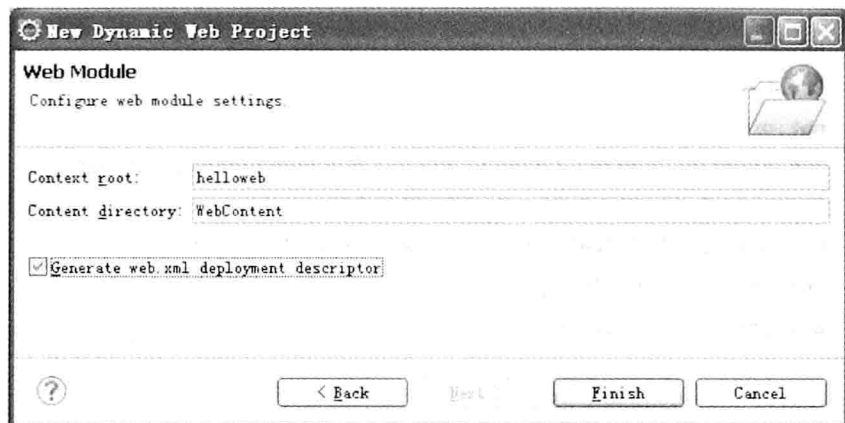


图 1-2 Web Module 对话框

实验题目 2: 使用 Eclipse IDE 创建和运行 Servlet。

(1) 右击 helloworld 项目,从弹出菜单中选择 New→Servlet,打开 Create Servlet 对话框。在 Java package 文本框中输入包名,如 com.demo,在 Class name 文本框中输入类名 HelloServlet。

(2) 单击 Next 按钮,进入下一对话框。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/helloServlet.do。

(3) 单击 Next 按钮,在出现的对话框中指定 Servlet 实现的接口以及自动生成的方法。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.io.*;

@WebServlet(name = "helloServlet", urlPatterns = { "/helloServlet.do" })
```

```
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>当前时间</title></head>");
        out.println("<h3>Hello, World!</h3>");
        out.println("现在的时间是:" + new java.util.Date());
        out.println("</body>");
        out.println("</html>");
    }
}
```

(4) 在 Eclipse IDE 中右击代码部分,在弹出菜单中选择 Run As→Run on Server 即可执行该 Servlet。

实验题目 3: 使用 Eclipse IDE 创建一个 JSP 页面。

(1) 右击 helloworld 项目的 WebContent 节点,从弹出菜单中选择 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。Eclipse 创建 hello.jsp 页面并在工作区中打开该文件,可以在<body>标签中插入代码。

```
<% @ page language = "java" contentType = "text/html; charset = UTF - 8"
    pageEncoding = "UTF - 8" %>
<html>
<head><title>简单的 JSP 页面</title></head>
<body>
    <h1>Hello, World!</h1>
    现在的时间是: <% = new java.util.Date() %>
</body>
</html>
```

(3) 要运行 JSP 页面,在 JSP 页面编辑区中右击鼠标,在打开的菜单中选择 Run As→Run on Server 即可执行该 JSP 页面。

【思考题】

1. 如何在 Eclipse 中配置 Tomcat?

2. 如何在 Eclipse 中创建动态 Web 项目? 如何创建和运行 Servlet 和 JSP?

1.3 习题解析

1. 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。

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

- 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。

3. 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. 401

【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。

4. 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。

5. 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。

6. MVC 设计模式不包括下面哪个? ()

- A. 模型 B. 视图 C. 控制器 D. 数据库

【答】 D。MVC 设计模式是模型(Model)、视图(View)和控制器(Controller)。

7. 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。

8. 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。

9. 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。

10. 什么是 Servlet? 什么是 Servlet 容器? 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 Servlet 容器中。Servlet 容器是运行 Servlet 的软件,主要用来扩展 Web 服务器的功能。

11. 什么是 MVC 设计模式,它有什么优点?

【答】 MVC 设计模式称为模型-视图-控制器,在这种模式中,将 Web 组件分为模型(Model)、视图(View)和控制器(Controller),每种组件完成各自的任務。该模式的 最大优点是将业务逻辑和数据访问从表示层分离出来。

2.1 本章要点

Servlet API 定义了若干接口和类,它是 Java Web 应用开发的基础,它由 4 个软件包组成: javax. servlet、javax. servlet. http、javax. servlet. annotation 和 javax. servlet. descriptor。

Servlet 接口是 Servlet API 中的核心接口,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()。

HttpServlet 类用来实现针对 HTTP 协议的 Servlet,它扩展了 GenericServlet 抽象类,我们编写的 Servlet 通常继承 HttpServlet 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()格式如下:

```
protected void doGet(HttpServletRequest request,
                      HttpServletResponse response)
```

该方法两个参数一个是请求对象,一个是响应对象。HttpServletRequest 接口对象是请求对象,使用它可以检索客户请求信息。例如,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。

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。

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。

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。

- 用户在浏览器的地址栏中输入 URL 并按回车键。
- 用户单击了 HTML 页面中的超链接。
- 用户在 HTML 页面中填写一个表单并提交。

如果需要从 Servlet 中将请求转发到其他资源(Servlet 或 JSP),可以通过请求对象的 `getRequestDispatcher()` 得到 `RequestDispatcher` 对象,然后调用它的 `forward()` 转发请求。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。

Web 应用程序具有严格定义的目录结构,每个 Web 应用程序都有一个文档根目录,目录名就是 Web 应用程序名,该目录中可以定义其他目录存放应用程序资源。每个 Web 应用程序在它的根目录中都必须有一个 `WEB-INF` 目录,该目录中主要存放供服务器访问的资源,其中 `classes` 目录存放支持该 Web 应用程序的类文件,`lib` 目录存放 Web 应用程序使用的全部库文件,该目录下还应该有一个 `web.xml` 文件,它叫 Web 应用程序部署描述文件,其中可以定义有关资源(如 Servlet、监听器、安全性约束等)。

在 Servlet 3.0 的 `javax.servlet.annotation` 包中定义了若干注解,使用 `@WebServlet` 注解可以定义 Servlet。

下面一行是为 `HelloServlet` 添加的注解。

```
@WebServlet(name = "HelloServlet", urlPatterns = {"/hello.do"})
```

这里,`name` 元素指定 Servlet 名称,`urlPatterns` 元素指定 URL。该注解还可包含其他元素。注解在应用程序启动时被 Web 容器处理,容器根据具体的元素配置将相应的类部署为 Servlet。

2.2 实验指导

【实验目的】

- (1) 掌握通过 Servlet 处理表单数据。
- (2) 掌握通过 Servlet 处理业务逻辑,实现请求转发等。
- (3) 掌握使用 Servlet 3.0 的新增功能实现文件上传。