



全国信息化计算机应用技术资格认证推荐教材

HandsOn
翰子昂实训体系第二阶段

JSP 案例教程

高寿柏 主编

山东友谊出版社





全国信息化计算机应用技术资格认证推荐教材

HandsOn
翰子昂实训体系第二阶段

JSP 案例教程

高寿柏 主编

山东友谊出版社



图书在版编目 (CIP) 数据

翰子昂实训体系第二阶段 / 高寿柏主编. — 济南:

山东友谊出版社, 2008.5

ISBN 978-7-80737-321-6

I. 翰子昂… II. 高… III. 软件开发—人才—培养—教材
IV. TP311.52

中国版本图书馆 CIP 数据核字 (2008) 第 060139 号

翰子昂实训体系第二阶段

JSP 案例教程

高寿柏 主编

主 管: 山东出版集团
集团网址: www.sdpress.com.cn
出版发行: 山东友谊出版社
地 址: 济南市胜利大街 39 号 邮政编码: 250001
电 话: 总编室 (0531) 82098756 82098142
发行部 (0531) 82098035 (传真)
印 刷: 济南希尔康印务有限公司
版 次: 2008 年 5 月第 1 版
印 次: 2008 年 5 月第 1 次印刷
规 格: 889mm×1194mm 16 开本
印 张: 12
字 数: 240 千字 (本册)
定 价: 260.00 元 (全五册)

(如印装质量有问题, 请与出版社总编室联系调换)

翰子昂实训体系软件理论教材丛书序

本套丛书是翰子昂实训体系在软件项目实训前的技术准备教材，与项目实训教材丛书一起组成了一个较完善的实训教材体系。丛书采用案例教学的形式，带领学员以解决项目问题为目标去学习知识，掌握技术细节。从而使学员在学习知识的同时，掌握整合技术点，为软件开发寻找解决方案的思路与途径。

软件开发要求从业人员熟悉软件开发的行业规范，具有良好的实践基础，有自主学习意识、独立思考的能力。案例教学法就是把软件开发对知识技术的要求带入理论知识点的学习过程，让学员带着要解决的问题去学习知识，逐步形成自己对问题的理解，建立问题与知识点间的联系，形成有效解决方案，从而提高自己分析和解决问题的能力。

案例教学法是一种推动式的教学方法，能创建良好宽松的学习氛围。传统的教学方法是拉动式的，使学员始终处于被动吸收的学习之中，不能培养学员主动思考及独立解决问题的能力。案例教学引进项目实际要求，学员解决问题的过程就是按照自己的理解去寻找相关知识与技术点的过程。在这一过程中，项目的规范与要求是学员要达到的目标，教师指导及相关资料的辅助是学员学习的推动力。项目完成后，学员掌握了问题所涉及的所有知识，形成了自己的解决方案，培养了自己主动思考的学习和工作习惯。

丛书以专题的形式呈现一系列知识结构，在每个专题中又以案例的形式描述了一个完整的实现过程，案例之后是所用知识点的讲解，最后是这个案例的扩展实现。学员在使用丛书时，应首先按照书中的步骤，逐步实现案例，遇到不了解的知识点时，再去参考案例后面的知识点讲解。通过这样主动探求，逐步深入的过程，来实现自己对知识点从感性认识到理性掌握的转变。案例完成后，学员还可以参考案例后的扩展内容进一步深化学习。由此，学员在完成一个案例时，不仅掌握了项目包含的技术知识，还可以了解与之相关的知识与技术。

软件可能是人类做出来的最复杂的东西了，软件开发也是一个复杂的过程，软件开发的讲授就更是复杂的工程了，笔者很高兴看到这套丛书将这—个复杂的过程变得清晰易懂，简单明了。这套丛书是通向正确方向的一条路径。

在丛书成书的思路整理、高校与企业调研、结构设计与编写成书的过程中，得到了许多高校研究与教学人员的大力支持，尤其是华东师范大学职业与成人教育研究所徐国庆博士的指导与建议，使我们能更加清晰地把握丛书定位与方向；

还有更多的企业中资深软件工程师和高校一线教师，亲自参加了技术知识的组织与书稿的写作。在此，对他们辛勤的劳动致以最真诚的感谢。

尽管我们始终追求尽善尽美，以便向您提供最新及最全面的知识与技术，但由于作者水平有限，疏漏之处仍在所难免，恳切希望得到您对本教材的建议。

翰子昂实训体系软件理论教材丛书编委成员，主编：高寿柏、宁东，副主编：任春梅、王爱军。

北京翰子昂教育科技有限公司

网址：<http://www.hands-on.com.cn>

2007年12月

前 言

JSP (Java Server Pages) 是一种新动态网页技术标准。在传统的网页 HTML 文件 (*.htm,*.html) 中加入 Java 程序片段 (Scriptlet) 和 JSP 标签, 构成了 JSP 网页 (*.jsp)。Servlet/JSP 容器收到客户端发出的请求时, 首先执行其中的程序片段, 然后将执行结果以 HTML 格式响应给客户端。

这几年, JSP 发展已经取得了巨大的成功, 它通过和 EJB 等 J2EE 组件进行集成, 可以编写出具有强大伸缩性、高负载的企业级应用, 推动了动态 Web 应用开发进程。

本书共分 9 个专题, 13 个案例, 通过实现不同的功能来讲解使用 JSP 开发技巧, 主要包括以下内容:

专题一: 使用不同的开发方式开发 Web 应用程序, 使学生对利用工具开发有深刻的认识, 并简单介绍一些著名的开发工具和服务器。

专题二: 详细介绍了 Servlet 的基本概念, Servlet 的生命周期, Servlet 的体系结构, ServletAPI 常用接口, Servlet 的通信方式, Servlet 上下文环境。通过案例的讲解和课后练习, 使学生熟练掌握利用 Servlet 开发应用程序。

专题三: 介绍了 JSP 的基础知识, JSP 页面的各种构成元素、JSP 页面的执行过程及其优点。使学生对 JSP 技术有了初步的认识。

专题四: 详细讲解了 JSP 声明、JSP 脚本元素、JSP 指令, 是对专题三基础知识的有益补充。

专题五: 详细讲解了 JSP 内置对象。利用 JSP 内置对象, 为程序开发提供了非常方便的强大功能。

专题六: 介绍了 JavaBean 组件基本知识和 JSP 中的标准动作标签, 利用 JavaBean 组件和 JSP 标签, 实现模式一开发模式, 通过本专题的学习, 学生建立架构模式开发的开发理念。

专题七: 介绍了使用 JSTL 标签、自定义开发标签的过程。JSTL 标签提供了强大、丰富的功能, 完全能应付常用的 WEB 应用开发。学生也可以根据实际需要, 编写自定义标签。

专题八: 过滤器和监听器, 自定义开发过滤器和监听器的步骤, 提供了验证, 监听处理, 根据需要实现不同的功能。

专题九: 讲解了不同的开发模式, 重点介绍了当今流行的 MVC 模式, 为以

后学生学习其他框架奠定了基础。

本书编委成员，主编：董红安、李国增，副主编：张永军、王喜年、赵雪峰。

北京翰子昂教育科技有限公司

2007年12月

目 录

专题一 Web 开发基础	1
1. 教学目标.....	1
2. 工作任务.....	1
3. 相关实践知识.....	1
3.1 使用“记事本+Tomcat”编写 HelloWorld	1
3.2 Eclipse+TomcatPlugin+Tomcat 开发环境的配置	4
3.3 使用 IDE 实现 HelloWorld	5
4. 相关理论知识.....	6
4.1 Web 应用程序的体系结构.....	6
4.2 Java Web 应用程序的目录结构.....	9
4.3 Java Web 应用程序的开发过程.....	10
4.4 常用开发工具以及应用服务器.....	10
5. 实验.....	16
6. 课后作业.....	16
专题二 Servlet 技术	17
教学目标.....	17
案例一 简单计数器.....	18
1. 教学目标.....	18
2. 工作任务.....	18
3. 相关实践知识.....	18
3.1 简单的计数器，记录个人访问此页面的次数	18
3.2 实现输出“Welcome to Servlet World”程序.....	20
4. 相关理论知识.....	21
4.1 Servlet 基础	21
4.2 Servlet 的生命周期	23
4.3 配置 Servlet	24
5. 实验.....	25
6. 课后作业.....	26
案例二 一个简单购物车.....	27
1. 教学目标.....	27
2. 工作任务.....	27
3. 相关实践知识.....	27
3.1 用 Servlet 实现购物车功能.....	27
3.2 用 Cookie 实现登录网页记录时间的功能	31
4. 相关理论知识.....	33
4.1 Servlet 体系结构	33
4.2 Servlet 常用接口的使用	34
4.3 会话跟踪.....	38
4.4 HttpSession 类主要方法	40
5. 实验.....	41

6. 课后作业.....	41
案例三 Servlet 通信方法和上下文状态.....	42
1. 教学目标.....	42
2. 工作任务.....	42
3. 相关实践知识.....	42
4. 相关理论知识.....	45
4.1 Servlet 通信方式	45
4.2 Servlet 上下文	48
5. 实验.....	49
6. 课后作业.....	49
专题三 JSP 基础知识.....	51
1. 教学目标.....	51
2. 工作任务.....	51
3. 相关实践知识.....	51
3.1 html 与 jsp 页面.....	51
3.2 写一个 Servlet 用于验证码的输出.....	57
3.3 在 message.jsp 中加入 include 指令	60
4. 相关理论知识.....	63
4.1 JSP 概念.....	63
4.2 JSP 与 SERVLET 的区别.....	64
4.3 JSP 的执行过程.....	64
4.4 JSP 的组成元素.....	66
4.5 JSP 编码的注意事项.....	68
5. 试验.....	71
6. 作业.....	71
专题四 JSP 脚本和指令.....	73
1. 教学目标.....	73
2. 工作任务.....	73
3. 相关实践知识.....	73
4. 相关理论知识.....	76
4.1 JSP 声明介绍.....	76
4.2 JSP 脚本元素.....	77
4.3 JSP 指令.....	79
5. 试验.....	82
6. 课后作业.....	82
专题五 JSP 隐含对象.....	83
1. 教学目标.....	83
2. 工作任务.....	83
3. 相关实践知识.....	83
3.1 使用 JSP 内置对象实现用户登录功能, 显示用户配置信息	83
3.2 使用 JSP 内置对象显示当前请求的容器信息.....	87
4. 相关理论知识.....	90

4.1 JSP 内置对象介绍.....	90
4.2 JSP 各个内置对象用法.....	91
5. 实验.....	94
6. 课后作业.....	94
专题六 JavaBean 和 JSP 中的标准动作	95
1. 教学目标.....	95
2. 工作任务.....	95
3. 相关实践知识.....	95
4. 相关理论知识.....	101
4.1 JavaBean 组件技术	101
4.2 JSP 中的标准动作标签.....	102
4.3 JavaBean 的 Scope 属性.....	106
5. 试验.....	108
6. 课后作业.....	109
专题七 JSTL 知识.....	111
教学目标.....	111
案例一 自定义标签显示时间.....	112
1. 教学目标.....	112
2. 工作任务.....	112
3. 相关实践知识.....	112
4. 相关理论知识.....	115
4.1 介绍标签基础知识.....	115
4.2 创建和使用自定义标签.....	117
4.3 简单标签生命周期.....	120
5. 试验.....	120
6. 课后作业.....	120
案例二 核心标签库的用法.....	122
1. 教学目标.....	122
2. 工作任务.....	122
3. 相关实践知识.....	122
4. 相关理论知识.....	132
4.1 EL 表达式.....	132
4.2 JSTL 简介	134
4.3 使用常用标签.....	134
4.4 使用条件标签.....	135
4.5 迭代标签.....	137
5. 试验.....	138
6. 课后作业.....	138
专题八 过滤和监听.....	139
教学目标.....	139
案例一 过滤器—Filter	140
1. 教学目标.....	140

2. 工作任务.....	140
3. 相关实践知识.....	140
4. 相关理论知识.....	148
4.1 过滤器简介.....	148
4.2 过滤器的运作方式.....	149
4.3 Filter 的定义与映射.....	149
4.4 Filter API 介绍.....	150
5. 实验.....	151
6. 练习.....	151
案例二 监听器实现在线用户统计.....	152
1. 教学目标.....	152
2. 工作任务.....	152
3. 相关实践知识.....	153
4. 相关理论知识.....	161
4.1 Servlet 监听器的概念.....	161
4.2 监听器类型.....	161
4.3 Servlet 监听器配置.....	164
5. 试验.....	164
6. 课后作业.....	164
专题九 MVC 模式.....	165
1. 教学目标.....	165
2. 工作任务.....	165
3. 相关实践知识.....	165
3.1 使用模式一 (JSP+JavaBean) 开发.....	165
3.2 使用模式二 (JSP+JavaBean+Servlet) 开发.....	172
4. 相关理论知识.....	178
4.1 JSP 技术使用的开发模式.....	178
4.2 MVC 模式介绍.....	178
4.3 应用 MVC 模式的必要性.....	179
4.4 比较两种模式.....	179
5. 实验.....	179
6. 课后作业.....	180

专题一 Web 开发基础

1. 教学目标

- 1.1 理解 Web 应用程序的体系结构
- 1.2 熟悉 Java Web 应用程序的目录结构
- 1.3 熟练掌握 Java Web 应用程序的开发过程
- 1.4 了解常用的开发工具以及应用服务器

2. 工作任务

- 2.1 使用“记事本+Tomcat”方式编写一个简单的 Web 应用程序——HelloWorld
- 2.2 Eclipse+TomcatPlugin+Tomcat 开发环境的配置
- 2.3 使用 IDE 实现 HelloWorld

3. 相关实践知识

3.1 使用“记事本+Tomcat”编写 HelloWorld

1. 在本地磁盘新建一个项目——HelloWorld。为了说明方便，我们使用\$MyAppPath 指代该项目的根路径。该项目目录结构如下图所示：



图 1-1 Web 应用程序目录结构

需要注意的是：WEB-INF 目录必须为大写。

2. 在目录\$MyAppPath/WEB-INF/src 下新建源程序：HelloWorldServlet.java。
3. 使用记事本编辑文件 HelloWorldServlet.java，输入以下代码并保存：

```
package handson;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
```

```
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class HelloWorldServlet extends HttpServlet {

    protected void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title> Hello World!</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Hello World!</h1>");
        out.println("</body>");
        out.println("</html>");
        out.close();
    }

    protected void doPost(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        this.doGet(request, response);
    }
}
```

4. 在目录\$MyAppPath/WEB-INF 下新建文件: web.xml (Java Web 应用程序部署描述文件)。在该文件中输入项目的描述信息并保存。代码如下:

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" version="2.5">
    <description>A simply Servlet example</description>
    <display-name>My first Servlet Application</display-name>
    <servlet>
        <servlet-name>HelloWorldServlet</servlet-name>
        <servlet-class>handson.HelloWorldServlet</servlet-class>
    </servlet>
    <servlet-mapping>
```

```
<servlet-name>HelloWorldServlet</servlet-name>
<url-pattern>/</url-pattern>
</servlet-mapping>
</web-app>
```

5. 在目录结构和相关源代码编写完成后, 需要将这些源文件编译成 JVM 使用的字节码文件。由于 HelloWorldServlet.java 使用了 javax.servlet 以及 javax.servlet.http 包中的 API, 因此在使用 javac 命令编译该文件之前, 必须先确定存放 Servlet API 的 JAR 文件的路径。

幸运的是, Tomcat 内部集成了这个 JAR (servlet-api.jar) 文件, 这样我们只需要将其设置进 classpath 参数就可以供 JVM 加载这些 API 了, 并最终成功编译 HelloWorldServlet.java 了。具体步骤如下:

- 单击“开始”菜单→“运行”(或 Win+R) 打开运行窗口;
- 输入 cmd, 单击“确定”;
- 将当前工作目录切换到 \$MyAppPath\WEB-INF;
- 运行 javac 编译 HelloWorldServlet.java, 如下:

```
javac -cp "%CATALINA_HOME%\lib\servlet-api.jar" -d classes
src\HelloWorldServlet.java
```

其中, CATALINA_HOME 是 Tomcat 的安装根目录, 如果之前没有设置该环境变量, 则使用 Tomcat 的安装根目录将其替换, 效果是一样的。

这时经过编译生成的字节码文件 HelloWorldServlet.class 会出现在目录: \$MyAppPath\WEB-INF\classes\handson 中。如果正确生成了该文件, 说明编译步骤完成。

6. 在编辑源码并编译成功之后, 就需要让程序运行起来。对于 Java Web 应用程序来说, 它必需运行在一个容器中。这里我们选择简单且具代表性的 Tomcat 6.0。具体步骤如下:

- 将项目 HelloWorld 的文件夹复制至 Tomcat 的 webapps 目录下。
- 启动 Tomcat 运行 HelloWorld 项目。

至此该项目的部署与启动流程已经完毕, 为了查看运行效果, 需要通过浏览器在浏览器端发出对该项目的请求。

- 打开浏览器, 在地址栏输入 URL: <http://localhost:8080/HelloWorld/>。
- 如果程序运行正常, 此时在浏览器页面中将会看到“Hello World!”显示。效果如图 1-2 所示:



图 1-2 运行效果

7. 使用 jar 命令将该项目打包到一个 WAR 文件，以方便项目的分发。在项目根目录\$My-AppPath 下执行如下命令：

```
jar cvf HelloWorld.war .
```

至此使用“记事本+Tomcat”编写 HelloWorld 的任务已经完成。使用这种方式编写程序，可以加深开发人员对 Web 应用程序开发步骤以及相关 API 的认识，但是，这样的开发速度根本无法适应企业的要求。现实中，项目组通常使用特定 IDE（集成开发环境）进行软件的开发过程。

下面介绍一种 IDE——Eclipse+TomcatPlugin+Tomcat 的配置方法。

3.2 Eclipse+TomcatPlugin+Tomcat 开发环境的配置

1. 获得相关软件正确版本。

软件名称	下载地址	版本
Tomcat	http://jakarta.apache.org	6.0.14 以上
Eclipse	http://www.eclipse.org	3.2.0 以上
Sysdeo/SQLI Eclipse Tomcat Launcher plugin	http://www.sysdeo.com/eclipse/Tomcatplugin	3.2.0 以上

表 1-1 软件配置

2. 安装 Eclipse，具体步骤这里不再详细介绍，需要强调的是：配置 Eclipse 中的 JRE 时，必须将其指定到已安装的 JDK 路径上。

3. 安装 Tomcat（具体步骤略）。

4. 将下载的 Sysdeo/SQLI Eclipse Tomcat Launcher plugin 插件释放到临时文件夹，这里得到一个新的文件夹 com.sysdeo.eclipse.tomcat_3.2.0.beta3，将这个文件夹剪贴到 Eclipse 的安装路径\plugins 目录下。

5. 重新启动 Eclipse。通常情况下，你马上会发现 Eclipse 有了两处变化：菜单栏中多了一个 Tomcat 选项，工具栏中多了三个按钮，上面是大家非常熟悉的 Tomcat 小猫。它们分别代表启动、停止和重新启动 Tomcat。如下图中的箭头所示：

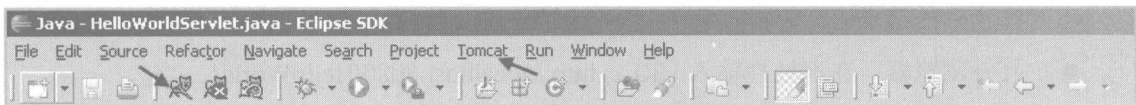


图 1-3 菜单栏和工具栏的变化

如果工具栏中没有出现三只小猫的按钮，你可以通过“窗口”→“定制透视图”→“命令”，在左侧的“可用的命令组”列表中，选择“Tomcat”选项让它现身。

6. 在进行完上述步骤后，需要对安装的插件进行设置，让它把 Eclipse 和 Tomcat 联系起来。点击“窗口”→“首选项”，选择 Tomcat 节点，选择 Tomcat 版本和设置 Tomcat 的安装目录，请参考下图：

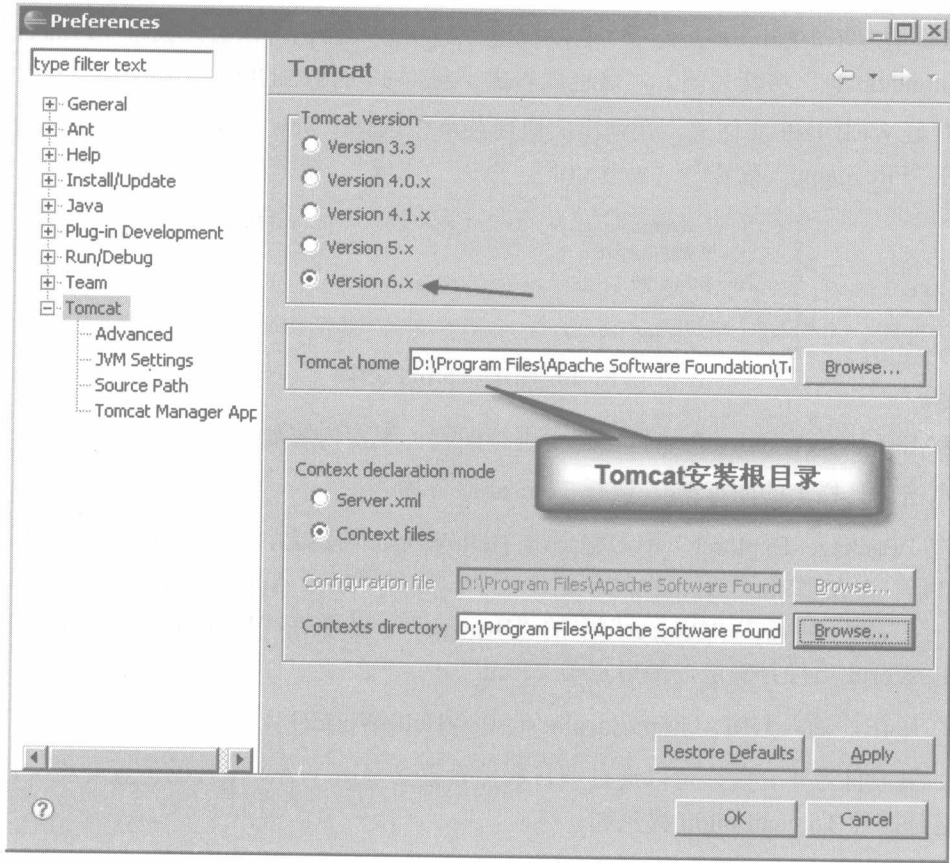


图 1-4 配置插件

至此，IDE 配置已经完成。

3.3 使用 IDE 实现 HelloWorld

有了 IDE，开发人员就可以在 Eclipse 中新建，编辑，调试并运行基于 Tomcat 的 Web 应用程序了。下面，在该 IDE 中重新编写一遍 HelloWorld 应用程序。具体步骤如下：

1. 点击菜单栏“File” → “New” → “Project” → “Java” → “Tomcat Project” → “New”，输入项目名 HelloWorld，然后点击“Finish”。这样就完成了项目的创建。生成的项目如下图所示：



图 1-5 新建 HelloWorld 项目

2. 打开“Java Perspective” → “Package Explorer”，右击 WEB-INF/src 文件夹，弹出快捷菜单，选择“New” → “Class”，按照提示新建 HelloWorldServlet.java 类文件，代码如前。

在默认情况下, Eclipse 会处在“自动构建”的状态(你也可以通过菜单“Project” → “Build Automatically”将其关闭), 这也就省去了我们手动构建的麻烦。

3. 右击 WEB-INF 文件夹, 在弹出的快捷菜单中选择“New” → “File”, 在“New File”对话框的“File name”处输入“web.xml”, 点击“完成”。结果如下图:

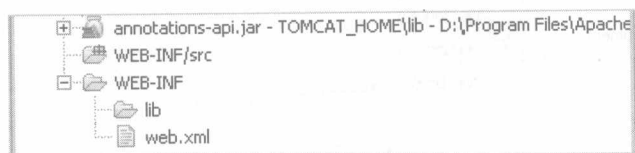


图 1-6 新建 web.xml 文件

4. 双击打开 web.xml, 在编辑窗口中进行配置。配置内容如前。

5. 点击工具栏上的图标, 运行 Tomcat 服务器。

6. 在“Package Explorer”中右键点击 HelloWorld 项目名, 选择“Run As” → “Open Run Dialog”, 在“Run”对话框的左侧一栏中, 展开“Java Application”一项, 点击 Tomcat 6.x, 在右侧一栏点击“Project”中的“Browse”, 接着在“Project Selection”对话框中选择 HelloWorld 项目并确定, 最后点击“Run”。

7. 打开 IE, 输入 URL (http://localhost:8080/HelloWorld/), 测试项目。

8. 如果项目需要分发, 可以通过项目属性设置要导出 WAR 文件的位置并将文件导出。这项功能是插件 TomcatPlugin 提供的。

至此使用 IDE 实现 HelloWorld 的任务完成。

4. 相关理论知识

4.1 Web 应用程序的体系结构

应用程序是什么, 简单来说, 它是现实世界的问题在计算机世界的解决方案。程序结构的选择在软件的开发过程中有着极其重要的作用, 它直接影响软件的开发过程以及后续的维护等环节, 因此选择合适的结构是软件开发的前提也是关键所在。

下面简单介绍应用程序体系结构的发展过程, 以及几种典型的应用程序结构。

组成应用程序的功能代码及其资源, 按照其在应用程序中的功能, 可以简单分解成为三个部分: 用户界面 (User Interface)、事务逻辑 (Business Logic) 和数据存取 (Data Access)。应用程序的基本功能单元如下图所示:

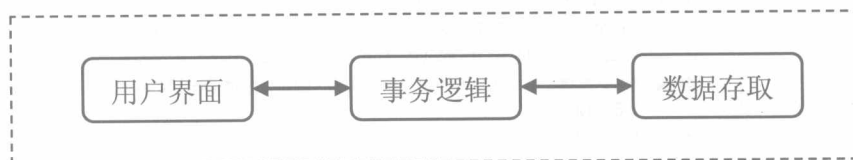


图 1-7 应用程序功能分解

4.1.1 单层体系结构

早期为大型机编写的应用程序, 没有把应用程序的三个组成部分分开, 即没有将用户界