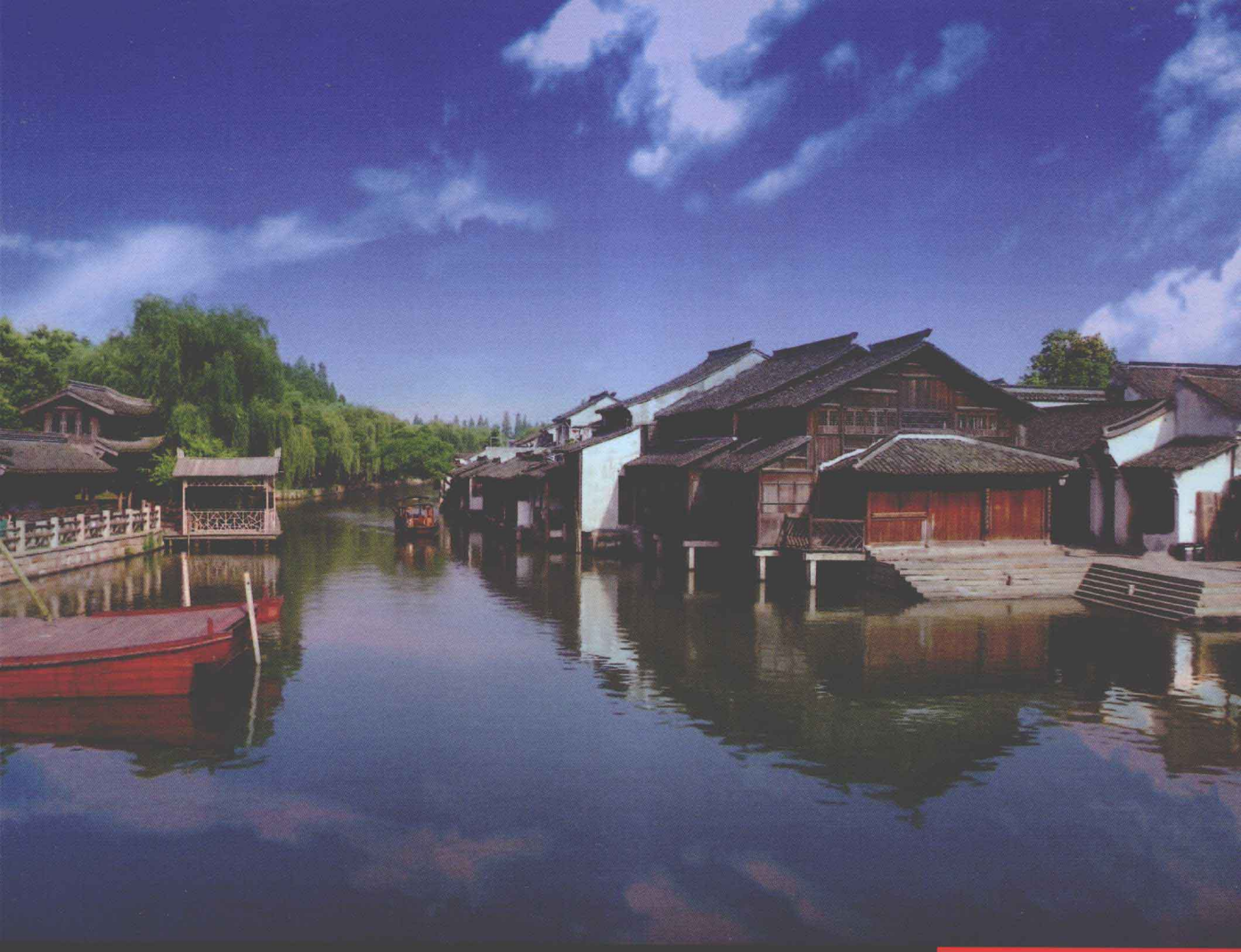

宋智军 邱仲潘 编著

JSP

从入门到精通（第二版）

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

内 容 简 介

本书坚持以语言为基础和以应用为主导的编写原则，理论联系实际，并通过大量的实例，循序渐进地为读者介绍了有关 JSP 开发所涉及的各类知识。全书共分为 11 章，首先从最基础的 JSP 开发环境的搭建开始，再通过介绍 JSP 基础、基本语法、内置对象、和 Servlet 结合、JavaBean 结合、使用 JSP 操作 XML 等，最后通过相应章节的知识点进行实例的讲解。

全书的基础知识介绍详细，理论联系实际，具有很强的操作性。本书还提供了大量的通过测试的可运行的完整实例代码，这些实例都有相应的设计步骤、代码详解、程序运行结果等，通过实例，不但可以使读者复习前面所学的内容，而且还增加了一定的创作技巧。本书适合技术支持人员和程序开发人员使用。

图书在版编目（CIP）数据

JSP 从入门到精通 / 宋智军, 邱仲潘编著. —2 版. —北京: 电子工业出版社, 2012.7
ISBN 978-7-121-17219-9

Ⅰ. ①J… Ⅱ. ①宋… ②邱… Ⅲ. ①JAVA 语言—网页制作工具 Ⅳ. ①TP393.092

中国版本图书馆 CIP 数据核字(2012)第 112286 号

责任编辑：戴　新
印　　刷：
装　　订：三河市鑫金马印装有限公司
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
　　北京市海淀区翠微东里甲 2 号　邮编：100036
开　　本：787×1092　1/16　印张：14　字数：360 千字
印　　次：2012 年 7 月第 1 次印刷
定　　价：29.00 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 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

JSP（Java Server Pages）是 Java EE 的基础和核心技术，是由 Sun 公司倡导并与许多其他公司参与一起建立的一种基于 Java 语言的动态网页技术，其实现方式是在传统的网页文件（*.html，*.htm）中插入 Java 程序段和 JSP 标记而形成的 JSP 文件（*.jsp）。为了帮助读者全面地了解 JSP 的知识体系及最新的 JSP 特性，笔者精心编著了本书。目前，关于 JSP 2.2 技术介绍方面的资料比较少，本书中关于 JSP 2.2 的新特性的介绍主要参考官方所发布的文档。而 JSP 2.2 版本和之前版本有诸多不同，对 JSP 2.2 的扩展性和简易性进行了很大的提升。关于 JSP 2.2 的最新特性可查看第一章的有关内容介绍。

本书依照循序渐进的学习规律，首先介绍了 JSP 的发展历史、基本概念和开发环境的搭建；在使读者掌握了这些基础知识之后，再结合大量的代码实例对各个知识点进行深入浅出的讲解，引领读者学习 JSP 和各种技术结合在项目开发中的实际应用。

本书由宋智军、邱仲潘编著，同时感谢冯姝慧、邹文、邓欣欣、王帅等给予的帮助和支持。另外，也非常感谢电子工业出版社的编辑和出版人员，他们为本书的选题策划、编辑加工和出版发行付出了辛勤的劳动。由于编者水平有限，加之时间仓促，书中难免有疏漏和不足之处，恳请专家和广大读者指正。

目　录

第 1 章　JSP 概述

JSP 是 Java Server Pages 的缩写，是由 Sun 公司倡导并与许多其他公司一起建立的一种基于 Java 语言的动态网页技术，其实现方式是在传统的网页文件（*.html,*.htm）中插入 Java 程序段和 JSP 标记而形成 JSP 文件（*.jsp）。

1.1　JSP 的诞生

在互联网的普及和发展的最初阶段，Web 应用全部是静态的 HTML 页面。这种静态网页的内容在设计时就确定好了，如要修改或维护页面就必须对 HTML 源代码进行修改。而客户端的浏览者只能浏览 HTML 页面里的一些固定的文本信息，不具备与用户交互的能力和动态显示功能。

随着 Internet 技术的不断深入，人们希望 Web 网页应该具备与用户交互的能力，因而出现了最早的 CGI（通用网关接口）技术。CGI 技术引入了“动态内容”的思想，能够根据需要随时生成 Web 网页内容，具有交互性。CGI 技术开启了动态 Web 应用的先河，但这种技术在开发动态 Web 应用时难度过大，而且在性能等各个方面也存在着很多限制。

1994 年，Rasmus Lerdorf 发明了专用于 Web 服务端编程的 PHP（Personal Home Page Tools）语言。它将 HTML 和 PHP 指令合成为完整的服务端动态页面，从而使开发者可以以更加简便、快捷的方式实现动态交互技术。

直到 1996 年，微软公司借鉴 PHP 的思想，开发出了 ASP（Active Server Page，意为“动态服务器页面”），并代替 CGI 技术成为了新一代的 Web 交互技术。ASP 是一种服务器端脚本编写环境，可以用来创建和运行动态网页或 Web 应用程序；可以与数据库和其他程序进行交互；可以包含 HTML 标记、普通文本、脚本命令以及 COM 组件等，是一种简单、方便的编程工具。

当 ASP 技术作为动态 Web 开发技术迅速成为了 Windows 系统下 Web 服务端的主流开发技术时，由 Sun 公司带领的 Java 团队也不甘示弱。1997 年，Servlet 技术问世；1998 年，Sun 公司发布了 JSP 标准，从某种程度上来说，这是为了对抗微软的 ASP 动态 Web 编程技术。JSP 和 ASP 两者都是动态 Web 编程技术，也都可以嵌入 HTML 中，但是它们的运行机制不同，这主要是因为 ASP 使用 VBScript 作为脚本语言，无需编译；而 JSP 则使用 Java 作为脚本语言，JSP 必须编译成 Servlet，才可以执行 JSP 页面。JSP 和 Servlet、JavaBean 技术的组合，大大提高了编译运行的执行效率，并逐渐发展成为 J2EE 平台的核心技术之一。

1.2　JSP 的特点

JSP 页面由 HTML 代码和嵌入其中的 Java 代码所组成。在页面被客户端请求以后，服务器开始对这些 Java 代码进行处理，然后将所生成的 HTML 页面返回给客户端浏览器。JSP 不

仅具备了 Java 技术的简单易用和完全面向对象等特点，而且具有平台无关性和安全可靠性。

JSP 1.0 规范版本是在 1999 年 9 月推出的，同年 12 月份又出现了 JSP 1.1 规范，目前最新的版本是 JSP 2.2 规范。JSP 2.2 主要针对 JSP 表达式语言（EL）以及与 JSF 的整合进行了改进，并把 JSP 2.2 构建在 J2SE1.5 的基础之上。它的重要新特性主要包括以下几部分。

■ 新的统一表达式语言。

新的统一表达式语言允许表达式延期求值，表达式既可以取值也可以赋值，还可以调用方法。

■ 简化配置访问资源或外界对象。

可以通过注解方式来实现资源注入，简化配置访问资源或外界对象，如访问 JNDI 对象。

■ 统一了 JSP 标签和 JSP 的实现。

通过统一表达式语言，统一了 JSF 标签和 JSP 的实现。

JSP 2.2 还有一些其他特性变化，具体的细节大家可以参考官方网所提供的相关资料，网址为：http://jcp.org/aboutJava/communityprocess/edr/jsr245/。这里不再赘述。

1.3 JSP 的运行机制

JSP 是服务器端技术，在服务器端，JSP 引擎解释并执行 JSP 页面的代码，然后将执行结果以 HTML 或 XML 页面的形式发送给客户端，而在客户端却看不到 JSP 页面本身的代码，只能看到 JSP 页面执行后的输出结果。下面通过具体的实例来说明 JSP 的运行机制。

● 首先新建一个 Web 项目，项目名为 “Hello World”，具体新建 Web 工程的方法请查看第 1.6.2 节常用操作中的“新建项目”内容。

● 修改 WebRoot 目录下的 index.jsp 文件，查找到如下代码。

```
<body>
    This is my JSP page. <br>
</body>
```

将以上代码修改如下内容。

```
<body>
    <%out.println("Hello World!");%>
</body>
```

● 最后把“Hello World!”项目打包发布（如何打包发布，请查看第 1.7 节打包和部署的相关内容），我们会看到如图 1.1 所示的结果。

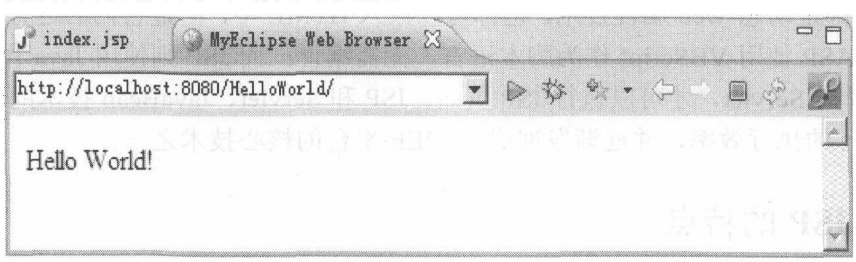

图 1.1 打包发布后运行结果

我们可以看到输出结果“Hello World!”，其中“<%” 和“%>”中的部分为 Java 语言代

码（服务器端的），而不是静态内容。通过这种方式就可以把 Java 代码嵌入到 HTML 页面中，并变成了动态的 JSP 页面。而我们查看客户端的 index.jsp 时，可以看到以下内容。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <base href="http://localhost:8080/HelloWorld/">

    <title>My JSP 'index.jsp' starting page</title>
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="expires" content="0">
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
<meta http-equiv="description" content="This is my page">
<!--
<link rel="stylesheet" type="text/css" href="styles.css">
-->
  </head>

  <body>
    Hello World!
  </body>
</html>
```

从表面上分析，JSP 页面已经不再需要 Java 类，但实现上 JSP 是 Servlet 的一种特殊形式，每个 JSP 页面都相当于一个 Servlet 的实例。因为 Web 项目中每个 JSP 页面首先都会由 Servlet 容器生成对应的 Servlet 类，这样才能被服务器响应给客户端。JSP 具体的运行机制如图 1.2 所示。

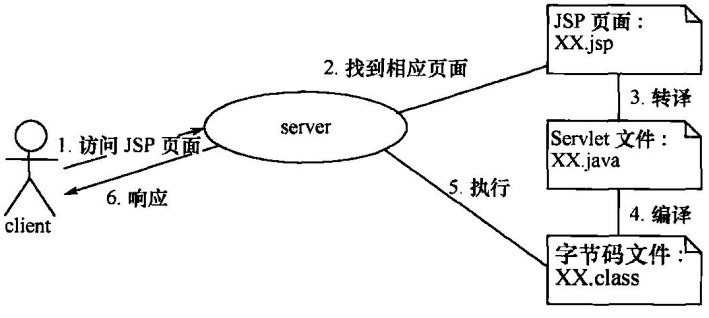

图 1.2　JSP 运行机制

图 1.2 所示的 JSP 流程如下。

■ 客户端（client）访问相应的 JSP 页面，并将访问请求发送到服务器端（Server）。

■ 服务器端接收到以“*.jsp”为扩展名的 HTTP 请求后，将这个请求交给一个 JSP 引擎，这个引擎是一个由 Tomcat 所提供的 Servlet 程序（org.apache.jsper.servlet.JspServlet）。

■ 当一个 JSP 页面第一次被访问时，JSP 引擎会把该页面翻译成为一个 Servlet 源程序，

再把这个源程序编译为“*.class”文件。

■ 最后，服务器将 class 文件交由 Servlet 容器加载到内存并执行，然后将执行后的 HTML 代码发送给客户端，因而我们看到客户端的代码与服务器的代码是不同的。

1.4 开发环境和工具

JSP 是 Java EE 的重要组成部分，要进行 JSP 开发，至少需要用到以下软件工具。

● JDK：Java 开发者工具包，用来对编写的 Java 源程序进行编译，对部署描述符、类文件等进行打包，生成“.jar”文件。

● Java EE 兼容的服务器：用来对上一步生成的“.jar”文件进行部署。

● Java 源代码编辑工具。

由于 Java 的开放性，各种免费、开源的 Java EE 开发工具也十分流行，它们具有占用内存小、运行速度快等优点，常用的免费 Java EE 工具有以下几种。

● Ant：用来对 Java EE 程序进行编译和部署。

● Eclipse：IBM 公司将 Websphere Application Developer Studio 基础源代码公开，交给开源社区后发展非常迅速的、免费的 Java 源代码编辑工具，并且 Eclipse 支持第三方的插件来扩展其功能。目前 Eclipse 可以通过安装第三方的 Java EE 插件来支持 JSP 的开发，并且 Eclipse 内置了支持 Ant 的插件。

● Netbeans：Sun 公司将 Java Enterprise Developer Studio 的基础源代码公开，交给开源社区后形成的免费 Java 开发工具。

本书采用 Eclipse 结合相应的 MyEclipse 插件来介绍 JSP 的开发和测试。

由于JSP 是服务器端技术，所以需要有相应的服务器端应用服务服务器。常见的应用服务器有以下几种。

● Sun 公司的 Java EE 企业版：免费的 Java EE 容器，可作为 Java EE 功能的演示和教学版。

● IBM 公司的 Websphere Application Server：市场占有率最高的应用服务器，因为其具有非常好的稳定性，常被用做重要电子商务场合的应用服务器。

● BEA 公司的 Weblogic：市场占有率仅次于 Websphere，对 Java EE 标准支持的效果比较好，具有较好的执行速度。

● 开源免费的 JBoss：无需安装，速度、性能都十分优异，对系统要求较低，部署 EJB 的速度非常迅速。

● Apache 软件基金会的 Jakarta 项目中的一个核心项目 Tomcat：运行时占用的系统资源较少，扩展性好，支持负载平衡与邮件服务等开发应用系统常用的功能。

本书采用 Tomcat 作为 JSP 的应用服务器，因为 Tomcat 是一个轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，技术先进、性能稳定且可免费使用，而且有了 Sun 公司的参与和支持，Tomcat 7.x 支持最新的 Servlet 3.0、JSP 2.2 和 EL 2.2 等，是开发和调试 JSP 程序的首选。

1.5 搭建开发环境

在前面已经介绍过，要进行 JSP 开发至少需要用到的软件工具，本书中的 JSP 实例采用

了 Tomcat 7.0 服务器，用 JDK 7、Eclipse IDE for Java EE Developers 以及 MyEclipse 10.0 插件进行构建 JSP 开发环境。

1.5.1　JDK 安装配置

JDK 的最新版本（目前 JDK 最新版本为 JDK 7）可以从官网下载（网址为：http://www.oracle.com/technetwork/java/javase/downloads/index.html）。下载完成后进行安装，安装步骤如下。

● 安装 JDK 7。

单击 jdk-7u1-windows-i586.exe，进入如图 1.3 所示界面。

如要采用默认安装，只需要按照安装向导一步一步单击【下一步】按钮即可，直到单击到【完成】按钮，JDK 安装成功。

● 设置环境变量。

安装完 JDK 后，还需要设置环境变量，在这里我们一共设置了三个环境变量：PATH、CLASSPATH 和 JAVA_HOME。设置方法如下。

■ 右击“我的电脑”→“属性”→“高级”→“环境变量”→“系统变量”，系统将出现如图 1.4 所示的界面。

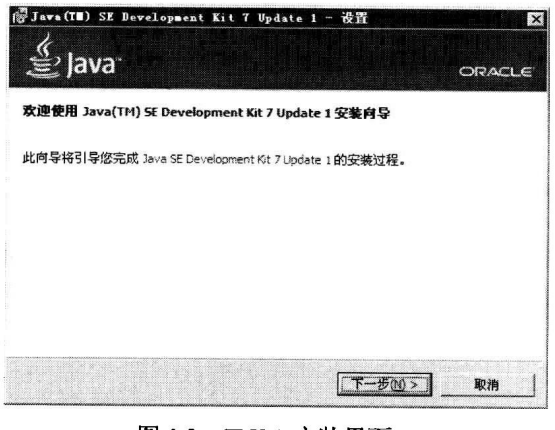

图 1.3　JDK 7 安装界面

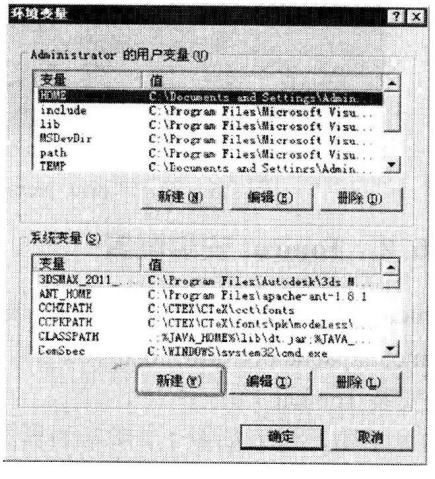

图 1.4　JDK 环境变量设置

■ 在“系统变量”中设置上面提到的三个环境变量，如果变量已经存在就选择“编辑”按钮，否则选“新建”按钮。单击系统变量下的“新建”按钮，系统将出现“新建系统变量”对话框，如图 1.5 所示。

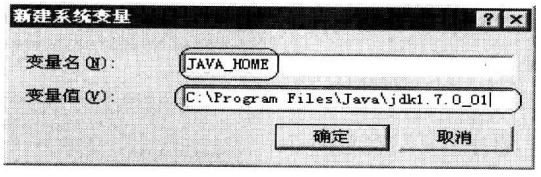

图 1.5　“新建系统变量”对话框

三个变量的设置如表 1.1 所示。

表 1.1 三个变量的设置

变量名	变量值
JAVA_HOME	指明 JDK 安装路径，就是刚才安装时所选择的路径（默认路径为：C:\Program Files\Java\jdk1.7.0_01），此路径下包括 lib，bin，jre 等文件夹
PATH	使得系统可以在任何路径下识别 java 命令，设置为：%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin
CLASSPATH	CLASSPATH 为 java 加载类路径，只有类在 CLASSPATH 中，java 命令才能识别，设置为：.;%JAVA_HOME%\lib;%JAVA_HOME%\lib\tools.jar 注意：CLASSPATH 变量是以“.;”开始的，它表示当前路径

■ 环境变量设置完成后，单击“确定”按钮，退出环境变量设置。

■ JDK 检验：JDK 安装完成后，可以检验一下，看 JDK 是否安装成功。在 DOS 命令行窗口中，键入“java –version”命令可以查看到安装的 JDK 的版本信息，如图 1.6 所示；键入 java 命令，可以看到此命令的帮助信息，如此则说明安装成功了。

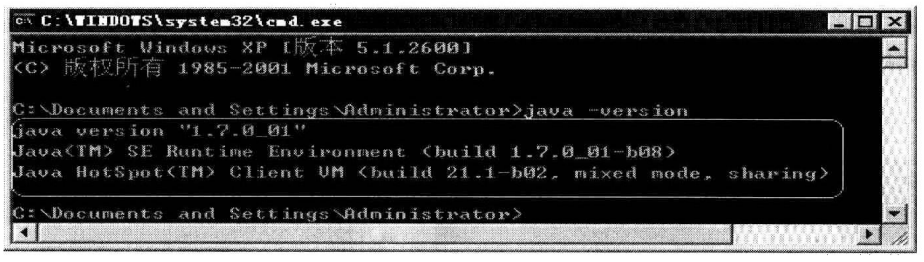

图 1.6 键入“java –version”命令查看到安装的 JDK 版本信息

1.5.2 Tomcat 安装配置

Tomcat 的最新版本（目前 Tomcat 最新版本为 Tomcat 7.0）可以从官网下载（网址为：http://tomcat.apache.org/download-70.cgi）。下载完成后进行安装，安装步骤如下。

● 安装 Tomcat 7.0。

（1）单击，进入如图 1.7 所示的界面。

（2）单击【Next】按钮，进入用户安装的许可协议确认界面，如图 1.8 所示。

图 1.7 Tomcat 7.0 安装界面

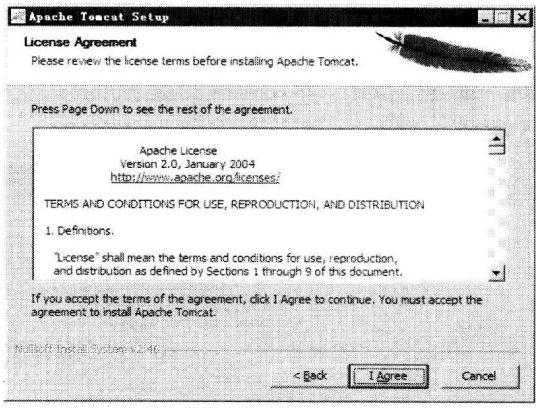

图 1.8 用户安装的许可协议确认界面

（3）单击【I　Agree】按钮，只需要按照安装向导一步一步单击【Next】按钮即可，直到单击到出现如图 1.9 所示的界面。

Tomcat 为我们预设了相应的 Port 端口，我们可以使用默认的端口，也可以根据需要修改相应的端口（一般选择默认即可）。Tomcat 有相应的可选属性，如设置相应的管理账号和密码，为了方便起见，这里的账号和密码都设置为“Admin”。最后按照安装向导进行相应的选择，即可完成 Tomcat 的安装。

● 启动 Tomcat。

安装完 Tomcat 后，我们就要启动 Tomcat。启动方法如下。

■ 单击任务栏的【开始】→【程序】→ Apache Tomcat 7.0 Tomcat7 → Configure Tomcat，系统将出现如图 1.10 所示的界面。

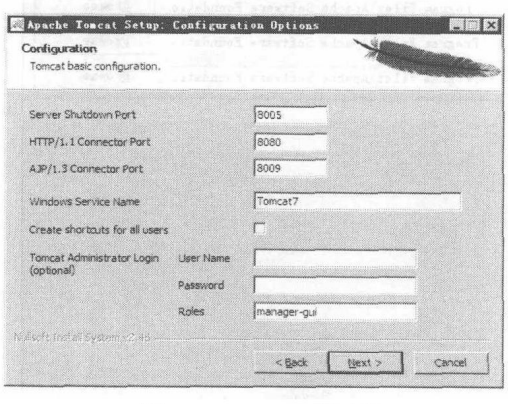

图 1.9　Tomcat 配置框

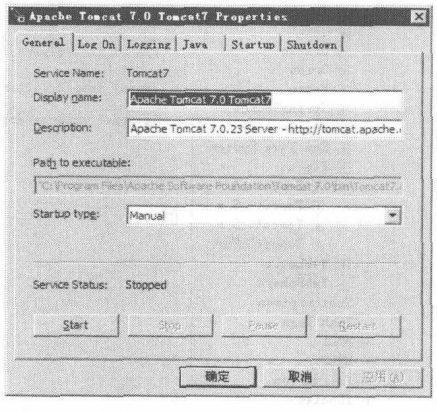

图 1.10　Tomcat 启动界面

■ 单击【Start】按钮，对 Tomcat 进行启动，启动成功后，打开浏览器输入本机 IP 再加上前面设置的 8080 端口号（如网址：http://localhost:8080/ 或 http://127.0.0.1:8080/ ），如出现如图 1.11 所示的欢迎界面，则 Tomcat 安装成功。

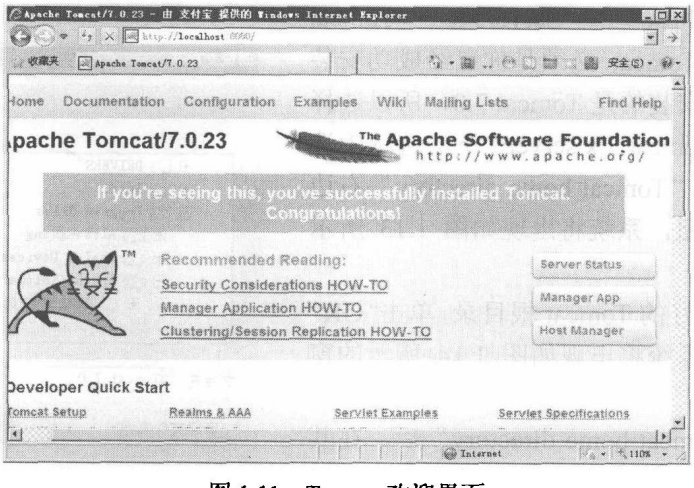

图 1.11　Tomcat 欢迎界面

1.5.3　MyEclipse 的安装配置

由于 MyEclipse 10.0 本身就集成了 Eclipse，所以我们只要安装 MyEclipse 即可。在安装

MyEclipse 时，只需要按照安装向导一步一步进行即可，直到单击到【完成】按钮，MyEclipse 安装成功。MyEclipse 最新版下载地址为：http://www.myeclipseide.com/。

Tomcat 和 MyEclipse 都安装成功后，下一步要在 MyEclipse 中配置 Tomcat，具体的步骤如下。

● 打开 MyEclipse，选择 MyEclipse 菜单栏下的“Window”→“Preferences”，打开 MyEclipse 的参数配置窗口。

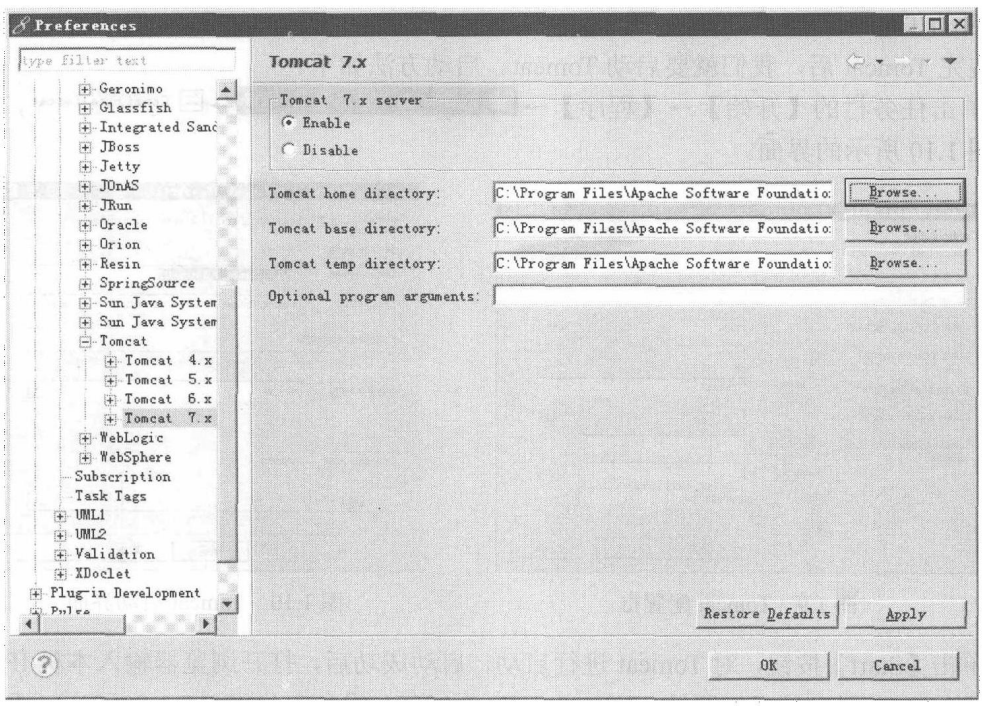

图 1.12　MyEclipse 的参数配置窗口

● 选择“MyEclipse”→“Servers”→“Tomcat 7.x”（这时选择的 Tomcat 必须是你已经成功安装的版本，由于我们安装的是 Tomcat 7.0，所以选择 Tomcat 7.x）。在“Tomcat 7.x server”下的选项中选择“Enable”，单击“Tomcat home directory”右边的“Browse...”按钮，系统将出现如图 1.13 所示的浏览文件夹窗口。

图 1.13　浏览文件夹窗口

● 选择已安装好的 Tomcat 根目录，单击“OK”按钮进入下一步，系统将出现如图 1.14 所示的窗口。

● 选择了“Tomcat home directory”后，在其下面的两项是系统自动生成的，不用更改，选择“Tomcat 7.x server”下的“Enable”单选按钮，再单击“Apply”按钮。然后选中 Tomcat 7.x 中的 JDK 一项，这里要特别注意，系统默认的是 JRE 的运行环境，这里要设定成 JDK 的。否则，MyEclipse

无法正常部署 Web 应用，也无法正常运行 Tomcat 服务器，我们单击“Add JDK”按钮，系统就会出现如图 1.15 所示的窗口。

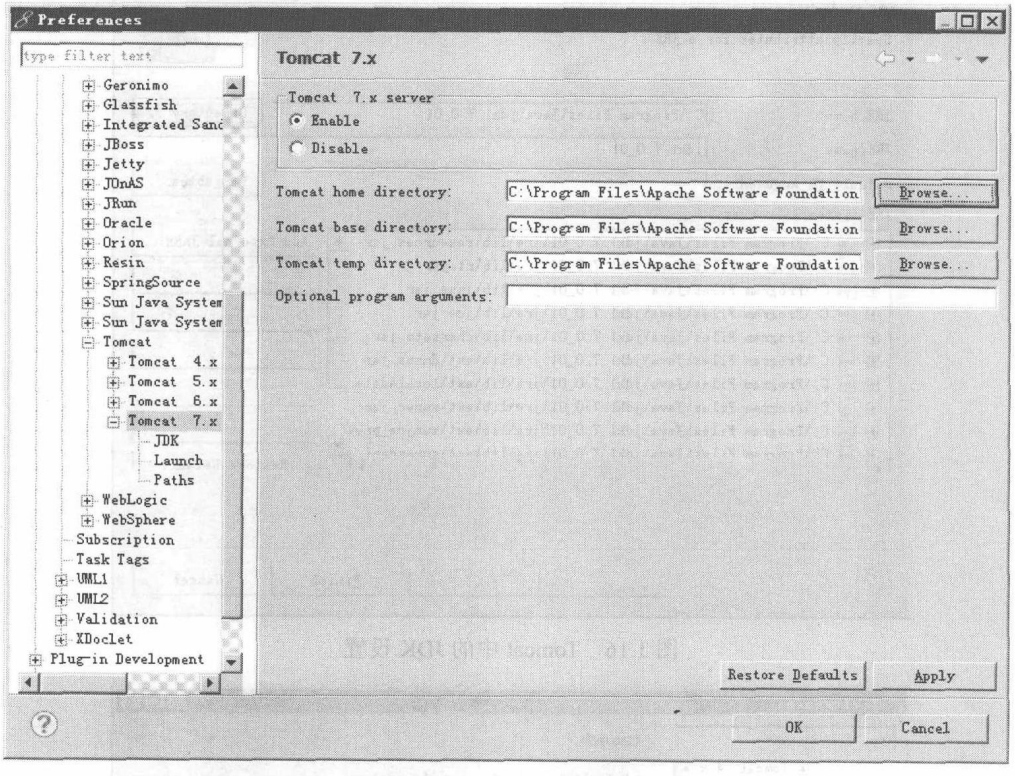

图 1.14　Tomcat 7.x 配置窗口

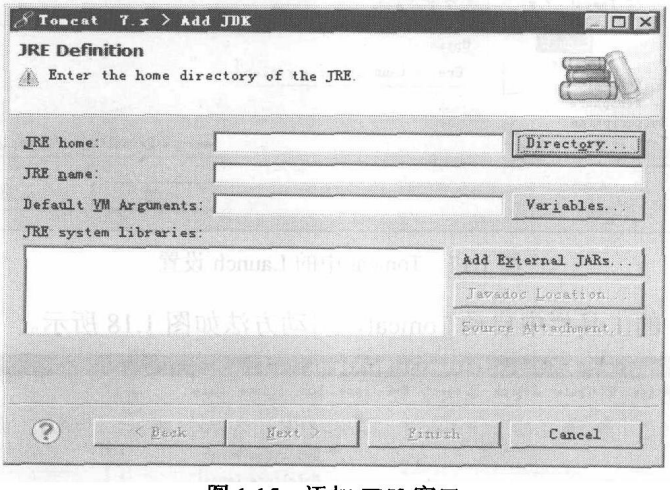

图 1.15　添加 JDK 窗口

- 单击“Directory...”按钮选择 JDK 的根目录，即可出现如图 1.16 所示的界面。
- 在图 1.17 所示的窗口中左侧单击“Launch”，在“Default Tomcat 7.x launch mode”下选择“Run mode”单选按钮，最后单击“Apply” →“OK”完成 Tomcat 在 Myecliipse 中的设置。

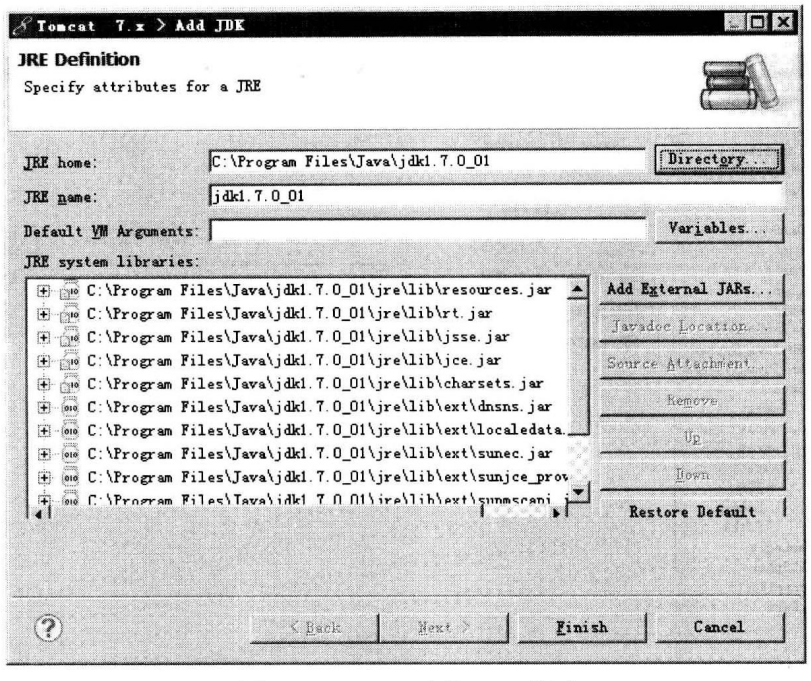

图 1.16　Tomcat 中的 JDK 设置

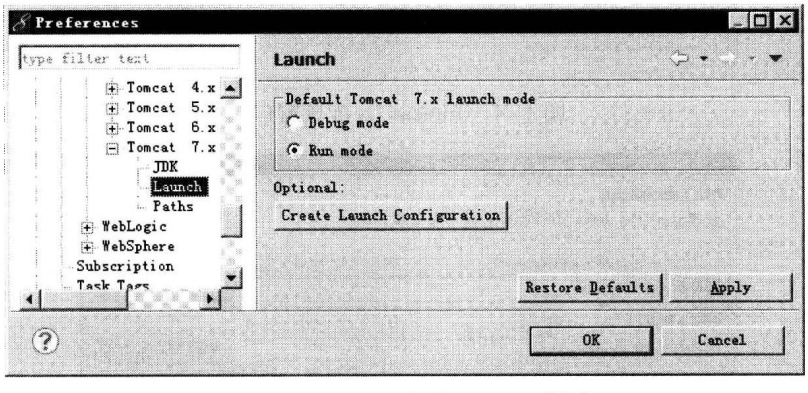

图 1.17　Tomcat 中的 Launch 设置

- 在 MyEclipse 的工具栏中启动 Tomcat，启动方法如图 1.18 所示。

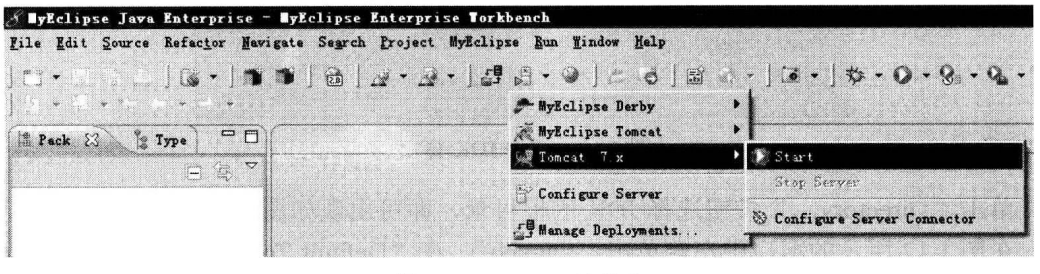

图 1.18　Tomcat 的启动

当然，我们也可以使用 MyEclipse 自带的 Tomcat 和 JDK，为了让大家更清楚地理解如何根据自己的需要进行环境的配置，我们在前面的内容中进行了详细的讲解。如果启动成功，

系统将会出现如图 1.19 所示的提示界面。

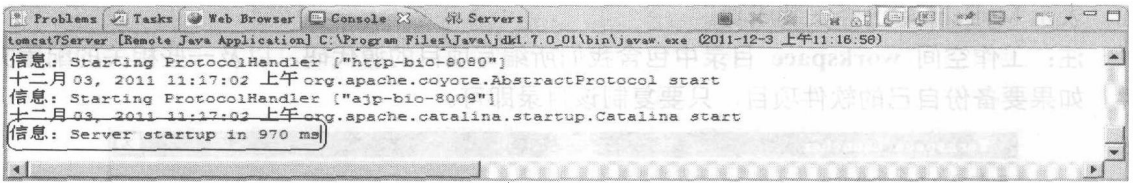

图 1.19　Tomcat 在 MyEclipse 中启动成功

如果系统没有显示如图 1.19 所示的信息，那么用户最好先在 Tomcat 的 bin 目录下运行“Tomcat 7.exe”启动一下，看是否能成功。如果这样能启动成功而不能从 MyEclipse 中启动成功，那这就是在 MyEclipse 中的配置存在问题。如果在 MyEclipse 下开启服务器时总会出现“Could not find the main class”的提示语，这很可能是因为 Tomcat 路径名中包含空格，请重新设置 Tomcat 路径。

Tomcat 在 MyEclipse 中配置并启动成功后，单击“Open MyEclipse Web Browser”按钮，在地址栏中输入“http://localhost:8080/”，即可显示出 Tomcat 欢迎主界面，如图 1.20 所示。

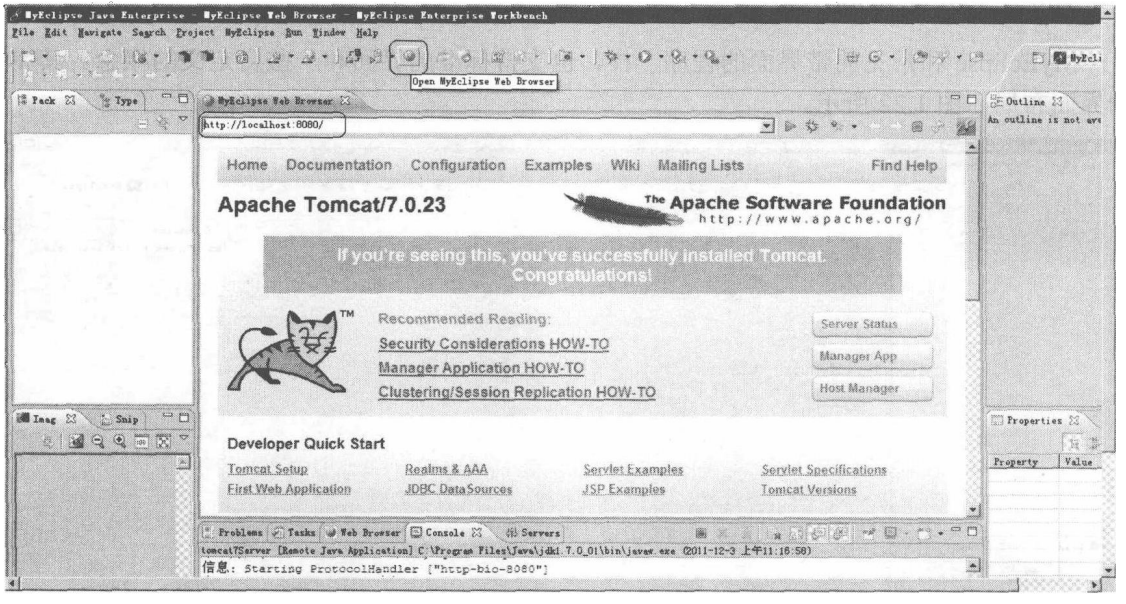

图 1.20　MyEclipse 中的 Tomcat 欢迎主界面

1.6　熟悉 MyEclipse 开发工具

本节内容让您对 MyEclipse 有一个快速的了解，便于以后进行 JSP 的开发。如果您已经很熟悉 MyEclipse 开发工具，那么就可以跳过这一节内容。前面我们已经成功安装了 MyEclipse，现在试运行一下 MyEclipse，检查它是否安装成功了。可以通过“开始”→“所有程序”→“MyEclipse”→“MyEclipse 10.0”来启动 MyEclipse。第一次启动时，系统将会弹出一个设置工作空间路径的对话框，如图 1.21 所示。

选择相应的工作路径，并且将其相关选项设置好，这样以后启动 MyEclipse 时就不会再弹