

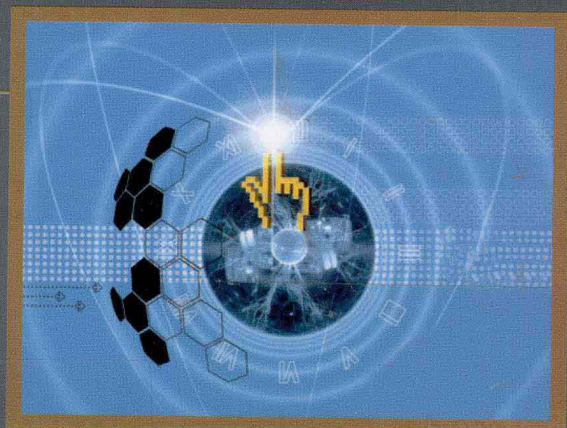


高等学校计算机科学与技术教材

JSP Web应用教程

(修订本)

□ 朱敏 黎方晟 朱晴婷 编著



- 原理与技术的完美结合
- 教学与科研的最新成果
- 语言精炼，实例丰富
- 可操作性强，实用性突出



清华大学出版社

● 北京交通大学出版社

[illegible]

高等学校计算机科学与技术教材

JSP Web 应用教程

(修订本)

朱 敏 黎方晟 朱晴婷 编著

清华大学出版社

北京交通大学出版社

• 北京 •

内 容 简 介

本书由浅入深, 详尽地介绍了 JSP 的基本原理与相关技术; 并从实际应用出发, 引导读者学习和掌握 JSP 这一网络开发技术。本书共分 8 章, 介绍了 JSP 的开发和运行环境, 以及 JSP、Java、JavaScript、Servlets、JavaBean、JDBC 等技术, 最后通过实际案例, 介绍了实际案例的设计、开发方法, 培养广大读者的实际应用能力。

本书可作为高等院校计算机应用专业的教学用书, 也适合已经具有初级计算机应用能力水平, 且具有一定的编程能力, 希望在网页网站开发和应用能力方面得到进一步提高的广大编程爱好者, 以及网页网站开发人员。

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 13501256678 13801310933

图书在版编目 (CIP) 数据

JSP Web 应用教程 / 朱敏, 黎方晟, 朱晴婷编著. —修订本. —北京: 清华大学出版社;
北京交通大学出版社, 2010.9

(高等学校计算机科学与技术教材)

ISBN 978-7-5121-0285-9

I. ①J… II. ①朱… ②黎… ③朱… III. ①JAVA 语言-主页制作-程序设计-高等学校-教材 IV. ①TP393.092

中国版本图书馆 CIP 数据核字 (2010) 第 167611 号

责任编辑: 谭文芳

出版发行: 清华大学出版社 邮编: 100084 电话: 010-62776969 <http://www.tup.com.cn>

北京交通大学出版社 邮编: 100044 电话: 010-51686414 <http://press.bjtu.edu.cn>

印刷者: 北京瑞达方舟印务有限公司

经 销: 全国新华书店

开 本: 185×260 印张: 13.75 字数: 346 千字

版 次: 2010 年 9 月第 1 版第 1 次修订 2010 年 9 月第 1 次印刷

书 号: ISBN 978-7-5121-0285-9/TP·615

印 数: 1~4000 册 定价: 24.00 元

本书如有质量问题, 请向北京交通大学出版社质监局反映。对您的意见和批评, 我们表示欢迎和感谢。

投诉电话: 010-51686043, 51686008; 传真: 010-62225406; E-mail: press@bjtu.edu.cn。

目 录

第 1 章 JSP 简介	1
1.1 JSP 的开发背景及发展历史	1
1.2 JSP 与其他动态网页技术	2
1.2.1 JSP 与 CGI 比较	2
1.2.2 JSP 与 ASP/ASP.NET 的比较	2
1.2.3 JSP 与 PHP 的比较	3
1.3 JSP 的优点	3
1.4 建立 JSP 的运行环境	4
1.4.1 JDK 的下载和安装	4
1.4.2 Tomcat 的下载和安装	6
1.4.3 Eclipse 的下载和安装	6
1.4.4 Ant 使用简介	8
1.4.5 MySQL 的下载和安装	9
小结	11
习题	11
第 2 章 Java 语言基础	12
2.1 Java 语言简介	12
2.1.1 什么是 Java	12
2.1.2 Java 语言的特点	13
2.1.3 Java 与 C/C++ 的差异	15
2.2 数据类型和表达式	16
2.2.1 标识符和关键字	16
2.2.2 简单数据类型	17
2.2.3 运算符和表达式	21
2.3 流程控制语句	22
2.3.1 分支选择语句	23
2.3.2 循环语句	26
2.3.3 转向语句	29
2.4 类、数组和字符串	31
2.4.1 类和对象	31
2.4.2 数组	35
2.4.3 字符串	37
2.5 程序包和异常处理	40

2.5.1	程序包	40
2.5.2	异常	41
小结		45
习题		45
第 3 章	HTML 和 JavaScript 基础	47
3.1	HTML 简介	47
3.1.1	HTML 的起源和发展现状	47
3.1.2	HTML 页面的结构	47
3.2	常用 HTML 标签和属性	48
3.2.1	超链接	48
3.2.2	表单和表单元素	49
3.2.3	表格	57
3.3	JavaScript 简介	58
3.3.1	起源、发展和现状	58
3.3.2	JavaScript 示例	58
3.4	HTTP 基础	59
3.4.1	GET	59
3.4.2	POST	60
3.5	JavaScript 基础	61
3.5.1	JavaScript 语法	61
3.5.2	JavaScript 事件	61
3.5.3	常用 JavaScript 对象和方法	62
3.6	实例	63
3.6.1	验证页面输入	63
3.6.2	动态显示、隐藏错误信息	64
小结		66
习题		66
第 4 章	JSP 语法和标签	67
4.1	JSP 基本语法	67
4.1.1	JSP 基本语法示例	67
4.1.2	注释	71
4.1.3	转义符	73
4.1.4	JSP 脚本元素 (Scriptlet)	74
4.1.5	JSP 指令元素	76
4.2	JSP 标准操作元素	80
4.2.1	include 操作	81
4.2.2	forward 操作	85
4.2.3	plugin 操作	86
4.2.4	Java Bean 的操作	88

4.3	JSP 内置对象	88
4.3.1	request 对象	88
4.3.2	response 对象	93
4.3.3	out 对象	97
4.3.4	application 对象	98
4.3.5	session 对象	100
4.4	其他内置对象	105
4.4.1	config 对象	105
4.4.2	exception 对象	105
4.4.3	page 对象和 pageContext 对象	105
小结		105
习题		107
第 5 章	Servlets 与 JavaBean	108
5.1	Servlets 的基本概念	108
5.1.1	Servlets 的结构	108
5.1.2	Servlets 的接口	109
5.1.3	Servlets 的生命周期	112
5.2	JSP 和 Servlets	113
5.2.1	JSP 和 Servlets 的关系	113
5.2.2	JSP 和 Servlets 的区别	113
5.2.3	JSP 使用 Servlets	114
5.2.4	Servlets 访问 JSP	115
5.3	Servlets 的应用	116
5.3.1	Servlets 的开发	116
5.3.2	Servlets 对数据库操作	120
5.4	JavaBean	129
5.4.1	JavaBean 简介	129
5.4.2	JavaBean 的属性	131
5.5	在 JSP 中使用 JavaBean	132
5.5.1	在 JSP 中使用 JavaBean 相关标签	132
5.5.2	在 JSP 中使用 JavaBean 应用实例	134
5.6	Servlets 与 JavaBean	136
小结		146
习题		146
第 6 章	数据库基础	148
6.1	基本概念	148
6.1.1	关系型数据库的起源、发展和现状	148
6.1.2	关系型数据库的结构	148
6.1.3	数据类型	148

6.1.4 数据库示例	150
6.2 结构化查询语言 (SQL)	152
6.2.1 查询 (SELECT)	152
6.2.2 插入 (INSERT INTO)	157
6.2.3 更新 (UPDATE)	157
6.2.4 删除 (DELETE)	158
6.3 数据库设计	158
6.3.1 未优化的表	158
6.3.2 优化后的结构	159
小结	159
习题	159
第 7 章 JSP 与 JDBC 数据库连接	161
7.1 JDBC 基础	161
7.1.1 JDBC 概要	161
7.1.2 数据库的连接	162
7.1.3 数据库操作	162
7.1.4 ResultSet 简介	167
7.2 用 JDBC 访问数据库	172
7.3 JSP 使用 JavaBean 操作数据库	174
7.4 访问数据库实现	178
7.4.1 检索数据分页处理	178
7.4.2 用 Hashtable 处理数据库操作	181
小结	184
习题	184
第 8 章 Web 应用——网上选课系统	185
8.1 系统介绍	185
8.1.1 背景介绍	185
8.1.2 需求分析	185
8.1.3 业务流程	186
8.2 系统设计	187
8.2.1 模块划分	187
8.2.2 界面设计	188
8.2.3 数据库设计	199
8.2.4 程序设计	202
小结	208
习题	208
参考文献	209

第 1 章 JSP 简介

JSP (Java Server Pages, Java 服务器主页) 技术为创建显示动态生成内容的 Web 页面提供了一个简捷而快速的方法。JSP 技术的设计目的使得构造基于 Web 的应用程序更加容易和快捷, 这些应用程序能够与各种 Web 服务器、应用服务器、浏览器和开发工具共同工作。

1.1 JSP 的开发背景及发展历史

在万维网短暂的历史中, 它已经从一个大部分显示静态信息的网络演化到动态地对信息进行交互和操作的网络。在各种各样的应用程序中, 对于使用基于 Web 的客户端, 几乎没有任何限制。

基于浏览器-服务器的应用程序比传统的基于客户-服务器的应用程序有许多好处。这些好处包括几乎没有限制的客户端访问及极其简化的应用程序部署和管理。如要更新一个应用程序, 管理人员只需要更改一个基于服务器的程序, 而不需要更改成千上万的安装在客户端的应用程序。因此, 软件工业正迅速地向着建造基于浏览器-服务器的多层次应用程序迈进。

这些快速增长的基于 Web 的应用程序要求开发技术上的改进, 静态 HTML (Hypertext Makeup Language, 超文本标记语言) 对于显示相对静态的内容是不错的选择; 新的挑战在于创建交互的基于 Web 的应用程序, 在这些程序中, 页面的内容是基于用户的请求或者系统的状态, 而不是预先定义的文字。

对于这个问题的一个早期解决方案是使用 CGI-BIN 接口; 开发人员编写与接口相关的单独程序, 以及基于 Web 的应用程序, 后者通过 Web 服务器来调用前者。这个方案存在严重的扩展性问题, 即每个新的 CGI (Common Gate Interface, 公共网关接口) 要求在服务器上新增一个进程。如果多个用户并发地访问该程序, 这些进程将消耗该 Web 服务器所有的可用资源, 并且系统性能降低到极其低下的地步。某些 Web 服务器供应商已经尝试通过为他们的服务器提供“插件”和 API (Application Programming Interface, 应用程序接口) 来简化 Web 应用程序的开发。这些解决方案与特定的 Web 服务器相关, 不能解决跨多个供应商的解决方案问题。例如, 微软的 ASP (Active Server Pages, 动态服务端页面) 技术使得在 Web 页面上创建动态内容更加容易, 但是它们也只能工作在微软的 IIS (Internet Information Server, Internet 信息服务器) 和 PWS (Personal Web Server, 个人 Web 服务器) 上。虽然还存在其他的解决方案, 但是都不能使一个普通的页面设计者能够轻松地掌握。例如, 像 Servlets 这样的技术就可以使得用 Java 语言编写交互应用程序的服务器端的代码变得容易, 但采用这种方法, 整个网页必须都在 Servlets 中制作, 如果开发人员或者 Web 管理人员想要调整页面显示, 就不得不编辑并重新编译该 Servlets, 采用这种方法, 生成带有动态内容的页面仍然需要应用程序的开发技巧。

显然,目前需要的是一个业界范围内的创建动态内容页面的解决方案。这个方案将解决当前方案所受到的限制,即:

- ✧ 能够在任何 Web 应用程序服务器上运行;
- ✧ 将应用程序逻辑和页面显示分离;
- ✧ 能够快速地开发和测试;
- ✧ 简化开发基于 Web 交互式应用程序的过程。

JSP 技术能够满足以上这些要求。JSP 规范是 Web 服务器、应用服务器、交易系统及开发工具供应商之间广泛合作的结果。Sun 公司开发出这个规范来整合和平衡已经存在的对 Java 编程环境(如 Servlets 和 JavaBeans)进行支持的技术和工具,其结果是产生了一种新的、开发基于 Web 应用程序的方法,给使用基于组件应用逻辑的页面设计者以强大的功能。

1.2 JSP 与其他动态网页技术

1.2.1 JSP 与 CGI 比较

JSP 与 CGI 最大的不同就是,编写 JSP 就好像编写 HTML 页面,只要了解简单的 JSP 语法就行了,无须熟悉 Java 等脚本语言。而 CGI 程序基本上就是用脚本语言直接写的,编写 CGI 程序,要求对脚本语言十分熟悉。另外,使用 CGI 程序时,它对每一次访问都要重新处理,这样在同时处理多个用户请求时,总显得力不从心。而 JSP 只编译一次,以后驻留于内存中,再次调用时不必重新处理,适于应付多用户情况。

1.2.2 JSP 与 ASP/ASP.NET 的比较

ASP 是由 Microsoft 提出的动态网页解决方案。它可以完成产生动态网页、处理 HTML 表单、创建与数据库相结合的网页、跟踪用户会话和利用定制的组件等众多的功能。

JSP 和 ASP 有很多共同之处,即:它们都提供在 HTML 代码中混合某种程序代码、由语言引擎解释执行程序代码的功能。HTML 代码主要负责描述信息的显示样式,而程序代码则用来描述处理逻辑。普通的 HTML 页面只依赖于 Web 服务器,而 JSP 和 ASP 页面还需要附加语言引擎来分析和执行程序代码。分析和执行完毕后,程序代码的执行结果以 HTML 代码形式发送给浏览器。JSP 和 ASP 都是面向 Web 服务器的技术,客户端浏览器则不需要任何附加的软件支持。总而言之,两者都为动态交互网页的制作提供了技术环境支持;两者都能够为程序开发人员提供程序的编写与自带组件设计从逻辑上分离的技术;两者都能够替代 CGI,使网站的建设发展变得较为简单和快捷。

JSP 的明显优势如下。

- ✧ 平台和服务器的独立性。基于 Java 的 JSP 与平台无关,它可以使用任何 Web 服务器(包括 Apache、IIS 和 PWS),而 ASP 技术主要依赖微软的技术支持,如果要跨平台的话,需要第三方 ASP 的引入产品。
- ✧ 使用的脚本语言。JSP 使用 Java 作为它的脚本语言,其面向对象的编程思想较为优秀,而 ASP 主要使用 VB、VBScript 和 JavaScript 等作为其脚本语言。

☞ JSP 跨平台的可重用性。JSP 组件 (Enterprise JavaBeans) 都是跨平台可重用的。EJB (Enterprise JavaBeans, 企业级 JavaBeans) 组件可以访问传统的数据库, 并能以分布式系统模式工作于 UNIX 和 Windows 平台, 而 ASP 还无法完全实现一些企业级的功能。

☞ 多样化和功能强大的开发工具支持。Java 已经有了许多非常优秀的开发工具而且有许多可以免费获取, 且其中有许多开发工具已经可以顺利地运行于多种平台之下。

Microsoft 在推出 ASP.NET 后提升了 ASP 对 JSP 的竞争力, 但是在跨平台的可重用性、开源框架方面仍然无法和 JSP 抗衡。

1.2.3 JSP 与 PHP 的比较

PHP 也是一门起源与 1995 年的动态网页编程语言。PHP 由于易于掌握、开发便捷和免费的 LAMP 架构 (Linux+Apache+MySQL+PHP) 受到中小网站开发团队的追捧。但是在建设大型商务网站方面 JSP 有 PHP 无法替代的优势。

☞ 数据库访问性能。同等条件下, JSP 访问 Oracle 数据库的时间开销大概是 PHP 的 20%。

☞ 分离内容的产生和显示。JSP 可以将重要的业务逻辑在 Servlets 中实现, 运行时需要编译成.class 文件, 有利于保护知识产权。

☞ 可重用组件。大型电子商务网站通常部署在多个服务器组成的服务器群组中, JSP 由于有 Servlets、JavaBean、Enterprise JavaBean 的支持, 使得跨服务器之间的调用有一个通用的规范。

1.3 JSP 的优点

1. 将内容的生成和显示进行分离

使用 JSP 技术, Web 页面开发人员可以使用 HTML 或者 XML 来设计页面的静态内容; 使用 JSP 生成页面的动态内容。JSP 是服务器端技术。在服务器端, JSP 引擎解释 JSP, 生成所有客户端请求的内容, 然后将结果以 HTML 或者 XML 页面的形式发送回客户端。这有助于开发人员既保护自己的代码, 又保证任何基于 HTML 的 Web 浏览器的完全可用性。

2. 强调可重用性组件

绝大多数 JSP 页面依赖于可重用的、跨平台的组件 (JavaBeans 或者 Enterprise JavaBean) 来执行应用程序所要求的更为复杂的处理。JavaBeans 已经是很成熟的技术, 基于组件的方法可供开发人员共享, 大大加速了总体开发过程。

3. 采用标识简化页面开发

JSP 技术封装了许多功能, 这些功能是在生成易用的、与 JSP 相关的 HTML 或者 XML 的动态内容时所需要的。标准的 JSP 标识能够访问和实例化 JavaBean 组件、设置或者检索组件属性、下载 Applet, 以及执行用其他方法更难以编码的耗时功能。通过开发定制标识

库, JSP 技术是可以扩展的。开发人员可以为常用功能创建自己的标识库, 从而使 Web 页面开发人员能够使用如同标识一样的工具来执行特定功能的构件。

4. 健壮性和安全性

由于 JSP 的内置脚本语言是基于 Java 语言的, 而且所有的 JSP 都被编译成 Servlets, JSP 就具有 Java 技术的所有好处, 包括健壮的存储管理和安全性。

5. 良好的移植性

作为 Java 平台的一部分, JSP 拥有 Java 编程语言“一次编写, 各处运行”的特点, 即具有良好的移植性。

6. 企业级的扩展性和性能

当与 J2EE (Java 2 Platform, Enterprise Edition, Java2 平台、企业级) 和 EJB 技术整合时, JSP 页面将提供企业级的扩展性和性能, 这对于在虚拟企业中部署基于 Web 的应用是必需的。

1.4 建立 JSP 的运行环境

由于 JSP 使用 Java 作为程序设计脚本语言, 因此需要建立 Java 的运行环境。编译和调试运行 Java 程序, 需要安装 JDK (Java Develop Kit, Java 开发工具包)。另外, JSP 是基于 Web 的 Java 应用程序, 因而它需要有特定的运行环境, 即解释器。由于 Java 语言是跨平台的, 所以能解释 Java 语言的 Web 服务器与平台无关。由 Apache 小组开发的 Tomcat 是一个开放源代码的自由软件, 可以自由获得而无须购买, 它是 Sun 公司推荐的产品, 并得到过 Sun 公司的鼎力帮助。

JSP 非常合适初学者, 以及在教学上使用。下面分别介绍 JDK 和 Tomcat 的安装。

本书所使用的平台是 Windows XP。

1.4.1 JDK 的下载和安装

1. 下载 JDK

JDK 是 Sun 公司制造的免费使用的软件包, 可以从 Sun 公司站点 <http://java.sun.com> 下载, 进入页面后会自动提示下载最新版本 (本书采用的是 JDK 6_18), 在页面的左边选择 Windows 操作系统后下载。

2. 安装 JDK

运行下载好的安装程序 `jdk-6u18-windows-i586.exe`, 选择安装路径后将会自动完成安装。

3. 设置环境变量

设置 Java 运行环境主要用到以下 3 个环境变量。

(1) JAVA_HOME

用于设定 Java 的安装路径。

(2) PATH

它是 Windows 所固有的, 需追加 JAVA_HOME\bin 目录, 这样在执行 JAVA_HOME\bin 文件夹下的命令时就不用输入长长的路径了, 如 javac、java 等命令。

(3) CLASSPATH

CLASSPATH 是运行 Java 非常重要的环境变量, Java 在编译和运行应用程序时都要通过它去找到需要的类文件。

设置环境变量的具体操作如下。

① 在桌面鼠标右键单击“我的电脑”, 在弹出的快捷菜单中选择“属性”, 在弹出的“系统属性”对话框中单击“高级”选项卡, 如图 1-1 所示。单击“环境变量”按钮, 出现如图 1-2 所示的“环境变量”对话框。

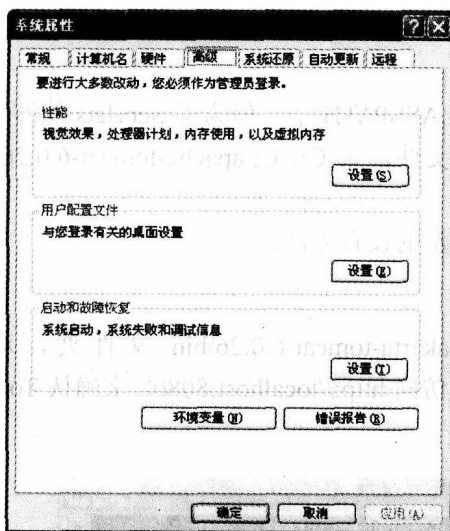


图 1-1 “高级”选项卡

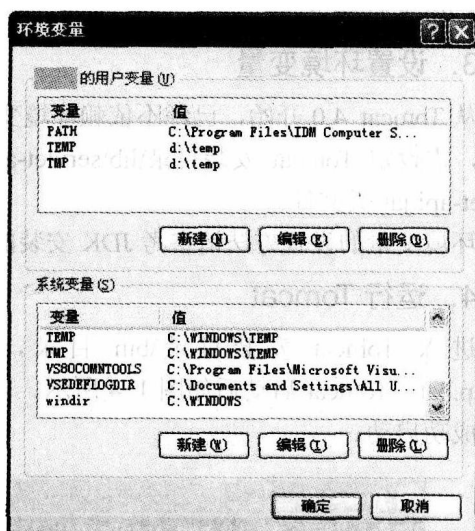


图 1-2 “环境变量”对话框

② 单击“系统变量”中的“新建”按钮, 打开如图 1-3 所示“新建系统变量”对话框, 添加 JAVA_HOME 环境变量, 并将其变量设置为 JDK 的安装目录, 如 C:\Program Files\Java\jdk1.6.0_18。

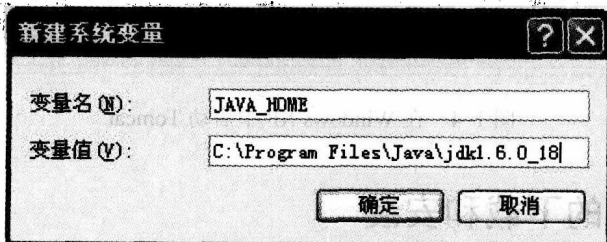


图 1-3 添加 JAVA_HOME 环境变量

③ 分别将 JAVA_HOME\bin 目录, 如 C:\Program Files\Java\jdk1.6.0_18\bin 追加至环境变量 PATH, JAVA_HOME\lib\tools.jar, 如 C:\Program Files\Java\jdk1.6.0_18\lib\tools.jar 追加至 CLASSPATH。

到此, Java 的运行环境安装设置完毕。

1.4.2 Tomcat 的下载和安装

1. 下载 Tomcat

可以从 Apache 站点 <http://tomcat.apache.org/> 下载, 本书使用 tomcat 6.0.26, 进入下载页面后选择下载 apache-tomcat-6.0.26.zip 文件。

2. 安装 Tomcat

Tomcat 的安装非常简单, 只要解压 apache-tomcat-6.0.26.zip 文件即可, 建议解压到盘区 \apache-tomcat-6.0.26 目录下。例如将文件 apache-tomcat-6.0.26.zip 解压到 C:\apache-tomcat-6.0.26.zip 文件夹下。

3. 设置环境变量

从 Tomcat 4.0 开始, 已经不依赖环境变量的 CLASSPATH 了。但为了 Servlets 能够顺利编译, 需设定 Tomcat 安装目录 \lib\servlet-api.jar 类文件, 如 C:\C:\apache-tomcat-6.0.26\lib\servlet-api.jar 类文件。

环境变量的设置方法请参考 JDK 安装的环境变量的设置方法。

4. 运行 Tomcat

进入 Tomcat 安装目录 \bin 目录, 如 C:\jakarta-tomcat-6.0.26\bin 文件夹, 双击 startup.bat。Tomcat 启动后如图 1-4 所示。可以通过访问 <http://localhost:8080/> 来确认 Tomcat 是否成功启动。

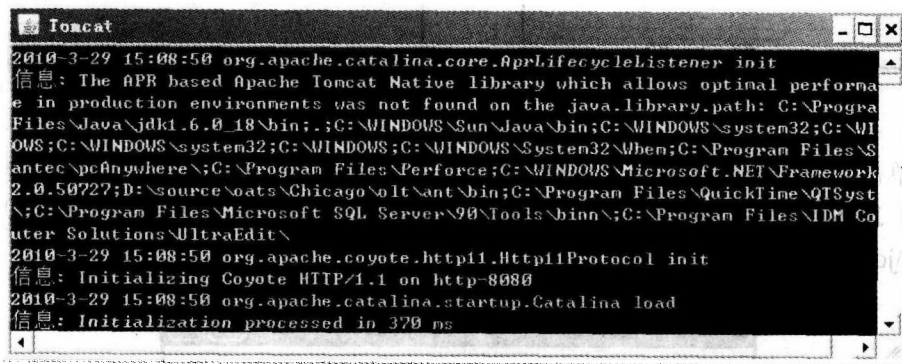


图 1-4 在 Windows XP 中启动 Tomcat

1.4.3 Eclipse 的下载和安装

1. 下载和安装 Eclipse

可以从 Eclipse 官方站点 <http://www.eclipse.org/downloads/> 下载 Eclipse IDE for Java Developers 版本, 单击下载页面后选择镜像站点下载。下载结束后, 解压下载的 zip 文件到指定目录, 本文的示例为 C:\eclipse。

2. 运行 Eclipse

进入 Eclipse 目录，运行 eclipse.exe，Eclipse 会弹出对话框让用户选择工作目录，输入 D:\JSPworkspace 后单击 OK 按钮，如图 1-5 所示。

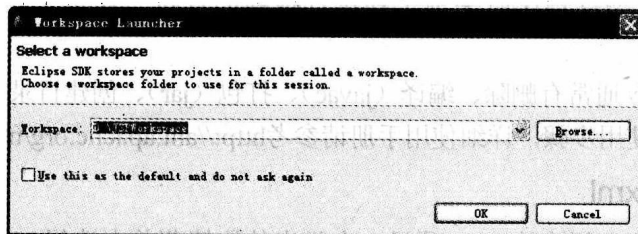


图 1-5 在启动 Eclipse 时输入工作目录

3. 操作 Eclipse

Eclipse 启动后，选择菜单项中的“File”→“New”→“Java Project”，在弹出的“New Java Project”窗口中输入项目名称 JSPHelloWorld 后选择“Finish”，如图 1-6 和图 1-7 所示。

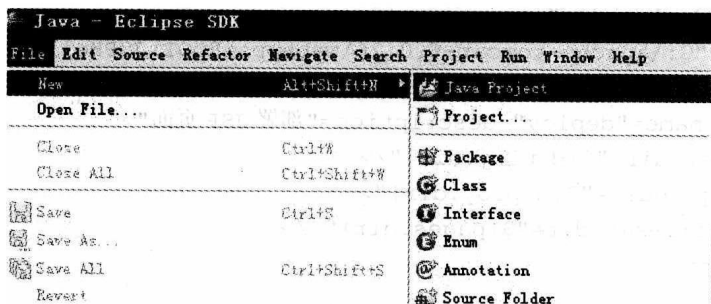


图 1-6 新建 Java 项目

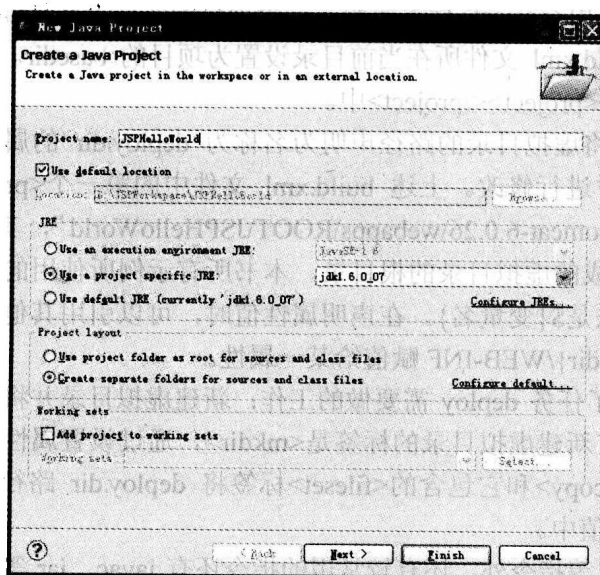


图 1-7 输入项目名称

1.4.4 Ant 使用简介

Ant 是 Eclipse 中最常用的自动化编译、部署工具。它的作用是能够按照 XML 配置文件自动化地编译单个或者多个 Java 项目中的文件,并将编译后的结果部署到指定的位置。使开发人员在调试程序时无须手动进行烦琐的打包、拷贝工作,极大地提高了工作效率。

Ant 能支持的指令通常有删除、编译 (javac)、打包 (jar)、创建目录和复制等操作。本书只介绍简单的创建、使用步骤,详细使用手册请参考<http://ant.apache.org/manual/index.html>。

1. 创建 build.xml

右键单击 1.4.3 节中创建的 Java 项目,在弹出的快捷菜单中选择 New→File, Eclipse 会弹出对话框请求输入文件名。在对话框中输入文件名 build.xml 后确定。

打开新建的 build.xml 文件,将以下内容复制到文件中并保存。

```
<?xml version="1.0" encoding="UTF-8" ?>
<project name="JSP Helloworld" default="deploy" basedir=".">
  <property
    name="deploy.dir"
    value="C:\apache-tomcat-6.0.26\webapps\ROOT\JSPHelloWorld" />
  <property name="pages.dir" value="${basedir}/pages" />
  <target name="deploy" description="部署 JSP 页面">
    <mkdir dir="${deploy.dir}" />
    <copy todir="${deploy.dir}">
      <fileset dir="${pages.dir}" />
    </copy>
  </target>
</project>
```

build.xml 文件中先用<project>标签声明了一个项目的名称 JSP Helloworld 和默认执行的任务 deploy,并将 build.xml 文件所在当前目录设置为项目的 basedir。Ant 项目中其他的定义都必须被包含在标签<project></project>中。

<property>标签中将虚拟目录的路径声明为名称为 deploy.dir 的属性,方便在其他地方引用和更改虚拟目录时进行修改。上述 build.xml 文件中的第一个<property>标签中的属性 value 值:“C:\apache-tomcat-6.0.26\webapps\ROOT\JSPHelloWorld”,其中的“webapps”是 Tomcat 安装完成后生成的虚拟目录的根目录,本书所有示例所使用的虚拟目录都在此根目录下。引用属性的方法是\${变量名}。在声明属性值时,可以引用其他的属性值进行声明,例如,可以将\${deploy.dir}/WEB-INF 赋值给某一属性。

<target>标签包含了任务 deploy 需要做的工作:新建虚拟目录并将当前项目中 JSP 页面复制到新建的目录中。新建虚拟目录的标签是<mkdir>,通过设置属性 dir 的内容来决定新建目录的位置。标签<copy>和它包含的<fileset>标签将 deploy.dir 路径中的所有文件复制到属性 dir 所设置的属性值中。

除了 copy、mkdir 等指令外,还有较常用的指令还有 javac、jar 等指令,在第 5 章中会介绍 javac 的使用。

2. 创建 JSP 页面

参照前述步骤在项目中创建目录 pages，在 pages 目录下创建文件 helloworld.jsp。打开 helloworld.jsp 将以下内容复制到文件中并保存。

```
<html><head></head><body><%= "JSP Hello world!"%></body></html>
```

3. 运行 build.xml

右键单击 Eclipse 左边项目文件列表中的 build.xml 文件名，在弹出的快捷菜单中选择 Run As→Ant Build，执行成功后在 Eclipse 中会输出成功复制文件的信息，如图 1-8 所示。

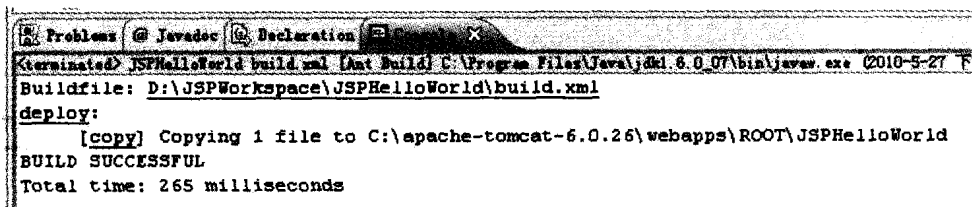


图 1-8 build.xml 运行结果

4. 运行 JSP 程序

运行 C:\apache-tomcat-6.0.26\bin\startup.bat 启动 Tomcat，在浏览器中打开网页 <http://localhost:8080/JSPHelloWorld/helloworld.jsp>，就能看到新部署的 JSP 页面了。

注意：Apache Tomcat 6.0.26 的默认设置使得 URL 是大小写敏感的，因此 URL 的大小写应与实际部署的目录名和 JSP 文件名一致。

1.4.5 MySQL 的下载和安装

后台数据库是 Web 应用程序的基础，因此掌握基础的数据库使用和设计十分重要。本书使用目前最流行的免费数据库 MySQL。

1. 下载和安装 MySQL

访问 <http://dev.mysql.com/downloads/mysql/> 下载最新版本的 MySQL Community Server 和 MySQL Workbench。前者为数据库服务器，后者为数据库管理工具。下载后运行安装包，按照默认设置进行安装。

2. 管理数据库

从“开始”菜单选择“MySQL”→“MySQL Workbench”启动数据库管理工具。在管理工具的菜单中选择“Database”→“Manage Connections”，如图 1-9 所示进行数据库连接设置。

设置完成后，单击右下角的“Test Connection”，测试成功后关闭连接设置窗口。在菜单中选择“Database”→“Query Database”，连接刚创建的数据库连接，提示输入密码时直接单击确定（MySQL 的 root 账号默认密码为空）。