普.通.高.等.学.校
计算机教育“十二五”规划教材

JSP 程序设计

（第 2 版）

JSP PROGRAMMING
(2nd edition)

范立锋 于合龙 孙丰伟 ◆ 主编
苏立明 李东明 王小佳 ◆ 副主编

人民邮电出版社
POSTS & TELECOM PRESS

普.通.高.等.学.校
计算机教育“十二五”规划教材

JSP 程序设计

（第2版）

JSP PROGRAMMING
(2nd edition)

范立锋 于合龙 孙丰伟 ◆ 主编
苏立明 李东明 王小佳 ◆ 副主编

人民邮电出版社
北京

图书在版编目（CIP）数据

JSP程序设计 / 范立锋，于合龙，孙丰伟主编. -- 2版. -- 北京 : 人民邮电出版社，2013.8
普通高等学校计算机教育“十二五”规划教材
ISBN 978-7-115-31400-0

Ⅰ. ①J… Ⅱ. ①范… ②于… ③孙… Ⅲ. ①JAVA语言－网页制作工具－高等学校－教材 Ⅳ. ①TP312②TP393.092

中国版本图书馆CIP数据核字(2013)第065092号

内 容 提 要

本书系统地介绍了 JSP 技术的概念、方法与实现过程，包括 JSP 运行环境、JSP 语法与组成元素、JSP 内置对象、JSP 对数据库的操作、JSP 对 JavaBean 的调用、JSP 对 Servlet 的调用等，最后还介绍了两个 JSP 综合实例。通过对本书的学习，读者可以系统地掌握 JSP 技术相关概念、方法、编程思路和技巧。

本书不要求面面俱到，也不追求博大精深，主要是面向大中专院校学生和没有开发经验或者仅有少量程序设计基础的读者，使读者能够在最短的时间内获得用 JSP 开发中小型网络系统的开发经验。同时，本书还有针对性地对一些技术的更新做了相关介绍，使读者能够掌握技术新动向，为以后更加深入地学习打下坚实基础。

◆ 主　　编　范立锋　于合龙　孙丰伟
副 主 编　苏立明　李东明　王小佳
责任编辑　刘　博
责任印制　彭志环　杨林杰

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京昌平百善印刷厂印刷

◆ 开本：787×1092　1/16
印张：18　　2013 年 8 月第 2 版
字数：469 千字　　2013 年 8 月北京第 1 次印刷

定价：39.80 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154

第 2 版前言

JSP（Java Server Pages）是近年来发展最迅速、最引人注目的 Web 应用开发技术之一，它是 Java SDK，Enterprise Edition（Java 企业版，Java EE）的重要技术。JSP 将 Java 语言的跨平台和开放性、Servlet 的强大功能与 HTML 以及脚本语言等简单易用的元素结合起来，解决了过去 Web 开发技术存在的各种不足和局限。

本书是作者在总结了多年开发经验与成果的基础上编写的。书中全面、翔实地介绍了 JSP 开发所需的各种知识和技巧。通过本书的学习，读者可以快速、全面地掌握使用 JSP 开发 Web 应用程序的方法，并且可以达到融会贯通、灵活运用的目的。

JSP 知识体系

JSP 的知识体系如下图所示。

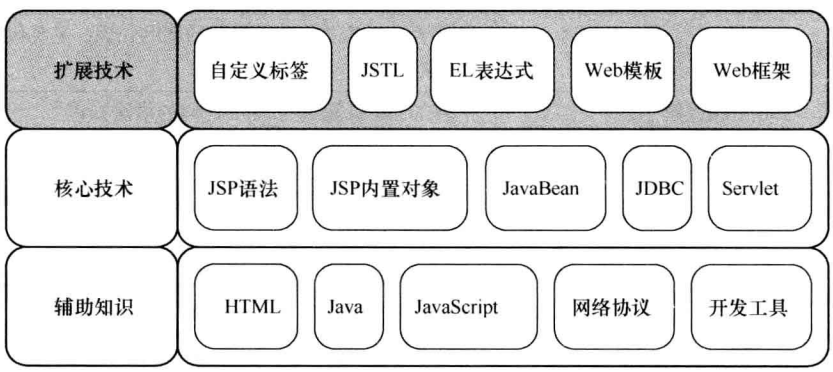

图 1-3　JSP 知识体系图

本书特点

（1）教材知识体系结构合理。知识安排强调整体性和系统性，知识表达强调层次性和有序性，便于读者学习和理解。

（2）理论与应用紧密结合。在每一章节，首先对相关知识点进行解释，然后用所学知识点和技术构建应用，通过实例来理解知识点，通过应用来熟悉技术。这样，理论与应用就可结合。

（3）语言通俗易懂，读者容易理解。书中采用程序结构、页面交互图、流程图、表格等多种方式描述问题及解决问题的过程，使读者从多个角度来理解问题。

（4）新的知识点。与其他教材区别在于，本书还加入了 Web 应用新的知识点，例如，本书的第 9 章中介绍的 Web 模板技术，主要介绍了两种流行的模板技术：Velocity 模板和 FreeMarker 模板。模板语言在现代的软件开发中占据着重要的地位，它的功能强大，而且学习起来又非常简单，即使不熟悉编程的人也能很快地掌握它。另外第 10 章中介绍的 JavaFx 组件，是快速构架 Java 富客户端的利器。

（5）新的技术点。JSP 相关技术的更新很快，在个别关键性更新版本中，往往会推出一些新技术、新用法。例如，Servlet 3.0 版本中增加了很多新特性，与之前版本比较是一个很大的创新。本书第 6 章中结合实际案例对 Servlet 3.0 的新特性进行了简单明了的介绍。

（6）扩展知识点。本书在介绍数据库连接池时，不仅介绍了连接池的原理与应用，而且还介绍了一些常用的连接池组件，这使得读者扩展思路和了解连接池不只是单一技术实现，而是可以通过多种技术进行实现的。

（7）案例的实用性。本书中最后两个案例是 Web 典型应用：购物车和论坛，前者用到的技术是“Servlet+MYSQL 5.0+JSP 表达式”，后者用到的技术是“Servlet+SQL Server 2000+JSTL”。通过两个不同的数据库介绍了两个案例的开发，使读者直接获取项目的开发经验。

本书结构

本书围绕 JSP 动态网页制作循序渐进地介绍相关知识，书中给出了许多实例，而且每章后附有习题，用来巩固所学内容。全书共分 13 章，各章具体内容如下表所示。

全书各个章节的主要内容

章　名	描　述
第 1 章　JSP 初步	介绍 JSP 技术、优点、缺点及应用
第 2 章　JSP 辅助知识	介绍学习 JSP 之前必须掌握的辅助知识，主要包括 HTML、JavaScript、网络协议
第 3 章　JSP 语法详解	介绍在 JSP 页面编写 Java 代码的语法知识
第 4 章　JSP 内置对象详解	介绍 JSP 页面中 9 个内置对象的方法与应用，并且给出相关实例
第 5 章　JavaBean 组件技术	介绍 JavaBean 在 Web 开发过程中的实质，并介绍 JavaBean 绑定属性的特点与应用
第 6 章　Servlet 核心技术	介绍 Servlet 相关的知识点，并同时讲解 Servlet 过滤器与监听器在实际开发中的作用，并对 Servlet 3.0 进行了讲解
第 7 章　JSP 操作数据库核心技术	介绍 JDBC 操作数据库中 SQL 语句的方法，并且介绍 JDBC 连接池在网页中的应用
第 8 章　JSP 核心表达式与标签	介绍 JSP2.0 新技术中核心表达式与标签的语法知识，并且给出各种标签的相关实例
第 9 章　Web 网页模板技术	介绍 Web 应用中两种模板技术：Velocity 模板和 FreeMarker 模板
第 10 章　JSP 实用组件技术	介绍 JSP 实用组件的相关知识点与应用，包括上传组件、E-mail 组件、JfreeChart 组件以及 JavaFx 组件
第 11 章　MVC 设计模式	介绍 MVC 设计模式，并对 Struts2 和 Spring 框架有一个初步的了解
第 12 章　JSP 实例开发 1——论坛	运用 JSP 技术和方法开发电子商城中购物车模块程序
第 13 章　JSP 实例开发 2——购物车	运用 JSP 技术和方法开发论坛程序

本书面向的读者

本书面向的是网络程序设计的初学者。读者只需要掌握一门高级计算机语言（例如，C 语言）就可以按照书中介绍的方法和实例构建互联网站，开发网络应用系统。同时，本书在内容编排上，由浅入深，循序渐进，也适合大中专院校的教师作为授课教材。

如果您具备一定的 JSP 开发基础，又希望提高自己的技术，使自己能够开发出中型 Web 应用程序，本书也非常适合。虽然本书的编写初衷是面向没有开发经验的读者，但是如果您具备以下方面的知识，学习起来将事半功倍：

★ 熟悉 HTML
★ 熟悉 Java 语言
★ 熟悉 Web 开发流程

本书实例的开发环境

本书用到了 JavaSE、Apache Tomcat、Ecplise 等软件，读者可以到相关网站下载。本书中实例使用的是 SQLServer 2000 和 MYSOL 5.0 数据库，因此要查看或修改数据库，计算机上必须安装这两个数据库。

技术支持

本书实例开发中用到的程序源代码，可以在人民邮电出版社教学资源与服务网（www. ptpedu. com.cn）上免费下载，以供读者学习和使用。

编　者

2013 年 2 月

目　录

第 1 章 JSP 初步

随着因特网和电子商务的普遍应用，各种动态网页语言陆续诞生。其中，JSP（Java Server Page）自从发布以来，一直受到密切的关注。JSP 是由 Sun Microsystems 公司倡导，并由许多公司一起参与建立的一种动态网页技术标准，该技术是在 Servlet 技术基础上发展而来的。如今，JSP 已经成为 Java 服务器编程的重要组成部分。

通过本章的学习，读者可以对 JSP 的技术特点、JSP 运行原理及 JSP 开发环境有一定的了解，并可以通过 Eclipse 工具开发一个简单的 Web 应用程序。

1.1 认识 JSP

JSP（Java Server Page）是运行在服务器的一种脚本语言。熟悉 HTML 或者其他动态页面技术的读者，在第 1 次看到 JSP 页面时可能会有一种似曾相识的感觉。这是因为，从本质上说，各种动态页面技术，都是通过在 HTML 中添加其他语言脚本的方式来实现的，而支持这些脚本的服务器可以执行这些脚本，然后生成 HTML 页面。

为了让读者直观认识 JSP 技术，先来看一段简单的 JSP 页面代码，该 JSP 页面名称为 sanyang.jsp，实现在页面中输出一句话的功能。具体代码如下：

```
<%@ page language="java" pageEncoding="GBK"%>
<html>
  <head>
    <title>第一个 JSP 程序</title>
  </head>
  <body>
   <%out.print("您好，三扬科技");%>
  </body>
</html>
```

上述代码的代码风格和普通的 HTML 页面的代码非常相似，不同的就是在“<%”和“%>”之间加入了 Java 代码。将页面部署到 Tomcat 服务器上，将该 Web 应用命名为 01，启动 Tomcat 服务器，在 IE 浏览器的地址栏中输入 URL 地址“http://localhost:8080/01/sanyang.jsp”，运行结果如图 1-1 所示。

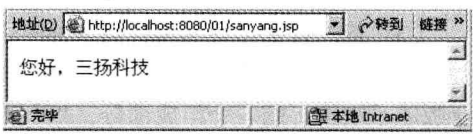

图 1-1 sanyang.jsp 页面的运行结果

1.2 JSP 技术特性

本节将介绍 JSP 的一些特性，如跨平台、分离静态内容和动态内容、强调可重用的组件等。

1. 跨平台

JSP 技术以 Java 为基础，不仅可以沿用 Java 强大的 API 功能，还具有跨平台的特性。不管是在何种平台下，只要服务器支持 JSP，就可以运行 JSP 开发的 Web 应用程序。例如，在 Windows NT 下的 IIS 通过添加 JRUN 或 ServletExec 插件就能支持 JSP。如今最流行的 Web 服务器 Apache 同样能够支持 JSP，而且 Apache 支持多种平台，从而使得 JSP 可以跨平台运行。

在数据库操作中，因为 JDBC 同样是独立于平台的，所以在 JSP 中使用的 Java API 中提供的 JDBC 来连接数据库，就不用担心平台变更时的代码移植问题。

2. 将内容的生成和显示分离

使用 JSP 技术，Web 页面开发人员可以使用 HTML 或 XML 标识来设计和格式化最终页面，通过使用 JSP 标识或者小脚本来生成页面上的动态内容。生成内容的逻辑被封装在标识和 JavaBean 组件中，并且捆绑在小脚本中，所有的脚本在服务器端运行。由于核心逻辑被封装在标识和 Bean 中， Web 管理人员和页面设计者可以专注于页面的编辑和设计，而无需考虑业务逻辑的处理。

当 JSP 应用运行时，服务器端 JSP 引擎负责解释 JSP 标识和小脚本，生成所请求的内容（如通过访问 JavaBean 组件，使用 JDBC 技术访问数据库或者包含文件），并将结果以 HTML（或者 XML）页面的形式发送回浏览器。这不仅有助于开发人员保护自己的代码，也保证了 Web 应用的通用性，任何基于 HTML 的 Web 浏览器的都可以正常使用该 Web 应用。

3. 强调可重用的组件

绝大多数 JSP 页面依赖于可重用的、跨平台的组件（JavaBean 或者企业级 JavaBean）来执行应用程序所要求的更为复杂的处理。开发人员能够共享和交换执行普通操作的组件，或者使得这些组件为更多的使用者或者客户团体所使用。基于组件的开发模式，加速了总体开发进程，并且使得各种组织在他们现有的技能和优化结果的开发努力中得到平衡。

4. 采用标识简化页面开发

JSP 采用标识简化页面开发，具有以下 4 个特点。

（1）Web 页面开发人员并不需要都是熟悉脚本语言的编程人员，因为 JSP 技术封装了许多功能，这些功能是在易用的、与 JSP 相关的 XML 标识中进行动态内容生成所需要的。

（2）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于编码和耗时的功能。通过开发定制化标识库，可以扩展 JSP 技术。第 3 方开发人员和其他人员可以为常用功能创建自己的标识库。这使得 Web 页面开发人员能够使用熟悉的工具和如同标识一样的执行特定功能的构件来工作。

（3）JSP 技术很容易整合到多种应用体系结构中，这样可以更好地利用现存的工具和技巧，并且可以扩展到能够支持企业级的分布式应用。作为采用 Java 技术家族的一部分，以及 J2EE（企业版体系结构）的一个组成部分，JSP 技术能够支持高度复杂的基于 Web 的应用。

（4）作为 Java 平台的一部分，JSP 拥有 Java 编程语言“一次编译，到处运行”的特点。随着越来越多的供应商将 JSP 支持添加到它们的产品中，用户可以使用自己所选择的服务器和工具，更改工具或服务器而并不影响当前的应用。

5. 存储的健壮性与安全性

由于 JSP 页面的内置脚本语言是基于 Java 编程语言的，且都编译成 Java Servlet，因此它具有 Java 技术的所有优势，包括健壮的存储管理和安全性。

1.3　JSP 工作原理

从本质上说，JSP 是结合 markup（HTML 或 XML）和 Java 代码来处理的一种动态页面。每个页面第 1 次被调用时，通过 JSP 引擎自动被编译成 Servlet 程序，然后被执行。例如，在 1.1 节中介绍的 sanyang.jsp 页面在 Tomcat 服务器运行时，该页面将会被编译成一个 Servlet 程序。该页面在 Tomcat 服务器上编译成的 Servlet 源代码如下：

```
package org.apache.jsp;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
public final class sanyang_jsp extends org.apache.jasper.runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent {
  private static final JspFactory _jspxFactory = JspFactory.getDefaultFactory();
  private static java.util.List _jspx_dependants;
  private javax.el.ExpressionFactory _el_expressionfactory;
  private org.apache.AnnotationProcessor _jsp_annotationprocessor;
  public Object getDependants() {
    return _jspx_dependants;
  }
  public void _jspInit() {
    _el_expressionfactory = _jspxFactory.getJspApplicationContext(getServletConfig().
getServletContext()).getExpressionFactory();
    _jsp_annotationprocessor = (org.apache.AnnotationProcessor) getServletConfig().
getServletContext().getAttribute(org.apache.AnnotationProcessor.class.getName());
  }
  public void _jspDestroy() {
  }
  public void _jspService(HttpServletRequest request, HttpServletResponse response)
      throws java.io.IOException, ServletException {
    PageContext pageContext = null;
    HttpSession session = null;
    ServletContext application = null;
    ServletConfig config = null;
    JspWriter out = null;
    Object page = this;
    JspWriter _jspx_out = null;
    PageContext _jspx_page_context = null;
    try {
      response.setContentType("text/html;charset=GBK");
      pageContext = _jspxFactory.getPageContext(this, request, response,
                  null, true, 8192, true);
      _jspx_page_context = pageContext;
      application = pageContext.getServletContext();
      config = pageContext.getServletConfig();
      session = pageContext.getSession();
      out = pageContext.getOut();
```

```
        _jspx_out = out;
        out.write("\r\n");
        out.write("<html>\r\n");
        out.write("  <head>    \r\n");
        out.write("    <title>第一个 JSP 程序</title>\t\r\n");
        out.write("  </head>  \r\n");
        out.write("  <body>\r\n");
        out.write("   ");
        out.print("您好，三扬科技");
        out.write("\r\n");
        out.write("  </body>\r\n");
        out.write("</html>\r\n");
      } catch (Throwable t) {
        if (!(t instanceof SkipPageException)){
          out = _jspx_out;
          if (out != null && out.getBufferSize() != 0)
            try { out.clearBuffer(); } catch (java.io.IOException e) {}
          if (_jspx_page_context != null) _jspx_page_context.handlePageException(t);
        }
      } finally {
        _jspxFactory.releasePageContext(_jspx_page_context);
      }
    }
  }
```

JSP 转换成 Servlet 的代码存放在 Tomcat 服务器的安装文件夹下。一般情况下，转换成 Servlet 的代码具体路径是：Tomcat 安装文件夹\work\Catalina\localhost\，然后可以通过该文件夹下的工程名去寻找。

在一个 JSP 文件第 1 次被请求时，JSP 引擎先把该 JSP 文件转换成一个 Java 源文件。如果转换过程中发现 JSP 文件有任何语法错误，转换过程将被中断，并向服务器端和客户端输出错误信息；如果转换成功，JSP 引擎调用 Java 虚拟机的 javac 程序把该 Java 文件的源文件编译成相应的 class 文件，该 class 文件也就是一个 Servlet 程序，然后创建一个该 Serlvet 的实例，提供服务响应用户的请求。

JSP 转换成 Servlet 的流程如图 1-2 所示。

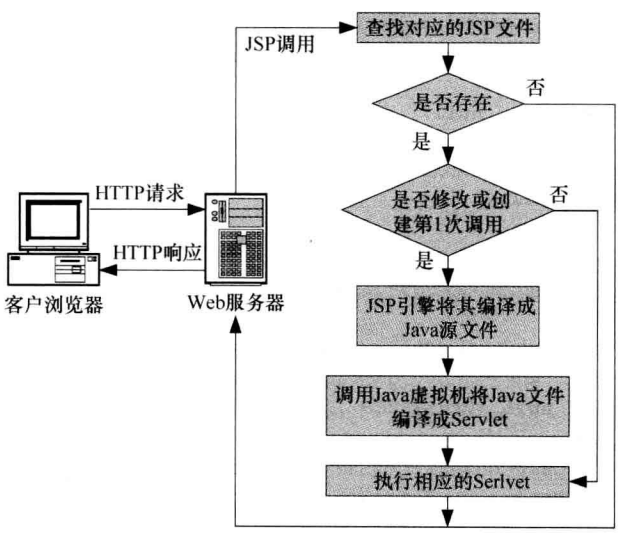

图 1-2　JSP 响应流程图

1.4　搭建 JSP 的运行环境

使用方便、高效快捷的 JSP 开发环境，对于学习 JSP 的读者来说会事半功倍。目前，比较流行的 JSP 开发平台和工具主要包括 JDK、Tomcat、Eclipse 等。其中，JDK 是 Java 语言的开发环境，Tomcat 是 Web 服务器，Eclipse 是一套简化的 Java EE 开发工具。即 JSP 应用开发的可视化 IDE。

1.4.1　JDK 的安装与配置

JDK 是 Java 语言的开发环境，由于 JSP 本身执行的计算机语言就是 Java，因此，想要开发与运行 JSP 程序也需要 JDK 的开发环境。JDK 的安装文件可以通过 http://java.sun.com 网站进行下载，目前的最新版本是 1.6。

下面介绍 JDK 的安装与配置。

1. JDK 的安装

从 Sun 公司官方网站中下载的安装文件的名称是 jdk-6u10-windows-i586-p.exe，下载完毕后，就可以在需要编译和运行 Java 程序的计算机安装 JDK 类，具体步骤如下。

（1）双击 jdk-6u10-windows-i586-p.exe 文件开始安装。安装向导会要求接受 Sun 公司如图 1-3 所示的许可协议。

（2）单击“接受”按钮接受许可协议后，打开设置 JDK 的安装路径及选择安装组件的对话框，如图 1-4 所示。

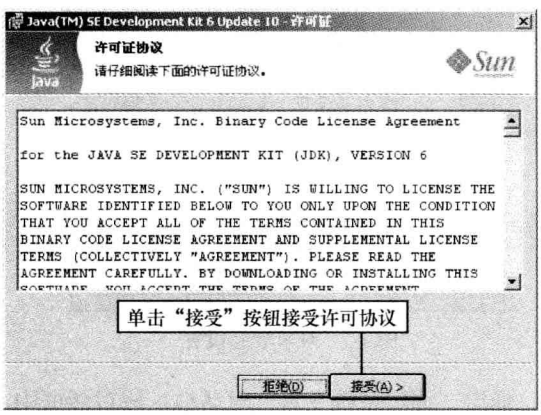

图 1-3　JDK1.6 的许可协议

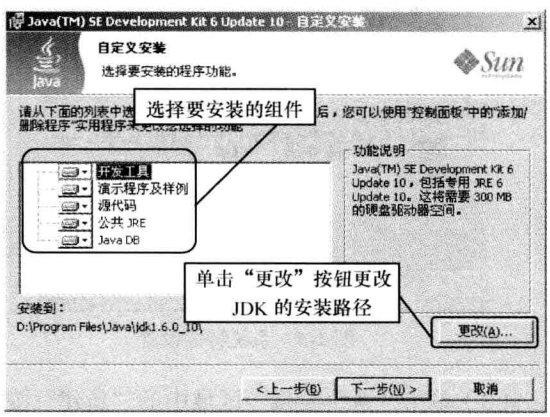

图 1-4　设置 JDK 的安装路径及选择安装组件对话框

（3）在图 1-4 所示的对话框中单击“更改”按钮，如更改安装路径为 D:\Program Files\Java\ jdk1.6.0_10，其他采用默认设置，单击“下一步”按钮将打开安装进度对话框安装 JDK。在安装过程中将打开图 1-5 所示的设置 JRE 安装路径的对话框。

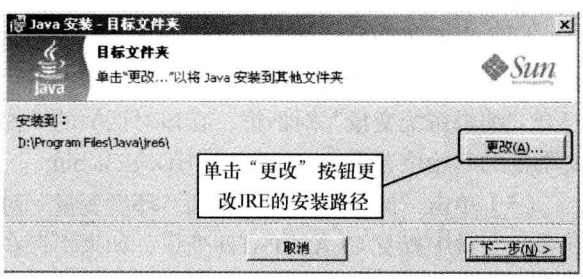

图 1-5　设置 JRE 的安装路径及选择安装组件对话框

由于 JDK 只是 Java 程序的开发环境，所以 JDK 的安装文件中还包含了一个 JRE（Java SE Runtime Environment，Java 运行环境），在默认情况下同 JDK 一起安装。

（4）在设置 JRE 安装路径的对话框中，单击“更改”按钮，在打开的对话框中将 JRE 的安装路径修改为 D:\Program Files\Java\jre6\，单击“下一步”按钮继续安装 JRE。在弹出的安装完成提示对话框中，取消“显示自述文件”复选框的勾选，单击“完成”按钮，即可完成 JDK 的安装。

在安装 JDK1.6 之前，关闭所有正在运行的程序，并确认系统中没有安装 JDK 的其他版本，否则，在进行配置时会有冲突。

2. 配置和测试 JDK

安装完 JDK 后，需要设置环境变量及测试 JDK 配置是否成功，具体步骤如下。

（1）在“我的电脑”上右击鼠标在弹出的快捷菜单中，选择“属性”选项。在打开的“系统特性”对话框中选择“高级”选项卡，如图 1-6 所示。

（2）单击“环境变量”按钮，打开“环境变量”对话框。在这里可以添加针对单个用户的“用户变量”和针对所有用户“系统变量”，如图 1-7 所示。

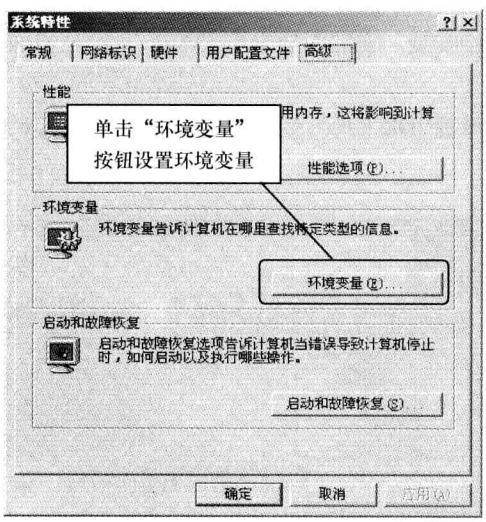

图 1-6　系统特性设置

图 1-7　环境变量设置

（3）单击“系统变量”区域中的“新建”按钮，弹出“编辑系统变量”对话框。该对话框中，在“变量名”文本框中输入“JAVA_HOME”，在“变量值”文本框中输入 JDK 的安装路径 D:\Program Files\Java\jdk1.6.0_10，单击“确定”按钮，完整环境变量 JAVA_HOME 的配置，如图 1-8 所示。

（4）在系统变量中查看 PATH 变量，如果不存在，则新建变量 PATH，否则选中该变量，单击“编辑”按钮，打开“编辑系统变量”对话框，在该对话框的“变量值”文本框的起始位置添加“%JAVA_HOME%\bin;”。

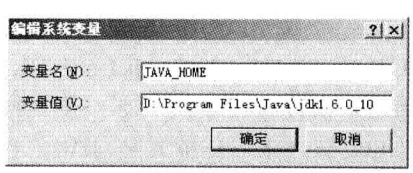

图 1-8　“JAVA_HOME”变量的设置

（5）单击“确定”按钮返回到“环境变量”对话框。在系统变量中查看 CLASSPATH 变量，如果不存在，则新建变量 CLASSPATH，变量值为%JAVA_HOME%\lib\dt.jar;%JAVA_HOME% \lib\tools.jar。