

Java EE

应用开发案例教程

卢守东 编著



+



+



+



- ◆ 以基础理论—实用技术—实训为主线
- ◆ 按照教与学的实际需要取材谋篇
- ◆ 精心设置了“小型案例实训”，旨在培养学生的实践能力
- ◆ 配备丰富的免费教学资源——电子教案、习题答案、案例素材



全国高等院校应用型创新规划教材·计算机系列

Java EE 应用开发案例教程

卢守东 编著

清华大学出版社
北 京

内 容 简 介

本书以技术需求为导向,以技术应用为核心,以开发模式为主线,以应用开发为重点,以能力提升为目标,全面介绍了基于 Java EE 的企业级应用开发的关键技术、流行框架、主要模式、实施要点与方法步骤。全书共 10 章,包括 Java EE 概述、JSP 基础、JDBC 技术、JavaBean 技术、Servlet 技术、Struts 2 框架、Hibernate 框架、Spring 框架、Ajax 应用与 Web 应用案例等内容,并附有相应的思考题与实验指导。

本书内容全面、实例翔实、案例丰富,注重应用开发能力的培养,可作为各高校本科或高职高专计算机、电子商务、信息管理与信息系统及相关专业软件开发技术课程的教材或教学参考书,也可作为 Java EE 应用开发与维护人员的技术参考书以及初学者的自学教程。

本书所有示例的代码均已通过调试,并能成功运行,其开发环境为 Windows7、JDK 1.7.0_51、Tomcat 7.0.50、MyEclipse 10.7.1 与 SQL Server 2008。

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 EE 应用开发案例教程/卢守东编著. —北京:清华大学出版社,2017

(全国高等院校应用型创新规划教材·计算机系列)

ISBN 978-7-302-46310-8

I. ①J… II. ①卢… III. ①JAVA 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 021365 号

责任编辑:孟 攀

封面设计:杨玉兰

责任校对:吴春华

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>, 010-62791865

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:27.75 字 数:675 千字

版 次:2017 年 3 月第 1 版 印 次:2017 年 3 月第 1 次印刷

印 数:1~2000

定 价:56.00 元

产品编号:073043-01

前 言

Java EE 是当前构建企业级 Java Web 应用的主流开发技术与平台之一,其应用已相当广泛。为满足社会不断发展的实际需求,并提高学生或学员的专业技能与就业能力,多数高校的计算机、电子商务、信息管理与信息系统等相关专业以及各地的相关培训机构均开设了 Java EE 程序设计或 Java EE 开发技术等课程。

本书以技术需求为导向,以技术应用为核心,以开发模式为主线,以应用开发为重点,以能力提升为目标,遵循案例教学、任务驱动与项目驱动的思想,结合教学规律与开发规范,按照由浅入深、循序渐进的原则,精心设计,合理安排,全面介绍了基于 Java EE 的企业级应用开发的关键技术、流行框架、主要模式、实施要点与方法步骤。全书实例翔实、案例丰富、编排合理、循序渐进、结构清晰,共分为 10 章,包括 Java EE 概述、JSP 基础、JDBC 技术、JavaBean 技术、Servlet 技术、Struts 2 框架、Hibernate 框架、Spring 框架、Ajax 应用与 Web 应用案例。各章均有“本章要点”“学习目标”与“本章小结”,既便于抓住重点、明确目标,也利于温故而知新。书中的诸多内容均设有相应的“说明”“注意”“提示”等知识点,以便于读者的理解与提高,并为其带来“原来如此”“豁然开朗”的美妙感觉。此外,各章均安排有相应的思考题,以便于读者的及时回顾与检测。书末还附有全面的实验指导,以便于读者的上机实践。

本书内容全面,结构清晰,语言流畅,通俗易懂,准确严谨,颇具特色,集系统性、条理性、针对性于一身,融资料性、实用性、技巧性于一体,注重应用开发能力的培养,可充分满足课程教学的实际需要,适合各个层面、各种水平的读者,既可作为各高校本科或高职高专计算机、电子商务、信息管理与信息系统及相关专业软件开发技术课程的教材或教学参考书,也可作为 Java EE 应用开发与维护人员的技术参考书以及初学者的自学教程。

本书所有示例的代码均已通过调试,并能成功运行,其开发环境为 Windows 7、JDK 1.7.0_51、Tomcat 7.0.50、MyEclipse 10.7.1 与 SQL Server 2008。

本书的写作与出版,得到了作者所在单位及清华大学出版社的大力支持与帮助,在此表示衷心感谢。在紧张的写作过程中,自始至终也得到了家人、同事的理解与支持,在此也一并深表谢意。

由于作者经验不足、水平有限,且时间较为仓促,书中不妥之处在所难免,恳请广大读者多加指正、不吝赐教,并将宝贵的意见或建议反馈至作者的电子邮箱 Lsd21cn@21cn.com。

编 者

目 录

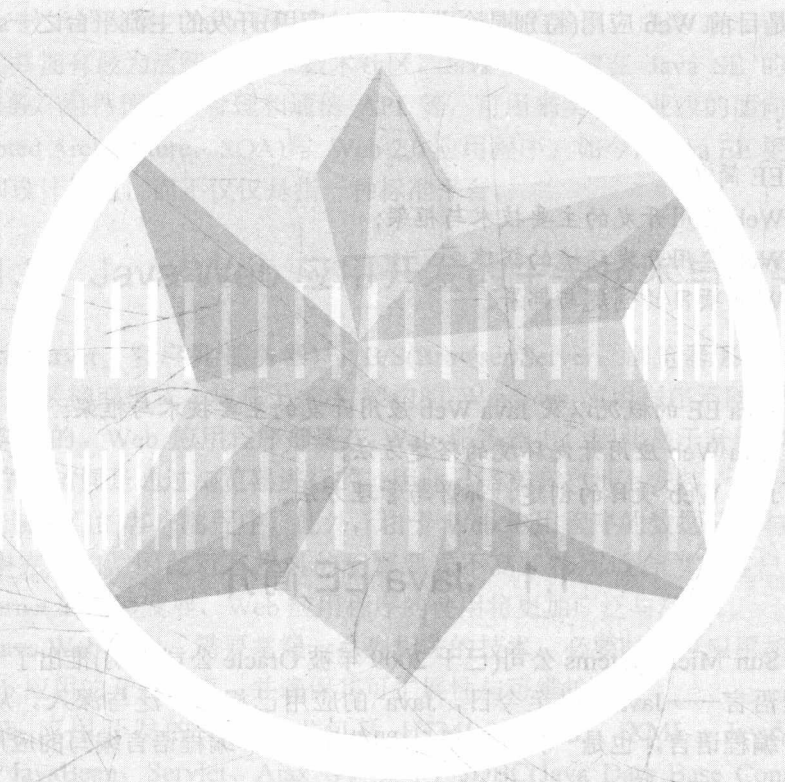
第 1 章 Java EE 概述	1	2.4.1 系统登录	68
1.1 Java EE 简介	2	2.4.2 简易聊天室	71
1.2 Java Web 应用开发的 主要技术与框架	3	本章小结	74
1.3 Java Web 应用开发环境的搭建	4	思考题	74
1.3.1 JDK 的安装与配置	4	第 3 章 JDBC 技术	77
1.3.2 Tomcat 的安装与配置	7	3.1 JDBC 简介	78
1.3.3 MyEclipse 的安装与配置	10	3.2 JDBC 的核心类与接口	78
1.3.4 SQL Server 的安装与配置	15	3.2.1 DriverManager 类	79
1.4 Java Web 项目的创建与部署	19	3.2.2 Driver 接口	79
1.5 Java Web 项目的导出、删除与导入	23	3.2.3 Connection 接口	80
本章小结	25	3.2.4 Statement 接口	81
思考题	25	3.2.5 PreparedStatement 接口	83
第 2 章 JSP 基础	27	3.2.6 CallableStatement 接口	84
2.1 JSP 简介	28	3.2.7 ResultSet 接口	85
2.2 JSP 基本语法	29	3.3 JDBC 基本应用	87
2.2.1 声明	29	3.4 JDBC 应用案例	97
2.2.2 表达式	29	3.4.1 系统登录	97
2.2.3 脚本小程序	30	3.4.2 数据添加	99
2.2.4 JSP 指令标记	32	3.4.3 数据维护	102
2.2.5 JSP 动作标记	35	本章小结	116
2.2.6 JSP 注释	40	思考题	116
2.3 JSP 内置对象	42	第 4 章 JavaBean 技术	117
2.3.1 out 对象	42	4.1 JavaBean 简介	118
2.3.2 request 对象	44	4.2 JavaBean 的规范	118
2.3.3 response 对象	49	4.3 JavaBean 的创建	119
2.3.4 session 对象	55	4.4 JavaBean 的使用	120
2.3.5 application 对象	57	4.4.1 <jsp:useBean>动作标记	120
2.3.6 exception 对象	62	4.4.2 <jsp:setProperty>动作标记	121
2.3.7 page 对象	63	4.4.3 <jsp:getProperty>动作标记	121
2.3.8 config 对象	65	4.5 JavaBean 的应用案例	125
2.3.9 pageContext 对象	66	4.5.1 系统登录	125
2.4 JSP 应用案例	68	4.5.2 数据添加	128
		4.5.3 数据维护	131

本章小结	132	6.7.1 单文件上传	206
思考题	132	6.7.2 多文件上传	208
第 5 章 Servlet 技术	133	6.8 Struts 2 文件下载	211
5.1 Servlet 简介	134	6.9 Struts 2 应用案例	213
5.2 Servlet 的技术规范	136	6.9.1 系统登录	213
5.3 Servlet 的创建与配置	136	6.9.2 数据添加	215
5.3.1 Servlet 的创建	136	本章小结	223
5.3.2 Servlet 的配置	140	思考题	223
5.4 Servlet 的基本应用	142	第 7 章 Hibernate 框架	225
5.5 Servlet 的应用案例	148	7.1 Hibernate 概述	226
5.5.1 系统登录	148	7.1.1 ORM 简介	226
5.5.2 数据添加	150	7.1.2 Hibernate 体系结构	227
本章小结	153	7.2 Hibernate 基本应用	227
思考题	153	7.2.1 Hibernate 基本用法	227
第 6 章 Struts 2 框架	155	7.2.2 Hibernate 相关文件	233
6.1 Struts 2 概述	156	7.3 Hibernate 核心接口	240
6.2 Struts 2 基本应用	157	7.3.1 Configuration 接口	240
6.2.1 Struts 2 开发包	157	7.3.2 SessionFactory 接口	240
6.2.2 Struts 2 基本用法	158	7.3.3 Session 接口	240
6.2.3 Struts 2 核心过滤器的配置	164	7.3.4 Transaction 接口	241
6.2.4 Struts 2 Action 的实现	166	7.3.5 Query 接口	241
6.2.5 Struts 2 Action 的配置	168	7.4 HQL 基本用法	241
6.3 Struts 2 拦截器	176	7.4.1 HQL 查询	242
6.3.1 拦截器的实现	176	7.4.2 HQL 更新	243
6.3.2 拦截器的配置	177	7.4.3 HQL 语句的执行	243
6.4 Struts 2 OGNL	181	7.5 Hibernate 对象状态	247
6.4.1 OGNL 表达式	181	7.5.1 瞬时态	247
6.4.2 OGNL 集合	183	7.5.2 持久态	247
6.5 Struts 2 标签库	184	7.5.3 脱管态	248
6.5.1 数据标签	184	7.6 Hibernate 批量处理	248
6.5.2 控制标签	189	7.6.1 批量插入	248
6.5.3 表单标签	193	7.6.2 批量修改	251
6.5.4 非表单标签	199	7.6.3 批量删除	253
6.6 Struts 2 数据验证	199	7.7 Hibernate 事务管理	254
6.6.1 数据校验	199	7.7.1 事务的基本概念	254
6.6.2 校验框架	201	7.7.2 基于 JDBC 的事务管理	254
6.7 Struts 2 文件上传	206	7.7.3 基于 JTA 的事务管理	256
		7.8 Hibernate 应用案例	257

7.8.1 数据查询.....	257	本章小结.....	325
7.8.2 系统登录.....	262	思考题.....	325
7.9 Hibernate 与 Struts 2 整合应用.....	266		
本章小结.....	269	第 9 章 Ajax 应用	327
思考题.....	269		
第 8 章 Spring 框架	271	9.1 Ajax 简介.....	328
8.1 Spring 概述.....	272	9.1.1 Ajax 的基本概念.....	328
8.2 Spring 基本应用.....	274	9.1.2 Ajax 的应用场景.....	328
8.2.1 工厂模式.....	274	9.2 Ajax 应用基础.....	330
8.2.2 Spring 基本用法.....	276	9.2.1 XMLHttpRequest 对象简介 ...	330
8.2.3 Spring 依赖注入.....	278	9.2.2 Ajax 的请求与响应过程.....	331
8.3 Spring 关键配置.....	281	9.2.3 Ajax 的基本应用.....	332
8.3.1 Bean 的基本定义.....	282	9.3 Ajax 开源框架 DWR.....	335
8.3.2 Bean 的依赖配置.....	282	9.3.1 DWR 简介.....	335
8.3.3 Bean 的别名设置.....	287	9.3.2 DWR 的工作原理.....	335
8.3.4 Bean 的作用域设置.....	288	9.3.3 DWR 的基本应用.....	336
8.3.5 Bean 的生命周期方法设置.....	290	9.3.4 DWR 与 Struts 2、Spring、 Hibernate 的整合应用.....	338
8.4 Spring 核心接口.....	292	本章小结.....	344
8.4.1 BeanFactory.....	293	思考题.....	344
8.4.2 ApplicationContext.....	294		
8.5 Spring AOP.....	295	第 10 章 Web 应用案例	347
8.5.1 AOP 简介.....	295		
8.5.2 AOP 的相关术语.....	295	10.1 系统简介.....	348
8.5.3 AOP 的实现机制.....	296	10.1.1 系统功能.....	348
8.5.4 Spring AOP 的基本应用.....	301	10.1.2 系统用户.....	349
8.6 Spring 事务支持.....	305	10.2 开发方案.....	349
8.6.1 使用 TransactionProxyFactoryBean 创建事务代理.....	306	10.2.1 分层模型.....	349
8.6.2 使用 Bean 继承配置事务 代理.....	307	10.2.2 开发模式.....	350
8.6.3 使用 BeanNameAutoProxyCreator 自动创建事务代理.....	309	10.2.3 开发顺序.....	351
8.6.4 使用 DefaultAdvisorAutoProxyCreator 自动创建事务代理.....	310	10.3 数据库结构.....	351
8.7 Spring 与 Struts 2 的整合应用.....	311	10.4 项目总体架构.....	352
8.8 Spring 与 Hibernate 的整合应用.....	314	10.5 持久层及其实现.....	353
8.9 Spring 与 Struts 2、Hibernate 的 整合应用.....	321	10.5.1 POJO 类与映射文件.....	353
		10.5.2 用户管理 DAO 组件及其 实现.....	355
		10.5.3 部门管理 DAO 组件及其 实现.....	358
		10.5.4 职工管理 DAO 组件及其 实现.....	360
		10.6 业务层及其实现.....	362



10.6.1 用户管理 Service 组件及其实现.....	362	10.7.4 系统主界面的实现.....	377
10.6.2 部门管理 Service 组件及其实现.....	364	10.7.5 当前用户功能的实现.....	381
10.6.3 职工管理 Service 组件及其实现.....	366	10.7.6 用户管理功能的实现.....	386
10.7 表示层及其实现.....	367	10.7.7 部门管理功能的实现.....	399
10.7.1 素材文件的准备.....	368	10.7.8 职工管理功能的实现.....	411
10.7.2 公用模块的实现.....	369	本章小结.....	424
10.7.3 登录功能的实现.....	372	思考题.....	424
		附录 实验指导.....	425
		参考文献.....	436



第1章

Java EE 概述

Java EE 是目前 Web 应用(特别是企业级 Web 应用)开发的主流平台之一,其应用是相当普遍的。

本章要点:

- Java EE 简介;
- Java Web 应用开发的主要技术与框架;
- Java Web 应用开发环境的搭建;
- Java Web 项目的创建与部署。

学习目标:

- 了解 Java EE 的概况以及 Java Web 应用开发的主要技术与框架;
- 掌握 Java Web 应用开发环境的搭建方法;
- 掌握 Java Web 项目的创建、部署与管理方法。

1.1 Java EE 简介

1996 年, Sun Microsystems 公司(已于 2009 年被 Oracle 公司收购)推出了一种新的纯面向对象的编程语言——Java。时至今日, Java 的应用已相当广泛与深入。从现在来看, Java 既是一种编程语言,也是一种开发平台(即使用 Java 编程语言编写的应用程序的运行环境)。

根据应用领域的不同, Java 平台分为 3 种,即 Java SE、Java EE 与 Java ME。其中, Java SE 为 Java 平台标准版(Java Platform Standard Edition),提供了 Java 编程语言的核心功能,主要用于开发一般的台式机应用程序(或桌面应用程序);Java EE 为 Java 平台企业版(Java Platform Enterprise Edition),是基于 Java SE 的一种企业级应用标准平台,主要用于快速设计、开发、部署和管理企业级的大型软件系统;Java ME 为 Java 平台微型版(Java Platform Micro Edition),包含了 Java SE 应用程序接口(Application Programming Interface, API)的一个子集,以及一个可运行于移动手机等微型设备之上的小型 Java 虚拟机(Java Virtual Machine, JVM),主要用于开发掌上电脑、智能手机等移动设备使用的嵌入式系统。在此,企业级应用特指具备多层结构、可扩展的、高可靠性的、大规模的网络应用软件,而 Java ME 应用软件则通常作为 Java EE 应用软件的客户端。

由于 Java 的所有平台都包含有一个 Java 虚拟机和一套应用程序接口,因此任何基于这些平台开发的应用都具备 Java 编程语言的优点,包括跨平台、安全性、可扩展性等。

 **说明:** Sun Microsystems 公司在 1998 年推出 JDK 1.2 时,使用了新名称 Java 2 Platform(即 Java 2 平台),修改后的 JDK 称为 Java 2 Platform Software Developing Kit(即 J2SDK),并分为标准版(Standard Edition, J2SE)、企业版(Enterprise Edition, J2EE)与微型版(Micro Edition, J2ME)。其中, J2SE 包含于 J2EE 中,而 J2ME 则包含了 J2SE 的核心,并添加了一些专有类。2005 年 6 月召开 Java One 大会时, Sun Microsystems 公司发布 Java SE 6。此时, Java 的各种版本已经更名,取消了其中的数字 2,分别为 Java SE、Java EE 与 Java ME。

随着 Java 技术的发展, Java EE 平台也得到了迅速的发展,成为 Java 语言中最为活跃的体系之一,并拥有最为活跃的 Java 技术社区。Java EE 构建在 Java SE 的基础之上,提供了 Web 服务、组件模型、管理和通信 API 等,可用来实现企业级的面向服务体系结构(Service-Oriented Architecture, SOA)与 Web 2.0 应用程序。如今,Java EE 更多地表达着一种软件架构和设计思想,而不仅仅是指一种标准平台。

1.2 Java Web 应用开发的主要技术与框架

C/S(Client/Server, 客户机/服务器)与 B/S(Browser/Server, 浏览器/服务器)是目前应用程序的两种主要架构或模式。与基于 C/S 架构的 Windows 应用程序不同,Web 应用程序是基于 B/S 架构的。Web 应用程序部署在 Web 服务器上,因此易于升级和维护。同时,Web 应用程序的访问是通过浏览器进行的,因此只需在客户机上安装一个标准的浏览器即可,而无须安装专门的客户端程序。此外,由于 Web 应用程序的数据分析与处理工作主要是在服务器中完成的,因此对客户机的配置要求不高,特别适合“瘦客户端”的运行环境。随着 Internet 的快速发展,Web 应用程序的使用将更加广泛与深入。

要开发 Java Web 应用,需要掌握一系列相关的技术。必要时,可应用相应的框架,以提高 Java Web 应用的开发效率,并确保其可扩展性与可维护性。

Java Web 应用开发的主要技术包括 HTML/XHTML、XML、JavaScript、Java、JDBC、JSP、JavaBean、Servlet、Ajax 等。其中,JDBC(Java Data Base Connectivity, Java 数据库连接)是一种用于执行 SQL 语句的 Java API(Application Programming Interface, 应用编程接口或应用程序编程接口),由一组用 Java 语言编写的类和接口组成,可让开发人员以纯 Java 方式连接数据库,并完成相应的操作。JSP(Java Server Pages)是由 Sun Microsystems 公司倡导、许多公司参与建立的一种动态网页技术标准(类似于 ASP 与 PHP 技术),可在 HTML 文档(*.htm、*.html)中嵌入 Java 脚本小程序(Scriptlet)和 JSP 标记(tag)等元素,从而形成 JSP 文件(*.jsp)。JavaBean 是 Java 中的一种可重用组件技术(类似于微软的 COM 技术),其本质是一种通过封装属性和方法而具有某种功能的 Java 类。Servlet 是一种用 Java 编写的与平台无关的服务器端组件,其实例化对象运行在服务器端,可用于处理来自客户端的请求,并生成相应的动态网页。Ajax(Asynchronous JavaScript and XML, 异步 JavaScript 和 XML)是一种创建交互式网页应用的开发技术,其核心理念为使用 XMLHttpRequest 对象发送异步请求,可实现 Web 页面的动态更新。

Java Web 应用开发的流行框架主要有 Struts 2、Hibernate、Spring、DWR 等。其中,Struts 2 是一个基于 MVC 设计模式的 Web 应用框架,本质上相当于一个 Servlet,可在 MVC 设计模式中作为控制器(Controller)建立模型(Model)与视图(View)之间的数据交互。Hibernate 是一个开源的 ORM(Object-Relation Mapping, 对象关系映射)框架,通过对 JDBC 进行轻量级封装(未完全封装),让 Java 编程者可以使用面向对象的编程思想来操纵数据库,是一种十分成功的 ORM 解决方案。Spring 是一个由 Rod Johnson 创建的开源框架,具有简单、可测试与松耦合的特征,可有效简化企业级应用的开发。DWR 是一种用于创建 Ajax Web 站点的开源框架,可让开发人员无须具备专业的 JavaScript 知识就可以轻松实现 Ajax 功能,并有效减少代码的编写量。目前,在实际的应用开发中,Struts 2、Hibernate

通常会与 Spring 框架整合在一起使用，从而构成十分流行的 SSH2 模式。在此基础上，可再配合使用 DWR 框架，为应用添加相应的 Ajax 功能，以改善用户的操作体验。

1.3 Java Web 应用开发环境的搭建

要进行 Java Web 应用的开发，首先要搭建好相应的开发环境。为此，需逐一完成 JDK 开发包、Web 服务器、IDE 开发工具以及数据库管理系统的安装与配置。在此，JDK 开发包使用 jdk1.7.0_51，Web 服务器使用 Tomcat 7.0.50，IDE 开发工具使用 MyEclipse 10.7.1，数据库管理系统使用 SQL Server 2008。下面，简要介绍一下在 Windows 7 中构建开发环境的基本步骤与关键配置。

1.3.1 JDK 的安装与配置

Java Web 应用程序的开发需要 JDK(Java Development Kit, Java 开发工具包)的支持。JDK 是整个 Java 的核心，内含 JRE(Java Runtime Environment, Java 运行环境)、Java 工具、Java 基础类库以及相关范例与文档，是一种用于构建在 Java 平台上发布的应用程序、Applet 与组件的开发环境。如今，Oracle 公司已取代 Sun 公司负责 JDK 的维护与升级工作，并定期在其官网上发布最新的版本。

1. 下载

各种版本的 JDK 均可从其官网免费下载，地址为

```
http://www.oracle.com/technetwork/java/javase/downloads/index.htm
```

或者

```
http://java.sun.com/javase/downloads/index.jsp
```

下载 JDK 7 Update 51 版，其安装文件为 jdk-7u51-windows-i586.exe。

2. 安装

JDK 的安装过程非常简单，只需双击安装文件 jdk-7u51-windows-i586.exe，启动 JDK 的安装向导(见图 1.1)，并根据向导完成后续的有关操作即可。在此，分别将 JDK 与 JRE 的安装目录设置为 C:\Program Files\Java\jdk1.7.0_51 与 C:\Program Files\Java\jre7，如图 1.2、图 1.3 所示。

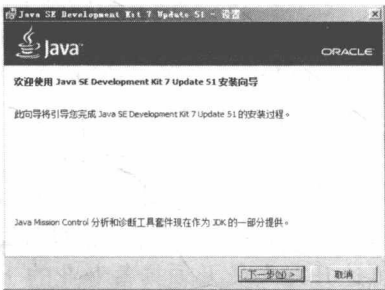


图 1.1 JDK 的安装向导

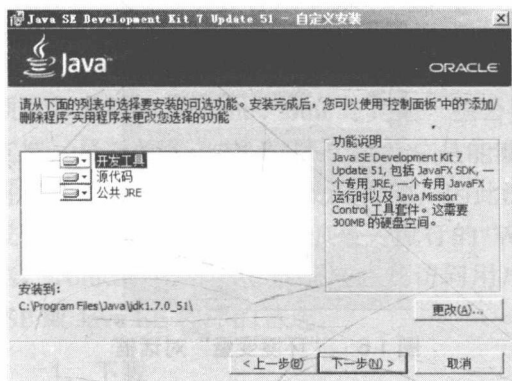


图 1.2 JDK 的安装目录

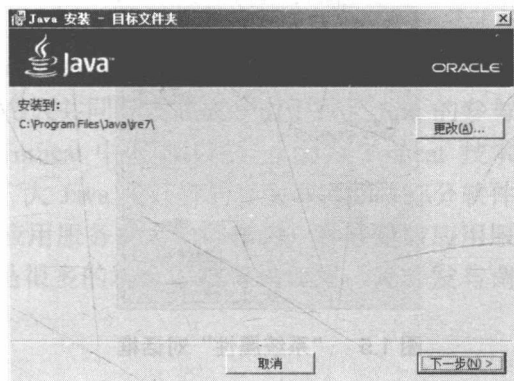


图 1.3 JRE 的安装目录

3. 配置

JDK 安装完毕后，还需进行相应的配置，主要是创建或设置有关的系统变量，以告知 Windows 系统 JDK 与 JRE 的安装位置。

以系统变量 `JAVA_HOME` 为例，其创建的基本步骤如下。

(1) 在桌面右击“计算机”图标，选择其快捷菜单中的“属性”菜单项，打开图 1.4 所示的“控制面板\系统和安全\系统”窗口。



图 1.4 “控制面板\系统和安全\系统”窗口

- (2) 单击“高级系统设置”链接，弹出图 1.5 所示的“系统属性”对话框。
- (3) 单击“环境变量”按钮，弹出图 1.6 所示的“环境变量”对话框。

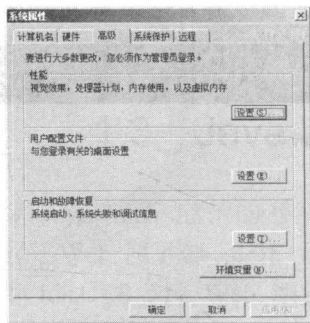


图 1.5 “系统属性”对话框

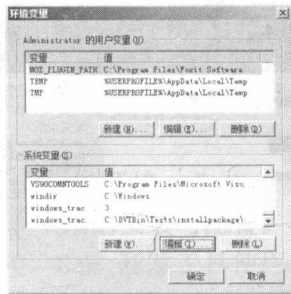


图 1.6 “环境变量”对话框

(4) 单击“系统变量”列表框下的“新建”按钮，弹出图 1.7 所示的“新建系统变量”对话框，并在其中输入变量名 JAVA_HOME、变量值“C:\Program Files\Java\jdk1.7.0_51”，最后单击“确定”按钮。

(5) 单击“确定”按钮，关闭“环境变量”对话框。

(6) 单击“确定”按钮，关闭“系统属性”对话框。

类似地，新建环境变量 Path，将其值设置为“.;%JAVA_HOME%\bin”（若该变量已经存在，则在其值前添加“.;%JAVA_HOME%\bin;”）；新建环境变量 CLASSPATH，将其值设置为“.;%JAVA_HOME%\lib”（若该变量已经存在，则在其值前添加“.;%JAVA_HOME%\lib;”）；新建环境变量 JRE_HOME，将其值设置为“C:\Program Files\Java\jre7”（若该变量已经存在，则在其值前添加“C:\Program Files\Java\jre7”）。

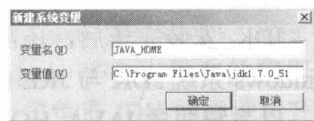


图 1.7 “新建系统变量”对话框

4. 测试

配置完毕后，为检验 JDK 是否正常，可按以下步骤进行操作。

(1) 单击“开始”按钮，选择“开始”菜单中的“运行”菜单项，弹出“运行”对话框（见图 1.8），然后输入 cmd 命令，单击“确定”按钮，打开命令提示符窗口（见图 1.9）。

(2) 输入并执行 java -version 命令，若能显示当前 JDK 的版本（在此为 java version “1.7.0_51”），则表示一切正常（见图 1.9）。

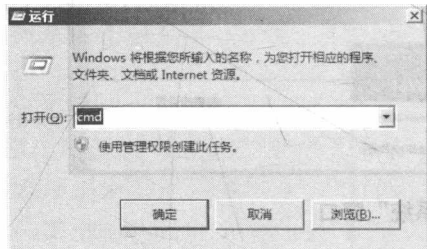


图 1.8 “运行”对话框

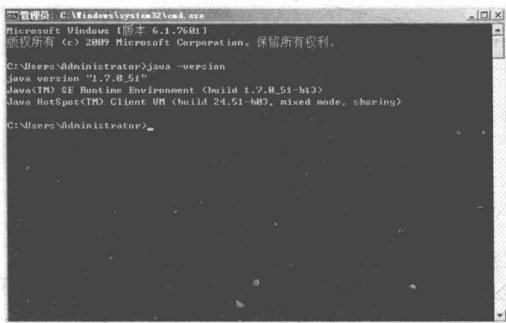


图 1.9 命令提示符窗口

1.3.2 Tomcat 的安装与配置

Tomcat 是 Apache 软件基金会(Apache Software Foundation)Jakarta 项目中的一个核心子项目,最初由 Apache、Sun 与其他一些公司及个人共同开发而成。由于有了 Sun 的参与和支持,最新的 Servlet 与 JSP 规范总是能够在 Tomcat 中得到体现。正因为 Tomcat 技术先进、性能稳定,而且是免费开源的,因此深受广大 Java 爱好者的喜爱并得到了部分软件开发商的认可,最终成为目前最为流行的 Web 应用服务器之一。作为一种轻量级应用服务器, Tomcat 在中小型系统与并发访问用户不是很多的场合下被普遍使用,是开发与调试 JSP 或 Java EE 程序的首选。

1. 下载

各种版本的 Tomcat 均可从其官网免费下载,地址为 <http://tomcat.apache.org/>。

下载安装版的 Tomcat-7.0.50,其安装文件为 apache-tomcat-7.0.50.exe。

2. 安装

Tomcat 的安装过程非常简单,只需双击安装文件 apache-tomcat-7.0.50.exe,启动 Tomcat 的安装向导(见图 1.10),并根据向导完成后续的有关操作即可。在此,选择 Normal 安装类型(见图 1.11),采用默认的端口设置,创建管理员登录账号 admin,并将其密码设置为 12345(见图 1.12),同时选定 JRE 的安装路径为 C:\Program Files\Java\jre7(见图 1.13),指定 Tomcat 的安装目录为 C:\Program Files\Apache Software Foundation\Tomcat 7.0(见图 1.14)。



图 1.10 Tomcat 的安装向导

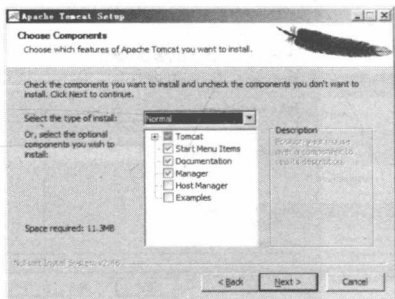


图 1.11 Tomcat 的安装类型

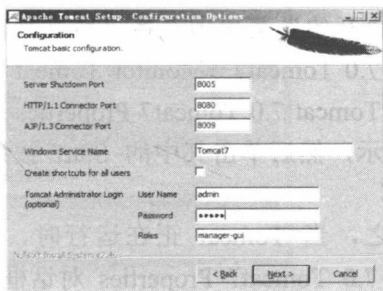


图 1.12 Tomcat 的基本配置

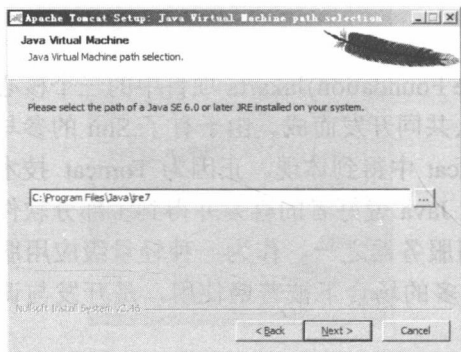


图 1.13 Tomcat 的 JRE 路径

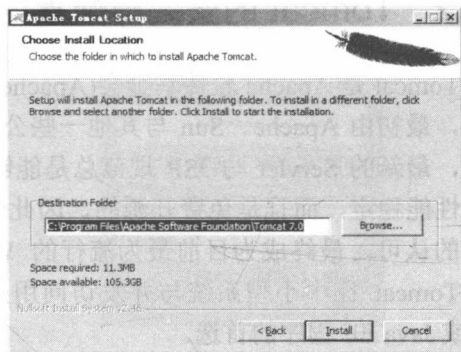


图 1.14 Tomcat 的安装路径

3. 配置

Tomcat 安装完毕后，必要时可对其进行相应的配置。

1) HTTP 端口号的修改

Tomcat 默认的 HTTP 端口号为 8080，若要使用其他端口号，可直接打开 Tomcat 的配置文件 C:\Program Files\Apache Software Foundation\Tomcat 7.0\conf\server.xml，找到如下代码，并将其中的 8080 改为所需要的端口号(如 8090 等)即可。

```
<Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
```

2) 管理员登录账号的设置

Tomcat 管理员登录账号的信息保存在 C:\Program Files\Apache Software Foundation\Tomcat 7.0\conf\tomcat-users.xml 文件中。通过修改该文件，即可实现管理员登录账号的添加、修改或删除。例如，以下代码表明当前 Tomcat 的管理员登录账号为 admin，密码为 12345。

```
<tomcat-users>
<role rolename="manager-gui"/>
<user username="admin" password="12345" roles="manager-gui"/>
</tomcat-users>
```

4. 启动与关闭

在“开始”菜单中选择“所有程序”→Apache Tomcat 7.0 Tomcat7→Monitor Tomcat 菜单项，打开 Apache Tomcat 7.0 Tomcat7 Properties 对话框，如图 1.15 所示，然后单击其中的 Start 按钮，即可启动 Tomcat。

反之，当 Tomcat 正在运行时，单击 Apache Tomcat 7.0 Tomcat7 Properties 对话框中的 Stop 按钮，可随即关闭 Tomcat。

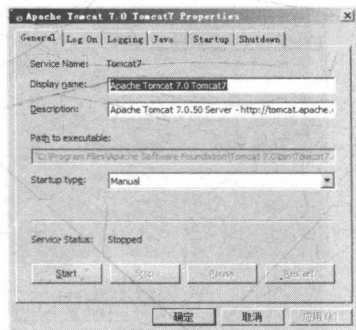


图 1.15 Apache Tomcat 7.0 Tomcat 7 Properties 对话框

5. 测试

为测试 Tomcat 是否正常，应先启动 Tomcat，然后打开浏览器，在地址栏输入“http://localhost:8080/”并按 Enter 键。若显示图 1.16 所示的 Apache Tomcat/7.0.50 页面，则表明 Tomcat 一切正常。单击其中的 Manage App 按钮，弹出图 1.17 所示的“Windows 安全”对话框，并在其中输入用户名 admin 与密码 12345，再单击“确定”按钮，即可打开图 1.18 所示的 Tomcat Web Application Manager 页面。在该页面中，可对当前发布的有关应用进行相应的管理，包括启动、停止等。

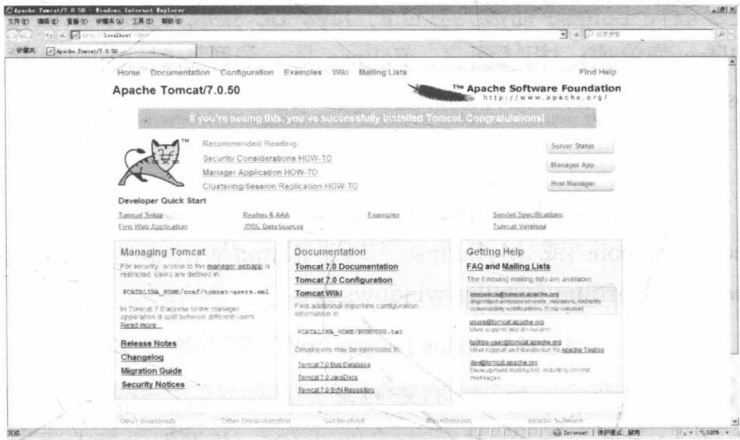


图 1.16 Apache Tomcat/7.0.50 页面

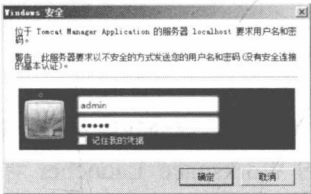


图 1.17 “Windows 安全”对话框

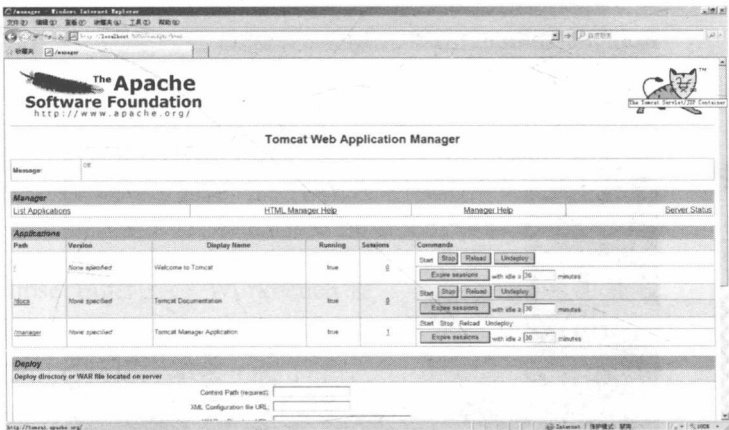


图 1.18 Tomcat Web Application Manager 页面