

精通

JSF — 基于 EJB Hibernate Spring 整合开发与项目实践

徐明华 邱加永 郑经煜 编著

 传智播客 推荐
ITCAST.CN

本书侧重于项目实践，帮助读者进一步提升实战技能，
领悟项目开发的技巧，是很值得阅读的佳作。

前端技术：JSF页面布局方案，JavaScript、Ajax等技术应用

框架整合：JSF、Spring、Hibernate、EJB等实战开发知识

四大案例：帮助读者提高项目实战水平

留言本系统（JSF + JDBC）

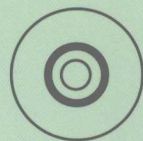
博客管理系统（JSF + Facelets + iBATIS）

新闻发布系统（JSF + EJB）

在线宠物店（JSF + Spring + Hibernate）

开发视频：300多分钟的开发视频，帮助读者顺利完成程序
调试

 人民邮电出版社
POSTS & TELECOM PRESS



源代码+开发视频

精通

JSF — 基于 EJB Hibernate Spring 整合开发与项目实践

徐明华 邱加永 郑经煜 编著

人民邮电出版社

北 京

图书在版编目 (CIP) 数据

精通JSF: 基于EJB、Hibernate、Spring整合开发与项目实践 / 徐明华, 邱加永, 郑经煜编著. —北京: 人民邮电出版社, 2009. 2

ISBN 978-7-115-19213-4

I. 精… II. ①徐…②邱…③郑… III. JAVA语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字 (2008) 第179580号

内 容 提 要

本书循序渐进、由浅入深地讲解了如何构架和开发 JSF 应用程序。全书共 16 章, 分为 JSF 基础篇、JSF 实战技术篇和 JSF 项目实战篇。其中第 1~2 章为 JSF 基础篇, 主要介绍 JSF 的基本结构和工作原理, 以及开发环境的安装与配置。第 3~12 章为 JSF 实战技术篇, 主要介绍了 JSF 组件、JSF 托管 Bean、JSF 导航、转换器、验证器、国际化、JSF 的页面布局方案、Facelets 使用、JSF 自定义组件以及和 JavaScript、Ajax 的整合等知识, 并结合实例把这些实战技术应用起来, 以便达到学通和会用的目的。第 13~16 章为 JSF 项目实战篇, 主要通过 4 个项目案例把 JSF 和目前流行的技术: Spring、Hibernate、EJB 3、iBATIS、Ajax 整合来完成企业开发。案例中讲解的许多通用的功能模块, 读者稍加修改就可以直接应用于自己的项目中。通过这些接近于企业真实项目的案例实践, 帮助读者进一步提升实战技能, 领悟项目开发中的技巧, 从而更加透彻地理解 JSF 框架, 为适应实战应用打下很好的基础。

本书适合正在开发 Java Web 应用的架构师、应用开发人员阅读, 也可作为大中专院校相关专业学生的教程及编程爱好者的自学用书。

精通JSF: 基于EJB、Hibernate、Spring整合开发与项目实践

◆ 编 著 徐明华 邱加永 郑经煜

责任编辑 屈艳莲

执行编辑 张 涛

◆ 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 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

三河市潮河印业有限公司印刷

◆ 开本: 787×1092 1/16

印张: 25.5

字数: 624 千字

印数: 1—3 500 册

2009 年 2 月第 1 版

2009 年 2 月河北第 1 次印刷

ISBN 978-7-115-19213-4/TP

定价: 49.00 元 (附光盘)

读者服务热线: (010)67132692 印装质量热线: (010)67129223

反盗版热线: (010)67171154

推 荐 序

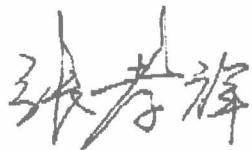
自从 Sun 公司把 JSF 从幕后推到了前台, IBM、Oracle、Red Hat 及一批重量级开发工具厂商也参与了 JSF 规范的制定, 而且 JSF 规范已经成为 Web 开发的标准。我们有理由相信, JSF 将会成为 Java Web 开发阵营中一颗璀璨的明星。

尽管 JSF 技术正在蓬勃发展, 然而市场上关于 JSF 的书籍简直是凤毛麟角, 芳踪难觅。当徐明华先生写了一本用 JSF 进行企业实战开发及和各种流行技术整合的书稿让我翻阅时, 我是比较惊喜, 在这样一个技术高度发达、到处都充满信息的年代, 身为一名开发人员, 我们有必要让自己掌握先进开发技术。《精通 JSF: 基于 EJB、Hibernate、Spring 整合开发与项目实践》无疑是带领开发者步入 Web 开发领域的最佳向导。这本书不但介绍了如何在企业环境中搭建完善的 JSF 开发方案, 而且对案例功能与实用技能深入浅出的阐述是其他同类书中较少见的。书中提供了很多一线开发人员多年积累下来的经验, 并以一系列生动的实例展现给读者, 是很值得关注和阅读的精品。

我在 IT 教育行业上坚持这么多年, 每当看到一些学员学业有成, 心中就比较释然, 虽然本人也编写过一些作品, 在教育培训中摸索出一些经验, 但我个人能力毕竟有限, 我希望更多的人才能与我同肩努力, 为中国的 IT 教育培训贡献自己的力量。

我也期待更多具有开发经验的程序员能编写相关的书籍, 为 IT 教育注入新鲜的力量。

中国著名 IT 培训教育专家
北京传智播客教育科技有限公司教学总监



2009 年 1 月

前 言

在 Java 技术领域中，Sun 公司人员已经将 J2EE 1.5 改名为 Java EE 5，这无疑是具有里程碑意义的一件大事。在 Java EE 5 的规范中，JSF 已经作为表现层框架被纳入 Web 开发之中，JSF 既然已经成为 Web 的开发标准，程序员当然都希望尽快学会，但是由于指导书籍匮乏，使人感到无所适从。本书正是满足读者学习和应用 JSF 进行实际开发的需求编写的。本书内容侧重于项目实战方面的知识，以便给正在使用 JSF 做项目开发或即将使用 JSF 的读者以实战指导。

本书内容

全书共 16 章，分三篇。其中第 1~2 章为 JSF 基础篇，主要介绍 JSF 的基本结构和工作原理以及开发环境的安装与配置。第 3~12 章为 JSF 实战技术篇，主要介绍了 JSF 组件、JSF 托管 Bean、JSF 导航、转换器、验证器、国际化、JSF 的页面布局方案、Facelets 使用、JSF 自定义组件以及和 JavaScript、Ajax 的整合等知识，并结合实例把这些实战技术应用起来，以便达到学以致用目的。第 13~16 章为 JSF 项目实战篇，主要通过 4 个项目案例把 JSF 和目前流行的技术：Spring、Hibernate、EJB 3、iBATIS、Ajax 整合来完成企业开发。

本书案例特色

本书的示例均已通过调试运行，配书光盘中有 5 个小时的开发视频，用以帮助读者顺利调试程序。书中大部分内容是从企业真实开发需求中提炼出来，具有很强的时效性、可操作性和实用性。读者经过学习，在短期内便可胜任有关 JSF 的开发工作。

适合的对象

本书适合正在开发 Java Web 应用的架构师、应用开发人员阅读，也可以作为大中专院校相关专业学生的教程及编程爱好者的自学用书。

关于作者

主编徐明华曾工作于著名 IT 企业，传智播客特约讲师。参与本书编写及代码调试的还有邱加永、郑经煜等同志，他们均有多年的项目开发经验，在此，对各位的辛勤工作表示衷心的感谢！

编者

2009 年 1 月

目 录

第一篇 JSF 基础

第 1 章 JSF 和开发环境配置基础知识···2	
1.1 获取 JSF·····2	
1.2 安装与配置 JDK·····3	
1.3 Tomcat 安装与配置·····5	
1.4 安装与配置 MyEclipse·····9	
1.5 安装与配置 MySQL·····11	
1.6 小结·····14	
第 2 章 创建 JSF 应用程序·····15	
2.1 JSF 开发环境准备·····15	
2.1.1 基本要求·····15	
2.1.2 选择 JSF 的实现·····15	
2.2 创建第一个 JSF 应用·····16	
2.2.1 运行效果·····16	
2.2.2 创建 Web 应用·····17	
2.2.3 映射一个 FacesServlet 的实例·····17	
2.2.4 创建 JSF 页面·····18	
2.2.5 定义页面导航·····22	
2.2.6 创建一个后台 Bean·····23	
2.2.7 添加一个托管 Bean 声明·····24	
2.2.8 运行调试·····24	
2.3 JSF 请求处理生命周期详解·····25	
2.3.1 恢复视图·····26	
2.3.2 应用请求值·····26	
2.3.3 处理验证·····26	
2.3.4 更新模型值·····27	
2.3.5 调用应用程序·····27	
2.3.6 呈现响应·····27	
2.4 第一个 JSF 应用的请求处理生命周期分析·····27	
2.5 关于用户界面组件的 immediate 属性·····30	
2.6 MVC 与 JSF·····30	
2.6.1 经典 MVC 设计模式·····30	
2.6.2 JSF 的模型、视图和控制器·····31	
2.7 小结·····32	

第二篇 JSF 实战技术

第 3 章 JSF 组件·····34	
3.1 HTML 标签组件说明·····34	
3.1.1 输入类标签·····36	
3.1.2 输出类标签·····37	
3.1.3 选择类标签·····38	
3.1.4 命令类标签·····42	
3.1.5 其他类标签·····42	
3.1.6 页面绘制实例·····43	
3.2 Core 标签组件说明·····45	
3.3 事件处理·····48	
3.3.1 值改变事件·····48	

3.3.2 动作事件	52	第 7 章 验证器	92
3.4 综合应用	54	7.1 验证器的基本知识	92
3.5 小结	60	7.1.1 为什么需要验证模型	92
第 4 章 JSF 托管 Bean	61	7.1.2 验证的时机	92
4.1 JavaBean 类型的托管 Bean	61	7.1.3 使用标准验证器	93
4.2 容器类型的托管 Bean	63	7.2 自定义验证器	93
4.3 初始化托管 Bean	65	7.2.1 定义一个类, 实现 javax.faces. validator.Validator 接口	93
4.3.1 初始化简单对象	65	7.2.2 在 Faces 上下文中注册 自定义验证器	94
4.3.2 初始化列表	66	7.2.3 用 <f.validator/> 标签在 页面中使用自定义验证器	94
4.4 托管 Bean 的有效范围	67	7.3 转换器和验证器综合示例	95
4.5 如何外部调用托管 Bean	68	7.3.1 创建用户注册的应用	95
4.5.1 在页面上调用 Bean	69	7.3.2 创建用户注册所需要绑定的 后台 Bean	95
4.5.2 在组件中调用 Bean	69	7.3.3 配置托管 Bean	95
4.5.3 在配置文件中调用托管 Bean	71	7.3.4 添加自定义转换器和自定义 验证器	96
4.6 小结	71	7.3.5 创建用户注册页面	96
第 5 章 导航规则	72	7.3.6 配置导航规则	98
5.1 静态导航实例	72	7.3.7 运行查看效果	98
5.1.1 创建 Bean 文件	73	7.4 小结	99
5.1.2 设置配置文件	73	第 8 章 JSF 应用的国际化	100
5.1.3 绘制页面文件	74	8.1 国际化和本地化的概念	100
5.1.4 设置 web.xml	75	8.2 Java 对国际化的支持	101
5.1.5 使用静态导航的疑问	76	8.2.1 Java 国际化原理	101
5.2 动态导航实例	76	8.2.2 Java 国际化实例	101
5.2.1 代码清单	76	8.3 JSF 对国际化的支持	102
5.2.2 配置清单	77	8.3.1 创建本地化资源文件	102
5.2.3 主页面清单	78	8.3.2 加载一个资源包	103
5.3 非常规导航	79	8.3.3 引用本地化静态数据	103
5.3.1 值改变事件	79	8.3.4 引用本地化动态数据	103
5.3.2 动作事件	83	8.3.5 引用错误消息	104
5.4 小结	85	8.3.6 图片、日期时间和数字的 国际化	105
第 6 章 转换器	86	8.4 JSF 国际化综合实例	105
6.1 转换的时机	86	8.4.1 实现用户登录验证的 国际化	105
6.2 使用标准转换器	87	8.4.2 实现自选语言栏	111
6.2.1 标准转换器的使用方法	87	8.5 小结	115
6.2.2 Java 基本类型转换器的使用	87	第 9 章 JSF 页面布局方案	116
6.2.3 DateTime 转换器的使用	88	9.1 使用 JSP 的 include 实现页面 布局	117
6.2.4 Number 转换器的使用	89	9.1.1 include 知识介绍	117
6.3 自定义转换器	89	9.1.2 include 实现页面布局	117
6.3.1 定义一个类, 实现 javax.faces. convert.Converter 接口	89		
6.3.2 在 Faces 上下文中注册 自定义转换器	91		
6.3.3 用 <f.converter/> 标签在页面中 使用自定义转换器	91		
6.4 小结	91		

9.2 使用 Tiles 框架实现页面布局	119	11.2 JSF 整合 Ajax	137
9.2.1 设置 Tiles 模板引擎	119	11.2.1 为什么要使用 Ajax	137
9.2.2 使用 XML 配置模板资源	120	11.2.2 Ajax 简介	137
9.2.3 创建用于布局的 JSP 文件	121	11.2.3 Ajax 的工作原理	137
9.2.4 创建内容 JSP 文件	121	11.2.4 Ajax 示例	138
9.2.5 使用 Tiles 框架的布局方案	122	11.2.5 Ajax4jsf 框架	141
9.2.6 运行查看效果	122	11.3 小结	147
9.3 小结	122	第 12 章 JSF 自定义组件	148
第 10 章 Facelets 的使用	123	12.1 JSF 组件模型	148
10.1 安装 Facelets	123	12.2 JSF 自定义组件开发步骤	149
10.2 使用 Facelets 进行页面模板化 布局	125	12.2.1 创建自定义的组件类	149
10.3 使用 JSFC	128	12.2.2 定制渲染器	151
10.4 复合组件	130	12.2.3 注册组件	151
10.4.1 创建复合组件定义文件	130	12.2.4 编写标签处理类	152
10.4.2 创建 Facelets 标签库描述 文件	131	12.2.5 创建标签库描述文件来登记 这个标签	153
10.4.3 在 web.xml 中声明标签库 描述文件	131	12.3 自定义组件实例: 数据分页	156
10.4.4 用命名空间导入标签库 描述文件	131	12.3.1 运行效果	156
10.5 小结	132	12.3.2 分页组件标签解析	156
第 11 章 JSF 与 Ajax 的整合	133	12.3.3 标签库描述文件	157
11.1 JSF 把输入组件解析成 HTML 标签的细节	133	12.3.4 标签处理类	158
		12.3.5 组件渲染器	159
		12.3.6 注册组件	163
		12.3.7 组件类	163
		12.3.8 一些思考	163
		12.4 小结	164

第三篇 JSF 项目实战

第 13 章 留言本 (JSF+JDBC)	166	iBATIS)	212
13.1 系统概述	166	14.1 系统设计	212
13.2 需求分析	166	14.1.1 功能说明	212
13.2.1 系统总体设计图	167	14.1.2 系统架构	214
13.2.2 用例图	167	14.1.3 技术架构	214
13.3 数据库设计	168	14.2 持久层技术 iBATIS 介绍	214
13.4 页面代码实现页面间转向的 关系	168	14.2.1 iBATIS 介绍	214
13.5 其他功能分析	184	14.2.2 iBATIS 的使用	215
13.6 配置文件	205	14.3 持久层设计	217
13.7 运行工程	208	14.3.1 数据库表设计	217
13.7.1 使用工具	208	14.3.2 创建实体类	218
13.7.2 工程部署	208	14.3.3 创建实体类的 SQL Map 映射文件	220
13.7.3 运行程序	209	14.3.4 DAO 组件设计	227
13.8 小结	211	14.3.5 DAO 接口定义	227
第 14 章 博客管理系统 (JSF+ Facelets+ iBATIS)	212	14.3.6 DAO 实现类	234
14.1 系统设计	212		
14.1.1 功能说明	212		
14.1.2 系统架构	214		
14.1.3 技术架构	214		
14.2 持久层技术 iBATIS 介绍	214		
14.2.1 iBATIS 介绍	214		
14.2.2 iBATIS 的使用	215		
14.3 持久层设计	217		
14.3.1 数据库表设计	217		
14.3.2 创建实体类	218		
14.3.3 创建实体类的 SQL Map 映射文件	220		
14.3.4 DAO 组件设计	227		
14.3.5 DAO 接口定义	227		
14.3.6 DAO 实现类	234		

14.4 业务逻辑层开发	245	15.7.2 工程部署	337
14.4.1 账户业务逻辑类		15.7.3 运行程序	338
AccountService	246	15.8 小结	339
14.4.2 文章分类业务逻辑类			
ArticleTypeService	247	第 16 章 在线宠物店 (Petstore)	
14.4.3 文章业务逻辑类		(JSF+Spring+Hibernate) ...	340
ArticleService	249		
14.4.4 文章回复业务逻辑类		16.1 实例功能说明及系统架构	340
FeedbackService	253	16.1.1 实例功能说明	340
14.5 Web 层设计	255	16.1.2 系统架构	341
14.5.1 Web 层概述	255	16.2 持久层开发	342
14.5.2 前台浏览各功能模块的		16.2.1 持久层域模型设计	342
设计与实现	261	16.2.2 持久层 DAO 组件设计	349
14.5.3 后台管理各功能模块的		16.3 数据层表结构生成	357
设计与实现	274	16.3.1 编写 hibernate.cfg.xml 配置	
14.6 分页组件的设计	279	文件	357
14.6.1 使用方式	279	16.3.2 编写方法调用 SchemaExport	
14.6.2 标签库描述文件	280	类生成数据库脚本	358
14.6.3 标签处理类	282	16.4 业务逻辑层开发	359
14.6.4 组件渲染器	283	16.4.1 业务逻辑接口设计	359
14.6.5 注册组件	286	16.4.2 异常处理	362
14.7 后台管理过滤设计	287	16.4.3 实现业务逻辑组件	362
14.8 运行本项目的说明	289	16.4.4 装配业务逻辑组件	368
14.8.1 开发环境说明	289	16.4.5 事务管理	369
14.8.2 运行环境说明	289	16.5 使用服务定位器模式	370
14.8.3 编码说明	289	16.5.1 ServiceLocator 模式介绍	370
14.8.4 Jar 包说明	289	16.5.2 ServiceLocator 模式实现	370
14.8.5 安装运行说明	289	16.6 JSF 与 Spring 的整合	372
14.9 小结	290	16.7 Web 层设计	372
第 15 章 新闻发布系统 (JSF+EJB) ...	291	16.7.1 Web 层概述	372
15.1 系统概述	291	16.7.2 首页的设计与实现	378
15.2 需求分析	291	16.7.3 浏览宠物模块的设计与	
15.2.1 系统总体设计图	292	实现	381
15.2.2 用例图	292	16.7.4 账户模块的设计与实现	385
15.3 数据库设计	292	16.7.5 购买宠物模块的设计与	
15.3.1 数据库需求分析	292	实现	390
15.3.2 数据库逻辑设计	293	16.8 安全性设计	395
15.3.3 事务处理浅析	294	16.9 运行本项目的一些说明	397
15.3.4 页面间转向关系	294	16.9.1 开发环境说明	397
15.4 页面代码实现	294	16.9.2 查看源码说明	397
15.5 业务层代码分析	318	16.9.3 运行环境说明	397
15.6 配置文件 web.xml	337	16.9.4 编码说明	397
15.7 运行工程	337	16.9.5 Jar 包说明	397
15.7.1 使用工具	337	16.9.6 安装运行说明	398
		16.10 小结	398

精通 JSF：基于 EJB、Hibernate、Spring 整合开发与项目实践

第一篇 JSF 基础

第 1 章 JSF 和开发环境配置基础知识

第 2 章 创建 JSF 应用程序

第 1 章

JSF 和开发环境配置基础知识

JSF 全称为 Java Server Faces，是 Java 技术的一种，专用于构建 Java Web 应用程序新的标准框架。它为 Web 应用开发提供了一个事件驱动的、基于组件的模型。使 Web 开发变得更快、更容易。并且允许开发人员按组件、事件、后台 Bean 以及它们之间的交互性来进行组织开发，避免了以往基于请求、响应和标记来考虑问题的弊端。

JSF 的开发流程和 ASP.NET 中所倡导的 code behind 方式很相似，其核心包括事件驱动、组件和标签等。封装程度非常高，它得到了 Sun、Oracle、IBM 和 BEA 等全球知名软件厂商的大力支持，正是因为如此，决定了 JSF 在许多 Java 流行开源框架中的领导地位。

JSF 是第一个以用户界面为核心的 Java Web 开发框架，与其他 Java Web 框架以“页面”为控制粒度不同的是，JSF 的控制粒度已经细化到页面上的“组件”，这是 JSF 简化 Java Web 开发的关键点之一。它已经正式成为 Java EE 规范的一部分。

1.1 获取 JSF

JSF 作为一个标准，它有很多参考实现。其中较为成熟的要数 Sun 公司和 Apache 的 MyFaces。本书所涉及的相关内容均以 Sun 公司的参考实现来进行。

目前 JSF 1.2 版已经成为 Java EE 的一个组成部分，如果计算机中已经安装了 Java EE 的 JDK，那么就不必再额外去下载 JSF 了。在本书实例中，建议大家还是单独下载 JSF 的参考实现。

下载 JSF 需要完成以下几个步骤。

(1) 在浏览器地址栏中输入地址 <http://java.sun.com/javaee/javaserverfaces/download.html>，单击如图 1.1 所示的链接。

(2) 在图 1.2 所示的界面中单击 Download 链接，将会显示所有已经发布的 JSF 各个版本的信息，如图 1.3 所示。

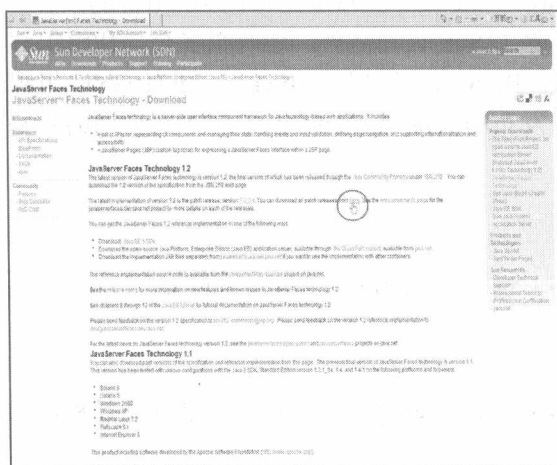


图 1.1 JSF 下载链接页面

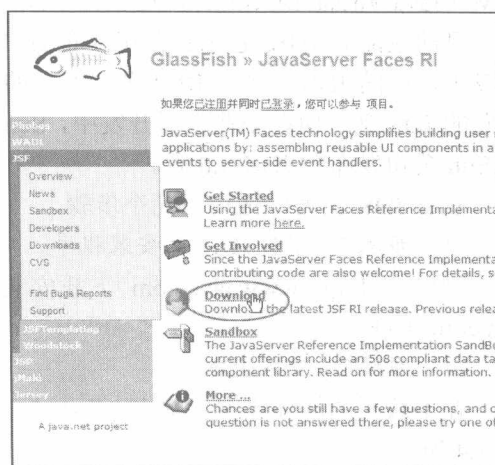


图 1.2 JSF 下载地址



图 1.3 JSF 版本发布信息

(3) 选择最新发布的二进制包（即在图 1.3 中排列在最上面的版本）下载，并解压所下载的文件，将 lib 子目录中的 jsf-api.jar 和 jsf-impl.jar 文件复制到自己构建的项目 WebRoot\WEB-INF\lib 目录下即可。

1.2 安装与配置 JDK

JDK 是 Java 开发工具包，是 Java Development Kit 的缩写。它是整个 Java 的核心开发工具，其中不仅包括了 Java 运行环境（Java Runtime Environment），跨平台的核心 Java 虚拟机

JVM (Java Virtual Machine), 还包括了众多的 Java 开发工具和 Java 基础的类库(rt.jar)。

目前主流的 JDK 是 Sun 公司发布的 JDK, 除了 Sun 公司之外, 还有很多公司和组织都开发了自己的 JDK, 例如 IBM 公司开发的 JDK, BEA 公司开发的 Jrocket, 还有 GNU 组织开发的 JDK 等。但在企业实际开发中, Sun 公司发布的 JDK 还是占主流, 本书所使用的是基于 Sun 公司发布的 JDK 5.0 版。

安装 JDK 需要完成以下几个步骤。

● 步骤 1 下载 JDK 的安装程序。

可以到 <http://www.java.com> 下载安装程序, 也可以到 Sun 公司官方网站 <http://www.sun.com/> 下载。

● 步骤 2 安装 JDK。

在下载后的软件包中, 双击 jdk-1_5_0_09-windows-i586-p.exe 文件, 启动安装过程, 如图 1.4 所示。



图 1.4 安装 JDK

安装 JDK 的过程中, 安装程序会出现一个是否接受许可证协议条款的选项, 选中【我接受……】项, 并单击【下一步】按钮, 表示接受许可证协议条款并继续安装, 如图 1.5 所示。

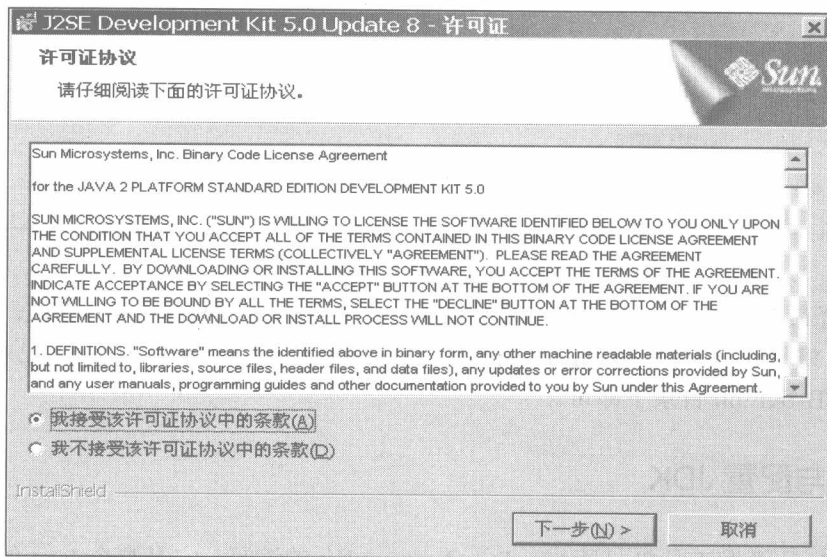


图 1.5 许可证协议提示

然后继续单击【下一步】按钮直至【完成】。

● 步骤3 配置 JDK 开发环境。

安装 JDK 以后,需要配置环境变量。在桌面“我的电脑”图标上单击右键,在弹出的菜单中选择【属性】项,在弹出的界面窗口中选择【高级】选项,并单击【环境变量】的按钮,在弹出的界面窗口的【系统变量】区域中,单击【新建】命令,在出现的文本框中添加以下环境变量:

```
JAVA_HOME=D:\Program Files\Java\jdk1.5.0_08 (JDK 安装的具体位置)
CLASSPATH=.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;
Path=;%JAVA_HOME%\bin;
```

● 步骤4 测试 JDK 是否配置成功。

执行【开始】菜单中的【运行】命令,在出现的文本框中输入 cmd 命令,将显示一个 DOS 命令的界面。在 DOS 提示符后输入 java 或 javac 命令,如果得到如图 1.6 所示的提示,则说明 JDK 配置成功。

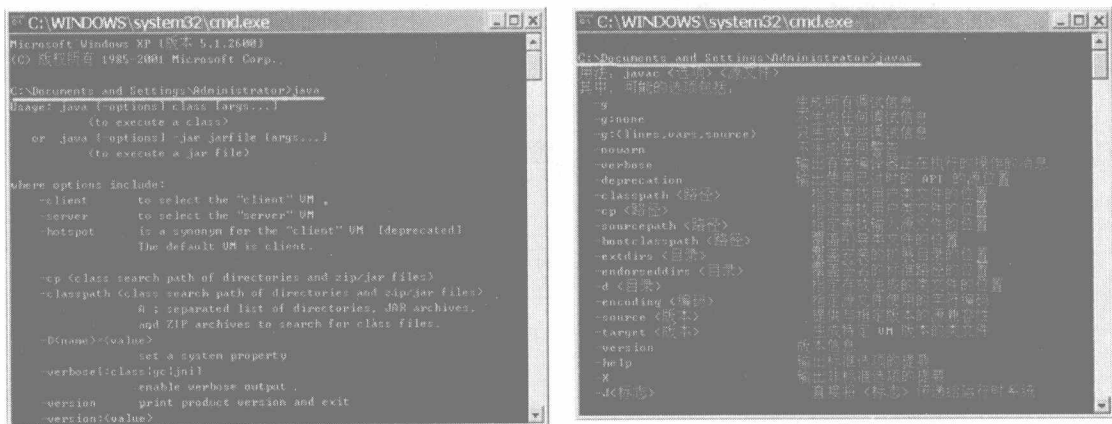


图 1.6 JDK 命令

Java 命令的作用是启动 JVM 执行.class 文件,而 javac 命令是编译 Java 文件,另外常用的命令还有 jar,即 Java 打包命令和 Java 文档生成器的 javadoc 命令。这些都是最基本的命令,读者可以查询 API 文档或其他相关资料以了解更多。

1.3 Tomcat 安装与配置

Tomcat 是 Sun 公司的 JSWDK(JavaServer Web Development Kit)中 Servlet 的运行环境(servlet 容器)。Tomcat 是一个 Server 容器,它不仅完全免费,而且功能强大,一般的中小型 Java Web 应用程序都用它来做服务器。本书中未特殊说明的示例均采用 Tomcat 服务器。

安装 Tomcat 需要完成以下几个步骤。

● 步骤1 下载 Tomcat。

可以在 <http://tomcat.apache.org> 网站下载 Tomcat,目前 Tomcat 已经是 6.0.16 版,但为了系统的稳定性和对 JSF 的支持,建议下载 Tomcat 6.0.14 版。单击如图 1.7 所示的超链接即可下载 Tomcat。

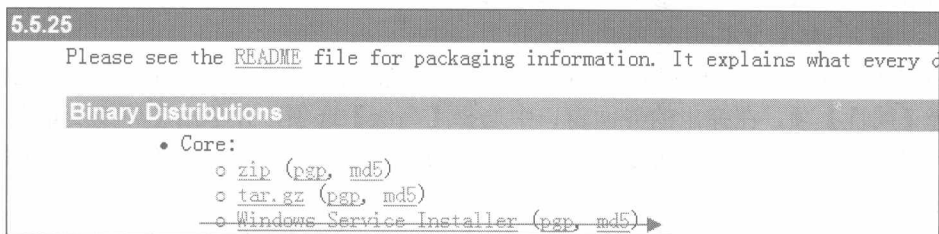


图 1.7 Tomcat 版本显示及下载链接

● 步骤 2 安装 Tomcat。

在下载后文件包中首先找到 `apache-tomcat-6.0.14.exe` 文件，双击此可执行文件。在显示的安装提示界面上单击【Next】按钮，直到出现如图 1.8 所示的管理员权限设置界面。

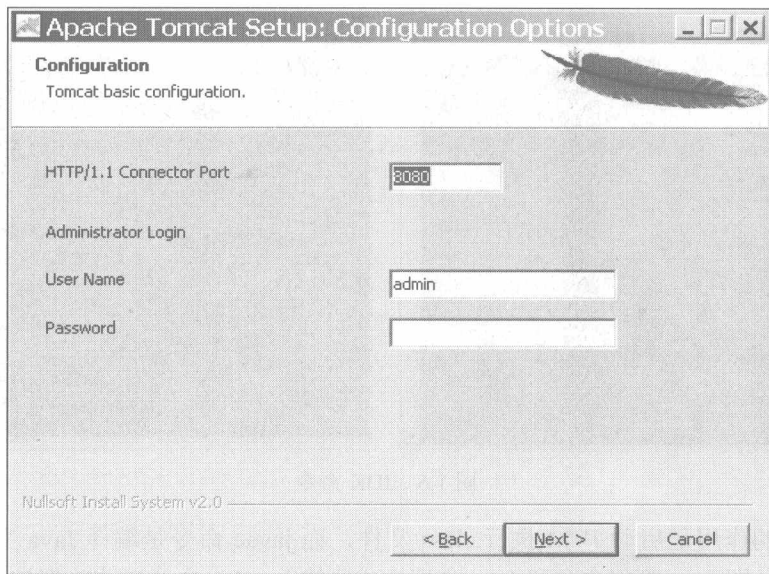


图 1.8 管理员权限设置

在图 1.8 所示的界面中。

第一个参数：是 Tomcat 所占用的端口，默认是 8080，也可以自定义。

第二个参数：是 Tomcat 的用户名。

第三个参数：是 Tomcat 的密码。

填写这些选项可以对访问 Tomcat 的管理页面的用户权限进行限定。如果对所有用户放开 Tomcat 的管理权限，可以按默认值进行设置。

单击【Next】按钮继续，JDK 安装位置如图 1.9 所示。

Tomcat 会自动检测到已经安装的 JDK 的路径，因为 Tomcat 最终还是依赖 Java 环境的。

单击【Install】按钮进行安装，然后在出现的界面单击【finish】按钮，整个安装过程完成。

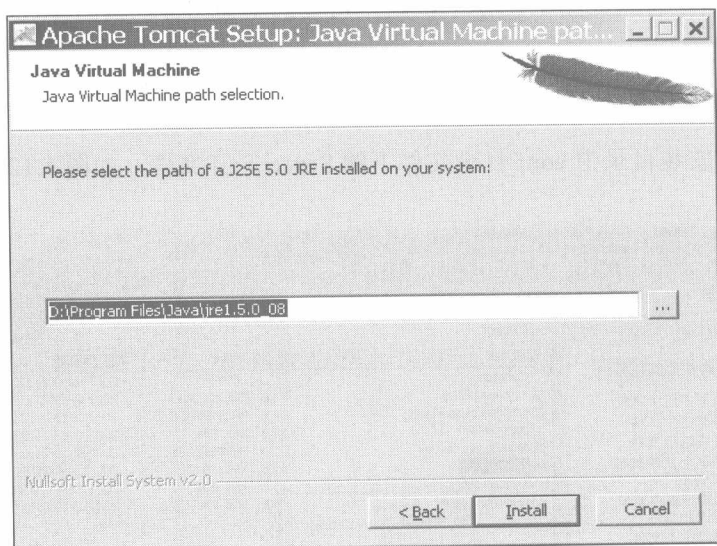


图 1.9 JDK 安装位置

● 步骤 3 配置 Tomcat 环境。

安装 Tomcat 后，依次单击【我的电脑】→【属性】→【高级】→【环境变量】项，在出现的环境变量配置窗口中添加以下环境变量（假定用户的 Tomcat 安装在 D:\Tomcat6.0 目录）。

添加环境变量：TOMCAT_HOME= D:\Tomcat6.0

修改 CLASSPATH 如下：

```
CLASSPATH=.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;%TOMCAT_HOME%;
```

● 步骤 4 测试。

配置完成后，可以到 D:\Tomcat6.0\bin 目录下：找到【tomcat6.exe】文件并双击执行，将出现如图 1.10 所示的状态提示信息。

如果没有抛出任何异常，则说明 Tomcat 启动成功。

信息：Server startup in 2686 ms

图 1.10 状态提示

然后打开浏览器键入：http://localhost: 8080/，出现 Tomcat 首页。如图 1.11 所示。

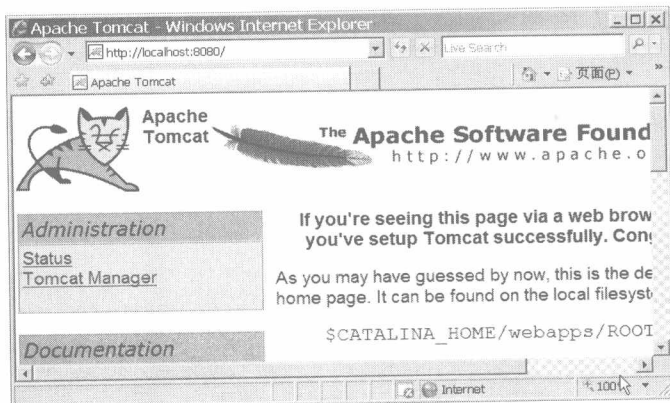


图 1.11 Tomcat 首页