

21世纪高等学校计算机规划教材

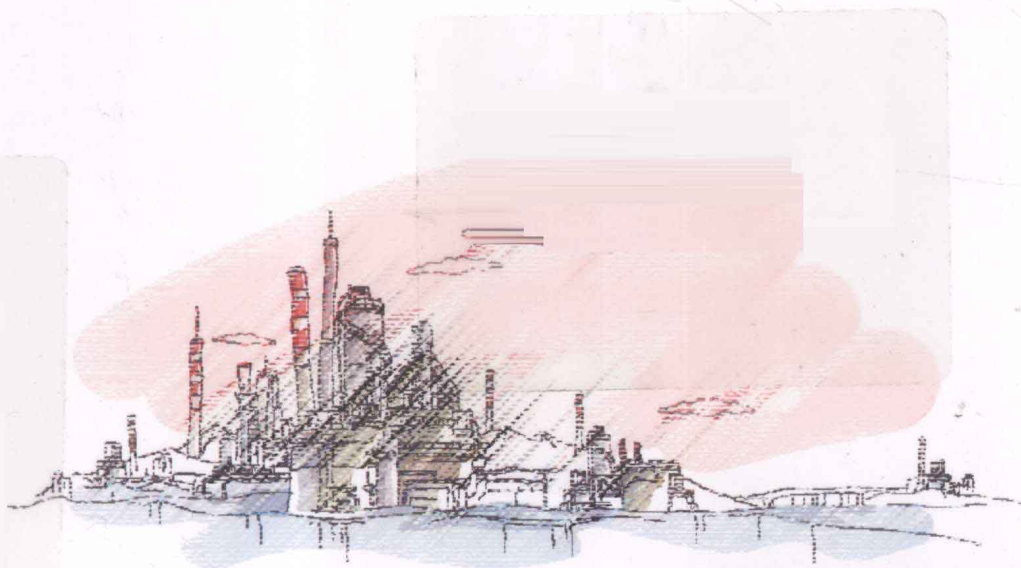
21st Century University Planned Textbooks of Computer Science

Java Web应用技术 与案例教程

Java Web Application Technology
and Case Tutorial

杨树林 胡洁萍 编著

- 涵盖主流技术和方法，体现教学需求
- 融合典型案例和知识，适应案例教学
- 突出项目引导和驱动，贴近实际应用



高校系列



人民邮电出版社
POSTS & TELECOM PRESS

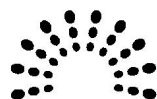
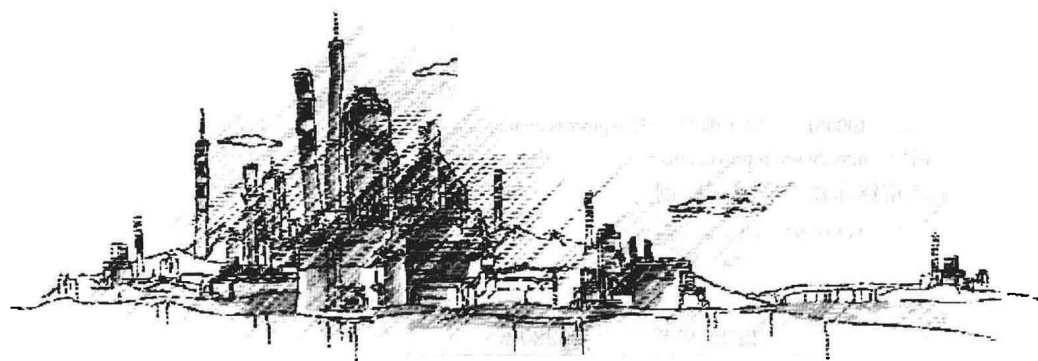
21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

Java Web应用技术 与案例教程

Java Web Application Technology
and Case Tutorial

杨树林 胡洁萍 编著



高校系列

人民邮电出版社

北京

图书在版编目(CIP)数据

Java Web应用技术与案例教程 / 杨树林, 胡洁萍编
著. — 北京: 人民邮电出版社, 2011.10
21世纪高等学校计算机规划教材
ISBN 978-7-115-26280-6

I. ①J… II. ①杨… ②胡… III. ①
JAVA语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2011)第185174号

内 容 提 要

Java Web 应用技术是当今最主流的 Web 程序开发技术之一。目前无论是高校的计算机专业, 还是 IT 培训学校都将 Java Web 应用技术作为教学内容之一, 这对于培养学生的计算机应用能力具有重要的意义。本书将带领读者进入 Java Web 应用技术世界, 由浅入深地学习各项知识。全书共分 9 章, 内容依次为 Java Web 应用技术概述、Java Web 应用技术基础、设计模式与分层架构、数据访问层与业务逻辑层设计技术、控制层实现技术、视图层设计技术、MVC 框架技术 Struts2、持久化技术 Hibernate、BBS 系统设计。每章内容与案例相结合, 有助于学生理解知识, 应用知识, 达到学以致用。书中引进一些新知识和新方法, 内容实用, 重点突出, 讲解精练, 案例典型, 既方便学习, 又便于应用。

本书内容丰富, 实例典型, 知识讲解系统, 适合作为大中专院校计算机及其相关专业的教材或参考书, 也适合软件开发人员及其他有关人员作为技术参考书。

21 世纪高等学校计算机规划教材

Java Web 应用技术与案例教程

- ◆ 编 著 杨树林 胡洁萍
责任编辑 李海涛
- ◆ 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 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京铭印刷有限公司印刷
- ◆ 开本: 787×1092 1/16
印张: 20 2011 年 10 月第 1 版
字数: 524 千字 2011 年 10 月北京第 1 次印刷

ISBN 978-7-115-26280-6

定价: 42.00 元

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

前 言

Java Web 应用技术经过多年的发展日趋成熟,已经成为最佳企业应用解决方案之一。Java Web 应用技术课程与市场衔接紧密,容易使学生掌握前沿应用技术,理解和掌握新的软件开发思想,有利于培养学生面向实际的综合应用开发能力。为了适应形势发展的需要,许多高校开始开设 Java Web 应用技术课程。但目前市场上有关 Java Web 的书多为技术参考书,教材类书籍仍以 JSP 为主的偏多,不能适应学生学习的需要。

应用技术类课程首要的问题是解决学生应用实践能力,为此,必须更新教学内容、改变教学模式。案例教学是计算机语言教学的最有效的方法之一。好的案例对学生理解知识,掌握如何应用知识十分重要。目前一些教材类书籍,内容繁杂,常用知识突出不够;新内容没有及时引入,与市场缺乏衔接;例子缺乏实用性和系统性,相互联系不够,对学生的技术指导不利,无法适应案例教学。而一些技术性较强的参考书,又过分强调技术,难度偏大,知识讲解不够系统,不适合于教学。为此,最好的办法就是要把案例和知识有机结合起来。一方面,跟踪 Java Web 技术发展,适应市场需求,精心选择内容,突出重点、强调实用,使知识讲解系统、精练;另一方面,设计典型的案例,将案例分解,融入到知识讲解中,使知识与案例相辅相承,达到既有利于学生学习知识,又有利于指导学生实践。为此,本书的编写,主要注意以下几点。

(1) 贯穿项目驱动、设计主导、案例教学的思想。作为实践性很强的课程,其主要目的是培养学生实战能力,因此本书注重贯穿项目驱动、设计主导、案例教学的思想。本书的前 3 章以简单的 helloapp 案例为主,重点讲解基本知识,做基础的铺垫,最后介绍了设计模式,并给出了博客网站的分析与总体设计。第 4 章到第 8 章主要围绕博客网站案例讲解知识,并将案例分解到知识当中,根据项目需要逐步引入知识。第 9 章给出相对完整的论坛系统的设计与实现,使设计清晰化,并到达较高的水平。案例与知识有机结合,整个过程都贯彻设计概念,体现了案例教学的思想。

(2) 基于需要选择内容,突出实用,讲解精练。本书在保证知识系统性的同时,注重精选内容,突出实用。从典型的项目出发,基于项目需要展开知识,围绕知识讲解案例,各项知识的讲解不求全而细,而是强调实用,突出重点。在案例选择上也不求多么复杂,而是求典型,注重案例之间的相互联系,案例与知识相辅相承,形成有机的整体,使知识讲解直接化,既有利于学生学习知识,又有利于指导学生实践。

(3) 注意适应市场需求,及时引进新内容,如 Struts2、Hibernate 等。本书选择目前较新的 NetBeans 平台。在设计方法上尽量与企业应用相符合。通过本书不仅可以学习到基本的设计模式,而且可以深入学习分层应用结构的设计方法。

本书不同于普通技术参考书,与一般教材的组织方式有所区别,它根据教学目标和市场需要精选内容,以培养学生应用能力为核心,将典型的、系统性的案例融于书中,将知识和能力有机统一起来,主要特点如下。

- (1) 根据市场应用精心选取教学内容,合理组织内容结构,突出重点。
 - (2) 注意新方法、新技术的引用,突出实用内容(主要指市场开发常用的)。
 - (3) 以案例为线索组织知识,围绕知识设计案例,相辅相成。
-
-
-
-

(4) 突出案例的典型性和针对性, 将系统性(案例之间尽可能有一定的联系, 组合在一起能形成完整系统)的案例融于知识中。

(5) 处理好具体实例与思想方法的关系, 局部知识应用与综合应用的关系。

(6) 知识讲解循序渐进, 难度适宜, 便于教学和学习。

全书共分 9 章, 内容依次为: Java Web 应用技术概述、Java Web 应用技术基础、设计模式与分层架构、数据访问层与业务逻辑层设计技术、控制层实现技术、视图层设计技术、MVC 框架技术 Struts2、持久化技术 Hibernate、BBS 系统设计。除包含许多配合知识学习的例子外, 全书还包含 29 个案例, 其中除了大型的论坛案例外, 其他案例主要是博客网站项目的分解案例, 随知识的讲解逐渐展开。

由于时间仓促, 作者水平有限, 书中难免存在疏漏和不足, 恳请读者批评指正, 使本书得以改进和完善。

编者

2011 年 10 月

目 录

第 1 章 Java Web 应用技术概述1

- 1.1 Java Web 技术简介1
 - 1.1.1 Java Web 应用技术的发展1
 - 1.1.2 Java Web 应用常用技术3
- 1.2 搭建 Java Web 开发环境4
 - 1.2.1 安装与配置 JDK4
 - 1.2.2 安装 Java Web 开发工具5
 - 1.2.3 安装 MySQL 及其开发环境7
 - 1.2.4 安装 Tomcat13
 - 1.2.5 安装 GlassFish16
- 1.3 Java Web 项目的建立与部署19
 - 1.3.1 在 NetBeans 下建立与部署 Java Web 项目19
 - 1.3.2 在 Tomcat 下部署 Web 应用程序22
- 本章小结23
- 习题与实验23

第 2 章 Java Web 应用技术基础25

- 2.1 HTML 网页设计25
 - 2.1.1 HTML 网页基本标记25
 - 2.1.2 表单28
 - 2.1.3 案例 2-1——登录和注册网页设计29
 - 2.1.4 用 JavaScript 验证表单32
 - 2.1.5 CSS34
 - 2.1.6 案例 2-2——用样式表修饰登录和注册网页37
- 2.2 Servlet 技术41
 - 2.2.1 什么是 Servlet41
 - 2.2.2 如何建立 Servlet41
 - 2.2.3 案例 2-3——建立用户登录 Servlet42

2.2.4 Servlet 的配置45

2.2.5 处理请求和响应45

2.2.6 案例 2-4——使用 Servlet 实现验证码47

2.3 JSP 页面设计50

2.3.1 什么是 JSP50

2.3.2 JSP 网页的组成51

2.3.3 JSP 隐含对象53

2.3.4 案例 2-5——使用 JSP 实现用户登录验证54

2.4 应用 JDBC55

2.4.1 JDBC 简介55

2.4.2 JDBC 的重要类和接口55

2.4.3 JDBC 访问数据库的基本步骤56

2.4.4 案例 2-6——实现基于数据库的登录验证58

本章小结60

习题与实验60

第 3 章 设计模式与分层架构62

- 3.1 JSP Model-1 设计模式62
 - 3.1.1 JSP Model-1 模式简介62
 - 3.1.2 JavaBeans 及其使用63
 - 3.1.3 案例 3-1——利用 JSP+JavaBeans 实现基于数据库的登录验证65
- 3.2 MVC 与 JSP Model-2 设计模式67
 - 3.2.1 MVC 概述67
 - 3.2.2 JSP Model-268
 - 3.2.3 Web 组件的 3 种关联关系68
 - 3.2.4 案例 3-2——使用 JSP+Servlet+JavaBeans 实现用户登录70
- 3.3 Java Web 应用中的分层架构72
 - 3.3.1 分层架构模式72

3.3.2	Java Web 应用中的三层结构	73
3.3.3	结合 MVC 模式的分层结构	73
3.3.4	案例 3-3——博客网站的分析与设计	74
	本章小结	88
	习题与实验	88
第 4 章	数据访问层与业务逻辑层设计技术	89
4.1	使用连接池连接数据库	89
4.1.1	数据源与连接池	89
4.1.2	案例 4-1——博客网站的辅助类设计	93
4.2	数据的基本操作	96
4.2.1	数据操作的基本原理	96
4.2.2	查询数据	97
4.2.3	更新数据	98
4.2.4	案例 4-2——博客网站数据访问类的设计	99
4.3	数据分页	104
4.3.1	按页查询数据	104
4.3.2	案例 4-3——分页工具类及几个数据访问类设计	105
4.4	使用存储过程	116
4.4.1	存储过程的定义	116
4.4.2	调用存储过程	117
4.4.3	案例 4-4——使用存储过程分页查询博文	118
4.5	大对象的存取	119
4.5.1	大对象存储的基本方法	119
4.5.2	案例 4-5——设计用户数据访问类 YbUserDao	120
4.6	工厂模式与业务逻辑类设计	125
4.6.1	属性文件及其读取	125
4.6.2	工厂模式建立对象	126
4.6.3	案例 4-6——博客网站工厂模式实现及业务逻辑层设计	126
	本章小结	134
	习题与实验	135

第 5 章 控制层实现技术 136

5.1	多请求处理与控制层实现	136
5.1.1	用一个 Servlet 处理多个请求	136
5.1.2	案例 5-1——实现博客网站的博文分类管理	137
5.2	过滤器与编码处理	141
5.2.1	过滤器	141
5.2.2	案例 5-2——设计一个编码过滤器	142
5.3	状态跟踪与侦听器	144
5.3.1	会话跟踪	144
5.3.2	Servlet 上下文	145
5.3.3	侦听器 Listener	145
5.3.4	案例 5-3——统计访问人数和在线人数	146
5.4	文件的上传和下载	149
5.4.1	上传文件和参数获取	149
5.4.2	保存文件	150
5.4.3	文件下载	151
5.4.4	案例 5-4——实现博客网站的用户功能	151
	本章小结	163
	习题与实验	163

第 6 章 视图层设计技术 165

6.1	自定义标签	165
6.1.1	自定义标签的设计原理	165
6.1.2	案例 6-1——设计博客网站的博文分类标签	166
6.2	表达式语言 EL 和标准标签库 JSTL	171
6.2.1	表达式语言 EL	171
6.2.2	标准标签库 JSTL	172
6.2.3	案例 6-2——实现博文管理	175
6.3	AJAX 技术	187
6.3.1	AJAX 基础	187
6.3.2	XMLHttpRequest 对象	187
6.3.3	AJAX 应用步骤	188
6.3.4	案例 6-3——实现无刷新登录	189
	本章小结	191

习题与实验	192	第 8 章 持久化技术 Hibernate	236
第 7 章 MVC 框架技术 Struts2	193	8.1 Hibernate 原理	236
7.1 Struts2 概述	193	8.1.1 Hibernate 简介	236
7.1.1 Struts 框架的历史	193	8.1.2 在项目中应用 Hibernate	237
7.1.2 Struts2 下载与配置	194	8.2 Hibernate 的实体映射	245
7.1.3 案例 7-1——使用 Struts2 实现博客 网站中的用户登录	195	8.2.1 实体映射基础	245
7.1.4 Struts2 的 MVC 原理	196	8.2.2 实体关系映射	246
7.1.5 Struts2 组件	197	8.2.3 案例 8-1——博客网站的实体类及 其映射设计	250
7.1.6 Struts2 的中文乱码问题	199	8.3 实体操作与数据查询	255
7.2 Action 与结果类型	199	8.3.1 实体操作	255
7.2.1 Action 的实现类	199	8.3.2 数据查询	256
7.2.2 在 Action 中访问 Web 对象	202	8.3.3 辅助类设计	258
7.2.3 多方法的 Action	203	8.3.4 案例 8-2——基于博客网站的 Dao 设计	261
7.2.4 Result 类型	204	本章小结	265
7.3 输入验证	205	习题与实验	265
7.3.1 使用 validate()方法	205	第 9 章 BBS 系统设计	267
7.3.2 使用验证文件验证	207	9.1 系统分析与设计	267
7.3.3 案例 7-2——实现客户登录和注册验证 (使用配置文件验证)	209	9.1.1 系统分析	267
7.4 Struts2 国际化、标签库和 OGNL 表达式	214	9.1.2 总体设计	268
7.4.1 Struts2 国际化	214	9.1.3 实体类(数据模型)设计	271
7.4.2 案例 7-3——实现登录界面的 国际化	216	9.1.4 接口设计	273
7.4.3 常用 Struts2 标签	219	9.1.5 环境配置与公共类实现	275
7.4.4 Struts2 的 OGNL	221	9.2 数据访问层实现	278
7.5 文件上传和下载	222	9.2.1 BbsUserDao 实现	278
7.5.1 利用 Struts2 上传文件	222	9.2.2 BbsLoginDao 实现	278
7.5.2 利用 Struts2 下载文件	223	9.2.3 BbsForumDao 实现	279
7.5.3 案例 7-4——实现博客网站的相册 管理	224	9.2.4 BbsThemeDao 实现	279
7.6 拦截器	231	9.2.5 BbsMessageDao 实现	280
7.6.1 拦截器	231	9.3 业务逻辑层实现	280
7.6.2 案例 7-5——使用拦截器实现博客 网站的权限验证	233	9.3.1 BbsUserService 实现	280
本章小结	234	9.3.2 BbsLoginService 实现	281
习题与实验	235	9.3.3 BbsForumService 实现	282
		9.3.4 BbsThemeService 实现	283
		9.3.5 BbsMessageService 实现	284
		9.4 控制层实现	285
		9.4.1 控制层基类	285

9.4.2 用户控制类	285	9.5.1 界面的总体设计	292
9.4.3 版块控制类	287	9.5.2 起始页与主页设计	293
9.4.4 主题控制类	287	9.5.3 主题视图设计	295
9.4.5 帖子控制类	289	9.5.4 帖子视图设计	299
9.4.6 用于访问控制的拦截器类	290	9.5.5 用户视图设计	303
9.4.7 配置 Action	291	本章小结	309
9.5 视图层设计	292	习题与实验	309

第 1 章

Java Web 应用技术概述

本章要点

- ☐ Java Web 应用技术的产生和发展
- ☐ Java Web 应用常用技术
- ☐ JDK 及其安装
- ☐ Netbeans 的安装方法
- ☐ Tomcat 及 GlassFish 服务器的安装方法
- ☐ Web 项目的建立和部署

Java Web 应用程序会生成包含各种类型的标记语言（HTML 和 XML 等）和动态内容的交互式 Web 页。它通常由 Web 组件（如 JavaServer Pages（JSP）、Servlet 和 JavaBeans）组成，用来修改和临时存储数据、与数据库和 Web 服务交互，以及根据客户端的请求呈现内容。

1.1 Java Web 技术简介

1.1.1 Java Web 应用技术的发展

1996 年，人们发现利用 Java 语言的优点来开发服务器端的应用程序非常有利。于是，许多软件厂商开始发展一些技术来帮助 Java 开发出更高效的服务端程序。但这些技术都必须搭配特殊的服务器或者专门设计。1997 年，JavaSoft 提出了 Java Servlet 概念，它是一个单一、标准、通用的机制，因而很快在服务器端的应用程序开发方面表现出明显优势。

Servlet 是用 Java 编写的 Server 端程序，与协议和平台无关，运行于某个支持 Java 语言的服务器之上，且可以访问服务器上的资源。Java Servlet 是一种可以动态加载的模块，是 Java Web 应用的一个组件。因此，可以动态地扩展服务器能力，并采用请求-响应模式提供 Web 服务。

为了支持企业级应用，并对数据封装、事务处理、交易控制等功能提供良好的技术基础。1998 年，Sun 发布了 EJB（Enterprise JavaBeans）1.0 标准，并在 1998 年发表 JDK1.2 版本的时候，开始使用名称 Java2 Platform，即 Java2 平台。修改后的 JDK 称为 Java2 Platform Software Developing Kit，即 J2SDK，并分为标准版（Standard Edition，J2SE），企业版（Enterprise Edition，J2EE）和微型版（Micro Edition，J2ME）。1999 年，Sun 正式发布了 J2EE 的第一个版本。

2000 年 Sun 公司又提出新的 JSP 技术。JSP 是 Java Server Page 的缩写，它是 Servlet 的扩展。

在传统的网页 HTML 文件 (*.htm, *.html) 中加入 Java 程序片段 (Scriptlet) 和 JSP 标签, 就构成了 JSP 网页。Java 程序片段可以操纵数据库、重新定向网页以及发送 E-mail 等, 实现建立动态网站所需要的功能。所有程序操作都是在服务器端执行, 网络上传送给客户端的仅是得到的结果, 这样大大降低了对客户浏览器的要求, 即使客户浏览器端不支持 Java, 也可以访问 JSP 网页。至此, J2EE 平台的三大核心技术 Servlet、JSP 和 EJB 都已先后问世。紧接着, 遵循 J2EE 标准, 为企业级应用提供支撑平台的各类应用服务软件相继涌现出来。IBM 的 WebSphere、BEA 的 WebLogic 都是这一领域里成功的商业软件平台。随着开源运动的兴起, JBoss 等开源的应用服务器新秀也吸引了许多用户的注意力。

到 2003 年, Sun 的 J2EE 版本已经升级到 1.4 版, 其中 3 个关键组件的版本也演进到了 Servlet 2.4、JSP 2.0 和 EJB 2.1, J2EE 体系及相关的软件产品已经成为 Web 服务端开发的一个强有力的支撑环境。由于 Web 应用程序开发中涉及的很多任务都可以重复或需要多种的模板代码, 因此这一阶段也产生许多框架, 以减少与常见活动相关联的开销。

但是, 从 1999 年诞生的第一个 J2EE 版本一直到 J2EE 1.4 版本, 各版本总被人们不断地抱怨。这并不是因为它不够强大。恰恰相反, 正是因为它太强大了, 强大得让人难以使用。实现一个简单的 J2EE 程序, 需要大量的配置文件, 尽管有些配置文件不是必需的。Sun 公司一直在试图改变这一切, 但一直未能如愿。2002 年, J2EE 1.4 推出后, J2EE 的复杂程度达到顶点。尤其是 EJB 2.0, 开发和调试的难度非常大。也许是要下决心改变这一切, 或者是受到市场和开发人员的压力, Sun 终于在 2006 年 5 月份正式发布了 J2EE 1.5(现改名为 Java EE 5)规范, 并宣称 Java EE 5 将是 Java EE 史上最简单的版本, 将大大降低开发难度。

Java EE(Enterprise Edition)是一种广泛使用的平台, 包含了一组协调技术, 可显著降低成本以及开发、部署和管理以服务器为中心的多层应用程序的复杂性。Java EE 是基于 Java SE 平台构建的, 并提供了一组用于开发和运行可移植、强健、可伸缩、可靠和安全的服务器端应用程序编程接口 (Application Programming Interface, API)。目前 Java EE 已经发展到 Java EE 6。图 1-1 说明了 Java Web 应用的发展过程。

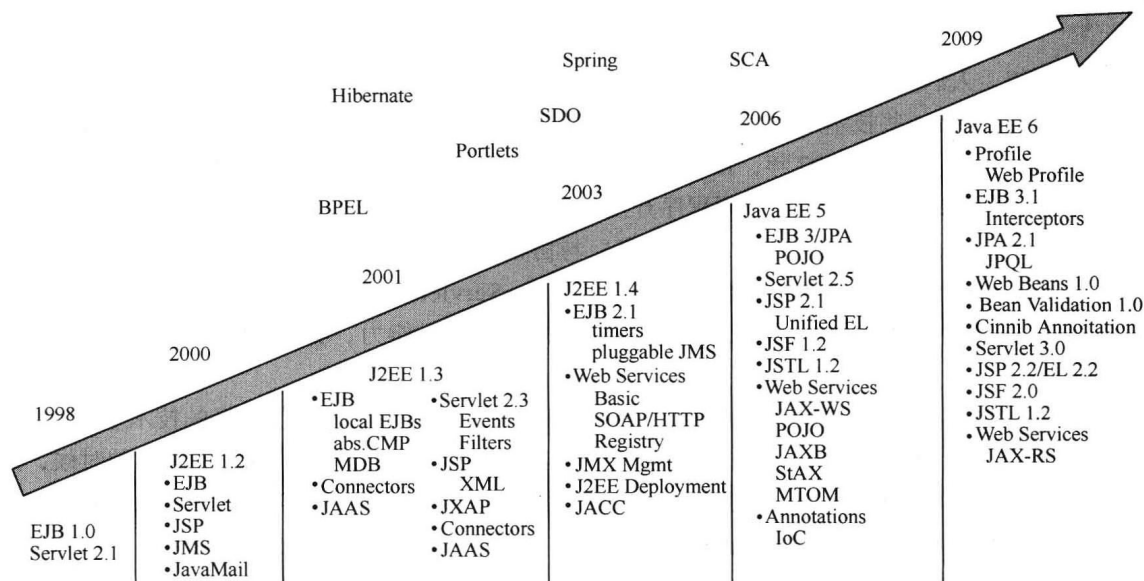


图 1-1 Java Web 应用的发展过程

1.1.2 Java Web 应用常用技术

1. Servlet

Servlet 是 Java 平台上的 CGI 技术。Servlet 在服务器端运行，动态地生成 Web 页面。与传统的 CGI 和许多其他类似 CGI 的技术相比，Java Servlet 具有更高的效率并更容易使用。对于 Servlet，重复的请求不会导致一个程序的多次转载，它是依靠线程的方式来支持并发访问的。

2. JSP

JSP(Java Server Page)是一种实现普通静态 HTML 和动态页面输出混合编码的技术。从这一点来看，非常类似 Microsoft ASP、PHP 等技术。借助形式上的内容和外观表现的分离，Web 页面制作的任务，可以比较方便地划分给页面设计人员和程序员，并方便地通过 JSP 来合成。在运行时，JSP 将会被首先转换成 Servlet，并以 Servlet 的形态编译运行，因此它的效率和功能与 Servlet 相比没有差别，一样具有很高的效率。

3. JDBC

数据库访问接口 (Java Database Connectivity, JDBC)，它使数据库开发人员能够用标准 Java API 编写数据库应用程序。JDBC API 主要用来连接数据库和直接调用 SQL 命令执行各种 SQL 语句。利用 JDBC API 可以执行一般的 SQL 语句、动态 SQL 语句及带 IN 和 OUT 参数的存储过程。Java 中的 JDBC 相当于 Microsoft 平台中的 ODBC(Open Database Connectivity)。

4. Hibernate

Hibernate 是一个面向 Java 环境的对象/关系数据库映射工具，即对象—关系映射 (Object-Relation Mapping, ORM) 工具。它对 JDBC API 进行了封装，负责 Java 对象的持久化，在分层的软件架构中它位于下持久化层，封装了所有数据访问细节，使业务逻辑层可以专注于实现业务逻辑。

5. Spring

Spring 是一个开源框架，是为了解决企业应用程序开发复杂性而创建的。它基于依赖注入和面向方面技术，大大地降低了应用开发的难度与复杂度，提高了开发的速度，为企业级应用提供了一个轻量级的解决方案。

6. EJB

EJB 定义了一组可重用的组件：Enterprise JavaBeans。开发人员可以利用这些组件，像搭积木一样建立分布式应用。在装配组件时，所有的 Enterprise JavaBeans 都需要配置到 EJB 服务器（一般的 Neblogic、WebSphere 等 J2EE 应用服务器都是 EJB 服务器）中。EJB 服务器作为容器和低层平台的桥梁管理着 EJB 容器，并向该容器提供访问系统服务的能力。所有的 EJB 实例都运行在 EJB 容器中。EJB 容器提供了系统级的服务，控制了 EJB 的生命周期。EJB 容器为它的开发人员代管了诸如安全性、远程连接、生命周期管理及事务管理等技术环节，简化了商业逻辑的开发。

7. JSF

JavaServer Faces (JSF) 是 Java Web 应用程序的一个用户界面 (User Interface, UI) 框架。设计 JSF 的目的在于，极大地缓解在 Java 应用服务器上运行的应用程序的编写和维护的压力，并将这些应用程序的 UI 重新呈现给目标客户端。

8. JMS

Java 消息服务 (Java Message Service, JMS) 是一组 Java 应用接口，它提供创建、发送、接收、读取消息的服务。JMS API 定义了一组公共的应用程序接口和相应语法，使得 Java 应用能够

和各种消息中间件进行通信。通过使用 JMS，能够最大限度地提升消息应用的可移植性。JMS 既支持点对点的消息通信，也支持发布/订阅式的消息通信。

9. JNDI

由于 Java EE 应用程序组件一般分布在不同的机器上，所以需要一种机制以便于组件客户使用者查找和引用组件及资源。在 Java EE 体系中，使用 Java 命名和目录接口（Java Naming and Directory Interface, JNDI）定位各种对象，这些对象包括 EJB、JDBC 数据源及消息连接等。JNDI API 为应用程序提供了一个统一的接口来完成标准的目录操作，如通过对象属性来查找和定位该对象。由于 JNDI 是独立于目录协议的，应用还可以使用 JNDI 访问各种特定的目录服务，如 LDAP、NDS、DNS 等。

1.2 搭建 Java Web 开发环境

1.2.1 安装与配置 JDK

1. JDK 简介

JDK (Java Development Kit)，是 Sun 公司最新提供的基础 Java 语言开发工具软件包。其中包含 Java 语言的编译工具、运行工具以及类库。其目录结构如下。

- ❑ **bin**：包含编译器、解释器和一些工具。
- ❑ **lib**：包含类库文件。
- ❑ **demo**：包含各种演示例子。
- ❑ **include**：包含 C 语言头文件，支持 Java 本地接口与 Java 虚拟机调试程序接口的本地编程技术。

- ❑ **jre**：包含 Java 虚拟机、运行时的类包和 Java 应用启动器。
- ❑ **sample**：Sun 配带的帮助学习者学习的 Java 例子。
- ❑ **src.zip**：源码压缩文件。

在 bin 目录下有如下常用工具。

- ❑ **javac.exe**：Java 语言编译器，输出结果为 Java 字节码。
- ❑ **java.exe**：Java 字节码解释器。
- ❑ **javadoc.exe**：帮助文档生成器。
- ❑ **jar.exe**：打包工具。
- ❑ **appletviewer.exe**：小应用程序浏览工具，用于测试并运行 Applet 小程序。

2. JDK 下载与安装

JDK 是一个开源、免费的工具。可以到 Sun 公司的官方网站上下载 JDK 最新版本，网址为 <http://java.sun.com/javase/downloads/index.jsp>。本书使用的 JDK 版本是 Java SE Development Kit 6u25。

下载后得到 jdk-6u25-windows-i586.exe 文件，直接双击运行即开始安装。在安装过程中可以选择安装路径和安装组件，如果没有特殊要求，保留默认设置即可。默认的安装路径是 C:\Program Files\Java\jdk1.6.0_25。之后，设置环境变量如下。

JAVA_HOME=<JSEDK 安装目录>

CLASSPATH=.;%JAVA_HOME%\lib;%JAVA_HOME%\lib\tools.jar

Path=<原 Path>;%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin

1.2.2 安装 Java Web 开发工具

比较流行的集成开发环境主要有 NetBeans、Eclipse、MyEclipse、JCreator、JBuilder。本书使用的是 NetBeans 7.0。

1. NetBeans 简介

NetBeans 是 Sun 公司推出的开放源码的 Java 集成开发环境 (Integrated Development Environment, IDE)。它是使用 Java 语言编写的, 具有很好的可移植性, 适用于各种客户机和 Web 使用, 是业界第一款支持创新型 Java 开发的开放源码 IDE。使用 NetBeans 可以更快地运行 Java Web 应用程序, 跟踪 Java EE 最新技术, 体验快速开发的便捷。

2. NetBeans 7.0 下载

NetBeans 的下载地址是 <http://netbeans.org/downloads/index.html>。可以选择 all 下载, 如图 1-2 所示。



图 1-2 NetBeans 7.0 下载界面

3. NetBeans 7.0 安装

下载后安装步骤如下。

(1) 运行 NetBeans-7.0-ml-windows.exe, 出现如图 1-3 所示对话框。在这个对话框中可以单击【定制】按钮, 打开定制对话框, 选择所要安装的组件。这里定制安装 Apache Tomcat 7.0.11, 其他保留默认设置, 单击【下一步】按钮。

(2) 在如图 1-4 所示的【许可证协议】对话框中, 选中【我接受许可证协议中的条款】复选框, 单击【下一步】按钮。

(3) 在如图 1-5 所示的【JUnit 许可证协议】对话框中, 选中【我接受许可证协议中的条款(A)。安装 JUnit】复选框, 单击【下一步】按钮。

(4) 在如图 1-6 所示的【NetBenas IDE 7.0 安装】对话框中, 可选择 NetBeans 安装位置, 这里保留默认设置, 单击【下一步】按钮。

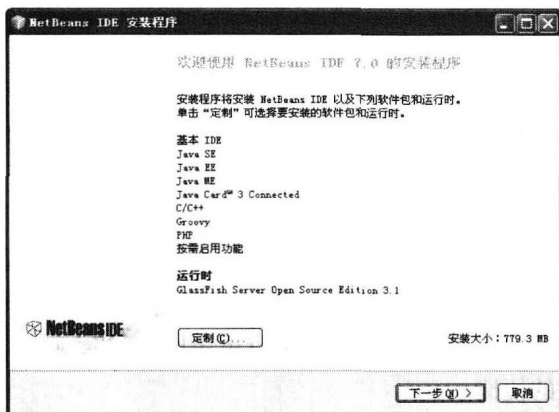


图 1-3 安装定制

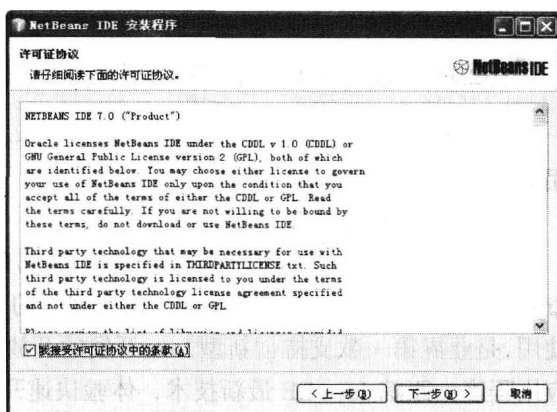


图 1-4 许可证协议

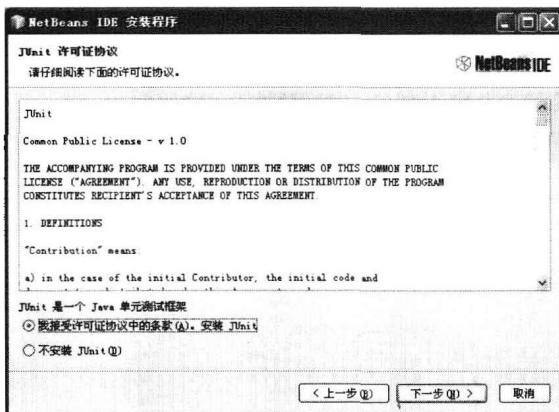


图 1-5 JUnit 许可证协议

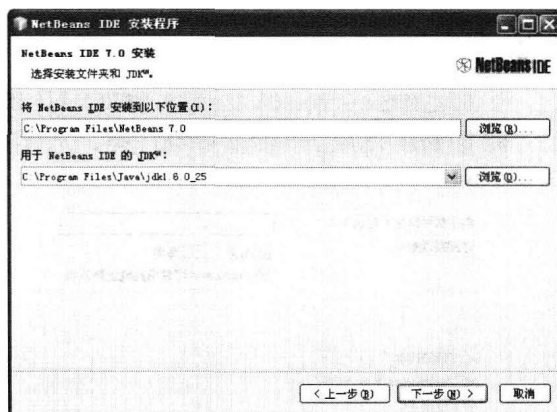


图 1-6 选择 NetBeans 安装文件夹

(5) 在如图 1-7 所示的【GlassFish 3.1 安装】对话框中, 可选择 GlassFish 服务器安装位置, 这里保留默认设置, 单击【下一步】按钮。

(6) 在如图 1-8 所示的【Apache Tomcat 7.0.11 安装】对话框中, 可选择 Tomcat 服务器安装位置, 这里保留默认设置, 单击【下一步】按钮。

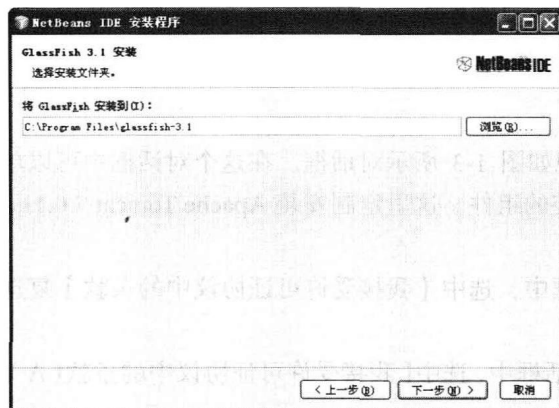


图 1-7 选择 GlassFish 3.1 安装文件夹

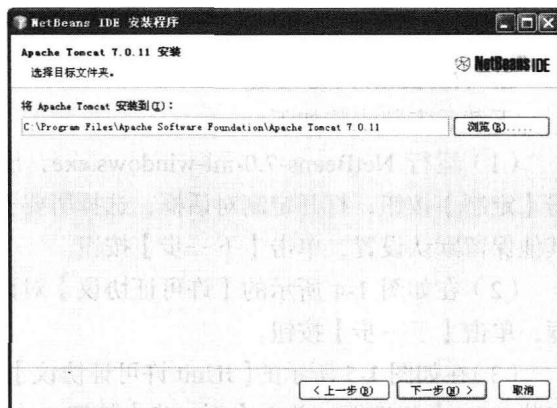


图 1-8 选择 Tomcat 安装文件夹

(7) 在如图 1-9 所示的【摘要】对话框中, 可查看安装的位置, 单击【安装】按钮。

(8) 在如图 1-10 所示的【安装完成】对话框中, 不选中复选框, 单击【完成】按钮。

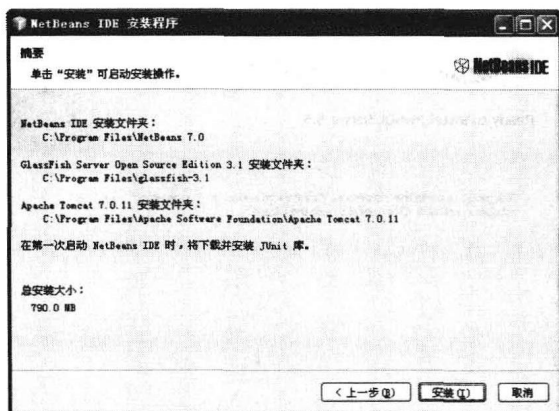


图 1-9 安装摘要

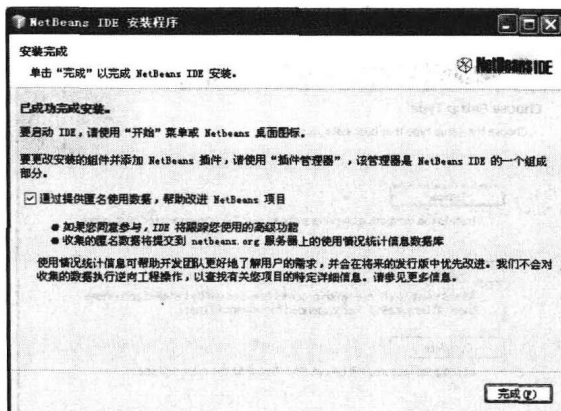


图 1-10 安装完成

(9) 为了能正常启动 NetBeans 7.0, 打开 NetBeans 安装目录下 etc 文件夹中的 netbeans.conf 文件, 将-J-XX:MaxPermSize=384m, 修改为-J-XX:MaxPermSize=256m。

1.2.3 安装 MySQL 及其开发环境

1. MySQL 简介

MySQL 是瑞典 MySQL AB 公司开发的一个小型关系型数据库管理系统, 在 2008 年被 Sun 公司收购。目前 MySQL 被广泛地应用在 Internet 上的中小型网站中。由于其体积小、速度快、总体拥有成本低, 尤其是开放源码这一特点, 许多中小型网站为了降低网站总体成本而选择了 MySQL 作为网站后台数据库系统。

2. 下载与安装 MySQL

本书使用的 MySQL 版本为 MySQL Community Server 5.5, 下载地址是 <http://www.mysql.com/downloads/mysql>。下载后得到安装文件 mysql-5.5.12-win32.msi, 具体安装步骤如下:

(1) 双击运行 mysql-5.5.12-win32.msi, 开始安装 MySQL, 打开如图 1-11 所示【MySQL Server 安装向导】对话框, 单击【Next】按钮。

(2) 在如图 1-12 所示的【安装许可协议】对话框中, 选中【I accept the terms in the Licence Agreement】复选框, 单击【Next】按钮。

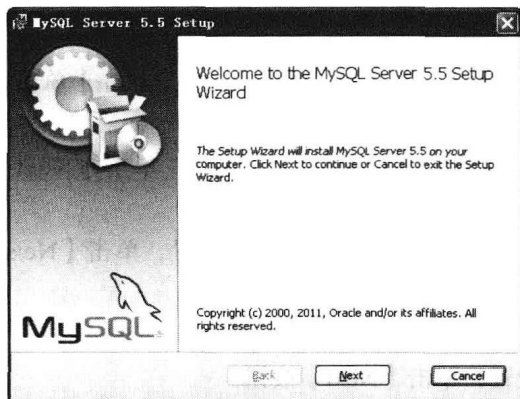


图 1-11 MySQL 安装向导

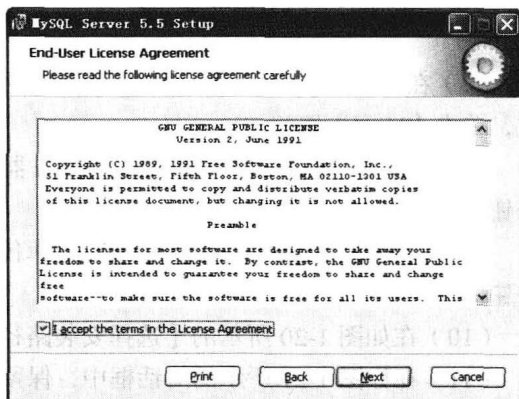


图 1-12 安装许可协议

(3) 在如图 1-13 所示的【选择安装类型】对话框中, 单击【Typical】按钮。

(4) 在如图 1-14 所示的【开始安装】对话框中, 单击【Install】按钮。

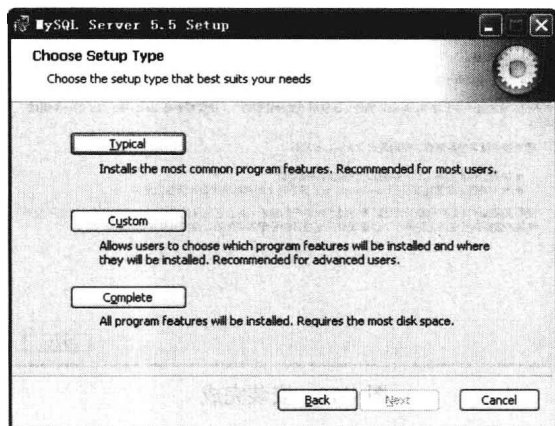


图 1-13 选择安装类型

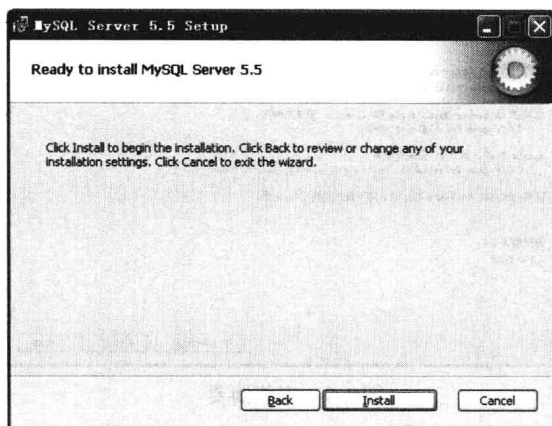


图 1-14 开始安装

(5) 连续单击【Next】按钮, 直到出现如图 1-15 所示的对话框; 选中【Launch the MySQL Instance Configuration Wizard】复选框, 单击【Finish】按钮。

(6) 在如图 1-16 所示的【安装配置向导】对话框中, 单击【Next】按钮。

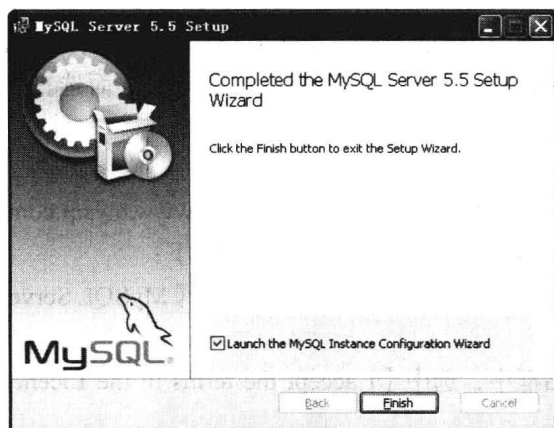


图 1-15 MySQL 安装完成

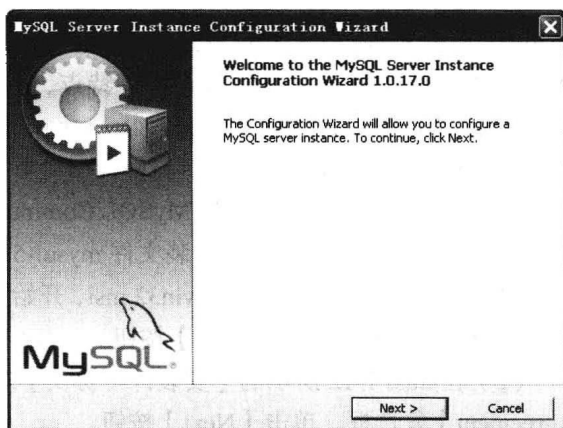


图 1-16 安装配置向导

(7) 在如图 1-17 所示的【选择配置类型】对话框中, 选择【Detailed Configuration】单选按钮, 单击【Next】按钮。

(8) 在如图 1-18 所示的【选择服务器类型】对话框中, 保留默认设置, 单击【Next】按钮。

(9) 在如图 1-19 所示的【选择数据库使用类型】对话框中, 保留默认设置, 单击【Next】按钮。

(10) 在如图 1-20 所示的【选择安装路径】对话框中, 保留默认设置, 单击【Next】按钮。

(11) 在如图 1-21 所示的对话框中, 保留默认设置, 单击【Next】按钮。

(12) 在如图 1-22 所示的【设置端口号】对话框中, 保留默认设置, 单击【Next】按钮。