



普通高等教育“十一五”国家级规划教材

林培光 耿长欣 张 燕 编著

# JavaEE简明教程

计算机科学与技术专业实践系列教材

清华大学出版社



普通高等教育“十一五”国家级规划教材

计算机科学与技术专业实践系列教材

# JavaEE简明教程

林培光 耿长欣 张 燕 编著

清华大学出版社  
北京

## 内 容 简 介

JavaEE 是一种企业应用的软件架构,是使用 Java 技术开发企业级应用的一种事实上的工业标准,是 Java 技术不断适应和促进企业级应用程序开发的产物。

本书是介绍 JavaEE 应用程序开发的入门级教材,主要介绍基于主流开发工具 Eclipse 进行 JavaEE 开发的关键技术和方法,共分 14 讲,分别从开发环境配置、基础 JSP/Servlet、JDBC 和 MySQL、JSTL/EL 到主流 MVC 开发框架 Struts 2,详细介绍了目前 JavaEE 开发中的关键技术。为了更加易于理解这些技术的实际用法,本书还由浅入深地给出了相关示例。

本书不仅可以作为高等院校计算机及相关专业的本科教材,而且也可以作为学习 JavaEE 开发技术的自学教材。

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

JavaEE 简明教程 / 林培光,耿长欣,张燕编著. —北京:清华大学出版社,2012.1

(计算机科学与技术专业实践系列教材)

ISBN 978-7-302-26558-0

I. ①J… II. ①林… ②耿… ③张… III. ①JAVA 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 174683 号

责任编辑:白立军 赵晓宁

责任校对:李建庄

责任印制:杨 艳

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954,jsjic@tup.tsinghua.cn

质 量 反 馈:010-62772015,zhiliang@tup.tsinghua.edu.cn

印 装 者:北京市清华园胶印厂

经 销:全国新华书店

开 本:185×260

印 张:14.75 字 数:367 千字

版 次:2012 年 1 月第 1 版

印 次:2012 年 1 月第 1 次印刷

印 数:1~3000

定 价:25.00 元

---

产品编号:041882-01

# 前 言

Java 语言作为一种优秀的面向对象的程序设计语言,具有平台无关性、安全机制、高可靠性和内嵌的网络支持等特点。目前,Java 语言已成为当前网络应用程序编写的首选工具之一。JavaEE 作为基于 Java 的企业级应用程序开发的事实标准,已经得到了广泛的应用。

JavaEE 是一种利用 Java 平台简化企业解决方案的开发、部署和管理相关的复杂问题体系结构。JavaEE 不仅巩固了标准版中的许多优点,如“编写一次、随处运行”的特性、方便存取数据库的 JDBC API、CORBA 技术以及能够在 Internet 应用中保护数据的安全模式等,同时还提供了对 EJB(Enterprise Java Beans)、Servlets、JSP(Java Server Pages)以及 XML 技术的全面支持,其最终目的就是成为一个能够使企业开发者大幅缩短投放市场时间的体系结构。JavaEE 为搭建具有可伸缩性、灵活性、易维护性的商务系统提供了良好的机制。

本书介绍了当今 JavaEE 平台开发技术,在注重系统性和科学性的同时,力求突出其实用性,在介绍相关的编程原理和基础知识的同时,着重利用丰富的实例演示 JavaEE 编程的细节。

本书作者均具有丰富的 JavaEE 项目开发和实训经验,因此在技术介绍过程中,能够结合具体应用实例,深入浅出地给出相关技术的核心内容,使读者真正掌握使用 JavaEE 进行系统分析、设计和实现的基本思路和方法。

本书由林培光、张燕、耿长欣主编,其中第 1~第 10 讲由林培光编写,第 11 讲由张燕编写,第 12~第 14 讲由耿长欣编写。虽然笔者都十分认真地编写、校对了本书全部内容,但书中不足之处在所难免,真诚希望广大读者批评指正。

在编写过程中,参考了网络上的许多资料,也得到了山东财政学院及计算机信息工程学院相关领导的大力支持以及研究生赵琳、胡耀斌等人的帮助,在此一并表示感谢。

编 者

2011 年 10 月

# 目 录

<b>第 1 讲 从零开始——开发环境配置</b> .....	1
1.1 JavaEE 简介 .....	1
1.1.1 JavaEE 规范 .....	2
1.1.2 JavaEE 平台的主流服务器简介 .....	2
1.1.3 JavaEE 的常用开发工具 .....	4
1.2 JDK 的安装 .....	8
1.2.1 JDK 简介 .....	8
1.2.2 安装.....	8
1.2.3 配置.....	8
1.2.4 测试 .....	10
1.3 Tomcat 的安装和配置 .....	10
1.3.1 Tomcat 的安装 .....	10
1.3.2 目录结构 .....	10
1.3.3 Tomcat 的启动和停止 .....	11
1.3.4 server.xml .....	11
1.4 Eclipse 的安装和使用 .....	12
1.4.1 Eclipse 安装 .....	12
1.4.2 使用 Eclipse 创建 Web 应用程序的过程 .....	13
1.4.3 Web 应用程序的一般目录结构及其初步配置 .....	16
1.5 开发一个简单的 Web 应用程序 .....	18
1.6 本讲小结.....	21
练习 1 .....	21
<b>第 2 讲 从 HTML 谈起</b> .....	22
2.1 HTML 简介 .....	22
2.2 学习 HTML 语言的必要性 .....	22
2.3 HTML 语言 .....	22
2.3.1 HTML 代码基础知识.....	22
2.3.2 常用的标签 .....	23
2.3.3 一段 HTML 示例代码 .....	24
2.4 HTML 的常用转义字符 .....	25
2.5 JavaScript .....	25
2.6 本讲小结.....	26
练习 2 .....	26

<b>第 3 讲 认识 JSP</b> .....	27
3.1 JSP 简介 .....	27
3.1.1 What is JSP .....	27
3.1.2 执行过程 .....	27
3.1.3 与 ASP 的比较 .....	28
3.1.4 JSP 2.0 的新功能 .....	28
3.2 JSP 的构成 .....	29
3.2.1 JSP 指令 .....	30
3.2.2 脚本元素 .....	31
3.2.3 动作元素 .....	31
3.3 本讲小结 .....	32
练习 3 .....	32
<b>第 4 讲 JSP 内置对象</b> .....	33
4.1 内置对象介绍 .....	33
4.2 out——输出对象 .....	33
4.3 request——请求对象 .....	34
4.3.1 request 对象简介 .....	34
4.3.2 实例演示 .....	34
4.4 response——响应对象 .....	35
4.4.1 response 对象简介 .....	35
4.4.2 实例演示 .....	35
4.5 session——会话对象 .....	36
4.5.1 对象简介 .....	36
4.5.2 实例演示 .....	37
4.6 application——应用程序对象 .....	37
4.6.1 application 对象简介 .....	37
4.6.2 实例演示 .....	38
4.7 exception——异常对象 .....	38
4.7.1 exception 对象简介 .....	38
4.7.2 实例演示 .....	39
4.8 Config——配置对象 .....	40
4.9 Page 和 PageContext: 页面和页面上下文对象 .....	40
4.10 使用 Cookie .....	41
4.10.1 Cookie 介绍 .....	41
4.10.2 实例演示 .....	41
4.11 本讲小结 .....	42
练习 4 .....	43



<b>第 5 讲 Servlet 入门</b> .....	44
5.1 Servlet 介绍 .....	44
5.2 Servlet 的技术特点 .....	44
5.3 开发一个简单的 Servlet .....	45
5.4 执行生成的 Servlet .....	45
5.5 Servlet 的配置 .....	46
5.6 Servlet 的生命周期 .....	47
5.7 本讲小结 .....	47
练习 5 .....	48
<b>第 6 讲 Servlet 高级应用</b> .....	49
6.1 Servlet 的初始化参数 .....	49
6.1.1 web.xml 中的参数定义 .....	49
6.1.2 在 Servlet 中读取参数 .....	49
6.1.3 实际应用 .....	50
6.2 Servlet 过滤器 .....	50
6.2.1 过滤器基本原理 .....	50
6.2.2 创建一个简单的过滤器 .....	50
6.2.3 实用案例 .....	51
6.3 Servlet 监听器 .....	53
6.3.1 对 request 对象进行监听 .....	53
6.3.2 对 session 对象进行监听 .....	54
6.3.3 对 ServletContext 对象(application)进行监听 .....	54
6.3.4 监听器实例——网站用户计数器 .....	56
6.4 本讲小结 .....	57
练习 6 .....	58
<b>第 7 讲 开发使用 JavaBean</b> .....	59
7.1 JavaBean 简介 .....	59
7.2 创建 JavaBean .....	59
7.3 在 JSP 中使用 JavaBean .....	60
7.3.1 定义 JavaBean 实例 .....	60
7.3.2 表单参数的封装 .....	61
7.3.3 读取 JavaBean 中的属性值 .....	61
7.4 JavaBean 程序示例 .....	61
7.5 JavaBean 高级应用 .....	63
7.5.1 JavaBean 的属性介绍 .....	63
7.5.2 企业 JavaBean 简介 .....	63
7.6 本讲小结 .....	64
练习 7 .....	64

<b>第 8 讲 MySQL 数据库的安装和使用</b> .....	65
8.1 简介 .....	65
8.2 安装 .....	66
8.2.1 MySQL 的安装 .....	66
8.2.2 安装后的目录结构 .....	66
8.3 操作 MySQL .....	66
8.4 MySQL 常用命令 .....	66
8.4.1 连接和退出 MySQL .....	66
8.4.2 修改密码 .....	67
8.4.3 操作数据库 .....	67
8.4.4 导出和导入数据 .....	68
8.5 MySQL-Front: MySQL 客户端工具 .....	68
8.5.1 MySQL-Front 简介 .....	68
8.5.2 创建到 MySQL 数据库的连接 .....	68
8.5.3 创建数据库 .....	69
8.5.4 创建数据表 .....	70
8.5.5 对数据表的 CRUD 操作 .....	71
8.5.6 创建存储过程 .....	72
8.6 其他客户端工具 .....	73
8.6.1 Navicat for MySQL .....	74
8.6.2 HeidiSQL .....	74
8.6.3 其他工具简介 .....	75
8.7 本讲小结 .....	76
练习 8 .....	76
<b>第 9 讲 数据库开发——JDBC</b> .....	78
9.1 JDBC 简介 .....	78
9.2 JDBC 的驱动程序 .....	78
9.3 JDBC 的主要对象及其工作方式 .....	80
9.4 加载数据库驱动程序和创建连接 .....	80
9.5 执行 SQL 语句 .....	81
9.6 处理结果集 .....	84
9.7 JDBC 其他应用 .....	87
9.7.1 数据库元数据 .....	87
9.7.2 结果集元数据 .....	87
9.7.3 事务 .....	88
9.8 关闭连接 .....	90
9.9 使用数据库连接池 .....	90
9.9.1 连接池的工作原理 .....	90



9.9.2	Tomcat 下连接池的配置 .....	91
9.9.3	通过连接池获得连接 .....	91
9.10	本讲小结 .....	92
	练习 9 .....	92
<b>第 10 讲</b>	<b>案例：用户管理系统(基于 JSP、JDBC 和 JavaBean)</b> .....	<b>93</b>
10.1	系统需求分析 .....	93
10.1.1	系统功能需求分析 .....	93
10.1.2	系统性能要求 .....	93
10.2	系统概要设计 .....	94
10.3	详细设计和实现 .....	96
10.3.1	数据库设计 .....	96
10.3.2	类及其功能设计 .....	97
10.3.3	类实现 .....	99
10.3.4	页面设计和实现 .....	108
10.4	本讲小结 .....	114
	练习 10 .....	114
<b>第 11 讲</b>	<b>Java 标准标签库</b> .....	<b>115</b>
11.1	表达式语言 .....	115
11.1.1	EL 简介 .....	115
11.1.2	EL 中的有效表达式 .....	116
11.1.3	表达式中的默认变量 .....	117
11.1.4	表达式中的操作符 .....	119
11.2	JSTL 标签库 .....	120
11.2.1	如何在 JSP 中使用 JSTL .....	120
11.2.2	Core 标签库 .....	121
11.2.3	IL8N Formatting 标签库 .....	127
11.2.4	JSTL XML processing 标签库 .....	131
11.2.5	数据库操作标签库 .....	132
11.3	实例：用 JSTL 实现 UserManager .....	133
11.3.1	JSTL 实现 UserManager 的开发环境 .....	133
11.3.2	各页面功能及流程 .....	133
11.3.3	各文件的源代码及说明 .....	134
11.4	本讲小结 .....	150
	练习 11 .....	150
<b>第 12 讲</b>	<b>J2EE 开发框架——Struts 2 入门</b> .....	<b>151</b>
12.1	Struts 概述 .....	151

12.1.1	Struts 1 .....	151
12.1.2	Struts 2 .....	152
12.1.3	Struts 1 和 Struts 2 的比较 .....	152
12.2	Struts 框架 .....	153
12.2.1	MVC 设计模式 .....	153
12.2.2	Struts 1 框架 .....	154
12.2.3	Struts 2 框架 .....	155
12.3	Struts 2 配置文件 .....	156
12.3.1	web.xml .....	156
12.3.2	struts.xml .....	157
12.3.3	struts-default.xml .....	158
12.3.4	struts.properties .....	158
12.3.5	struts-plugin.xml .....	159
12.4	Struts 2 开发环境的搭建 .....	160
12.5	开发一个简单的 Struts 2 应用程序——Hello World .....	160
12.6	小结 .....	163
练习 12	.....	163
<b>第 13 讲</b>	<b>J2EE 开发框架——Struts 2 标签库</b> .....	<b>164</b>
13.1	Struts 2 标签概述 .....	164
13.1.1	Struts 2 标签库分类 .....	164
13.1.2	Struts 2 标签的使用配置 .....	165
13.2	Struts 2 的 OGNL .....	165
13.2.1	值栈 .....	165
13.2.2	命名对象 .....	166
13.2.3	#, %、\$ 符号 .....	166
13.3	表单标签 .....	167
13.3.1	表单标签的通用属性 .....	167
13.3.2	常用的表单标签 .....	168
13.4	非表单标签 .....	174
13.5	数据标签 .....	176
13.6	控制标签 .....	182
13.7	Ajax 标签 .....	186
13.8	小结 .....	190
练习 13	.....	190
<b>第 14 讲</b>	<b>Struts 2 实例——用户管理系统</b> .....	<b>191</b>
14.1	系统概述 .....	191
14.1.1	系统分析 .....	191

14.1.2	系统设计	192
14.2	配置文件	192
14.3	业务处理逻辑	193
14.3.1	数据库连接	193
14.3.2	建立业务对象	194
14.3.3	业务逻辑	195
14.4	建立业务功能模块	199
14.4.1	主页面	199
14.4.2	用户注册	199
14.4.3	用户登录	203
14.4.4	查看个人信息	205
14.4.5	修改个人信息	206
14.4.6	权限控制	208
14.4.7	查看所有用户	209
14.4.8	用户删除	212
14.4.9	查找某一用户	212
14.4.10	退出系统	212
14.5	本讲小结	213
	练习 14	213
<b>附录 A</b>	<b>图索引</b>	<b>214</b>
<b>附录 B</b>	<b>表索引</b>	<b>217</b>
<b>附录 C</b>	<b>示例代码索引</b>	<b>219</b>

# 第 1 讲 从零开始——开发环境配置

“工欲善其事，必先利其器”。作为 J2EE 开发的重要准备，本讲重点介绍 J2EE 开发环境的搭建过程。本讲涉及的重要知识点如下：

- 什么是 JDK；
- 如何安装和配置 JDK；
- 测试 JDK 的安装情况；
- J2EE 常用服务器介绍；
- Tomcat 服务器介绍；
- Tomcat 的安装和配置方法；
- J2EE 开发工具——Eclipse 的安装和使用；
- J2EE 应用程序基本结构。

## 1.1 JavaEE 简介

Java 2 平台有 3 个版本，它们是适用于小型设备和智能卡的 Java 2 平台 Micro 版 (Java 2 Platform Micro Edition, J2ME)、适用于桌面系统的 Java 2 平台标准版 (Java 2 Platform Standard Edition, J2SE)、适用于创建服务器应用程序和服务的 Java 2 平台企业版 (Java 2 Platform Enterprise Edition, J2EE)。2005 年 6 月，Sun 公司公开 Java SE 6，此时，Java 的各种版本更名取消其中的数字“2”，即：J2EE 更名为 JavaEE，J2SE 更名为 Java SE，J2ME 更名为 Java ME。

Java 微版 (Java Platform Micro Edition, Java ME)，是一个技术和规范的集合，为移动设备 (包括消费类产品、嵌入式设备、高级移动设备等) 提供了基于 Java 环境的开发与应用平台。Java ME 目前分为两类配置，一类是面向小型移动设备的 CLDC (Connected Limited Device Profile)；另一类是面向功能更强大的移动设备 (如智能手机和机顶盒)，称为 CDC (Connected Device Profile CDC)。Java ME 有自己的类库，其中 CLDC 使用的是专用的 Java 虚拟机叫做 KVM。

Java 平台标准版 (Java Platform Standard Edition, Java SE) 用于开发和部署桌面、服务器以及嵌入设备和实时环境中的 Java 应用程序。Java SE 包括用于开发 Java Web 服务的类库，同时 Java SE 为 JavaEE 提供了基础。

Java 平台企业版 (Java Platform Enterprise Edition, JavaEE)，用于开发便于组装、健壮、可扩展、安全的服务器端 Java 应用。JavaEE 基于 Java SE，此外新加了企业应用所需的类库，具有 Web 服务、组件模型，以及通信 API 等特性，这些为面向服务的架构 (Service-oriented architecture, SOA) 以及开发 Web 2.0 应用提供了支持。JavaEE 技术的基础就是核心 Java 平台或 Java 2 平台的标准版。JavaEE 体系结构提供中间层集成框架，用来满足无须太多费用而又需要高可用性、高可靠性以及可扩展性的应用的需求。通过提供统一的开发

平台,JavaEE 降低了开发多层应用的费用和复杂性,同时提供对现有应用程序集成强有力支持,完全支持 Enterprise JavaBeans,有良好的向导支持打包和部署应用,添加目录支持,增强了安全机制,提高了性能。

### 1.1.1 JavaEE 规范

JavaEE 规范有的是标准的,有的不是标准的。标准的含义就是被 Sun 公司 JDK 体现与兼容,成为业界遵循的准则。

比如 Servlet,当它成为标准规范以后,就会在 JDK 里得到支持和体现,中间件开发商就会去实现它,用户就要按照这种标准开发应用。Struts 是开源组织 Apache 开发的规范,不是标准的,但以后可能会成为标准。所以,合格的 JavaEE 程序员,不仅应掌握 Java 的编写规范,而且应掌握 JavaEE 开发平台的规范(包括标准和非标准的)。

各种中间件是一种标准。中间件的意义在于,它已经实现了每个应用都需要的底层基础部分。比如,WebLogic 和 JBoss 已经提供了应用容器和 Servlet、JDBC、JNDI、JMS 和 WebService 等接口。开发者在这个基础上,可以很容易而且可以很好去实现这些功能,从而使开发者集中精力,去解决应用的业务逻辑。通常,业务逻辑是千变万化的,但是底层实现却有迹可循,中间件开发商承担了底层实现的责任。目前,绝大部分应用都是基于现成的中间开发的,作为 JavaEE 程序员,掌握主流的中间件是基本的职业要求之一。

各种框架也是一种标准。中间件使开发者从底层实现中解放出来,基于框架的开发使开发者从程序基本架构中解放出来,从而更彻底地将精力转到业务逻辑领域。所以,出色的开发者是那些对业务能透彻领悟的人,程序构架则由建模行家担当。但是,基于框架开发,不是忽略框架,因为大部分业务逻辑的实现,都是基于框架的。而且,当现有的框架不能满足需求时候,还需要扩展框架甚至被迫另外开发框架。JavaEE 程序员,熟悉主流的框架也是职业要求,还应知道各框架的优劣和特性。

### 1.1.2 JavaEE 平台的主流服务器简介

目前流行的 Web 服务器有 tomcat(推荐给初学者)、apache、weblogic(商业用)、websphere(商业用)、resin 和 JBoss。

#### 1. Tomcat

Tomcat 是一个开放源代码、运行 servlet 和 JSP Web 应用软件的基于 Java 的 Web 应用软件容器。Tomcat Server 是根据 servlet 和 JSP 规范进行执行的。

Tomcat 是 Java Servlet 2.2 和 JavaServer Pages 1.1 技术的标准实现,是基于 Apache 许可证下开发的自由软件。Tomcat 是完全重写的 Servlet API 2.2 和 JSP 1.1 兼容的 Servlet/JSP 容器。随着 Catalina Servlet 引擎的出现,Tomcat 第 4 版的性能得到提升,成为一个值得考虑的 Servlet/JSP 容器。因此,目前许多 Web 服务器都是采用 Tomcat。

#### 2. Apache

Apache 仍然是世界上用得最多的 Web 服务器,市场占有率达 60%左右。它源于 NCSA httpd 服务器,当 NCSA WWW 服务器项目停止后,那些使用 NCSA WWW 服务器的人们开始交换用于此服务器的补丁,这也是 Apache 名称的由来(pache 是补丁的意思)。世界上很多著名的网站都是 Apache 的产物,Apache 的成功之处主要在于它的源代码开放、有一支开放的开发队伍、支持跨平台的应用(可以运行在几乎所有的 UNIX、Windows、

Linux 系统平台上)以及它的可移植性等方面。

Apache 只支持静态网页,要集成 tomcat 才能支持 JSP。

### 3. WebLogic

WebLogic 是美国 BEA 公司出品的一个 application server,是一个基于 JavaEE 架构的中间件,BEA WebLogic 是用于开发、集成、部署和管理大型分布式 Web 应用、网络应用和数据库应用的 Java 应用服务器。将 Java 的动态功能和 Java Enterprise 标准的安全性引入大型网络应用的开发、集成、部署和管理之中。

BEA WebLogic Server 是一种多功能、基于标准的 web 应用服务器,为企业构建自己的应用提供了坚实的基础。各种应用开发、部署所有关键性的任务,无论是集成各种系统和数据库,还是提交服务、跨 Internet 协作,起始点都是 BEA WebLogic Server。由于它具有全面的功能、对开放标准的遵从性、多层架构、支持基于组件的开发,基于 Internet 的企业都选择它开发、部署最佳的应用。

BEA WebLogic Server 在使应用服务器成为企业应用架构的基础方面处于领先地位。BEA WebLogic Server 为构建集成化的企业级应用提供了稳固的基础,以 Internet 的容量和速度,在连网的企业之间共享信息、提交服务,实现协作自动化。

### 4. WebSphere

WebSphere Application Server 是一种功能完善、开放的 Web 应用程序服务器,是 IBM 公司电子商务计划的核心部分,是基于 Java 的应用环境,用于建立、部署和管理 Internet 和 Intranet Web 应用程序。这一整套产品进行了扩展,以适应 Web 应用程序服务器的需要,范围从简单、高级直到企业级。

WebSphere 针对以 Web 为中心的开发人员,这些开发人员都是在基本 HTTP 服务器和 CGI 编程技术上成长起来的。IBM 公司提供的 WebSphere 产品系列,通过提供综合资源、可重复使用的组件、功能强大并易于使用的工具,以及支持 HTTP 和 IIOP 通信的可伸缩运行时环境,帮助用户从简单的 Web 应用程序转移到电子商务世界。

### 5. Resin

Resin 提供了最快的 JSP/Servlets 运行平台。在 Java 和 JavaScript 的支持下,Resin 可以为任务灵活选用合适的开发语言。Resin 的一种先进的语言 XSL(XML stylesheet language)可以使得形式和内容相分离。如果选用 JSP 平台作为 Internet 商业站点的支持,那么速度、价格和稳定性都是要考虑到的。Resin 十分出色,表现更成熟,具备商业软件的要求,而且是全免费的。

J2EE 是基于 Java 服务器端大系统的基础。目前 Resin 可以支持 Sun 公司的 J2EE,而 Tomcat 不能直接支持。但 Tomcat 结构非常合理,而且是 Apache 组织的产品,因此有着很好的远景。

Resin1.1 的特性如下:

- (1) 支持 jsp1.1 和在服务器端编译的 javascript。
- (2) 比 mod\_perl、mod\_php 更快,比 Jakarta Tomcat 快 3 倍。可以在 Caucho 的网站看到很全面的各种测试报告。
- (3) 使用 servlet2.2 引擎。
- (4) 自动的 servlet/bean 编译。

- (5) Servlet、XSL Filtering。
- (6) 支持 IIS、Apache、Netscape 和其他内置了 HTTP 1.1 的 Web 服务器。
- (7) XSLT 和 XPath1.0 引擎。
- (8) 企业级的共享软件(基于一个开放源码的协议)。

Resin 安装特别简单,只需将 resin-1.1.3.zip 解压缩到某个目录即可。在运行之前需要有 JDK 的支持。启动方式是执行: resin-1.1.3/bin/httpd(在 Windows NT)系统已经按照默认设置启动,系统将监听 8080 端口。JSP 文件可放在 resin-1.1.3/doc 目录下;Servlet 可以放在 resin-1.1.3/doc/web-inf/classes 下面。测试从 http://localhost:8080/中可以看到 Resin 的欢迎界面。

## 6. JBoss

JBoss 是一个运行 EJB 的 J2EE 应用服务器。它是开放源代码的项目,遵循最新的 J2EE 规范。从 JBoss 项目开始至今,它已经从一个 EJB 容器发展成为一个基于 J2EE 的一个 Web 操作系统,体现了 J2EE 规范中最新的技术,并且它还在 2002 年的 JavaWorld Editors' Choice Awards(JavaWorld 编辑大奖)评选中获得“最佳 Java 应用服务器”大奖。无论是学习还是应用,JBoss 提供了一个非常优秀的平台。JBoss 作为一个基于 J2EE 的开放源代码的应用服务器,是全世界开发者共同努力的成果。因为 JBoss 代码遵循 LGPL 许可,可以在任何商业应用中免费使用,而不用支付费用。

### 1.1.3 JavaEE 的常用开发工具

现在常用的 Java 项目开发环境有 JBuilder、VisualAgeforJava、Eclipse、NetBeansIDE、JCreator+J2SDK、JDK+记事本、EditPlus+J2SDK 等。

#### 1. NetBeans 与 SunJavaStudio5

NetBeans 是开放源码的 Java 集成开发环境(Integrated Development Environment, IDE),适用于各种客户机和 Web 应用。

SunJavaStudio 是 Sun 公司最新发布的商用全功能 JavaIDE,支持 Solaris、Linux 和 Windows 平台,适于创建和部署 2 层 JavaWeb 应用和 n 层 J2EE 应用的企业开发人员使用。

NetBeans 是业界第一款支持创新型 Java 开发的开放源码 IDE。开发人员可以利用业界强大的开发工具来构建桌面、Web 或移动应用。同时,通过 NetBeans 和开放的 API 的模块化结构,第三方能够非常轻松地扩展或集成 NetBeans 平台。

#### 2. Oracle 的 JDeveloper

Oracle9i JDeveloper(最新为 10g)为构建具有 J2EE 功能、XML 和 Webservices 的复杂的、多层的 Java 应用程序提供了一个完全集成的开发环境。它为运用 Oracle 9i 数据库和应用服务器的开发人员提供特殊的功能和增强性能,另外,它也有资格成为用于多种用途 Java 开发的一个强大的工具。

Oracle 9i JDeveloper 的主要特点如下:

(1) 具有一体化建模语言(Unified Modeling Language,UML)建模功能。可以将业务对象及 e-business 应用模型化。

(2) 配备有高速 Java 调试器(Debugger)、内置 Profiling 工具、提高代码质量的工具 CodeCoach 等。



(3) 支持简单对象访问协议 (SimpleObjectAccessProtocol, SOAP)、统一描述、发现和集成协议 (UniversalDescription, DiscoveryandIntegration, UDDI)、Web 服务描述语言 (Web Services Description Language, WSDL) 等 Web 服务标准。

JDeveloper 不仅仅是很好的 Java 编程工具,而且是 OracleWeb 服务的延伸,支持 ApacheSOAP、9iAS,可扩充的环境,与 XML、WSDL 语言紧密相关。

Oracle 9i JDeveloper 完全利用 Java 编写,能够与以前的 Oracle 服务器软件以及其他厂商支持 J2EE 的应用服务器产品相兼容,而且在设计时着重针对 Oracle 9i,能够无缝化跨平台之间的应用开发,提供了业界第一个完整的、集成了 J2EE 和 XML 的开发环境,允许开发者快速开发可以通过 Web、无线设备及语音界面访问的 Web 服务和交易应用。以往只能通过将传统 Java 编程技巧与最新模块化方式结合到一个单一集成的开发环境中之后才能完成 J2EE 应用开发生命周期管理的事实,从根本上得到改变。它的缺点是对于初学者来说,较复杂,也比较难。

### 3. IBM 公司的 Visual Age for Java

Visual Age for Java 是一个非常成熟的开发工具,它的特性对于 IT 开发者和业余的 Java 编程人员来说都是非常有用的。它提供对可视化编程的广泛支持,支持利用 CICS (Customer Information Control System, 客户信息控制系统) 连接遗传大型机应用,支持 EJB 的开发应用,支持与 Websphere 的集成开发,方便的 Bean 创建和良好的快速应用开发 (Rapid Application Development, RAD) 支持和无文件式的文件处理。

IBM 公司为建设 Web 站点所推出的 Web Sphere Studio Advanced Edition 及其包含的 Visual Age for Java Professional Edition 软件已全面转向以 Java 为中心。这样,Java 开发人员对 Web Sphere 全套工具会更加习惯。Studio 是个不错的 HTML 站点页面编辑环境,它所提供的工具有 Web 站点管理、快速开发 JDBC 页向导程序、HTML 编辑器和 HTML 语法检查等。Studio 和 VisualAge 集成度很高,菜单中提供了在两种软件包之间快速移动代码的选项。这就让使用 Studio 的 Web 页面设计人员和使用 VisualAge 的 Java 程序员可以相互交换文件、协同工作。Visual Age for Java 支持团队开发,内置的代码库可以自动地根据用户做出改动而修改程序代码,这样就可以很方便地将目前代码和早期版本做出比较。与 Visual Age 紧密结合的 Websphere Studio 本身并不提供源代码和版本管理的支持,它只是包含了一个内置文件锁定系统,当编辑项目的时候可以防止其他人对这些文件的错误修改,软件还支持诸如 Microsoft Visual Source Safe 这样的第三方源代码控制系统。Visual Age for Java 完全面向对象的程序设计思想使得开发程序非常快速、高效。可以不编写任何代码就可以设计出一个典型的应用程序框架。Visual Age for Java 作为 IBM 电子商务解决方案其中产品之一,可以无缝地与其他 IBM 产品(如 WebSphere、DB2)融合,迅速完成从设计、开发到部署应用的整个过程。

但是,Visual Age for Java 独特的管理文件方式使其集成外部工具非常困难,无法让 Visual Age for Java 与其他工具一起联合开发应用。

### 4. BEA 的 WebLogic Workshop

BEA WebLogic Workshop 是一个统一、简化、可扩展的开发环境,使所有的开发人员都能在 BEA WebLogic Enterprise Platform 之上构建基于标准的企业级应用,从而提高了开发部门的生产力水平,加快了价值的实现。WebLogic Workshop 除了提供便捷的 Web

服务之外,还能够用于创建更多种类的应用。作为整个 BEA WebLogic Platform 的开发环境,不管是创建门户应用、编写 workflow,还是创建 Web 应用,Workshop 8.1 都可以帮助开发人员更快更好地完成。

WebLogic Workshop 的主要特点如下:

(1) 使 J2EE 开发切实可行,提高开发效率。

BEA WebLogic Workshop 使开发人员远离 J2EE 内在的复杂性,集中精力专注业务逻辑,无须操心单调乏味的基础结构代码。这种创新意味着,已被企业验证的 J2EE 的强大功能,最终被大多数不熟悉 Java 和 J2EE 的应用开发人员所掌握,从而使 IT 部门的工作效率提高一个数量级。

Workshop 的可视化设计器以及直观的概念,如事件、属性和控件等,实现了基于事件的开发。Workshop 简化的程序设计模型,使开发人员不必掌握复杂的 J2EE API 和面向对象的程序设计原理。所有开发人员,包括 J2EE 专家和具有可视化和过程化语言技能的应用开发人员在内,都可以共同工作在 BEA WebLogic Enterprise Platform 之上。Workshop 的可视化开发环境,可创建带有代码注释的标准 Java 文件,用来说明由运行时框架实施的企业级需求。J2EE 和其他高级开发人员,借助功能强大的代码编辑功能,可以访问 Java 源代码,从而弥补了可视化设计器的不足。

(2) 构建企业级应用。

通过在可伸缩、安全可靠的企业级架构上实施各种应用,BEA WebLogic Workshop 大大降低了开发风险。而且,所有应用的创建都使用标准的 J2EE 组件,既保护了用户的技术投资,又保持了最大的灵活性。

BEA WebLogic Workshop 运行框架,是统一整个架构的汇聚层,使单一、简化的程序设计模型扩展到所有的 BEA WebLogic Enterprise Platform 应用类型。通过解释设计时创建的注释代码,运行时框架可以实现必要的 J2EE 组件,并且提取与 J2EE 应用开发有关的所有底层细节。

(3) 降低 IT 复杂性。

BEA WebLogic Workshop 提供各种 Java 控件,使得与 IT 资源的连接更轻而易举。另外,在构建任何 BEA WebLogic Platform 的应用中,Java 控件不仅可扩展而且完全相同。这种强大、有效的方法能够降低 IT 技术的复杂性,优化信息的可用性,推动包含“最佳业务方案”的可重用服务的开发,使开发人员能以更低的成本、更短的时间实现更大的产出。

利用 BEA WebLogic Workshop,任何开发人员都能以最大的生产效率,构建各种 Web 服务、Web 应用、门户和集成项目。BEA WebLogic Workshop 是 BEA 的产品战略核心,帮助客户接触和利用面向服务架构的强大功能。

BEA Weblogic Workshop 8.1 极大简化了当前实际企业集成环境中企业级应用和服务的构建,并成为全面支持关键企业级应用(如异步、真正松耦合和粗粒度消息传送等)的自然选择。它的缺点是过于复杂,对于初学者来说,理解起来较为困难。

## 5. JCreator

JCreator 是一个 Java 程序开发工具,也是一个 Java 集成开发环境。无论是要开发 Java 应用程序或网页上的 Applet 元件都难不倒它。在功能上比 Sun 公司所公布的 JDK 等文字模式开发工具容易操作,还允许使用者自定义操作窗口界面及无限 Undo/Redo 等功能。