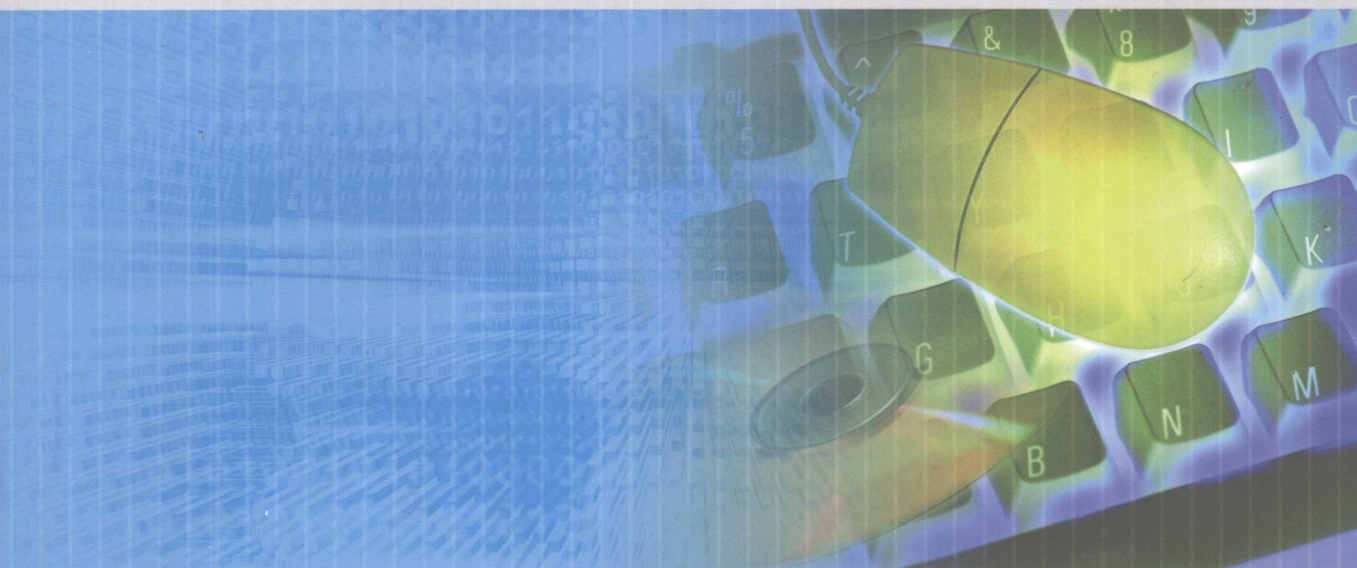




21 世纪全国高校应用人才培养信息技术类规划教材



Eclipse 与 J2EE 应用开发

李长云 饶居华 主 编
李 鹏 王志兵



北京大学出版社
PEKING UNIVERSITY PRESS

TP311.56/422

2008

内容简介

21 世纪全国高校应用人才培养信息技术类规划教材

Eclipse 与 J2EE 应用开发

李长云 饶居华
李 鹏 王志兵 主 编



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

本书详细地介绍了在当前最流行的开发环境 Eclipse 环境下如何进行 J2EE 应用开发,主要内容有: Eclipse 开发环境、WebLogic Server 开发环境、J2EE 体系结构及设计模式、EJB 开发、JSP 技术、struts2.0 开发,最后讨论了一个电子商务应用的大型实例,具有实际的应用价值。本书实践性很强,在对每种技术的介绍中都穿插了实际的应用例子。

本书适合作为高校计算机专业教材,也可供广大企业应用开发人员阅读,无论是 Eclipse 的初学者,还是有经验的 J2EE 开发人员,都能从书中获益。

图书在版编目(CIP)数据

Eclipse 与 J2EE 应用开发/李长云,饶居华,李鹏,王志兵,主编. —北京:北京大学出版社, 2008.1

(21 世纪全国高校信息类规划教材)

ISBN 978-7-301-13068-1

I. 实… II. ①李… ②饶… ③李… ④王… III. ①软件工具—程序设计—高等学校—教材
②JAVA 语言—程序设计—高等学校—教材 IV. TP311.56 TP312

中国版本图书馆 CIP 数据核字(2007)第 192183 号

书 名: Eclipse 与 J2EE 应用开发

著作责任者: 李长云 饶居华 李鹏 王志兵 主编

责任编辑: 温丹丹 桂 春

标准书号: ISBN 978-7-301-13068-1

出 版 者: 北京大学出版社

地 址: 北京市海淀区成府路 205 号 100871

电 话: 邮购部 62752015 发行部 62750672 编辑部 62765126 出版部 62754962

网 址: <http://www.pup.cn>

电子信箱: xxjs@pup.pku.edu.cn

印 刷 者: 世界知识印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×980 毫米 16 开本 24.75 印张 620 千字

2008 年 6 月第 1 版 2008 年 6 月第 1 次印刷

定 价: 44.00 元

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究

举报电话: 010-62752024; 电子信箱: fd@pup.pku.edu.cn

前言

Eclipse 是著名的跨平台的自由集成开发环境 (IDE)，最初用于 Java 语言开发。Eclipse 本身只是一个框架平台，但是对插件的支持使得它拥有其他功能相对固定的 IDE 软件很难具有的灵活性。许多软件开发商以 Eclipse 为框架开发自己的 IDE。目前最广泛应用的版本为 3.2。J2EE (Java 2 Enterprise Edition) 是利用 Java 2 平台来简化企业解决方案的体系结构。J2EE 技术的基础是 Java 2 平台，不但有 J2SE 平台的所有功能，同时还提供了对 EJB、Servlet、JSP、XML 等技术的全面支持，其最终目标是成为一个支持企业级应用开发的体系结构，简化企业解决方案的开发，部署和管理等复杂问题。事实上，J2EE 已经成为企业级开发的工业标准和首选平台。而目前在 Eclipse 环境下进行 J2EE 的应用开发的书籍甚少，据此，我们编写了该书。

本书从 Eclipse 的基础知识讲起，这样方便没有基础的读者学习。全书共 13 章，先简单介绍一下本书的组织结构和主要内容。

第 1 章从总体上介绍了 J2EE 的基本概念、应用程序模型、安装与配置等。

第 2 章是 Eclipse 概述及开发环境的搭建，详细介绍了 Eclipse 的安装与运行以及插件集成。

第 3 章介绍了 WebLogic Server 开发环境，详细介绍了 WebLogic 的安装与配置。

第 4 章主要通过简单的 J2EE 开发例子，带领大家熟悉 J2EE 的开发环境和开发过程。

第 5 章介绍了 J2EE 的结构设计与解决方案，具体介绍了 J2EE 的分层结构模型以及各层之间的关系。

第 6 章介绍了 J2EE 设计模式的基本概念和特点。

第 7 章介绍会话 Bean 的细节，如何开发、如何部署和如何在客户程序中使用 Bean。

第 8 章介绍了实体 Bean，然后用大量篇幅介绍了 CMP 与 BMP 的开发和部署，最后列举了开发实体 Bean 的一些实例。

第 9 章介绍了 Java 消息服务的基本理论和消息驱动 EJB 的基本理论，消息驱动 EJB 是实现异步消息传输的一种简单有效的方法。然后介绍了 JMS 及其他 EJB 的关系，并以实例说明了消息驱动 EJB 的开发过程，以及消息驱动 EJB 与会话 EJB 和实体 EJB 的集成应用开发。

第 10 章首先介绍了软件测试的基础知识，包括软件测试的目的、原则、测试对象及测试方法。然后着重介绍了 JUnit 测试工具，以及如何在 Eclipse 中结合 JUnit 测试工具进行单元测试。本章通过几个简单的实例，逐步指导读者如何利用 JUnit 测试工具来编写测试程序。

第 11 章介绍了 Servlet 技术和 JSP 编程，并以实际的例子来说明 JSP 编程的具体过程，还介绍了 JSP 中的 Java Bean 组件以及 JSP 的部署过程。

第 12 章全面的介绍了最新的框架 Struts2.0，包括经典的 MVC 模式的分析，Struts2.0 的历史由来，Struts2.0 的安装配置和基本使用，对 Struts2.0 的流程也作了深入的分析。最后详细讲解了 Struts2.0 的常用标签，并对每一种标签都给出比较详细的参数表和简单的示例代码，

能使读者迅速掌握这些常用标签的用法。

第 13 章通过一个综合案例讲述了电子商务系统的开发。

本书由李长云、饶居华、李鹏、王志兵主编，同时，在本书的编写过程中，湖南工业大学软件新技术实验室的何频捷、李伟和吴岳忠提出了大量的参考性意见。本书的编写也得到国家自然科学基金“开放环境下的软件动态演化研究”、湖南省教育厅优秀青年科研项目“基于高阶 π 演算的动态体系结构语言研究”、湖南省学位与研究生教改课题“应用导向型的研究生专业课程教学新模式”和湖南工业大学重点教改课题“计算机基础实验教学示范中心建设”的资助。在此，我们表示衷心的感谢。

由于编者水平有限，加之时间仓促，书中不足之处在所难免，敬请广大读者批评指正。

编者

2007 年 5 月

序 言

当代社会正在步入知识经济时代。知识经济的特点在于创造价值的源泉已不再主要是依赖于资源、资本和人的简单劳动，而是依赖于科技。科技的更新将引起新兴产业的产生和发展，而软件产业是典型的高科技产业。在新的形势下，软件技术有何新的发展动向？软件产业的发展面临怎样的机遇？我们能否抓住此机遇推动我国的软件产业发展？本书所介绍的 Eclipse 技术可较好地解决这些问题。

1999 年，美国 IT 委员会出台了一个非常重要而且很有影响的报告，报告认为软件是信息时代社会的重要基础设施。这个报告引发了我们对软件的新认识，将软件提高到和电力、交通同样的地位，然而实际上这个基础却相当脆弱。软件越来越普及而且越来越复杂，但软件的生产能力远远满足不了飞速发展的实际需求，开发可靠安全的各种软件的适用技术还较弱。在我国，这种矛盾表现尤为突出：一方面是社会对软件的巨大需求；另一方面我们缺乏高端软件研发能力，核心软件技术被国外巨头垄断。中国作为一个大国，必须要有自主的软件产业体系，否则信息安全没有保障，甚至有可能扩大数字鸿沟，我国庞大的制造业也会缺乏竞争力，“中国制造”不能提升到“中国创造”。

倪光南院士指出，为了中国软件产业的迅速发展，为了建立中国自主的软件产业体系，大力推进开源软件是一个关键举措。由于开源软件的开放性，它有利于创新，有利于打破垄断，中国自主操作系统等许多自主软件正是基于开源软件发展起来的。今后，随着建设创新型国家和和谐社会构建的整体推进，开源软件在中国将会有更大的发展和更多的推广应用。

本书所介绍的 Eclipse 开发平台就是目前非常流行的一种开源软件，最初用于 Java 语言开发。Eclipse 之技术先进，得到遍布全球的开发组织的普遍认可，许多重要公司的著名软件都正在使用 Eclipse 来进行开发。Eclipse 具有与众不同的视野——它运用的是“以平台为中心”而非“以工具为中心”的思维方式。在 Eclipse 领域，世界上各种各样的软件技术都可以相互借鉴和参考，也可以在这个平台上发展。

J2EE (Java 2 Platform Enterprise Edition, Java 2 平台企业版) 提供了跨越系统的单一标准，该标准使用统一的、基于组件的模型，包含了多层应用程序需要的资源，并孕育了用来满足企业战略需求的新一代组件、工具、系统和应用程序。但很多人在学习和使用 J2EE 的过程中，需要经历一条陡峭的学习曲线。为缓解这个问题，就需要借助稳定的开发平台和内容丰富的书籍来了解 J2EE 平台的关键环节。

本书系统介绍了 Eclipse 环境下进行 J2EE 应用开发的技术，详细阐述了 J2EE 应用程序的开发过程。我相信，读者使用本书一定会收获颇丰，是为序。



2008 年 3 月

目 录

第1章 J2EE 概述.....	1
1.1 J2EE 的发展历程.....	1
1.2 J2EE 概念及优势.....	2
1.2.1 J2EE 的概念.....	2
1.2.2 J2EE 的优势.....	3
1.3 J2EE 的应用程序模型.....	4
1.3.1 客户端.....	4
1.3.2 J2EE 服务器.....	5
1.3.3 企业信息系统层.....	7
1.4 J2EE 的核心 API 与组件.....	7
1.5 J2EE 安装与配置.....	8
1.5.1 J2EE SDK 1.4 的安装.....	8
1.5.2 配置 J2EE SDK 1.4 开发环境.....	9
1.6 小结.....	11
第2章 Eclipse 概述及开发环境的搭建.....	12
2.1 Eclipse 简介.....	12
2.2 Eclipse 的安装和运行.....	13
2.2.1 Eclipse 的获取.....	13
2.2.2 Eclipse 的安装.....	13
2.3 Eclipse 插件集成.....	15
2.3.1 Eclipse 插件简介及安装方法.....	15
2.3.2 Eclipse 多国语言包的安装.....	17
2.3.3 MyEclipse 插件的安装.....	18
2.3.4 其他几个常用的插件.....	20
2.4 小结.....	21
第3章 WebLogic Server 开发环境.....	22
3.1 WebLogic 概述.....	22
3.1.1 BEA WebLogic Server 在企业中的角色.....	22
3.1.2 BEA WebLogic Enterprise Platform.....	24
3.1.3 J2EE 的能力与基于标准的解决方案.....	24
3.2 安装 WebLogic Server.....	25
3.2.1 系统需求.....	26
3.2.2 WebLogic Server 的安装过程.....	26
3.2.3 WebLogic 域和域配置向导.....	29

3.3	配置 WebLogic JDBC.....	34
3.3.1	安装 Microsoft SQL Server 2000 JDBC 驱动程序	35
3.3.2	配置 JDBC 连接池	35
3.3.3	配置 JDBC 多池	40
3.3.4	配置 JDBC 数据源	41
3.3.5	配置 JDBC 数据源工厂	43
3.3.6	JDBC 配置和服务器性能	43
3.4	小结	44
第 4 章	开发 J2EE 应用入门	45
4.1	配置 MyEclipse 的 WebLogic 服务器	45
4.2	简单 Java 程序开发	47
4.2.1	使用工程向导	47
4.2.2	调试 Java 程序	49
4.3	Web 应用开发	49
4.3.1	创建 MyEclipse Web 项目	49
4.3.2	创建 JSP 文件	50
4.3.3	部署和测试 Web 应用程序	52
4.4	EJB 开发	53
4.4.1	创建一个 EJB Project	53
4.4.2	创建一个 Session EJB	54
4.4.3	XDoclet EJB 配置	56
4.4.4	运行 XDoclet 的代码生成并部署到 WebLogic	58
4.5	发布一个 EJB 项目	59
4.6	小结	60
第 5 章	J2EE 体系结构	61
5.1	J2EE 的模型-视图-控制 (MVC) 体系结构	61
5.2	J2EE 客户端	63
5.2.1	Web 客户端	63
5.2.2	小应用程序 (Applet)	63
5.2.3	应用程序客户端	63
5.2.4	JavaBean 组件架构	64
5.2.5	J2EE 服务的通讯	65
5.3	J2EE 服务器	65
5.4	后台数据库	66
5.4.1	JDBC 概述	66
5.4.2	数据库连接技术	68
5.5	小结	69
第 6 章	J2EE 设计模式	70
6.1	设计模式概述	70
6.1.1	什么是模式	70
6.1.2	模式的类型	71

6.1.3 J2EE 设计模式.....	71
6.2 表示层设计模式.....	73
6.2.1 截取过滤器.....	73
6.2.2 前端控制器.....	74
6.2.3 视图帮助.....	76
6.3 业务层设计模式.....	78
6.3.1 会话外观模式.....	78
6.3.2 值对象模式.....	81
6.4 集成层设计模式.....	82
6.4.1 数据访问对象模式.....	82
6.4.2 服务激发器模式.....	84
6.5 小结.....	84
第 7 章 会话 Bean.....	85
7.1 会话 Bean 介绍.....	85
7.1.1 无状态会话 Bean.....	85
7.1.2 有状态会话 Bean.....	85
7.1.3 SessionBean 接口.....	86
7.2 开发无状态会话 Bean.....	87
7.2.1 StatelessFundManager 的实现类.....	88
7.2.2 StatelessFundManager 的主接口.....	90
7.2.3 StatelessFundManager 的组件接口.....	90
7.2.4 StatelessFundManager 的部署描述项.....	91
7.2.5 部署.....	92
7.3 开发客户程序.....	92
7.3.1 客户程序.....	93
7.3.2 运行客户程序.....	95
7.4 无状态会话 Bean 的寿命周期.....	97
7.4.1 从客户角度看会话 Bean 的寿命周期.....	97
7.4.2 无状态会话 Bean 的状态图.....	98
7.4.3 无状态会话 Bean 的顺序图.....	99
7.5 开发有状态会话 Bean.....	101
7.5.1 StatefulFundManager 的实现类.....	101
7.5.2 StatefulFundManager 的主接口.....	104
7.5.3 StatefulFundManager 的组件接口.....	104
7.5.4 StatefulFundManager 的部署描述项.....	104
7.6 客户程序.....	105
7.6.1 修改客户程序.....	106
7.6.2 运行客户程序.....	108
7.7 有状态会话 Bean 的生命周期.....	108
7.8 小结.....	110

第 8 章	开发实体 Bean	111
8.1	实体 Bean 介绍	111
8.1.1	实体 Bean 与会话 Bean 的比较	112
8.1.2	容器管理与 Bean 管理的持久性	112
8.1.3	主键	112
8.1.4	Entity Bean 接口	113
8.2	开发 CMP 实体 Bean	115
8.2.1	CMP 的介绍	115
8.2.2	CMP 实体 Bean 工程	117
8.2.3	Bean 实现类 (Address.java)	119
8.2.4	CMP 实体 Bean 的主接口	124
8.2.5	CMP 实体 Bean 的组件接口	125
8.2.6	部署描述项和部署	127
8.3	开发客户程序	131
8.3.1	客户程序	131
8.3.2	运行客户程序	135
8.4	开发 Bmp 实体 Bean	136
8.4.1	Bean 实现类	136
8.4.2	BMP 实体 Bean 主接口	143
8.4.3	BMP 实体 Bean 远程接口	143
8.4.4	部署描述项	144
8.5	开发客户程序	146
8.6	实体 Bean 寿命周期	147
8.7	小结	149
第 9 章	消息驱动 EJB	150
9.1	Java 消息服务	150
9.1.1	JMS 概述	150
9.1.2	JMS 的编程接口	151
9.1.3	JMS 编程模型	154
9.1.4	JMS 和 J2EE	157
9.2	消息驱动 Bean 简介	157
9.2.1	消息驱动 Bean 概述	157
9.2.2	消息驱动 Bean 的编程接口	159
9.2.3	消息驱动 Bean 的特点	160
9.3	消息驱动 Bean 的生命周期	160
9.3.1	消息驱动 Bean 环境	161
9.3.2	创建消息驱动 Bean	162
9.3.3	使用消息驱动 Bean	162
9.3.4	删除消息驱动 Bean	163
9.4	开发消息驱动 Bean	163
9.4.1	消息驱动 Bean 的构成	163

9.4.2 消息驱动 Bean 的实现.....	163
9.4.3 部署消息驱动 Bean.....	165
9.4.4 测试消息驱动 Bean.....	166
9.5 消息 Bean 与会话 Bean 及实体 Bean 的集成应用.....	167
9.5.1 创建消息 Bean.....	169
9.5.2 创建会话 Bean.....	172
9.5.3 创建实体 Bean.....	174
9.5.4 集成消息 Bean、会话 Bean 和实体 Bean.....	176
9.6 小结.....	181
第 10 章 测试 EJB.....	182
10.1 软件测试基础.....	182
10.1.1 软件测试的目的.....	182
10.1.2 软件测试的原则.....	183
10.1.3 软件测试的对象.....	183
10.1.4 软件测试的种类.....	184
10.1.5 软件测试的基本方法.....	185
10.2 单元测试基础.....	185
10.2.1 单元测试概述.....	186
10.2.2 单元测试的种类.....	187
10.2.3 单元测试的必要性.....	188
10.2.4 单元测试的误区.....	188
10.3 JUnit.....	190
10.3.1 JUnit 简介.....	190
10.3.2 JUnit 测试作用.....	190
10.3.3 JUnit 测试设计原则.....	192
10.3.4 JUnit 测试起步.....	194
10.3.5 JUnit 测试用例.....	195
10.3.6 JUnit 测试套件.....	200
10.4 EJB 单元测试.....	203
10.4.1 EJB 单元测试基础.....	203
10.4.2 会话 EJB 的单元测试.....	205
10.4.3 实体 EJB 的单元测试.....	226
10.4.4 消息驱动 EJB 的单元测试.....	228
10.5 小结.....	230
第 11 章 Servlet/JSP 技术.....	231
11.1 Servlet 技术.....	231
11.1.1 Servlet 概述.....	231
11.1.2 Servlet API 介绍.....	233
11.1.3 Servlet 处理请求信息.....	234
11.1.4 Cookies 管理.....	235

11.2 JSP 概述.....	236
11.2.1 JSP 运行机制.....	237
11.2.2 JSP 运行环境.....	237
11.2.3 JSP 生命周期.....	237
11.2.4 JSP 网络应用服务框架.....	237
11.2.5 JSP 开发的几种典型方式.....	242
11.3 JSP 编程.....	243
11.3.1 JSP 基本语法.....	243
11.3.2 JSP 标记.....	244
11.3.3 JSP 指令.....	244
11.3.4 JSP 动作.....	245
11.3.5 JSP 隐含对象.....	249
11.3.6 JSP 注释.....	254
11.3.7 JSP 中的 JavaBean 组件.....	255
11.4 JSP 部署.....	258
11.4.1 JSP 编译器.....	258
11.4.2 部署 Web 应用.....	259
11.4.3 编辑部署描述器.....	261
11.5 小结.....	264
第 12 章 Struts2.0 开发.....	265
12.1 Struts2.0 的由来.....	265
12.2 Struts2.0 的体系结构.....	266
12.2.1 经典 MVC 模式.....	266
12.2.2 Struts2.0 框架简介.....	268
12.3 Struts2.0 的安装与配置.....	269
12.3.1 下载与安装.....	269
12.3.2 配置文件.....	270
12.4 Struts2.0 的基本功能.....	274
12.4.1 Action.....	275
12.4.2 result.....	279
12.4.3 类型转换(Type Conversion).....	282
12.4.4 数据校验.....	288
12.4.5 拦截器.....	293
12.4.6 国际化.....	301
12.5 Struts2.0 的标签.....	305
12.5.1 通用标签.....	306
12.5.2 UI 标签.....	322
12.6 小结.....	331
第 13 章 综合案例.....	332
13.1 电子购物系统数据模型分析.....	332
13.1.1 概述.....	332

13.1.2 账户模型.....	332
13.1.3 产品模型.....	334
13.1.4 订单模式.....	336
13.2 总体设计.....	338
13.2.1 购物系统模块设计.....	338
13.2.2 各个模块在设计方面的深入.....	339
13.3 购物系统中的设计模式.....	351
13.3.1 简单工厂模式.....	351
13.3.2 业务代表模式.....	375
参考文献.....	378

第1章 J2EE 概述

为了加速设计和开发企业级的应用程序，Sun 公司推出了一种全新的概念模型——Java 2 Platform Enterprise Edition (J2EE)，它与传统的互联网应用程序模型相比有着不可比拟的优势。

J2EE 平台提供了一个多层结构的分布式应用程序模型，该模型具有重用组件的能力、基于扩展标记语言 (XML) 的数据交换、统一的安全模式和灵活的事务控制；使开发者不仅可以比以前更快地发布对市场的新的解决方案，而且其独立于平台、基于组件的 J2EE 解决方案不再受任何提供商的产品和应用程序编程界面的限制。提供商和买主都可以选择最合适于其商业应用和所需技术的产品和组件。

本章将主要从如下几个方面介绍 J2EE。

- J2EE 的发展历程
- J2EE 概念及优势
- J2EE 的应用程序模型
- J2EE 的核心 API 与组件
- J2EE 安装与配置

1.1 J2EE 的发展历程

在“J2EE”这个缩略语被第一次介绍给世人的时刻，也许没有几个人可以预料出它在日后的奇特历程。那是在 1999 年 6 月的 JavaOne 年会上，时任 Sun 公司 Java 企业开发部门主管的 Mala Chandra 兴奋地预告了 Java 世界的这位新成员的来临。

那些不熟悉背景的听众们，揣摩着她演说中出现的一串串全新术语，表情大概又是惊喜又是迷惑：一个完整的“多层企业开发架构”，以“容器”和“组件”的形式提供服务，一套“厂商中立的开放技术规范”，对开发者隐藏了不同平台和“中间件”的技术细节，实现了企业级应用间的“无缝集成”等。在今天的开发者看来，这些似乎都已经是老生常谈，但在当时的场景下，闪动在幻灯片上的每一个口号，都意味着听众们事后又要经历一段困难的学习过程。幸亏 Chandra 有一副了不起的口才，这位本科念建筑学的印度裔高层主管，谈起软件架构来也有很强的空间想象力。她清晰地说明了设计 J2EE 架构的两个初衷：

首先，对于厂商来说，J2EE 意味着一套开放标准，加入这个标准，他们的产品就可以运行在各种不同的操作系统和工作环境下，成为一个成熟的企业运算体系中可替换的部件。

其次，对于开发者来说，J2EE 是一套现成的解决方案，采用这个方案，企业应用开发中的很多技术难题（包括跨平台移植、事务处理、安全性等等）就会迎刃而解，“信息像一条不间断的河流，经过各种各样的平台和设备，从企业应用系统的这一端流向那一端”。

换言之，在 J2EE 诞生伊始的环境中，市面上已经存在着很多程度不一的“准 J2EE 中间

件”了。它们主要用于解决三大类问题：事务处理、分布式对象管理和 Web 请求处理。首先，事务处理管理器（Transaction Processing Monitor）一直是高端企业计算领域的热门产品，著名的应用服务器厂商 BEA，正是通过收购事务处理软件 Tuxedo 进入中间件市场的。另一方面，从 90 年代初开始，越来越多的人把“N 层分布式对象架构”当成传统的客户端/服务器架构的替代方案。那时刚刚兴起的 CORBA 技术是推动这一趋势的重要力量。最后，Java 技术在 Web 领域中的应用也是当时初露头角的热点。

1997 年 6 月，Sun 在发布一款“Java Web Server”的同时第一次公布了 Servlet API；没想到这项技术副产品（连同 1998 年问世的 JSP）正好迎合了厂商的战略需要。对于上面提到的 N 层架构来说，HTTP 服务是一个非常理想的前端；所以基于 Java 的 Web 引擎，也在此时成了企业级 Java 解决方案的一个必不可少的部分。

Java、Web、事务、分布式对象，这几股开发潮流汇合在一起，形成了当时最热门的产品“应用服务器（Application Server）”或“中间件（Middleware）”。

说到这里，我们才梳理出了 J2EE 技术规范的第一个版本在 1999 年 12 月问世的实际意义。首先，它为 Java 企业开发提供了一幅清晰的全景，各项分支技术在这个领域中的地位和作用得到了客观、准确的定义。至此大家才对一个 Java 企业解决方案的构成要素有了基本共识。其次，它使用“容器”和“组件”等概念描绘了 Java 企业系统的一般架构，明确地划分了中间件厂商和应用开发者的职责所在。最后，J2EE 通过一套公开标准规定了应用服务器产品的具体行为，在执行此标准的厂商产品之间实现了一定程度的可替换性和互操作性。

1.2 J2EE 概念及优势

本节将从五个方面对 J2EE 进行比较全面的介绍。先从 J2EE 的概念说起，再谈它的优势，然后介绍 J2EE 典型的四层模型及其框架结构，最后简单介绍 J2EE 十三种核心技术。

1.2.1 J2EE 的概念

目前，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）。

J2EE 是一种利用 Java 2 平台来简化企业解决方案的开发、部署和管理相关的复杂问题的体系结构。J2EE 技术的基础就是核心 Java 平台或 Java 2 平台的标准版，J2EE 不仅巩固了标准版中的许多优点，例如“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等，同时还提供了对 EJB（Enterprise JavaBeans）、Java Servlets API、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。

J2EE 体系结构提供中间层集成框架用来满足无需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，J2EE 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的支持，完全支持 Enterprise JavaBeans，通过

良好的向导来支持打包和部署应用。

1.2.2 J2EE 的优势

J2EE 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。

(1) 保留现存的 IT 资产。企业必须适应新的商业需求,利用已有的企业信息系统方面的投资,而不是重新制定全盘方案。这样,一个以渐进的(而不是激进的,全盘否定的)方式建立在已有系统之上的服务器端平台机制是公司所需要的。J2EE 架构可以充分利用用户原有的投资,如一些公司使用的 BEA Tuxedo、IBM CICS, IBM Encina, Inprise VisiBroker 以及 Netscape Application Server。这之所以成为可能,是因为 J2EE 拥有广泛的业界支持和一些重要的企业计算领域供应商的参与。每一个供应商都对现有的客户提供了不用废弃已有投资,进入可移植的 J2EE 领域的升级途径。由于基于 J2EE 平台的产品几乎能够在任何操作系统和硬件配置上运行,现有的操作系统和硬件也能被保留使用。

(2) 高效的开发。J2EE 允许公司把一些通用的、很繁琐的服务端任务交给中间件供应商去完成。这样开发人员可以集中精力在如何创建商业逻辑上,从而相应地缩短开发时间。

(3) 能提供复杂的中间件服务。

① 状态管理服务。让开发人员写更少的代码,不用关心如何管理状态,这样能够更快地完成程序开发。

② 持续性服务。让开发人员不用对数据访问逻辑进行编码就能编写应用程序,能生成更轻巧,与数据库无关的应用程序,这种应用程序更易于开发与维护。

③ 分布式共享数据对象 CACHE 服务。让开发人员编制高性能的系统,极大提高整体部署的可伸缩性。

(4) 支持异构环境。J2EE 能够开发部署在异构环境中的可移植程序。基于 J2EE 的应用程序不依赖任何特定操作系统、中间件、硬件。因此设计合理的基于 J2EE 的程序只需开发一次就可部署到各种平台。这在典型的异构企业计算环境中是十分关键的。J2EE 标准也允许客户订购与 J2EE 兼容的第三方的现成的组件,把他们部署到异构环境中,节省了由自己制订整个方案所需的费用。

(5) 可伸缩性。企业必须选择一种服务器端平台,这种平台应能提供极佳的可伸缩性去满足那些在他们系统上进行商业运作的大批新客户。基于 J2EE 平台的应用程序可被部署到各种操作系统上。例如可被部署到高端 UNIX 与大型机系统,这种系统单机可支持 64 至 256 个处理器(这是 Windows NT 服务器所望尘莫及的)。J2EE 领域的供应商提供了更为广泛的负载平衡策略。能消除系统中的瓶颈,允许多台服务器集成部署。这种部署可达数千个处理器,实现可高度伸缩的系统,满足未来商业应用的需要。

(6) 稳定的可用性。一个服务器端平台必须能全天候运转以满足公司客户、合作伙伴的需要。因为 INTERNET 是全球化的、无处不在的,即使在夜间按计划停机也可能造成严重损失。若是意外停机,那会有灾难性后果。J2EE 部署到可靠的操作环境中,他们支持长期的可用性。一些 J2EE 部署在 Windows 环境中,客户也可选择健壮性能更好的操作系统如 Sun Solaris、IBM OS/390。最健壮的操作系统的可用性可达 99.999% 或每年只需 5 分钟停机时间。这是实时性很强商业系统理想的选择。

1.3 J2EE 的应用程序模型

J2EE 使用多层的分布式应用模型，应用逻辑按功能划分为组件，各个应用组件根据它们所在的层分布在不同的机器上。事实上，Sun 设计 J2EE 的初衷正是为了解决两层模式（client/server）的弊端，在传统模式中，客户端担当了过多的角色而显得臃肿，在这种模式中，第一次部署的时候比较容易，但难于升级或改进，可伸展性也不理想，而且经常基于某种专有的协议——通常是某种数据库协议。它使得重用业务逻辑和界面逻辑非常困难。现在 J2EE 的多层企业级应用模型将两层化模型中的不同层面切分成许多层。一个多层化应用能够为不同的每种服务提供一个独立的层，图 1-1 是 J2EE 典型的四层结构：

- (1) 运行在客户端机器上的客户层组件；
- (2) 运行在 J2EE 服务器上的 Web 层组件；
- (3) 运行在 J2EE 服务器上的业务逻辑层组件；
- (4) 运行在 EIS 服务器上的企业信息系统（Enterprise Information System）层软件。

注意：事实上 J2EE 应用程序既可以是三层结构（去除图 1-1 中的 Web 层，由客户端直接和运行在 J2EE 服务器中的业务逻辑层通信），也可以是四层甚至更多层结构。但很多时候总是将 J2EE 应用程序的多层结构考虑为三层结构。这是因为它们分布在三个不同的位置：客户端机器、J2EE 服务器机器和在后端的传统的机器。三层结构的应用程序可以理解为在标准的两层结构的客户端/服务器模式的客户端应用程序和后端存储资源中间增加了一个多线程的应用程序服务器。

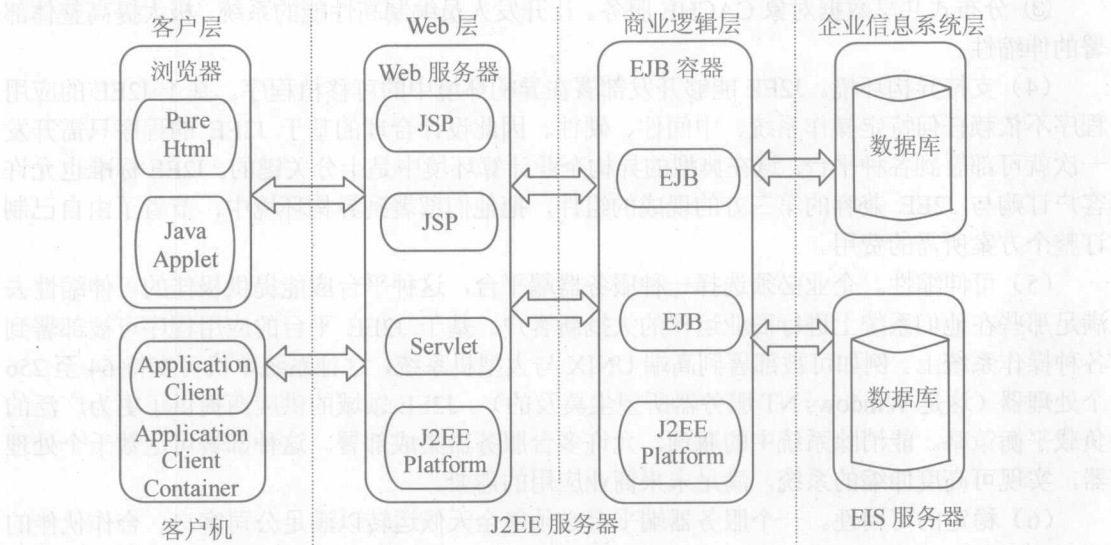


图 1-1 多层结构的应用程序

下面各小节将分别介绍 J2EE 应用程序的各层。

1.3.1 客户端

一个 J2EE 客户端既可以是一个 Web 客户端，也可以是一个应用程序客户端。