



J2EE

构建企业级应用 解决方案

陈华军 编著

人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS



附光盘



J2EE

构建企业级应用 解决方案

陈华军 编著

人民邮电出版社

图书在版编目 (CIP) 数据

J2EE 构建企业级应用解决方案 / 陈华军编著. ——北京: 人民邮电出版社, 2002.3
ISBN 7-115-10189-2

I. J... II. 陈... III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 010127 号

内 容 提 要

本书全面介绍了 J2EE (Java 2 Enterprise Edition) 的相关技术, 如 EJB、JSP/Servlet、JMS、JTA/JTS 等。同时, 本书从体系结构的角度, 分析讲解开发企业级应用程序的各个层次, 以便让读者理解 J2EE 相关技术之间的关系以及它们在体系结构中所处的位置和发挥的作用。

本书将帮助读者理解基于组件的开发, 学会用 EJB 开发企业 Internet/Intranet 应用程序; 全面了解 J2EE 相关技术, 理解它们之间的关系以及它们在 J2EE 体系结构中所处的地位; 通过实际的 J2EE 开发, 让读者了解开发的各个重要环节和基本方法。

本书面向具备了一定的 Java 基础且希望用 J2EE 进行高级 Java 应用程序开发的读者。

J2EE 构建企业级应用解决方案

- ◆ 编 著 陈华军
责任编辑 张立科
- ◆ 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>
读者热线 010-67180876
北京汉魂图文设计有限公司制作
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 27.25
字数: 665 千字 2002 年 3 月第 1 版
印数: 1-5 000 册 2002 年 3 月北京第 1 次印刷

ISBN 7-115-10189-2/TP · 2821

定价: 42.00 元 (附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

前 言

一、概述

J2EE 是目前市场上主流的企业级分布式应用程序开发平台，它是在 Sun 公司的领导下，多家公司参与共同制定的应用程序开发规范，正是因为它是一种规范，所以能够得到业界的支持，从而具备了灵活的可移植性；同时因为 Java 本身所具有的“一次编写，到处执行”的特性，也具备了应用程序的跨平台性。使用 J2EE 架构，能够快速、高效地开发出可移植的、跨平台的、具有强大功能的企业级应用程序。

本书全面介绍了 J2EE (Java 2 Enterprise Edition) 的相关技术，如 EJB、JSP/Servlet、JMS、JTA/JTS 等。同时，本书从体系结构的角度，分析讲解了开发企业级应用程序的各个层次，以便让读者理解 J2EE 相关技术之间的关系以及它们在体系结构中所处的位置和所发挥的作用。

二、本书结构

本书侧重 J2EE 的核心编程，而没有展开介绍应用程序服务器。

在结构上，本书可分为两个部分，第一部分介绍 J2EE 的编程基础。这部分逐一介绍了 J2EE 相关技术的基础知识，其中的实例采用的是 J2SDKEE 1.3 自带的 J2EE 应用服务器和 Deploytool 部署工具，这是 Sun 公司提供的官方参考版本，主要用于学习使用。

在实际的应用程序开发过程中是不会使用 NotePad 的，应用服务器和部署工具也不会使用学习版本，所以在本书的第二部分详细介绍了如何使用 Jbuilder 和 Borland 公司的 BAS 应用服务器开发和部署 J2EE 应用程序。使用 Jbuilder 是因为它是业界的首选 IDE，使用 BAS 是因为 BAS 是和 Jbuilder 结合得最好的应用服务器。但是这并不意味着这是最合理的选择，IDE 和应用服务器的选择依赖于用户的资金投入、开发人员的水平、各个应用服务器对 J2EE 规范的支持水平等因素。用户应该根据实际情况作出选择。

三、本书的读者对象

本书面向具有一定 Java 基础而希望用 J2EE 开发企业应用程序的读者。

首先，读者必须具备一定的 Java 基础编程经验，所以本书不会讲解如何配置 CLASSPATH、如何打包应用程序等。在阅读本书之前，读者应该对 Java 的基本语法和 JVM 有所了解，并且应该了解如何使用 Java 的基础类库中最核心的部分，如输入输出流等。

此外本书还涉及到了 XML 相关的知识，读者在阅读本书之前，还应该对 XML 的作用和基本语法有所了解。但 XML 并不是 J2EE 的核心内容，所以本书并没有对 XML 作专门介绍。

四、本书的特点

- ◆ 丰富的实例，不同层次、不同难度、循序渐进。

- ◆ 侧重于 J2EE 的整体架构，全面介绍 J2EE 相关技术以及它们之间的关系和它们所发挥的作用。
- ◆ 介绍全新的计算模式：分布式计算。

五、本书的目的

- ◆ 理解基于组件的开发，学会用 EJB 开发企业 Internet/Intranet 应用程序。
- ◆ 全面了解 J2EE 相关技术，理解它们之间的关系以及它们在 J2EE 体系结构中所处的地位。
- ◆ 通过实际的 J2EE 开发，让读者了解开发的各个重要环节和基本方法。

由于作者水平有限，疏漏之处在所难免，请读者批评指正。

编 者

目 录

第 1 章 J2EE 概述	1
1.1 J2EE 的产生背景	1
1.1.1 企业级应用程序——电子商务	1
1.1.2 多层体系结构	1
1.1.3 服务器端组件体系结构	2
1.1.4 实现多层应用程序的解决方案	2
1.2 J2EE 技术	3
1.2.1 J2EE 的优势	3
1.2.2 J2EE 的体系结构	4
1.2.3 J2EE 的服务器端运行环境	5
1.2.4 J2EE 相关技术概览	7
1.2.5 J2EE 的开发角色	8
1.3 J2EE 的应用	9
1.3.1 多层 J2EE 应用程序	9
1.3.2 独立客户端应用程序	10
1.3.3 简单 Web 应用程序	10
1.3.4 BTB (Business to Business) 企业对企业应用程序	10
1.4 本章小结	11
第 2 章 服务器端组件	13
2.1 Enterprise Bean	13
2.1.1 Enterprise Bean 的分类	13
2.1.2 EJB 的客户端	14
2.1.3 几种 EJB 的适用场合	14
2.2 中间层与分布式对象	15
2.2.1 分布式对象	15
2.2.2 中间层	16
2.3 Enterprise Bean 的组成	18
2.3.1 Enterprise Bean 类	18
2.3.2 EJB 对象 (EJB Object)	19
2.3.3 Remote Interface	20
2.3.4 Local Interface	21
2.3.5 Home 对象和 Home Interface	21
2.3.6 部署描述文件 (Deployment Descriptors)	22
2.3.7 集成	23
2.4 Enterprise Bean 的调用	23

2.4.1	使用 JNDI 寻找 Home 对象	24
2.4.2	Enterprise Bean 调用实例	25
2.5	本章小结	26
第 3 章	Session Bean	27
3.1	Session Bean 基础	27
3.1.1	Session Bean 的生命周期	27
3.1.2	Session Bean 的分类	28
3.2	Hello World 无状态 Session Bean	29
3.2.1	Home Interface	29
3.2.2	Remote Interface	30
3.2.3	Bean 类	31
3.2.4	部署描述文件	32
3.2.5	Enterprise Bean 的集成	33
3.2.6	编写客户端代码	33
3.2.7	HelloBean 小结	35
3.3	有状态 Session Bean	36
3.3.1	有状态 Session Bean 池	36
3.3.2	CountBean	37
3.4	本章小结	41
第 4 章	永久性存储基础	43
4.1	永久性存储	43
4.1.1	Java 对象序列化	43
4.1.2	对象—关系映射	43
4.1.3	对象数据库	44
4.2	JDBC 基础	45
4.2.1	JDBC 概要	45
4.2.2	数据库的连接	46
4.2.3	数据源	48
4.2.4	Statement 简介	52
4.2.5	ResultSet 简介	54
4.2.6	PreparedStatement 简介	56
4.2.7	CallableStatement 简介	57
4.3	Entity Bean 基础	58
4.3.1	什么是 Entity Bean	58
4.3.2	Entity Bean 所包含的文件	59
4.3.3	Entity Bean 的一些特点	60
4.3.4	Entity 上下文对象	64
4.4	本章小结	64
第 5 章	Entity Bean	65
5.1	BMP Entity Bean	65

5.1.1	Entity Bean 基础	65
5.1.2	Entity Bean 的 finder 方法	67
5.1.3	BMP 实例——Account Bean	69
5.1.4	BMP Entity Bean 的生命周期	85
5.2	CMP Entity Bean	87
5.2.1	CMP Entity Bean 的特点	87
5.2.2	编写 CMP Entity Bean 的基本规则	91
5.2.3	CMP Entity Bean 实例	92
5.2.4	CMP Entity Bean 的生命周期	103
5.3	本章小结	104
第 6 章	数据库表关系	107
6.1	关系基数 (Cardinality)	107
6.1.1	关系基数	107
6.1.2	CMP 和 BMP 的不同点	107
6.1.3	用 BMP 实现一对一关系	108
6.1.4	使用 CMP 实现一对一关系	109
6.1.5	用 BMP 实现一对多关系	111
6.1.6	用 CMP 实现一对多关系	112
6.1.7	用 BMP 实现多对多关系	113
6.1.8	用 CMP 实现多对多关系	116
6.2	关系的方向性 (Directionality)	119
6.2.1	在 BMP 中实现方向性	119
6.2.2	在 CMP 中实现方向性	120
6.3	数据库表关系的其他问题	121
6.3.1	对象的推迟生成	121
6.3.2	聚和关系和组合关系	122
6.3.3	引用完整性	123
6.4	本章小结	124
第 7 章	Java Servlets 技术	125
7.1	Java Servlet 技术基础	125
7.1.1	Servlet 简单实例	125
7.1.2	用 HttpSession 对象跟踪保存客户端状态变量	127
7.1.3	Cookies	128
7.1.4	在 Servlet 中使用其他资源	129
7.1.5	在 Servlet 之间共享资源	131
7.2	Servlet 技术在 J2EE 中的应用	132
7.2.1	搭建简单的开发平台	132
7.2.2	Adder 实例	133
7.2.3	设置 ContextRoot	137
7.2.4	设置 JNDI 名	137

7.2.5	部署和运行	138
7.3	本章小结	138
第 8 章	Java Server Pages 技术	139
8.1	Web 应用程序设计的 JSP 方法	139
8.1.1	JSP 技术的作用	139
8.1.2	JSP 页面的外观	140
8.2	JSP 基本语法	141
8.2.1	JSP 小脚本	141
8.2.2	JSP 指令	141
8.2.3	JSP 动作	142
8.2.4	JSP 页面中的隐含对象	144
8.3	JSP 页面中的 Java Bean 组件	146
8.3.1	Java Bean 编写规范	146
8.3.2	在 JSP 页面中使用 Java Bean	146
8.3.3	使用 Java Bean 作为 JSP 和 Enterprise Bean 的中介	147
8.4	本章小结	159
第 9 章	自定义标签库	161
9.1	自定义标签库简介	161
9.1.1	自定义标签	161
9.1.2	使用自定义标签库的好处	161
9.1.3	自定义标签的使用方法	162
9.1.4	标签体	162
9.2	编写自定义标签	163
9.2.1	标签处理类	163
9.2.2	标签库描述文件 (Tag Library Descriptors)	164
9.2.3	标签属性	166
9.2.4	属性值的有效性验证	167
9.2.5	带有标签体的标签	168
9.2.6	脚本变量	168
9.2.7	标签之间的协同操作	171
9.3	iterate 标签实例	173
9.3.1	iterate 标签的使用	173
9.3.2	标签处理类	174
9.3.3	标签辅助信息类	176
9.4	通过自定义标签访问 EJB	176
9.4.1	编写 JSP 文件	176
9.4.2	编写标签库描述文件	177
9.4.3	编写标签处理类	178
9.5	本章小结	180
第 10 章	事务处理	181

10.1	事务处理基础	181
10.1.1	为什么需要事务处理	181
10.1.2	基本术语	182
10.1.3	事务处理模型	183
10.2	EJB 中的事务处理	184
10.2.1	Bean 实现的事务处理	185
10.2.2	容器管理的事务处理	185
10.2.3	客户端控制的事务处理	186
10.2.4	三种管理事务的方式的比较	187
10.2.5	Entity Bean 的事务处理	187
10.3	容器管理的事务处理	187
10.3.1	事务属性	187
10.3.2	容器管理的事务的滚回 (Roll Back)	190
10.3.3	同步 Session Bean 的成员变量值	191
10.3.4	容器管理的事务中不能使用的方法	192
10.4	Bean 管理的事务处理	192
10.4.1	JDBC 事务处理	193
10.4.2	JTA 事务处理	193
10.4.3	Bean 管理的事务中不能使用的方法	195
10.4.4	Bean 管理的事务和容器管理的事务	195
10.4.5	客户端控制的事务处理	196
10.5	事务的并发控制	197
10.6	本章小结	200
第 11 章	安全性	201
11.1	认证	201
11.1.1	J2EE 用户、域和组	201
11.1.2	管理用户和组	202
11.1.3	客户端认证	202
11.2	授权	203
11.2.1	定义角色 (Role)	203
11.2.2	设置方法的访问权限	204
11.2.3	把角色映射到用户或组	204
11.2.4	安全授权的传递	205
11.2.5	编码实现安全逻辑	205
11.3	认证与授权的实例描述	209
11.3.1	J2EE 应用程序客户端	209
11.3.2	Web 浏览器客户端	210
11.4	本章小结	211
第 12 章	JMS 和 Message-Driven Bean	213
12.1	JMS 基础	213

12.1.1	概述	213
12.1.2	JMS API 的基本概念	214
12.1.3	JMS API 编程模型	216
12.1.4	一个简单的 JMS 应用程序	221
12.1.5	提高 JMS 应用程序的可靠性	224
12.2	Message-Driven Bean	228
12.2.1	MDB 概述	228
12.2.2	编写 Message-Driven Bean	228
12.2.3	一个 Message-Driven Bean 的实例	230
12.2.4	Message-Driven Bean 中的分布式事务处理	231
12.3	Message-Driven Bean 与 Session Bean 的协同工作	231
12.3.1	概述	231
12.3.2	编写 Publisher Session Bean	232
12.3.3	编写 Message-Driven Bean	235
12.3.4	编写客户端程序	236
12.3.5	应用程序的部署和运行	237
12.4	Message-Driven Bean 与 Entity Bean 的协同工作	239
12.4.1	概述	239
12.4.2	编写应用程序客户端: HumanResourceClient.java	240
12.4.3	编写两个 Message-Driven Bean	243
12.4.4	编写 SetupOffice Entity Bean	248
12.4.5	编写 ScheduleMsgBean	254
12.4.6	应用程序的部署和运行	255
12.5	本章小结	258
第 13 章	J2EE 高级话题	259
13.1	CORBA 模型	259
13.1.1	CORBA 含义	259
13.1.2	理解 CORBA	259
13.1.3	为什么要关心 CORBA	260
13.1.4	CORBA 的工作原理	262
13.2	RMI-IIOP	268
13.2.1	RMI-CORBA 互操作的必要性	268
13.2.2	RMI 与 CORBA 的合并	270
13.2.3	解析 RMI 与 CORBA 的差别	271
13.3	结合 RMI 和 CORBA 的应用开发步骤	272
13.3.1	RMI-IIOP 客户端与 CORBA 对象实现	272
13.3.2	CORBA 客户端与 RMI-IIOP 对象实现	273
13.4	本章小结	274
第 14 章	开发前的准备工作	275
14.1	系统开发环境	275

14.2 应用服务器——BAS	276
14.2.1 应用服务器简介	276
14.2.2 应用服务器的适用领域	277
14.2.3 应用服务的各种性能	278
14.2.4 Application Server 种类	278
14.2.5 Borland Application Server 4.5 简介	278
14.3 开发工具 Borland Jbuilder 5	282
14.3.1 初识 Jbuilder 5 环境	282
14.3.2 开发前的环境配置	291
14.4 JDataStore 简介	298
14.4.1 JDataStore Explore 的功能简介	299
14.4.2 JDataStore Explore 的操作	299
14.5 本章小结	300
第 15 章 Esite 实例分析	301
15.1 Esite 实例简介	301
15.1.1 运行实例前的设置	301
15.1.2 预览 Esite 实例	305
15.2 数据库结构分析	308
15.3 EJB 结构分析	309
15.3.1 Entity Bean	309
15.3.2 Session Bean	310
15.4 功能模块分析	311
15.5 分析汇总	311
15.6 本章小结	312
第 16 章 EJB 层的开发	313
16.1 Entity Bean 的开发	313
16.1.1 新建项目和空 EJB 组	313
16.1.2 Entity Bean “User”的开发	315
16.1.3 Entity Bean “Inventory” 的开发	320
16.1.4 Entity Bean “Order” 和 “Orderitem” 的开发	321
16.1.5 Entity Bean “ShoppingCart” 的开发	324
16.2 Session Bean 的开发	327
16.2.1 Session Bean “ServerDataModule” 的开发	327
16.2.2 ServerDataModuleBean 类深入分析	338
16.2.3 Session Bean “Cart” 的开发	349
16.3 本章小结	357
第 17 章 Web 层的开发	359
17.1 开发 ClientDataModule 类	359
17.1.1 建立 ClientDataModule 框架	359
17.1.2 ClientDataModule 类源代码分析	361

17.2	Servlet 相关开发	366
17.2.1	开发 SessionBean 类	366
17.2.2	Login 类的开发	371
17.2.3	NewUserServlet 类的开发	373
17.2.4	Search 类的开发	376
17.2.5	AddItem 类的开发	377
17.2.6	RemoveItem 类的开发	379
17.2.7	Purchase 类的开发	380
17.2.8	Logout 类的开发	382
17.3	JSP 页面的开发	383
17.3.1	welcome.jsp 页面的开发	385
17.3.2	searchresults.jsp 页面的开发	386
17.3.3	shoppingcart.jsp 页面的开发	390
17.4	本章小结	392
第 18 章	集成 Esite 系统	393
18.1	在 BAS 中部署 EJB	393
18.2	配置 Tomcat 服务器的运行	395
18.3	运行 Esite	396
18.4	本章小结	397
第 19 章	客户端应用程序	399
19.1	开发 Application 类和 Frame 类	399
19.1.1	建立 Application 和 Frame 框架	399
19.1.2	添加库存管理功能	401
19.2	BookstoreManagerFrame 的源代码分析	405
19.3	本章小结	407
附录 1	EJB 快速参考手册	409
附录 2	OMG IDL 语法	419

第 1 章 J2EE 概述

本章将讲述 J2EE 在现代计算机应用领域中所扮演的角色以及 J2EE 所包含的关键技术。

1.1 J2EE 的产生背景

1.1.1 企业级应用程序——电子商务

在电子商务迅猛发展的今天，商业社会的各个部门需要依赖企业级应用程序来共同完成自己的工作，比如：企业之间需要在线交易；企业的电子商务网站需要使用银行的系统进行在线支付和结算；企业要进行网上分销和采购；政府部门的城市规划需要使用网络来进行招标投标……可以预见，在不久的将来，企业级的应用程序将大大影响社会经济的发展。

开发企业级应用程序的关键是在强调时间性（即适时性）的同时，还要求程序部署方便、移植灵活、容易升级和更新等。这对应用程序开发人员提出了两个基本的开发要求：快速和有效。那么快速开发和部署高质量的应用程序的关键又是什么呢？是“体系结构”，或者说是编程的模型。在计算机领域里，“Model”和“Architecture”两词占有相当重要的地位：好的模型常常能够让程序的开发工作事半功倍；好的体系结构通过合理的层次划分不仅可以提高开发效率，而且还可以为程序的升级和移植提供便利。

1.1.2 多层体系结构

传统的 C/S 体系结构的弊端就不多赘叙了，先谈谈目前已广为人知的三层或多层应用程序。

三层应用程序的起源早于 Web，但 Web 极大地推动了三层应用程序的发展，最初人们用 ASP、JSP、PHP 等脚本程序实现浏览器客户端和服务端端的交互。这种结构封装了简单事务逻辑的脚本程序，使得该程序可在 Web 服务器中运行并完成诸如数据库访问、安全认证、数值分析计算等任务，再把结果用 HTML 的格式返回给浏览器客户端，而 Web 服务器充当中间层完成商业逻辑计算。但是，这样的三层应用程序也存在以下弊端：第一，包含有大量商业逻辑运算的脚本程序执行效率不高；第二，脚本代码被嵌入到 HTML 标签中，尽管很多新技术提供了简化的编程方式，但开发调试效率仍然不高；第三，代码无法重用和移植；第四，以脚本代码的形式提供给用户的程序不利于版权保护。除此以外，随着电子商务的快速发展，人们对应用程序的要求也越来越高。比如远程调用、负载均衡、集中高效的事务和安全处理、集成企业已有系统（比如 ERP）以及最重要的“软件复用”和分布式计算，所有这些新的需求都需要一个更为复杂的体系结构、更为细分的层次和功能更为强大的服务器中间层来支持。

1.1.3 服务器端组件体系结构

无论是三层还是多层体系结构，都需要一个强大的中间层，在这个中间层中运行着服务器端组件，这些中间件（Middleware）完成大部分的企业计算工作。服务器端组件需要一个运行的环境，这就是应用服务器，应用服务器完成很多底层的工作，如事务处理、安全性、线程调度、数据库连接池、组件之间的通信等。服务器端组件体系结构如图1-1所示。

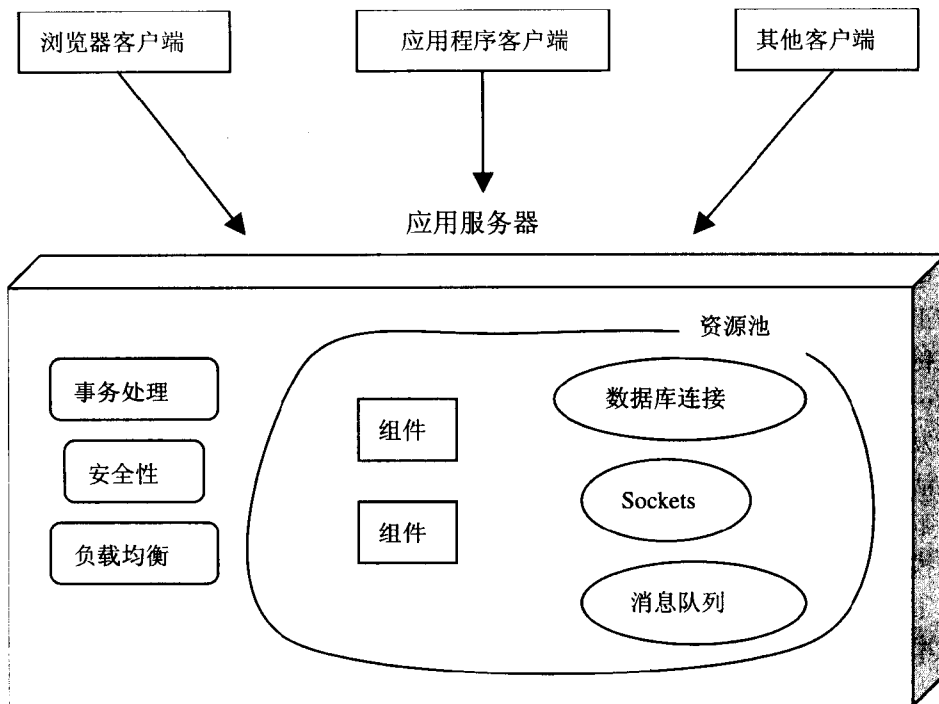


图 1-1 服务器端组件运行环境——应用服务器

1.1.4 实现多层应用程序的解决方案

1. 微软.NET 体系结构

微软公司的.NET体系结构是Windows DNA体系结构的升级版本。.NET平台最重要的部分是.NET Framework，这是一个与操作系统紧密相关的综合运行环境，它包括面向组件的中间层基础结构（COM+）、Common Language Runtime（CLR）环境、一个运行时的编译器和一组使用.NET组件模型打包的操作系统库。

程序员的开发工具是 Visual Studio.NET。表示层程序员使用 Visual Studio.NET 来定义向瘦客户端系统提交 HTML 页面格式的显示逻辑；商务层程序员使用 Visual Studio.NET 以多种语言实现商务逻辑，然后将商务逻辑打包为 COM+ 组件。Visual Studio.NET 是中性的语言，也就是说应用程序可以采用多种语言开发，然后集成到一块，这一点可以说是.NET 一大优势。.NET 的服务器端组件可用微软公司开发的 VB.NET、VC.NET、C#等语言编写。微软公

司把不同语言编写的组件（COM）翻译成一种中间语言（Microsoft Intermediate Language），实现了程序的“跨语言”特性，这种中间语言通过 Common Language Runtime（CLR）执行（CLR 类似于 Java 中的 JRE）。将语言特性合并到一种通用运行语言（CLR）中，而不是合并到一种特定的语言中，这使得各种语言可以自由地在整个 .NET 平台内互用。

客户端（浏览器、无线设备等）采用 ASP.NET 与服务器端组件交互；服务器端组件通过 ADO.NET 访问数据库；组件运行于 MTS 或 COM+ 中；消息处理采用 MSMQ。此外 .NET 还支持多种服务器：最新的 .NET 企业级服务器是 Application Center Server（应用中心服务器）；Internet Security and Acceleration Server（Internet 安全与加速服务器，ISA Server）重点解决表示层的需求；BizTalk Server 是一个综合的集成产品，主要用来将各种组织机构的操作联系在一起，并且允许该组织机构与合作伙伴的操作互用；Commerce Server 是用于创建电子商务站点的架构，它重点是解决电子商务零售业务的需求，使用 Commerce Server 提供的组件，并进行专门化，就可以快速地创建电子商务零售网站。

.NET 的缺点是从开发到部署到运行，都只能采用微软公司的平台。此外，由于 .NET 还不够成熟，所以它的市场份额并不高。

2. Sun J2EE 体系结构

这是本书所要介绍的内容。J2EE 是由 Sun 公司领导、多家公司共同参与制定的一个企业应用程序开发标准。熟悉 Java 语言的人都知道，Java 2 平台有 3 个版本，它们分别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）。

Sun J2EE 并不像微软公司的 .NET 那样是一系列的产品，而是一个规范和标准，任何一家公司都可以依据这些规范和标准来开发自己的产品。Sun 公司的目的就是让多个厂家共同参与这个规范的制定，这样，只要以这样的规范来开发自己的产品就能给用户带来更大的开放性和可移植性，正是因为如此，基于 Java 的应用服务器要比 .NET 多很多。Java 的开放性和 Java 本身具有的“一次编写、到处执行”的跨平台性使得 J2EE 在企业分布式计算领域中占据了很大的优势。后面的章节将详细介绍 J2EE 的相关技术。

1.2 J2EE 技术

J2EE 包含多种技术，这些技术为用户提供了稳定安全的开发平台、功能强大的应用服务器、方便移植和跨平台的应用程序。下面将对这些技术进行介绍。

1.2.1 J2EE 的优势

J2EE 拥有 Java 固有的跨平台特性，为开发具有可伸缩、灵活、易维护的企业电子商务系统提供了良好的机制。

■ 保留并集成企业已有 IT 资产

在利用已有的企业信息系统的基礎之上开发新的系统，而不是重新制定全盘方案。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 都可以非常方便地集成到 J2EE 应用程序中。这是因为 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升级途径。由于基于 J2EE 平台的产品对软硬件的要求不高,因此现有的操作系统和硬件也能被保留使用。

■ 高效地开发

J2EE 允许企业程序开发人员把一些通用的、很繁琐的服务端任务交给中间件供应商完成。这样开发人员可以把精力集中在商业逻辑上,相应地缩短了开发时间,甚至只需要像搭积木一样集成一些现成的组件就可以完成整个应用程序的开发,这就是“软件复用”的思想。此外,高级中间件供应商还提供以下复杂的中间件服务:

- ◆ 状态管理服务: 此服务使得开发人员不用编写太多代码,不用关心如何管理状态;

- ◆ 永久性存储服务: 此服务使得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;

- ◆ 分布式共享数据对象缓存服务: 利用该服务,开发人员能够编制出高性能的系统,极大提高整体部署的伸缩性。

■ 支持异构环境

J2EE 拥有 Java 固有的跨平台特性,用 J2EE 开发的应用程序能够非常方便地部署在不同的硬件和操作系统中,此外由于不同的应用服务器开发商都遵循相同的 J2EE 规范开发他们的产品,所以同一个 J2EE 应用程序可以灵活地部署在不同的应用服务器中。因为不同的客户完全可能会根据他们的喜好和实际情况(资金投入、已有软硬件平台等)选择不同的应用程序运行环境,所以设计只需开发一次就可部署到各种平台的应用程序在典型的异构企业计算环境中是十分关键的。J2EE 标准还允许客户订购与 J2EE 兼容的第三方的现成的组件,把这些组件部署到异构环境中,从而节省由自己完全实现所需的费用。

■ 可扩展性

企业选择的服务器端平台应能提供极佳的可扩展性,以满足新客户的需求。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 64~256 个处理器(这是 Windows 服务器不能达到的)。此外, J2EE 应用服务器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,实现可高度扩展的系统,满足未来商业应用的需要。

■ 稳定性和持续性

Internet 是全球化的,所以服务器必须 24 小时全天候运转,若是意外停机,就有可能丢失一笔大交易的机会,损坏企业形象,甚至导致灾难性后果。J2EE 应用程序可以部署到可靠的操作环境中,它们支持长期的可用性,是实时性很强的商业系统理想的选择。

1.2.2 J2EE 的体系结构

J2EE 的体系结构可以分为 4 层,如图 1-2 所示。