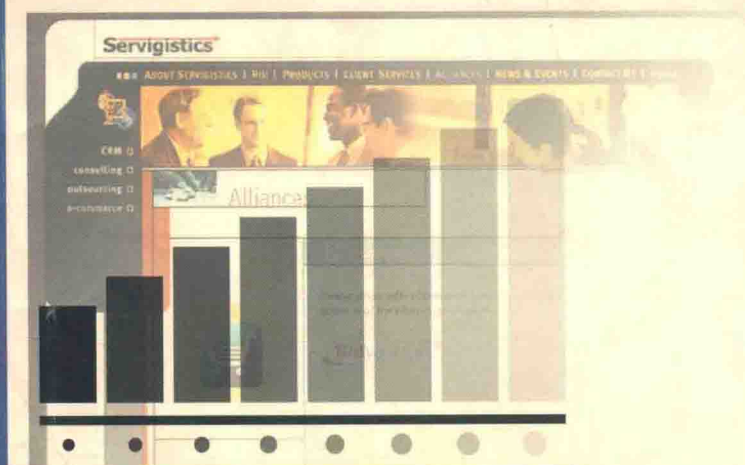


Java 编程实例系列丛书

# J2EE EJB 应用编程实例



刘 特 编著

- ◆ 全面解析Java技术
- ◆ 丰富、简洁的实例，使晦涩的概念和理论变得轻松易读
- ◆ 超凡的实战经验，帮您进入更广阔的编程空间

# Java



清华大学出版社

Java 编程实例系列丛书

# J2EE EJB 应用编程实例

刘 特 编著

清华大学出版社  
北 京

## 内 容 简 介

J2EE 使用 Java 语言作为基于 Web 应用程序的开发工具, 不仅融合了 Java 语言易移植、跨平台、安全性和面向对象的特点, 而且还具有强大的事务处理能力。J2EE 的核心是 Enterprise Java Bean 组件体系, 它给出了如何开发和部署服务器端(Server)应用商业逻辑(Business Logic)组件的一种规范。本书着重讨论了 J2EE 体系机构和 Enterprise Java Bean 组件的开发和部署。不仅给出了详细的理论叙述, 而且通过分析一个个具体的实例, 来演示如何构架和发布一项具体的应用。

本书在介绍 J2EE Enterprise Java Bean 技术时兼顾理论和应用, 使得初学者能够比较容易上手, 而不会感到枯燥难懂。除此以外, 本书中的实例都是来源于实际应用中, 比较实用和具有代表性。

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

### 图书在版编目(CIP)数据

J2EE EJB 应用编程实例/刘特编著.—北京: 清华大学出版社, 2003

(Java 编程实例系列丛书)

ISBN 7-302-07196-9

I. J… II. 刘… III. Java 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 078274 号

出 版 者: 清华大学出版社

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

社 总 机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

组稿编辑: 彭 欣

文稿编辑: 刘 颖

封面设计: 一克米工作室

印 刷 者: 北京嘉实印刷有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 18.25 字数: 435 千字

版 次: 2003 年 9 月第 1 版 2003 年 9 月第 1 次印刷

书 号: ISBN 7-302-07196-9/TP·5238

印 数: 1~4000

定 价: 28.00 元

# 丛 书 序

《Java 编程实例系列丛书》在参与编写的作者、合作者，以及丛书编辑的共同努力下，近日即将与读者见面了，首先向付出辛勤劳动的丛书作者和编辑们致以崇高的敬意。

Java 语言是一种与平台无关的编程语言，它具有“一次编写，随处运行”的特点，所以，非常适合于分布式的网络编程。随着 Internet 网络在我国的迅速普及，参与和从事网络编程的人员也在不断地增加，同时，也将有更多的网络开发者选用 Java 语言作为编程工具，而一些 C 和 C++ 程序员也在逐步转向应用 Java 语言编写程序。为在较短时间内学会并掌握 Java 语言的编程方法和技巧，我们从实际应用出发，编写了《Java 编程实例》丛书系列，该套丛书包括：《Java 基础编程贯通教程》、《Java Applet 编程实例》、《Java 图形与动画编程实例》、《J2EE EJB 编程实例》、《Java 安全性编程实例》、《Java 数据库编程实例》、《Java 网络编程实例》、《Java 服务器编程实例》等。丛书的每一个实例都是一个具体的应用，具有较强的实用性和目的性，这套丛书的实例包括了 Java 语言应用的方方面面，而且实例难易结合、应用面广，非常适合初、中、高级的学习 Java 语言编程的读者。

在这里有一点需要说明的是，如果您在阅读本套丛书之前，没有接触过 Java 语言，建议您先认真阅读一下《Java 基础编程贯通教程》这本书，该书叙述全面、重视教学方法，同时将丛书实例涉及到的内容贯通起来进行讲解，不但有利于把问题说清楚，也可以减轻初学者的学习负担。从头到尾，一本书基本解决问题，读者学习之前不需要有编程基础，只要读通本书，就能立即参与编程。这也是我们之所以把《Java 基础编程贯通教程》列入该套丛书的主要原因。

如果您以前接触过 Java 语言或已经具备一定的编程基础，那么您可以直接进入实例演练的世界，相信本套丛书将会成为您学习 Java 编程的良师益友。

《Java 编程实例系列丛书》在编写过程中，充分体现了理论联系实际，所有参加编写的人员都是多年使用 Java 语言开发网络实际应用项目、具有丰富的实际应用经验的研究员、高级工程师、工程师以及从事 Java 教学和培训的教师等，有些参编人员已经获得了 Sun 公司的 Java 程序员的资格认证。在丛书中的多数实例都是作者从实际应用项目中提炼出来的，它们具有很强的代表性和实用性，并由浅入深地介绍给读者，通过丛书的实例能够给读者一定的启发，可以说，该系列丛书为 Java 程序员开发各种实际应用项目提供了可参考的解决方案。

《Java 编程实例系列丛书》的内容涉及了计算机应用的前沿技术，例如，网络应用技术、图形动画以及多媒体技术、数据库技术、大型企业级应用开发技术、网络服务器应用技术、EJB 技术、网络安全技术等，每一个实例都有明确的目标，即解决一类问题。由于 Java 代码的可移植性，读者完全可以将丛书中的实例应用到自己的项目中。

在《Java 编程实例系列丛书》内的每一个实例源程序代码中都添加了详细的注释，增强了程序的可读性，便于读者学习和使用。

本系列丛书体系结构分类合理、各单册层次清晰、深入浅出、通俗易懂，相信一定能够得到 Java 程序员和 Java 爱好者的欢迎。

《Java 编程实例系列丛书》编委会

2003 年 3 月

## 编委会名单

主编：孙一林

编委：(按照姓氏笔划为序)

|     |     |     |     |     |
|-----|-----|-----|-----|-----|
| 丁友东 | 王 强 | 史斌星 | 史 佳 | 孙一林 |
| 刘 特 | 张 莉 | 何 梅 | 李 敏 | 李华彪 |
| 陈 雷 | 郑立华 | 赵文昉 | 侯晓强 | 徐迎晓 |
| 彭 波 | 黎晓冬 | 冀荣华 |     |     |

# 前言

近几年来, Internet 发展突飞猛进。随着网络应用的不断拓展,越来越多的企业开始寻求一种能够快速、便捷又具有强大功能的语言工具来开发企业级的 Web 应用程序。Java 语言可以说是到目前为止推广最快的一种应用程序开发语言,它的一个最显著的特点就是“平台无关性”,在一种操作系统上编译的 Java 代码可以非常方便地在另一种(相同或者完全不同)平台上运行,惟一的要求便是这些操作系统都安装了 Java 虚拟机并且拥有 JRE。这一特点使 Java 程序易于移植,容易实现跨平台运行,真正实现了“一次编译,到处可用”的功能。另外,Java 语言是一种纯面向对象的编程语言,Java 语言在实现 OOP 的开发上依据“元类模型”的构思,通过反射机制和对象序列化特性,使其更容易实现永久性的对象。不仅如此,Java 语言的另一个特点就是擅长开发网络应用程序。Java 语言将开发重点放在分布网络环境的程序设计中,支持分布网络对象计算和移动计算技术(RMI 和 COTBA 技术),而且 Java 语言提供了广泛的类库,可以方便地处理 HTTP 和 FTP 等 TC/IP 协议。Java 应用程序可以通过 Socket 来建立网络通信、利用 RMI 来调用远程对象中的方法。Java 语言拥有强大的安全性以及健壮的错误捕捉和垃圾回收机制,使其所开发的应用程序更加可靠。

随着 Internet 的飞速发展和新一代电子商务(E-Business)的快速崛起,人们对于分布网络应用程序的要求越来越高,其中的安全性、可靠性和运行性能是开发者追求的首要目标。为此, Sun 公司所推出的 J2EE 三层网络应用构架和 Enterprise Java Bean 的组件模式不仅实现了上述目标,而且在开发和部署整体应用程序时,降低了人力和时间的开支,缓解了系统开发的复杂程度,成为企业级分布网络应用程序开发的强有力工具。J2EE 是一个标准的网络应用体系结构,它使用 Java 语言作为基于 Web 应用程序的开发工具,不仅融合了 Java 语言易移植、跨平台、安全健壮性和面向对象的特点,而且还具有强大的事务处理能力。J2EE 的核心是 Enterprise Java Bean 组件体系,它给出了如何开发和部署服务器端(Server)应用商业逻辑(Business Logic)组件的一种规范。除此以外, J2EE 包含的 JSP 动态脚本和 Java Servlet 程序使 Client/Server 之间的交互更加快捷。

本书以实例的形式,详细地阐述了 J2EE Enterprise Java Bean 的体系结构和开发部署的方法,描绘了如何使用 Enterprise Java Bean 组件来开发和部署各种企业应用程序。本书由浅入深,层层推进,对 J2EE 中的重要组件(Java Bean、JSP 和 Java Servlet、RMI 和 Enterprise Java Bean 等)都给出了详细的分析和实例说明。本书在介绍 J2EE Enterprise Java Bean 技术时兼顾理论和应用,使得初学者比较容易上手,而不会感到枯燥难懂。除此以外,本书中的实例都是来源于实际应用中,比较实用和具有代表性。

本书由 6 部分组成:第 1 章主要介绍编程环境的准备和设置, J2EE 三层模式以及 Enterprise Java Bean 的组成结构;第 2 章通过分析 Java Bean 的计算实例和 Java Bean 的进度条组件实例介绍了 Java Bean 的构成和实现;第 3 章通过分析 Java Servlet 的网页计数器和基于 WebPage 的 mail 发送程序以及 Applet 与 Servlet 之间的通信实例,详细介绍了 Java Servlet 服务器端程序的工作原理和实现方法;第 4 章和第 5 章共演示了 3 个实例——利用

Cookie 和 Session 对象实现用户身份验证和投票程序,利用 JDBC-ODBC 和 JavBean 技术实现数据的处理和分页显示功能的实例,以及实现 JSP 同 Java Servlet 之间的通信实例,通过这些实例阐述了 JSP 的工作原理和实现方法;第 6 章着重分析了有关 RMI 的程序,包括利用 Java RMI 包来创建一个分布式的计算引擎服务器的实例,通过建立 Factory 机制实现在客户端启动 RMI 服务器程序并且创建它的实例,通过使用 Serializable 接口实现对象序列化实例以及通过利用 Security 包来对 RMI 对象进行数字签名的实例;第 7 章、第 8 章和第 9 章是最核心的 3 章,通过分析 3 个完整的 Enterprise Java Bean 实例——图书馆用户登录实例、网上花店订购实例和员工福利信息登记实例,分析了 J2EE Enterprise Java Bean 的结构和构成,以及开发和部署一项应用的全过程。

在本书出版之际,特别感谢徐迎晓、李敏、何梅三位同事,他们是我的良师益友,在技术方面给予了我很大的帮助;同时感谢刘必健、姜巧英夫妇和童争雄老师,在我写书期间给予我很大的支持和鼓励;还要感谢张帆、卢屹、唐宁、张以菁、朱小江、丛佳、邱海亮、秦梅仙、袁凌翔、庄静、封烨等同窗好友的支持。

本书所举实例,均只列出了关键代码。完整的代码读者请访问 [www.wenyuan.com.cn](http://www.wenyuan.com.cn) 下载。

由于作者水平有限,书中错误和疏漏之处在所难免,恳请广大读者批评指正。希望本书能对各位读者学习和应用 J2EE Enterprise Java Bean 有所帮助。

作者

2003 年 2 月

# 目 录

|                                              |     |
|----------------------------------------------|-----|
| 第 1 章 准备上手 .....                             | 1   |
| 1.1 开发环境的准备 .....                            | 2   |
| 1.2 基于 J2EE 的三层模式 .....                      | 4   |
| 1.3 Enterprise Java Bean 的组成结构 .....         | 7   |
| 1.4 环境设置 .....                               | 10  |
| 1.4.1 JDK 的环境设置 .....                        | 10  |
| 1.4.2 J2EE Server 的环境设置 .....                | 12  |
| 1.4.3 JSWDK Web Server 的环境设置 .....           | 16  |
| 1.5 小结 .....                                 | 17  |
| 第 2 章 Java Bean 组件设计实例 .....                 | 18  |
| 2.1 一个实现计算的 Java Bean 程序实例 .....             | 19  |
| 2.2 一个文件读取进度条 Java Bean 的设计实例 .....          | 34  |
| 2.3 小结 .....                                 | 43  |
| 第 3 章 Java Servlet 瘦客户端/服务器端的程序实例 .....      | 44  |
| 3.1 基于 Java Servlet 的网页计数器 .....             | 45  |
| 3.2 基于 WebPage 的 mail 发送程序 .....             | 49  |
| 3.3 Applet 与 Servlet 之间的通信实例 .....           | 56  |
| 3.4 小结 .....                                 | 64  |
| 第 4 章 基于 JavaServerPage 的会员身份验证和投票系统实例 ..... | 65  |
| 4.1 会员身份验证和投票系统实例 .....                      | 66  |
| 4.2 小结 .....                                 | 92  |
| 第 5 章 Java Server Page 的高级应用 .....           | 93  |
| 5.1 网上图书查询及订购系统实例 .....                      | 94  |
| 5.2 随机抽样结果的图形化显示实例 .....                     | 116 |
| 5.3 小结 .....                                 | 128 |
| 第 6 章 远程方法调用 Java RMI 程序实例 .....             | 129 |
| 6.1 计算引擎服务器实例 .....                          | 130 |
| 6.2 各大城市天气预报服务器实例 .....                      | 137 |
| 6.3 对象序列化的实例 .....                           | 146 |

|       |                                           |     |
|-------|-------------------------------------------|-----|
| 6.4   | 数字签名报文的 RMI 实现.....                       | 152 |
| 6.5   | 小结.....                                   | 159 |
| 第 7 章 | 基于 Enterprise Java Bean 的图书馆用户登录实例.....   | 160 |
| 7.1   | Enterprise Java Bean 的组成与结构.....          | 161 |
| 7.2   | 图书馆读者信息登录实例.....                          | 171 |
| 7.3   | 小结.....                                   | 209 |
| 第 8 章 | 基于 Enterprise Java Bean 的网上花店的订单部署实例..... | 209 |
| 8.1   | 网上花店的订单部署实例.....                          | 210 |
| 8.2   | 小结.....                                   | 249 |
| 第 9 章 | 基于 Enterprise Java Bean 的员工福利信息登记实例.....  | 250 |
| 9.1   | 员工福利信息登记实例.....                           | 251 |
| 9.2   | 小结.....                                   | 282 |

# 第1章

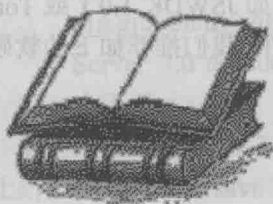
## 准备上手

### 本章要点

Enterprise Java Bean(EJB)是 Sun 公司制定的标准,它遵循 J2EE 应用程序平台的体系,用来执行于 J2EE 三层(Three-Tier)网络结构的中间层(Middle-Tier)的应用程序服务器(Application Server)上。EJB 是一种容易开发并可重复使用的服务器端网络信息存取组件。由于 EJB 所执行的环境——Enterprise Java Bean Server 是一个兼备安全、稳定和可信赖的数据事物存取结构,加上其本身属于 Java RMI 的体系,并支持 CORBA/IIOP 协议,使 EJB 不但具备远程存取的能力,而且易移植、扩充性强,并且可以跨越不同的 EJB Server 平台进行相互合作。本章主要介绍 J2EE 的构架和组成,以及 EJB 的基本概念和组成原理,并且给出了运行 EJB Server 的一些环境要求。

### 本章主要内容

- ① 编程环境的准备
- ② J2EE 三层模式的介绍
- ③ Enterprise Java Bean 的组成结构
- ④ 编程环境的设置



## 1.1 开发环境的准备

无论是设计 Java Applet 或是 Java Application, 还是开发 Enterprise Java Bean 程序, 都需要构建一个编译、解释和执行 Java 程序的环境。Java 平台的一个基本原则就是: “一次编译, 到处可用”。无论哪一个操作系统, 只要安装了 Java 虚拟机(Java Virtual Machine, JVM), 正确配置了 Java 运行环境(Java Running Environment, JRE), 那么就可以方便地运行在其他相同或不同操作系统平台上编译过的 Java 程序。充分体现了 Java 程序易移植、跨平台的独特优势。

近几年, 许多公司依据 Sun Microsystem 公司所定义的 Java 语言标准, 开发了自己的 Java 程序设计调试和封装发布的工具。其中比较著名的除了 Sun 公司本身的 Java 开发工具 JDK(Java Development Kit, 如图 1.1 所示)以及 J2EE 的开发环境 J2SDKEE(Java 2 Platform Software Development Kit, Enterprise Edition)以外, 还有 Microsoft 公司的 Visual J++和 Borland JBuilder(如图 1.2 所示), 以及 Oracle 公司的 JDeveloper 等。在支持 Java Servlet 和 Java Server Pages(JSP)方面, 除了 Sun 公司的 JSWDK 1.0.1 以外, 还有 Tomcat Web Server 和 WebLogic Server 等第三方服务器组件。由于各种工具都有其优点和不足之处, 所以在程序设计中应该灵活运用各种工具来完成不同 Java 程序的设计和开发, 而不拘泥于一种工具, 这样才能发挥各种工具的优势, 方便快捷地编写出许多难度较大的 Java 程序。



图 1.1 Sun Java Development Kit 的环境

本书所有程序是基于 JDK 1.3 和 J2SDKEE 1.3 开发的, 同时为了方便学习, 希望您安装合适的 Java 编辑器(如 JBuilder 等)和 Servlet Web Server(如 JSWDK 1.0.1 或 Tomcat Web Server 等), 以及数据库引擎(如 Microsoft SQL Server 等)软件, 我们推荐如下的软硬件环境:

- 硬件环境
  - ◆ CPU: 主频 550 MHz 以上;

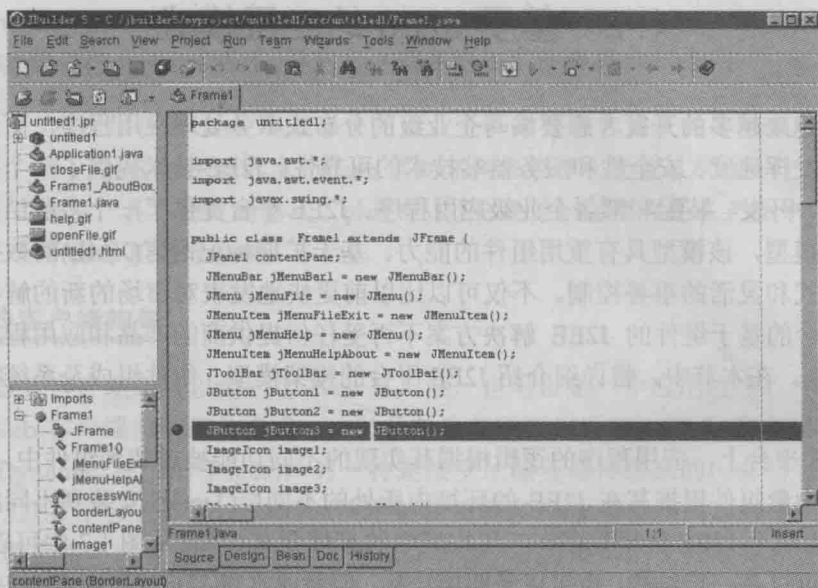


图 1.2 Borland Jbuilder 的环境

- ◆ 内存：128 MB 以上，256 MB 最佳；
- ◆ 硬盘：10 GB 左右自由空间；
- ◆ 显示器：14 英寸或更高；
- ◆ 分辨率：800×600 像素或更高；
- ◆ 网卡或调制解调器。
- 网络环境
  - ◆ 网络结构：Ethernet Network 或其他结构的 LAN 或 WAN；
  - ◆ 计算机：至少有一台服务器端计算机和一台客户端计算机。
- 软件环境
  - ◆ 操作系统：Windows 98/Windows 2000 专业版或 Server 版/Windows XP；
  - ◆ 开发 EJB 所需软件：JDK 1.3 版本和 J2SDKEE 1.3 版本，JDK 和 J2SDKEE 的各种版本都可以在网站 <http://java.sun.com> 上免费下载；
  - ◆ Java Web Server 软件：JSWDK 1.0.1 或更高的版本(JavaServer Web Development Kit)，JSWDK 的各种版本都可以在网站 <http://java.sun.com> 上免费下载；Tomcat Web Server；BEA WebLogic Server；
  - ◆ 数据库引擎：MySQL 的 Windows 版本，MySQL 是共享版的数据库，可以免费从网站 <http://www.mysql.com> 上下载；MySQL 的 JDBC 驱动程序 `org.gjt.mm.mysql.Driver`，也可以免费从网站 <http://www.mysql.com> 上下载；SQL Server 7.0 或 2000，其 JDBC 驱动程序可以从网站 <http://www.microsoft.com> 上下载；
  - ◆ Java 集程开发工具：Borland JBuilder 5.0 或更高的版本。

以上给出的软件中，Java Web Server 软件和数据库引擎都可以选取其中的一项或多项来调试程序。

## 1.2 基于 J2EE 的三层模式

今天,越来越多的开发者想要编写企业级的分布式事务处理应用程序,而这些应用程序必须可以发挥速度、安全性和服务器端技术的可靠性。J2EE 技术提供了一个基于组件的方法来设计、开发、装配和部署企业级应用程序。J2EE 平台提供了一个多层结构的分布式的应用程序模型,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。不仅可以比以前更快地发表对市场的新的解决方案,而且独立于平台的基于组件的 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(APIs)的限制。在本节中,将详细介绍 J2EE 平台的构架模型、组件组成及系统特性等重要内容。

在 J2EE 平台上,应用程序的逻辑根据其实实现的不同功能被封装到组件中,组成 J2EE 应用程序的大量组件根据其在 J2EE 的环境中所处的不同层(Tier)被安装到不同的机器中。图 1.3 所示为两个 J2EE 应用程序在三层构架中的具体分布。尽管从图 1.3 中可以看到 J2EE 应用程序既可以是三层结构,也可以是四层结构,但是通常将 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:客户端计算机、J2EE 服务器计算机和在后端的传统的计算机。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(Application Server)。

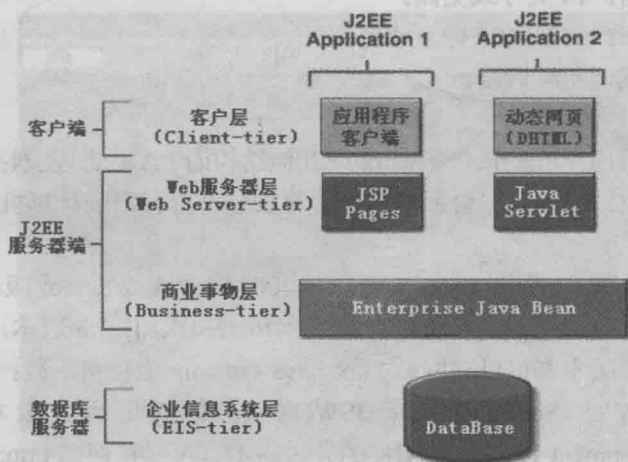


图 1.3 J2EE 的三层体系结构

J2EE 是一个标准的体系结构,它特别面向使用 Java 语言基于 Web 的企业应用程序的开发和部署。ISV 和企业能够使用这样一个结构体系进行内部网络应用的开发和部署,从而有效地替代了传统的二层模型。

J2EE 应用程序由组件组成。一个 J2EE 组件就是一个自带功能的软件单元,它随同相关的类和文件被装配到 J2EE 应用程序中,并实现与其他组件的通信。J2EE 规范是这样定义 J2EE 组件的:

- Java Application 和 Java Applet 是运行在客户端的组件。

- Java Servlet 和 Java Server Page (JSP)是运行在服务器端的 Web 组件。
- EJB 组件(Enterprise Bean)是运行在服务器端的商业软件。



**注意：** J2EE 组件由 Java 编程语言写成，并和用该语言写成的其他程序一样进行编译。J2EE 组件和标准的 Java 类的不同点在于：它被装配在一个 J2EE 应用程序中，具有固定的格式，并遵守 J2EE 规范。它被部署在产品中，由 J2EE 服务器对其进行管理。

### J2EE 的客户端构架

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。

一个 Web 客户端由两部分组成：一种是包含各种标记语言(HTML 和 XML 等)的动态 Web 页面(Dynamic HTML Pages)，另一种是接受从服务器传送来的，并将它显示出来的 Web 页面。J2EE 平台上的 Web Server 应用被称为瘦客户/服务器应用。当使用一个瘦客户端时，服务器会将一些重量级的操作(如数据库查询、执行复杂的商业规则和连接传统应用程序等)交给 J2EE 服务器的 Enterprise Bean 执行。这样就可以充分发挥 J2EE 服务器端技术在安全性、速度、耐用性和可靠性方面的优势(图 1.3 是 Web 层与 J2EE 应用程序之间的关系)。

应用程序客户端直接访问运行在商业层的 Enterprise Bean。然而，如果应用程序需要授权，一个 J2EE 应用程序客户端可以打开一个 HTTP 连接来与一个运行在 Web 层的 Java Servlet 或 JSP 建立通信。

J2EE 的 Web 组件既可以是 Servlet 也可以是 JSP 页面。Servlets 是一个 Java 编程语言类，它可以动态地处理请求并作出响应。JSP 页面是一个基于文本的文档，它以 Servlet 的方式执行，但是它更方便建立静态内容。

### 服务器端构架

在商业事物层中所包含的商业代码，表示了与特定的商业领域相适应的逻辑。它由运行在商业层的 Enterprise Bean 处理。一个 Enterprise Bean 也可以从存储器获取数据对它进行处理，并将其发送到客户端应用程序。Enterprise Beans 有 3 种类型：Session Beans、Entity Beans 和 Message-Driven Beans。一个 Session Bean 描述了与客户端的一个短暂的会话。当客户端的执行完成后，Session Bean 和它的数据都将消失，与之相对应的是一个 Entity Bean 描述了存储在数据库的表中的一行持久稳固的数据。如果客户端终止或者服务结束，底层的服务会负责 Entity Bean 数据的存储。一个 Message-Driven Bean 结合了一个 Session Bean 和一个 Java 信息服务(JMS)信息监听者的功能，它允许一个商业组件异步地接受 JMS 消息。

### 企业信息系统层的构架

企业信息系统层处理企业信息系统软件，并包含诸如企业资源计划(ERP)、主机事务处理、数据库系统和其他传统系统这样的底层系统。J2EE 应用程序组件可能需要访问企业信息系统，例如获得一个数据库连接等。

后台数据库(知识库)引擎的访问和数据处理方法：一个 J2EE 执行并不需要支持特定类

型的数据库,这意味着不同的 J2EE 产品所支持的数据库可以改变。JDBC API 让你可以从 Java 编程语言的方法中调用 SQL 命令。在一个 Enterprise Bean 中,当覆盖了默认的容器管理持续化或是在一个 Ssession Bean 中访问数据库时,必须使用 JDBC API。也可以使用 JDBC API 从一个 Servlet 或 JSP 页面中直接访问数据库,而不通过一个 Enterprise Bean 来完成。JDBC API 有两个部分:一个由应用程序组件用来访问数据库的应用程序级的接口,一个用来将 JDBC 驱动整合到 J2EE 平台中的服务提供商接口。

其他的一些 J2EE 服务机制和 API 包括:

- J2EE 服务机制

- ◆ Java Servlet 和 Java Server Page(JSP)技术

Java Servlet 技术允许你定义一个特定的 HTTP 的 Servlet 类。一个 Servlet 类为服务器扩展了这样一个性能:可以通过请求-响应这样的程序模式访问主机应用程序。尽管 Servlet 可以响应任何形式的请求,但是它们通常用于 Web 服务器上的应用程序。Java Server Page(JSP)技术使你可以将 Servlet 代码嵌入到基于文本的文档中。一个 JSP 页面就是一个基于文本的文档,它包含两个类型的文本:静态模板的数据,它可以表示为任何基于文本的格式,例如 HTML、WML 和 XML; JSP 单元,它决定页面如何建立动态的内容。

- ◆ Java 消息服务(JMS)

JMS 是一个消息标准,它允许 J2EE 应用程序建立、发送、接受和阅读消息。它使得建立连接简单的、可靠的和异步的公布式通信成为可能。

- ◆ Java 命名目录接口(JNDI)

JNDI 提供命名的目录功能。它为应用程序提供标准的目录操作的方法,例如获得对象的关联属性、根据它们的属性搜寻对象等。使用 JNDI,一个 J2EE 应用程序可以存储和重新得到任何类型的命名 Java 对象。因为 JNDI 不依赖于任何特定的执行,应用程序可以使用 JNDI 访问各种命名目录服务,包括现有 LDAP、NDS、DNS 和 NIS 等命名目录服务。这使 J2EE 应用程序可以和传统的应用程序和系统共存。

- ◆ Java 认证和授权服务(JAAS)

Java 认证和授权服务(JAAS)为 J2EE 应用程序提供了一个方法,以为一个特定的用户或一组用户进行认证和授权。JAAS 是标准的可插入认证模块(PAM)结构的 Java 版本,它对 Java 2 平台的安全认证框架进行了扩展以支持基于用户的安全认证。

- J2EE 中所包含的 API

- ◆ Java Transactions API

Java 事务 API(JTA)提供了划分事务的标准接口。J2EE 体系结构提供了一个默认的自动提交以处理事务提交和回滚。一个自动提交意味着在每一个数据库读写操作之后任何其他应用程序显示数据时都会看到更新了的数据。然而,如果你的应用程序执行两部分相互依赖的数据库访问操作,你可能会想要用 JTA API 去确定整个事务,这个事务将包含两个操作的开始、回滚和提交。

- ◆ JavaMail API 和 JavaBeans 激活架构 API

J2EE 应用程序可以使用 JavaMail API 来发送 e-mail 告示。JavaMail API 包含两部分：一个由应用程序组件用来发送 mail 的应用程序级的接口和一个服务提供接口。J2EE 平台包含作为服务提供的 JavaMail，使应用程序组件可以发送 mail。之所以要包含 JavaBeans 激活架构(JAF)是因为 JavaMail 要使用到它。它提供标准的服务以确定任意数据段的类型、如何对它进行访问、找出在其上可应用的操作并建立适当的 JavaBeans 组件以执行那些操作。

#### ◆ Java XML 处理 API

XML 是一种描述基于文本的数据的语言，使用 XML 使数据可以被任何程序和工具读取和处理。程序和工具可以生成其他程序和工具可以读取和处理的 XML 文档。Java XML 处理 API(JAXP)支持使用 DOM、SAX 和 XSLT 对 XML 文档进行处理。JAXP 使应用程序可以不依赖于特殊的 XML 处理执行来解析和转换 XML 文档。例如，一个 J2EE 应用程序可以使用 XML 来生成报表，而不同的公司都可以获得这个报表并使用各自最适宜的方法来处理它。一个公司可能会通过程序将 XML 数据导入到 HTML 中以便其可以在网站中公布，另一个公司可能会通过工具导出 XML 数据以制定销售预算，而另一个公司可能会将 XML 数据导入它的 J2EE 应用程序中对其进行处理。

#### ● J2EE Server 中的常用命令

表 1.1 列出了 J2EE 实现标准中所包含的命令。

表 1.1 J2EE 命令集合

| 命 令        | 作 用                                                |
|------------|----------------------------------------------------|
| j2ee       | 启动和终止 J2EE 服务                                      |
| Cloudscape | 启动和终止默认的数据库                                        |
| j2eeadmin  | 增加 JDBC 驱动、JMS 目的文件以及不同资源的连接 factory               |
| Keytool    | 建立公钥和私钥，并生成 X509 自签署证书                             |
| Realmtool  | 导入证书文件。为一个 J2EE 应用程序的认证和授权列表中增加或删除 J2EE 用户         |
| Packager   | 将 J2EE 应用程序组件打包到 EAR、EJB JAR、应用程序客户端 JAR 或 WAR 文件中 |
| Verifier   | 校验 EAR、EJB JAR、应用程序客户端 JAR 和 WAR 文件是否符合并遵守 J2EE 规范 |
| Runclient  | 运行一个 J2EE 应用程序客户端                                  |
| Cleanup    | 从 J2EE 服务器中删除所有已部署的应用程序                            |

## 1.3 Enterprise Java Bean 的组成结构

Enterprise Java Bean(以下简称 EJB)是一种面向对象的组件，其底层实现了 Java 标准的 RMI-IIOP(Remote Method Invocation/Internet Inter-ORB Protocol)协议机制，被运行于分布式结构的网络平台之上。EJB 主要是用来包含企业之间数据存取的商业逻辑(Business Logic)，通过调用其中的 business()方法来实现客户端的各种数据存取的事物活动。EJB 是一种容易开发实现、功能强大并且可以重复使用的网络 transaction 组件。由于 EJB Server 提供了 4

种强大的服务机制: persistence、security、concurrency 和 transaction 服务, 所以对于客户端来说, 其实是一种安全稳定和具有相当高的信任度的数据交换存取机构。

### 1. EJB技术的设计目标和优越性

J2EE 为 EJB 技术提供了工作环境。事实上, Sun 公司把若干项软件技术都设想为这样的构件块, 它们将使大型企业把以任务为关键的业务系统移植到 Java 环境中, 而 EJB 技术正是这些技术之一。EJB 组件是按它们自己的规范定义的, 但 EJB 技术并不是一项独立的技术。它建立在其他 Java 技术之上, 这些技术由 Sun 和其他 IT 公司联合规定, 它们一起提供了这个框架的内容, 该框架就称为 J2EE。Sun 公司在推出 EJB 时, 为 EJB 体系结构制定了以下目标:

- EJB 体系结构将是标准的组件体系结构, 用于以 Java 编程语言构建分布式面向对象的商务应用程序。
- EJB 体系结构将使编写应用程序变得容易。应用程序开发人员将不必了解低层次的事务和状态管理的细节、多线程、资源共享和其他复杂的低级 API。
- EJB 应用程序将遵循 Java 编程语言的“一次编写, 到处可用”的原则。
- EJB 体系结构将处理企业应用程序生命周期中的开发、部署和运行等方面。并且该体系结构将定义一些约定, 这些约定使多个供应商提供的工具能够开发并部署可在运行时互操作的组件。
- EJB 体系结构将实现一系列的兼容性, 例如, 与现有的服务器平台兼容、与用 Java 编程语言编写的其他 API 兼容以及 CORBA 兼容等。

使用 EJB 技术开发基于分布式网络应用程序, 将会使企业和开发人员获得许多好处。EJB 组件不但使编写应用程序更为简单, 而且服务器端商业逻辑(Business Login)可以移植, 这使开发人员能够将 Bean 从一种操作环境移植到另一种操作环境, 而无需重新编写其源代码。除此之外, EJB 体系结构内置了对典型企业级系统服务的支持, 包括分布式对象、事务处理、数据库、安全和全局命名机制。

### 2. EJB的组件

EJB 按照其性能和所具有的功能不同可分为两大类的 Bean 对象: Entity Bean 和 Session Bean。前者主要是对后台数据库中的数据记录进行存取操作, 而后者是用来对客户端复杂的商业逻辑(Business Logic)进行处理。

#### ● Entity Bean

Entity Bean 是一个标准的 Java 类, 它主要是利用对象的形式来展现数据库中的数据记录, 利用对象所定义的 business()方法, 来执行对应数据库的存取操作, 如插入记录、修改记录和删除记录等操作。而对于客户端而言, 不用去考虑使用哪一种 SQL 语言, 只要通过 Entity Bean 中所包含的 business()方法来执行相应的数据库操作便可以了。在客户端, 必须通过远程接口来存取 Entity Bean, 而实现 Remote interface 的对象被称为 EJB 对象。

按照存取数据表中数据记录的方式不同, 可分为 Container-managed persistence 的 Entity Bean 和 Bean-managed persistence 的 Entity Bean。前者主要是由 EJB 容器自