

# 《深入掌握J2EE编程技术》

## 图书基本信息

书名：《深入掌握J2EE编程技术》

13位ISBN编号：9787030103864

10位ISBN编号：7030103866

出版时间：2002-6-1

出版社：科学出版社

作者：杨绍方编著

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：[www.tushu000.com](http://www.tushu000.com)

# 《深入掌握J2EE编程技术》

## 内容概要

本书全面深入地介绍了当今最热闹的编程技术——J2EE平台开发技术，该技术主要用于开发基于Internet的多层结构应用程序。本书全面介绍了JSP层，EJB层和EIS层的设计思想与编程技术，涉及的内容包括：JDBC，JNDI，LDAP，Servlet，JSP，Taglib，EJB，J2EE，Connector，SAX与DOM，RMI，CORBA，JavaMail和JMS。这些内容基本覆盖了J2EE平台开发的主要内容。

本书的特点是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来演示J2EE编程的细节。

本书主要为J2EE编程技术的开发人员而写，既可作为J2EE的入门教材，也可作为J2EE开发人员的参考书。

## 书籍目录

### 第1章 绪论

#### 第1节 概述

##### 1.1.1 J2EE技术

##### 1.1.2 J2EE平台的结构

##### 1.1.3 J2EE平台的特点

##### 1.1.4 J2EE平台的优点

#### 第2节 J2EE的角色

#### 第3节 J2EE应用程序的开发

##### 1.3.1 J2EE应用程序的组成

##### 1.3.2 J2EE应用程序的开发过程

### 第2章 J2EE快速入门

#### 第1节 源程序及环境设置

##### 2.1.1 Enterprise Bean的源代码

##### 2.1.2 编译Enterprise Bean

#### 第2节 配置Enterprise Bean

##### 2.2.1 第1步：在DOS命令行启动J2EE服务器

##### 2.2.2 第2步：运行应用程序配置工具

##### 2.2.3 第3步：创建一个新的J2EE应用程序

##### 2.2.4 第4步：打包Enterprise Bean

##### 2.2.5 第5步：指定Enterprise Bean的JNDI名

#### 第3节 J2EE应用程序客户机

##### 2.3.1 ConverterClient.java源程序

##### 2.3.2 打包J2EE应用程序客户机

##### 2.3.3 指定应用程序客户机对Enterprise Bean的引用

##### 2.3.4 指定Enterprise Bean的JNDI名

#### 第4节 部署J2EE应用程序

#### 第5节 运行J2EE应用程序客户机

#### 第6节 创建Web客户机

##### 2.6.1 源程序：index.jsp

##### 2.6.2 打包Web客户机

##### 2.6.3 Web客户机如何引用Enterprise Bean

##### 2.6.4 指定JNDI名

##### 2.6.5 部署

##### 2.6.6 运行Web客户机

### 第3章 JNDI与LDAP

#### 第1节 命名与目录

##### 3.1.1 命名的概念

##### 3.1.2 目录的概念

#### 第2节 JNDI基础

##### 3.2.1 JNDI的结构

##### 3.2.2 naming包

##### 3.2.3 directory包

##### 3.2.4 事件

#### 第3节 LDAP编程技术

##### 3.3.1 理解LDAP

##### 3.3.2 在Java中使用Netscape Directory SDK

##### 3.3.3 快速入门

3.3.4 LDAP URL

3.3.5 常见操作

第4章 JDBC编程技术

第1节 JDBC编程技术综述

4.1.1 JDBC的概念

4.1.2 JDBC的用途

4.1.3 JDBC - ODBC桥

4.1.4 JDBC URL

4.1.5 odbc子协议

4.1.6 事务

第2节 DriverManager类

第3节 Statement接口

第4节 PreparedStatement接口

第5节 ResultSet接口

第6节 JDBC编程实例：JDBCDemo.java

4.6.1 创建新的ODBC数据资源

4.6.2 ODBC编程的典型步骤

4.6.3 源程序：JDBCDemo.java

第7节 JDBC数据库连接池

4.7.1 原理

4.7.2 JDBC数据库连接池的实现

4.7.3 在Servlet中使用连接池

第5章 Servlet编程技术

第1节 Java Servlet概述

5.1.1 关于Servlet

5.1.2 javax.servlet API的特征

第2节 体会Servlet编程：一个简单的网页计数器

5.2.1 源程序：SimpleCounter.java

5.2.2 编译、打包、部署与运行

第3节 Servlet的相关类和接口

5.3.1 HttpServlet类

5.3.2 HttpServletRequest接口

5.3.3 HttpServletResponse接口

5.3.4 ServletConfig接口

5.3.5 ServletContext接口

5.3.6 ServletInputStream类

5.3.7 ServletOutputStream类

第4节 探查Servlet服务器的环境信息

5.4.1 源程序

5.4.2 添加Web组件、部署和运行

第5节 Cookie编程技术

5.5.1 概述

5.5.2 ServletCookieAPI

5.5.3 Cookie的常用操作

5.5.4 实例

第6节 session编程技术

5.6.1 HTTP状态的维护

5.6.2 Session API的使用

5.6.3 实例：ShowSession.Java

## 第6章 JSP编程技术

### 第1节 概述

#### 6.1.1 什么是JSP页

#### 6.1.2 Java Server Page技术的特点

#### 6.1.3 JSP页的语义

### 第2节 Web应用程序及其模式

#### 6.2.1 应用程序的模式

#### 6.2.2 JSP技术在XML中的使用

#### 6.2.3 重定向请求

#### 6.2.4 包含请求

### 第3节 JSP页的标准语法和语义

#### 6.3.1 一般的语法规则

#### 6.3.2 错误处理

#### 6.3.3 注释

#### 6.3.4 转义的约定

#### 6.3.5 语义概述

### 第4节 指令 (directive)

#### 6.4.1 page指令

#### 6.4.2 include指令

#### 6.4.3 taglib指令

### 第5节 隐含对象

#### 6.5.1 隐含对象的类型

#### 6.5.2 隐含对象与范围

### 第6节 使用脚本元素

#### 6.6.1 如何增加脚本

#### 6.6.2 `<%` , `<%=` 和 `<%!` 之间的区别

### 第7节 标准的action

#### 6.7.1 `<jsp:useBean>`

#### 6.7.2 `<jsp:setProperty>`

#### 6.7.3 `<jsp:getProperty>`

#### 6.7.4 `<jsp:include>`

#### 6.7.5 `<jsp:forward>`

#### 6.7.6 `<jsp:param>`

#### 6.7.7 `<jsp:plugin>`

### 第8节 JSP容器

#### 6.8.1 JSP页的模式

#### 6.8.2 JSP页实现类

#### 6.8.3 缓冲

#### 6.8.4 预编译

### 第9节 JSP技术类

#### 6.9.1 JspPage接口和HttpJspPage接口

#### 6.9.2 JspWriter抽象类

#### 6.9.3 PageContext抽象类

#### 6.9.4 JspFactory抽象类

### 第10节 实例：显示页面所提交的内容

#### 6.10.1 源程序

#### 6.10.2 运行结果

### 第11节 JSP中的JavaBeans

#### 6.11.1 JavaBeans及其用途

## 6.11.2 创建JavaBeans

## 第7章 JSP taglib编程技术

### 第1节 taglib基础

#### 7.1.1 概述

#### 7.1.2 使用标记

#### 7.1.3 JSP tag API

### 第2节 taglib编程快速入门

#### 7.2.1 taglib的实现体系

#### 7.2.2 tag handler类

#### 7.2.3 实例

#### 7.2.4 taglib的配置、部署与运行

### 第3节 JSP tag API及其生命周期

#### 7.3.1 Tag接口及其生命周期

#### 7.3.2 IterationTag接口及其生命周期

#### 7.3.3 BodyTag接口及其生命周期

#### 7.3.4 TagSupport类

#### 7.3.5 BodyTagSupport类

#### 7.3.6 BodyContent类

#### 7.3.7 TagData类

#### 7.3.8 PageData类

#### 7.3.9 TagAttributeInfo类

#### 7.3.10 TagExtraInfo类

#### 7.3.11 TagInfo类

#### 7.3.12 TagLibraryInfo类

#### 7.3.13 TagLibraryValidator类

#### 7.3.14 TagVariableInfo类

#### 7.3.15 ValidationMessage类

#### 7.3.16 VariableInfo类

### 第4节 在标记中使用属性

#### 7.4.1 带属性的标记

#### 7.4.2 实例

### 第5节 包含标记的体

#### 7.5.1 含有体的标记

#### 7.5.2 tag handler类

#### 7.5.3 标记库描述符文件

#### 7.5.4 JSP文件（HeadingTag.jsp）

#### 7.5.5 运行

### 第6节 操作标记的体

#### 7.6.1 简介

#### 7.6.2 实例

## 第8章 EJB概述

### 第1节 简介

#### 8.1.1 Enterprise JavaBean的起源

#### 8.1.2 Enterprise JavaBean的特点

#### 8.1.3 Enterprise JavaBean的角色

#### 8.1.4 Java Beans组件和Enterprise JavaBeans

#### 8.1.5 Enterprise JavaBeans编程的限制

#### 8.1.6 场景：开发，装配和部署

### 第2节 Enterprise JavaBean基础

- 8.2.1 作为组件的Enterprise JavaBean
- 8.2.2 Enterprise JavaBean的约定
- 8.2.3 Session和Entity对象
- 8.2.4 标准的CORBA映射
- 8.2.5 Enterprise JavaBeans对session状态的维护
- 8.2.6 Helper对象
- 第3节 J2EE应用程序的设计思想
- 8.3.1 概述
- 8.3.2 选择系统结构
- 8.3.3 应用程序对象
- 8.3.4 数据访问对象
- 8.3.5 值对象
- 8.3.6 Session Beans作为Entity Beans的前台
- 第9章 Session Bean
- 第1节 Stateful Session Beans实例
- 9.1.1 Stateful Session Bean类：CartEJB.java
- 9.1.2 Home接口：CartHome.Java
- 9.1.3 Remote接口：Cart.Java
- 9.1.4 Helper类
- 9.1.5 客户机：CartClient.java
- 9.1.6 在Stateful和Stateless Session Beans之间做出选择
- 第2节 剖析SessionBean
- 9.2.1 简介
- 9.2.2 EJB容器
- 9.2.3 EJB对象
- 9.2.4 session对象的标识符
- 9.2.5 客户机角度的session bean生命周期
- 9.2.6 创建和使用session bean
- 第3节 Session Bean组件的约定
- 9.3.1 简介
- 9.3.2 会话的状态
- 9.3.3 Session Bean与其容器之间的协议
- 9.3.4 Stateful Session Bean状态图
- 9.3.5 Stateful Session Bean的序列图
- 9.3.6 Stateless Session Bean
- 9.3.7 Stateless Session Bean的状态图
- 9.3.8 Stateless Session Bean的序列图
- 9.3.9 Enterprise Bean提供者的责任
- 9.3.10 容器提供者的责任
- 第4节 Session Bean的例子
- 9.4.1 session bean提供者的责任
- 9.4.2 容器提供者提供的类
- 9.4.3 容器提供者的责任
- 第10章 Entity Bean
- 第1节 概述
- 10.1.1 简介
- 10.1.2 Entity Beans的特征
- 10.1.3 异常处理
- 第2节 Bean管理持久存储（BMP）的实例

- 10.2.1 Entity Bean类：AccountEJB.java
- 10.2.2 Home接口：AccountHome.java
- 10.2.3 Remote接口：Account.java
- 10.2.4 部署并运行AccountEJB实例
- 10.2.5 运行客户机
- 第3节 容器管理持久存储（CMP）的实例
- 10.3.1 Entity Bean类：ProductEJB.java
- 10.3.2 Home接口：ProductHome.java
- 10.3.3 Remote接口：Product.java
- 10.3.4 客户机类：ProductClient.java
- 10.3.5 创建表
- 10.3.6 运行ProductEJB实例
- 第4节 Entity Bean组件的约定
- 10.4.1 运行时执行模式
- 10.4.2 Entity的持久存储
- 10.4.3 实例的生命周期
- 10.4.4 Entity Bean组件的约定
- 10.4.5 多个事务的并行访问
- 10.4.6 不可重入和可重入的实例
- 10.4.7 Enterprise Bean提供者的责任
- 10.4.8 容器提供者的责任
- 10.4.9 容器管理的Entity Bean
- 第5节 序列图
- 10.5.1 创建一个entity对象
- 10.5.2 在一个事务中钝化和活化一个实例
- 10.5.3 提交一个事务
- 10.5.4 启动下一个事务
- 10.5.5 移走一个entity对象
- 10.5.6 发现一个对象
- 10.5.7 从池中增加或移走实例
- 第6节 Entity Bean的例子
- 10.6.1 Enterprise Bean提供者的责任
- 10.6.2 容器提供者提供的类
- 10.6.3 容器提供者的责任
- 第11章 打包与部署
- 第1节 概述
- 11.1.1 简介
- 11.1.2 角色与任务
- 第2节 打包J2EE应用程序
- 11.2.1 EJB模块
- 11.2.2 将组件打包放进EJB模块
- 11.2.3 Web模块
- 11.2.4 将组件打包放进Web模块
- 11.2.5 应用程序的客户端模块
- 第3节 部署描述符
- 11.3.1 简介
- 11.3.2 部署描述符的一般元素
- 11.3.3 Enterprise Bean的元素
- 11.3.4 Web组件元素



## 第4节 部署工具

### 11.4.1 部署工具的行为

### 11.4.2 部署工具的要求

## 第12章 安全

### 第1节 概述

#### 12.1.1 J2EE的安全

#### 12.1.2 J2EE的角色与安全

#### 12.1.3 资源的登录

### 第2节 认证

#### 12.2.1 J2EE的用户、领域和组

#### 12.2.2 客户机认证

#### 12.2.3 管理J2EE用户和组

### 第3节 授权

#### 12.3.1 声明角色

#### 12.3.2 声明角色的引用

#### 12.3.3 映射角色到J2EE的用户和组

### 第4节 其他安全措施

#### 12.4.1 Bean管理的安全性

#### 12.4.2 证书及其工具

#### 12.4.3 设置J2EE服务器证书

## 第13章 事务

### 第1节 EJB的事务

#### 13.1.1 简介

#### 13.1.2 EJB事务的案例

#### 13.1.3 事务的属性

### 第2节 认识JTS

#### 13.2.1 简介

#### 13.2.2 事务管理器的功能

#### 13.2.3 事务管理器的实现

### 第3节 认识JTA

#### 13.3.1 JTA与其他Java API的关系

#### 13.3.2 Java Transaction API

#### 13.3.3 在应用程序服务器中使用JTA

## 第14章 EIS与J2EE Connector

### 第1节 企业信息系统

#### 14.1.1 企业信息系统集成案例

#### 14.1.2 关系数据库管理系统的访问

#### 14.1.3 访问对象

#### 14.1.4 连接

### 第2节 J2EE Connector体系结构

#### 14.2.1 简介

#### 14.2.2 基本原理

#### 14.2.3 Connector体系结构

#### 14.2.4 案例

#### 14.2.5 连接管理的目标

### 第3节 J2EE Connector的连接管理

#### 14.3.1 体系结构：连接管理

#### 14.3.2 应用程序编程模型

#### 14.3.3 接口 / 类的规范

- 14.3.4 错误日志与跟踪
- 14.3.5 对象图
- 14.3.6 序列图
- 14.3.7 体系结构：非管理的环境
- 第4节 CCI简介
- 第15章 XML Java编程
- 第1节 XML入门
  - 15.1.1 概述
  - 15.1.2 XML如何使用
  - 15.1.3 XML的语法
  - 15.1.4 XML的元素
  - 15.1.5 XML的属性
  - 15.1.6 XML文档：cd\_catalog.xml
- 第2节 DTD简介
  - 15.2.1 元素
  - 15.2.2 属性
- 第3节 XML文件的分析
  - 15.3.1 XML解析器及其种类
  - 15.3.2 校验XML文档的合法性
- 第4节 常用的接口和类
  - 15.4.1 DocumentBuilderFactory类
  - 15.4.2 DocumentBuilder类
  - 15.4.3 Document接口
  - 15.4.4 Element接口
  - 15.4.5 Node接口
  - 15.4.6 Attr接口
  - 15.4.7 Text接口
  - 15.4.8 HandlerBase类
  - 15.4.9 DefaultHandler类
  - 15.4.10 XMLReaderFactory类
  - 15.4.11 XMLReader接口
  - 15.4.12 Attributes接口
  - 15.4.13 SAXParseException类
  - 15.4.14 SAXException类
  - 15.4.15 XmlDocument类
  - 15.4.16 ElementNode类
  - 15.4.17 Resolver类
- 第5节 实例
  - 15.5.1 使用DOM创建XML文档
  - 15.5.2 使用Swing JTree显示XML文档
  - 15.5.3 使用SAX统计XML文件的标记
- 第16章 RMI编程技术
- 第1节 概述
  - 16.1.1 分布式应用程序的责任
  - 16.1.2 RMI的编程思想
- 第2节 RMI的接口和类
  - 16.2.1 java.rmi.Remote接口
  - 16.2.2 java.rmi.RemoteException类
  - 16.2.3 Java.rmi.server.RemoteObject类及其子类

- 16.2.4 java.rmi.registry.LocateRegistry类
- 16.2.5 java.rmi.Naming类
- 16.2.6 Java.rmi.server.RemoteServer类
- 16.2.7 Java.rmi.server.UnicastRemoteObject类
- 16.2.8 java.rmi.RMI SecurityManager类
- 第3节 stub与skeleton
  - 16.3.1 stub的功能
  - 16.3.2 skeleton的功能
- 第4节 Java RMI编译器：rmic
- 第5节 RMI版的HelloWorld
  - 16.5.1 编写RMI服务器
  - 16.5.2 编写RMI客户端程序
  - 16.5.3 批处理文件：RmiHello.bat
  - 16.5.4 安全策略文件
  - 16.5.5 本地运行
  - 16.5.6 在不同机器上运行RMI软件
- 第6节 使用RMI技术实现分布式数据库操作
  - 16.6.1 远程接口
  - 16.6.2 实现远程接口
  - 16.6.3 RMI服务器程序
  - 16.6.4 RMI客户端程序
  - 16.6.5 编译程序：compile.bat
  - 16.6.6 安全策略文件：RmiJDBCServer.policy
  - 16.6.7 启动RMI服务器
- 第17章 Java CORBA编程技术
  - 第1节 CORBA与Java IDL
    - 17.1.1 关于CORBA
    - 17.1.2 关于Java IDL
    - 17.1.3 Nutshell中的CORBA
    - 17.1.4 Java IDL瞬态名字服务器：tnameserv
    - 17.1.5 关于idlj.exe
  - 第2节 相关的类和接口
    - 17.2.1 org.omg.CORBA.ORB类
    - 17.2.2 org.omg.CORBA.Object接口
    - 17.2.3 org.omg.CosNaming.NamingContext接口
    - 17.2.4 org.omg.CosNaming.NameComponent类
    - 17.2.5 org.omg.CORBA.Request类
    - 17.2.6 org.omg.CORBA.NVList类
  - 第3节 一个简单的实例
    - 17.3.1 批处理文件：CorbaDemo.bat
    - 17.3.2 定义并转换接口文件：CorbaDemo.idl
    - 17.3.3 实现CORBA服务器：TheDateServant.java
    - 17.3.4 实现客户机：CorbaDemoClient.java
    - 17.3.5 本地运行
    - 17.3.6 在网络中不同计算机上运行
    - 17.3.7 在applet中实现CORBA客户机
- 第18章 JavaMail
  - 第1节 概述
    - 18.1.1 JavaMail的体系结构

- 18.1.2 JavaMail类的层次结构
- 18.1.3 JavaMail框架 -
- 18.1.4 JavaMail事件模型
- 18.1.5 使用JavaMail API
- 第2节 Message及其相关的接口和类
- 18.2.1 简介
- 18.2.2 Message类的方法
- 18.2.3 相关的接口和类
- 18.2.4 消息的创建与发送
- 第3节 Mail Session
- 18.3.1 简介
- 18.3.2 提供者的登记
- 18.3.3 实例
- 18.3.4 安全管理
- 18.3.5 Store URL与Folder URL
- 第4节 消息的存储与获得
- 18.4.1 Store类
- 18.4.2 Folder类
- 18.4.3 Search的步骤
- 第5节 JavaBeans活动框架 ( JAF )
- 18.5.1 概述
- 18.5.2 对消息内容的操作
- 第6节 消息的写作
- 18.6.1 创建Message对象
- 18.6.2 消息的创建
- 18.6.3 设置消息的属性
- 18.6.4 设置Message的内容
- 18.6.5 创建MIME Multipart消息
- 第7节 传送的协议与机制
- 18.7.1 得到Transport对象
- 18.7.2 Transport的方法
- 18.7.3 Transport的事件
- 18.7.4 使用Transport类
- 第8节 Internet邮件
- 18.8.1 简介
- 18.8.2 MimeMessage类
- 18.8.3 MimeBodyPart类
- 18.8.4 MimeMultipart类
- 18.8.5 MimeUtility类
- 18.8.6 ContentType类
- 第9节 使用JavaMail API收发邮件
- 18.9.1 接收邮件
- 18.9.2 发送普通的邮件
- 18.9.3 发送带附件的邮件
- 第19章 JMS编程技术
- 第1节 概述
- 19.1.1 Messaging的定义
- 19.1.2 JMS API
- 第2节 JMS API的基本概念

- 19.2.1 JMS API的结构体系
- 19.2.2 消息域
- 19.2.3 Message消费
- 19.2.4 JMS API编程模型
- 第3节 编写简单的JMS客户机应用程序
  - 19.3.1 简单的point - to - point
  - 19.3.2 简单的Publish / Subscribe
- 第4节 创建强壮的JMS应用程序
  - 19.4.1 使用基本的可靠性机制
  - 19.4.2 使用先进的可靠性机制
- 第5节 在J2EE应用程序中使用JMS API
  - 19.5.1 使用Enterprise Bean产生并同步地接收消息
  - 19.5.2 使用Message - Driven Bean
  - 19.5.3 管理分布式事务
- 第6节 使用JMS API的J2EE应用程序
  - 19.6.1 编写组件
  - 19.6.2 创建和打包
  - 19.6.3 部署和运行
- 附录A 配置Jbuilder 5.0和Weblogic 6.X的开发环境
- 附录B Netscape目录服务器的安装
- 附录C 部署描述符文件
  - C.1 application.xml
    - C.1.1 语法和格式
    - C.1.2 实例
  - C.2 ejb-jar.xml
    - C.2.1 语法和格式
    - C.2.2 实例
  - C.3 web.xml
    - C.3.1 语法和格式
    - C.3.2 实例
- 主要参考文献

# 《深入掌握J2EE编程技术》

## 精彩短评

1、也太慢了2号订购23号才到手我曾想不要了呢

# 《深入掌握J2EE编程技术》

## 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:[www.tushu000.com](http://www.tushu000.com)