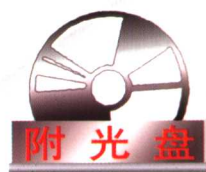


精通

J2EE

网络编程

● 邓子云 主 编
肖 峰 谢英辉 副主编



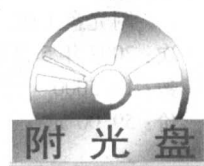
清华大学出版社

精通

J2EE

网络编程

● 邓子云 主 编
肖 峰 谢英辉 副主编



清华大学出版社

·北京·

内 容 简 介

随着 Internet 的深入人心, J2EE 技术的应用也越来越广泛, 这项技术也被大量的程序员所青睐。

本书共分 15 章, 内容涉及 J2EE 的基本概念、Java 语言基础、JSP 开发、文件处理、数据库操作、JNDI 开发、RMI 开发、XML 操作、EJB 编程、Web Services 开发、JMS 与消息 EJB、JavaMail 开发、常用安全技术、多线程技术和通信技术。书中各章节的内容相对独立, 读者可根据需要选读。

随书附带的光盘中包括本书用到的所有软件程序以及全书实例的源代码。本书使用的 J2EE 应用中间件和 Web 中间件是 Weblogic 8.1, Java 开发工具是 Eclipse 3.0, 数据库是 SQL Server 2000。

本书的适用面较广, 初、中、高级读者均能找到自己感兴趣的知识。本书可作为大中专院校计算机专业的 Java 课程教材, 也可作为 Java 培训班的教材, 亦方便对 J2EE 感兴趣的读者自学使用。

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 13501256678 13801310933

图书在版编目 (CIP) 数据

精通 J2EE 网络编程 / 邓子云主编; 肖峰, 谢英辉副主编. —北京: 清华大学出版社, 2007.5
ISBN 978-7-302-14692-6

I. 精… II. ①邓… ②肖… ③谢… III. Java 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 021297 号

责任编辑: 魏江江 赵晓宁

责任校对: 李建庄

责任印制: 李红英

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

c-service@tup.tsinghua.edu.cn

社 总 机: 010-62770175 邮购热线: 010-62786544

投稿咨询: 010-62772015 客户服务: 010-62776969

印 刷 者: 清华大学印刷厂

装 订 者: 三河市金元印装有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 34.75 字 数: 823 千字

附光盘 1 张

版 次: 2007 年 5 月第 1 版 印 次: 2007 年 5 月第 1 次印刷

印 数: 1~5000

定 价: 59.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 022658-01

作者简介



邓子云，硕士，高级工程师，中国系统分析员协会专业顾问；现供职于湖南现代物流职业技术学院，主持或参与了小额支付系统、财库行信息管理系统等大大小小的软件项目 20 余个，在学术刊物上发表论文 20 余篇；用 Java 开发了多个项目，具有丰富的理论与工程实践经验。E-mail: dengziyun@126.com。



肖峰，硕士，深圳美亚通公司技术总监；主持或参与了 Ebills 单证中心系统、863 计划 CZ-CIMS 等多个大型项目；曾获得省级科技进步二等奖，在核心刊物上发表多篇论文；擅长金融类软件项目的设计与实施管理。E-mail: xfygj@126.com。



谢英辉，硕士，软件工程师；现供职于长沙民政学院，目前致力于企业信息化、软件项目管理方向的研究；曾经在创智软件园日本事业部从事过两年软件外包工作，主持和参与了项目管理过程信息化系统、日本选举系统、资产继承等 10 余个大型软件项目；在学术刊物上发表过多篇关于项目过程管理信息化方面的论文。E-mail: xie-ying-hui@163.com。

Java 语言正受到越来越多程序员的青睐。由于 Java 语言具有良好的跨平台性,丰富的技术体系,编写出来的信息系统安全、稳定,大量的企业业务或管理信息系统都采用 J2EE (Java 2 企业版) 框架来构建。

目前市面上有关 J2EE 的书籍不少,大多为国外著作译成中文出版。由于 J2EE 涉及的技术内容相当广泛,学习起来有一定的难度,所以本书将以通俗易懂的语言由浅入深地介绍 J2EE 技术的基本理论及应用实例。

1. 本书的内容

全书共 15 章,每个章节内容相对独立,但也有所关联。在工程实践中,常常是多种技术的结合,如果本书中的内容之间具有一定的相关性,会用文字指出。

第 1 章介绍 J2EE 的一些基本概念,以引领读者快速入门,并列出本书中使用较多的开发工具,接着讲述如何安装与配置开发环境。

第 2 章介绍的内容是 Java 语言基础,主要介绍 Java 的 5 种基本数据类型、常量和变量、运算符与变量、数组、面向对象的类、对象、封装、类的创建、对象的创建、类的继承和多态性等基本概念。

第 3 章从介绍 JSP 的基本概念开始,逐步讲解 JSP 的常用开发技术: JSP 指令、JSP 内置对象、JavaBean 技术、Servlet 开发技术、JSTL 技术等,这些都是进行 Java 的 Web 开发常用的技术。

第 4 章讲述的是 Java 中的文件操作处理的相关内容。

第 5 章讲述数据库的基本操作,包括 JDBC、数据的查询与更新、事务处理、连接池的配置与使用、Web 开发中的数据分页处理等。

第 6 章将在学习了解 JNDI 的基本特点后,用代码示例的方式详细讲解 JNDI。

第 7 章主要讲述的内容包括 RMI 特点、一个简单 RMI 会话的实现、带有回调的 RMI 会话的实现、基于 IIOP 的 RMI 会话的实现、JNDI、LDAP 和 RMI 的结合使用。

第 8 章讲述的内容是在 Java 中如何操作 XML 文件,主要说明两种接口读写 XML 方式: DOM 和 SAX。

第 9 章讨论 EJB 的基本概念、框架及基本原理,再介绍会话 EJB 和实体 EJB 的开发,并在 JSP 中调用 EJB。

第 10 章介绍如何开发 Web Services,并在客户端调用。

第 11 章首先介绍 JMS 和消息 EJB 的基本概念,然后用实例来指导如何开发。

第 12 章首先讲述 JavaMail 中使用的邮件协议及基本的概念,再介绍与 JavaMail 开发相关的类及其常用方法,然后再编写发送和接收 E-mail 的程序。

第 13 章将从一个简单的加解密实例入手分析,逐步深入学习 J2EE 常用的安全技术。

第 14 章将介绍多线程技术。当系统需要满足大量并发、同时访问的情况下，多线程的 Java 服务器端系统将着力于解决单线程系统的速度和效率瓶颈。

第 15 章的内容是 Java 的常见通信技术的实现，如 Socket 通信方式、UDP 通信等。

2. 如何阅读本书

阅读本书，并不需要预先掌握 Java 的基本知识，因为在书中已带有 Java 语言的基础内容。本书的适用面较广，初、中、高级读者均可阅读。本书可作为大中专院校计算机专业的 Java 课程教材，也可作为 Java 培训班的教材，亦方便对 J2EE 感兴趣的读者自学使用。

对 Java 比较熟悉或已有一定的 J2EE 基础的读者可跳过第 1 章、第 2 章的学习；已了解 JSP 开发的读者可跳过第 3 章的学习，其他章节可根据需要选读。如果对 JSP 感兴趣的读者，可参考我的 JSP 方面的著作——《JSP 网络编程从基础到实践》。

随书附带的光盘中包括本书用到的所有开源的软件以及全书实例的源代码，以供参考学习。本书使用的 J2EE 应用中间件和 Web 中间件是 Weblogic 8.1，Java 开发工具是 Eclipse 3.0，数据库是 SQL Server 2000。其实，如果没有专门的 Java 开发工具，用记事本也可以编写程序，使用工具的目的只是为了提高开发的效率；数据库的品种也可以不用 SQL Server，因为 JDBC 已经为程序员屏蔽了这些差异。当然，建议做数据库操作时使用标准的 SQL 语句。

本书的第 3、8、9、11~13 章由湖南现代物流职业技术学院高级工程师邓子云编写；第 4、10、14、15 章由深圳美亚通科技有限公司技术总监肖峰编写；第 1、2、5~7 章由长沙民政学院谢英辉老师编写；湖南科技职业技术学院的段玲玲老师参与了第 5、9 章的部分内容的编写工作，并作了大量的资料整理工作，全书由邓子云统稿。我们在繁重的项目开发任务和教学任务中抽出时间来编写此书，目的就是想要奉献给读者一本通俗易懂的、理论与实践紧密结合的 J2EE 书籍。

由于作者的水平有限，书中的错误和不妥之处在所难免，敬请读者批评指正，我们的 E-mail 如下：

dengziyun@126.com（邓子云）、xfygi@126.com（肖峰）、xie-ying-hui@163.com（谢英辉）。

有关本书的意见和反馈，读者可在希赛网（<http://www.csai.cn>）社区“书评在线”版块中的“希赛技术/管理图书”栏目与作者进行交流。

邓子云

2007 年 3 月于长沙

第 1 章 概述	1	1.8 习题	28
1.1 Java 简介	1	第 2 章 Java 语言基础	29
1.1.1 Java 的由来	1	2.1 数据类型	29
1.1.2 Java 的定义	1	2.1.1 整型	29
1.1.3 Java 的特点	2	2.1.2 浮点型	30
1.1.4 Java 与 C/C++ 语言	2	2.1.3 布尔型	30
1.1.5 Java 与 Internet	3	2.1.4 字符型	30
1.2 J2EE 框架	3	2.1.5 字符串	31
1.2.1 MVC 框架	4	2.2 变量与常量	32
1.2.2 Struts 框架	4	2.2.1 变量	32
1.2.3 Spring MVC 框架	5	2.2.2 常量	33
1.3 开发工具	5	2.3 运算符和表达式	34
1.3.1 JDK 介绍	5	2.3.1 运算符	34
1.3.2 Eclipse 介绍	5	2.3.2 表达式	37
1.3.3 BEA Weblogic 介绍	6	2.4 程序控制	38
1.3.4 Lomboz 介绍	7	2.4.1 真和假	38
1.3.5 EasyStruts 介绍	8	2.4.2 if-else 语句	38
1.3.6 Struts 介绍	8	2.4.3 switch 语句	39
1.4 开发环境的安装与配置	8	2.4.4 while 语句	40
1.4.1 JDK 的安装与配置	8	2.4.5 do-while 语句	41
1.4.2 SQL Server 2000 的安装 与配置	9	2.4.6 for 语句	41
1.4.3 Weblogic 8.1 的安装与配置	10	2.4.7 转移语句	42
1.4.4 Eclipse 及其插件的安装	16	2.4.8 多重循环	43
1.5 Eclipse 使用指南	17	2.5 数组	43
1.5.1 文件存放	17	2.5.1 一维数组	44
1.5.2 Eclipse 开发环境	19	2.5.2 多维数组	46
1.5.3 编译、调试与运行	21	2.6 面向对象	48
1.5.4 Eclipse 插件使用	23	2.6.1 类	48
1.5.5 Eclipse 中程序的版本管理	23	2.6.2 对象	48
1.5.6 Eclipse 键盘快捷键的使用	24	2.6.3 封装	48
1.6 Java 程序	27	2.6.4 类的创建	49
1.7 小结	28	2.6.5 对象的创建与使用	52

2.6.6 类的继承	53	4.2 目录及文件操作	114
2.6.7 类的多态性	57	4.2.1 java.io.File 类	114
2.6.8 接口	58	4.2.2 用 File 类进行目录遍历	116
2.6.9 内部类	60	4.3 二进制文件读写	117
2.6.10 包	63	4.3.1 流介绍	117
2.7 小结	66	4.3.2 InputStream 和 Output- Stream	118
2.8 习题	66	4.3.3 字节流文件读取	119
第 3 章 JSP 开发	67	4.3.4 字节流文件写入	121
3.1 JSP 概述	67	4.3.5 格式化字节流文件读写	122
3.1.1 B/S 开发模式	67	4.4 文本文件读写	126
3.1.2 JSP 运行原理	68	4.4.1 Reader 和 Writer	126
3.1.3 编写 JSP 代码的方法	68	4.4.2 字符流文件读取	127
3.2 JSP 指令与动作指令	73	4.4.3 字符流文件写入	129
3.2.1 JSP 指令	73	4.4.4 文本行处理	131
3.2.2 JSP 动作指令	74	4.5 小结	132
3.3 JSP 内置对象	79	4.6 习题	133
3.3.1 request	80	第 5 章 数据库操作	134
3.3.2 response	83	5.1 SQL 基础	134
3.3.3 session	85	5.1.1 表操作	134
3.3.4 application	86	5.1.2 查询语句	139
3.3.5 out	87	5.1.3 插入、更新与删除语句	141
3.3.6 Cookie	88	5.1.4 存储过程	142
3.4 JavaBean 开发	90	5.2 JDBC 基础	143
3.4.1 编写 JavaBean	91	5.2.1 JDBC 包	143
3.4.2 编译和部署 JavaBean	93	5.2.2 JDBC API 应用结构图	144
3.5 Servlet 技术	96	5.2.3 JDBC 驱动程序	144
3.5.1 什么是 Servlet	96	5.3 查询数据库	145
3.5.2 Servlet 的工作原理	96	5.3.1 建立数据库	145
3.5.3 编译与部署 Servlet	97	5.3.2 初始化数据库	146
3.6 JSTL 技术	102	5.3.3 访问数据库	147
3.6.1 JSTL 技术概述	102	5.4 更新数据库	152
3.6.2 安装 JSTL	103	5.4.1 Statement 对象	152
3.6.3 核心标签库	103	5.4.2 更新数据表	152
3.7 小结	112	5.4.3 可更新结果集	154
3.8 习题	113	5.5 调用存储过程	155
第 4 章 Java 文件处理	114	5.6 事务处理	157
4.1 Java 的 I/O 类介绍	114	5.6.1 数据库事务	158

5.6.2 执行一个数据库事务	158	7.1.2 分布式对象应用程序	191
5.6.3 在事务里使用保存点	160	7.1.3 术语的定义	191
5.6.4 将 SQL 语句成批放入一个 事务中	161	7.1.4 分布式和非分布式模型的 比较	192
5.7 连接池技术	163	7.1.5 RMI 相关的接口和类	192
5.7.1 理解数据源	163	7.1.6 远程方法调用中的线程 使用	194
5.7.2 理解连接池	164	7.1.7 远程对象的垃圾收集	194
5.7.3 配置连接池和数据源	164	7.1.8 动态类加载	195
5.8 Web 查询中的分页处理	166	7.2 一个简单 RMI 会话实现	196
5.8.1 通过存储过程完成分页 处理	167	7.2.1 远程接口	196
5.8.2 通过 Java 完成分页处理	169	7.2.2 实现远程接口的类	197
5.8.3 使用【上一页】或【下一页】 按钮进行翻页	172	7.2.3 运行 rmic 编译器	200
5.9 小结	174	7.2.4 创建安全策略	200
5.10 习题	174	7.2.5 启动 RMI 注册表	201
		7.2.6 RMI 客户程序	202
第 6 章 JNDI 开发	175	7.3 远程对象激活	203
6.1 什么是 JNDI	175	7.4 带有回调的 RMI 会话	204
6.1.1 命名服务	175	7.4.1 服务器程序	204
6.1.2 目录服务	176	7.4.2 客户程序	206
6.1.3 LDAP 的介绍	176	7.4.3 运行程序	207
6.2 使用 JNDI	179	7.5 基于 IIOP 的 RMI	208
6.2.1 JNDI 服务提供者	179	7.5.1 服务器程序	208
6.2.2 JNDI 的包	180	7.5.2 客户端程序	210
6.2.3 常用的 JNDI 操作	180	7.5.3 运行程序	211
6.2.4 JNDI 操作步骤	181	7.6 JNDI 与 LDAP 和 RMI	212
6.2.5 JNDI 允许存储的对象 类型	181	7.7 小结	215
6.2.6 JNDI 存储查询串行化的 Java 对象	181	7.8 习题	216
6.2.7 JNDI 查询修改 LDAP 目录 条目	185	第 8 章 XML 操作	217
6.3 小结	188	8.1 XML 概述	217
6.4 习题	189	8.2 XML 基础知识	217
第 7 章 Java RMI 开发	190	8.2.1 如何编写 XML 文档	217
7.1 RMI	190	8.2.2 XML 文档的结构	218
7.1.1 RMI 的优点	190	8.2.3 DTD	220
		8.3 使用 DOM 接口读写 XML	224
		8.3.1 DOM 相关类	225
		8.3.2 读取 XML 文档	228
		8.3.3 修改 XML 文档	231

8.4 使用 SAX 接口读写 XML	234
8.4.1 SAX 相关类	235
8.4.2 SAX 解析 XML 文档的过程	236
8.4.3 读取 XML 文档	236
8.5 使用 SAX 还是 DOM	239
8.6 小结	239
8.7 习题	239

第 9 章 EJB 编程

9.1 EJB 概述	240
9.1.1 EJB 分类	240
9.1.2 EJB 应用模型	241
9.1.3 开发 EJB 的步骤	241
9.2 搭建 EJB 开发环境	242
9.3 Lomboz 使用指南	242
9.3.1 Eclipse 开发环境配置	242
9.3.2 建立一个 J2EE 工程	244
9.3.3 建立一个 J2EE 模块	244
9.3.4 建立一个 EAR 模块	246
9.3.5 建立一个 EJB	247
9.4 会话 EJB 开发	248
9.4.1 无状态会话 EJB	248
9.4.2 有状态会话 EJB	267
9.5 实体 EJB 开发	279
9.5.1 实体 EJB 概述	279
9.5.2 何时使用实体 EJB	280
9.5.3 相关的接口与类	280
9.5.4 主键类	305
9.5.5 Bean 管理数据关系	306
9.5.6 容器管理数据关系	343
9.5.7 EJB-QL	355
9.6 EJB 的环境与资源配置	356
9.6.1 环境变量	356
9.6.2 引用 EJB	356
9.6.3 引用数据库资源	358
9.6.4 引用邮件资源	359
9.6.5 引用 URL 资源	360
9.7 EJB 的事务管理	361

9.7.1 Bean 管理事务	361
9.7.2 容器管理事务	375
9.8 小结	378
9.9 习题	379

第 10 章 Web Services 开发

10.1 了解 Web Services	380
10.2 Web Services 技术框架	380
10.3 SOAP 协议介绍	381
10.3.1 SOAP 协议组成	381
10.3.2 SOAP 消息示例	382
10.3.3 协议绑定	383
10.3.4 SOAP 调用	384
10.4 J2EE 与 Web Services	386
10.4.1 Web Services for J2EE	386
10.4.2 JAX-RPC	387
10.4.3 JAXR	388
10.4.4 JAXM	388
10.4.5 SAAJ	388
10.4.6 JAXP	389
10.4.7 JAXB	389
10.5 Weblogic 与 Web Services	389
10.5.1 用自定义 ant 任务装配 Web 服务	389
10.5.2 servicegen 使用说明	390
10.5.3 用 Java 类实现 Web 服务	391
10.5.4 用 EJB 实现 Web 服务	394
10.5.5 编写 Stub 方式客户端	398
10.5.6 动态代理方式客户端	400
10.5.7 DII 方式客户端	402
10.5.8 Weblogic 的 UDDI 服务	403
10.5.9 客户端 UDDI API	405
10.5.10 编写 UDDI 发布程序	405
10.5.11 编写 UDDI 查询程序	411
10.6 小结	413
10.7 习题	414

第 11 章 JMS 与消息 EJB

11.1 JMS 基本概念	415
---------------------	-----

11.1.1 同步与异步	415	12.3.1 Session	444
11.1.2 JMS	416	12.3.2 InternetAddress	445
11.1.3 消息	416	12.3.3 MimeMessage	446
11.1.4 消息的生产者与消费者	418	12.3.4 Transport	448
11.1.5 点对点的消息传递模型	418	12.3.5 Store	448
11.1.6 发布/订阅消息传递模型	418	12.3.6 Folder	448
11.1.7 连接工厂与连接	418	12.4 发送邮件	449
11.1.8 会话	419	12.4.1 发送文本格式的邮件	449
11.2 配置 JMS	419	12.4.2 发送 HTML 邮件	452
11.2.1 创建连接工厂	419	12.4.3 发送带附件的邮件	454
11.2.2 定义文件存储	420	12.5 接收邮件	456
11.2.3 定义数据库存储	421	12.6 删除邮件	459
11.2.4 定义 JMS 服务器	422	12.7 小结	460
11.2.5 在 JMS 服务器中创建消息队列和主题	423	12.8 习题	460
11.3 点对点模型的实现	423	第 13 章 J2EE 常用安全技术	461
11.3.1 发送消息到队列	424	13.1 加密与解密技术	461
11.3.2 从队列同步接收消息	426	13.1.1 恺撒密码	461
11.3.3 从队列异步接收消息	429	13.1.2 对称加密和解密	463
11.4 发布订阅模型的实现	431	13.1.3 基于密码的加密和解密	465
11.4.1 发布消息到主题	431	13.1.4 基于输入输出流作加密和解密	469
11.4.2 从主题同步接收消息	433	13.1.5 非对称加密和解密	471
11.4.3 从主题异步接收消息	435	13.1.6 数字信封技术	475
11.5 过滤与浏览消息	437	13.2 消息摘要和签名	480
11.5.1 过滤消息	437	13.2.1 消息摘要	480
11.5.2 浏览消息	437	13.2.2 消息验证码	482
11.6 消息 EJB 开发	438	13.2.3 数字签名与验签	483
11.6.1 MessageDrivenBean 接口	438	13.3 数字证书	486
11.6.2 编写消息 EJB	439	13.3.1 生成数字证书	486
11.7 小结	442	13.3.2 读取数字证书	490
11.8 习题	442	13.3.3 签发数字证书	491
第 12 章 JavaMail 开发	443	13.4 小结	495
12.1 JavaMail 基础	443	13.5 习题	495
12.1.1 邮件协议	443	第 14 章 Java 多线程技术	496
12.1.2 JavaMail 概述	443	14.1 什么是多线程	496
12.2 安装与配置 JavaMail	444	14.2 线程类 Thread	497
12.3 相关的类介绍	444	14.3 创建并启动线程	498

14.4	线程的状态	500	15.2.6	通信中的编码与字节 顺序	523
14.5	停止线程	501	15.3	非阻塞式 Socket 通信	525
14.6	线程的同步	503	15.3.1	java.nio 包介绍	525
14.7	线程的阻塞与唤醒	505	15.3.2	SocketChannel 和 Server- SocketChannel	527
14.8	线程池	506	15.3.3	用 nio 实现阻塞式通信	528
14.8.1	为什么需要线程池	506	15.3.4	Selector 与非阻塞	531
14.8.2	建立线程池的方法	507	15.3.5	非阻塞客户端程序	533
14.8.3	一个简单的线程池	507	15.3.6	非阻塞服务器	536
14.9	小结	510	15.4	UDP 通信	539
14.10	习题	510	15.4.1	使用 Java 操作 UDP	539
第 15 章	Java 通信技术	511	15.4.2	UDP 客户端	541
15.1	IP 地址处理	511	15.4.3	UDP 服务端	542
15.2	Socket 通信	513	15.5	小结	543
15.2.1	相关类介绍	513	15.6	习题	543
15.2.2	Socket 客户端程序	514	参考文献	544	
15.2.3	Socket 服务端程序	517			
15.2.4	多线程 Socket 服务器	520			
15.2.5	线程池应用	523			

第1章 概 述

美国硅谷有一句行话，每 10~15 年有一次轮回。最近的一次轮回就是从 Java 开始的。

Java 是一种由 Sun 公司开发而成的新一代编程语言。使用它可在不同种机器、不同种操作系统平台的网络环境中开发软件。不论用户使用的是哪一种万维网浏览器、哪一种计算机、哪一种操作系统，只要浏览器上面注明了“支持 Java”，用户就可以看到生动的主页。Java 正在逐步成为 Internet 应用的主要开发语言。它彻底改变了应用软件的开发模式，带来了自 PC 机以来的又一次技术革命，为迅速发展的信息世界增添了新的活力。

Java 从 1995 年开始在计算机业界就受到了高度重视，特别是在跨平台性方面，是 Internet 和多媒体（multimedia）等相关技术与软件产品所无法相比的。Java 作为一种软件开发技术和编程语言，在网络开发方面具有独特的优势。

本章主要介绍的内容包括：Java 的由来、定义和特点；J2EE 框架介绍；在 J2EE 企业级应用项目方面本书中采用的开发工具的介绍；本书中开发环境的安装与配置；在 Eclipse 中运行程序的操作介绍。

1.1 Java 简介

1.1.1 Java 的由来

Java 语言诞生于 1991 年，起初被称为 OAK 语言，工程师们最初的目的只是为了开发一种独立于平台的软件技术。在 Java 出现以前，Internet 上的信息内容都是一些乏味死板的 HTML 文档，在 Web 中看不到交互式的内容，用户和技术人员很希望能够在 Web 上创建一个无须考虑软硬件平台就可以执行的并且具有极大安全保障的应用程序。对于用户的这种要求，传统的编程语言显得无能为力，而 Sun 公司的工程师敏锐地察觉到了这一点，从 1994 年起，他们开始将 OAK 技术应用于 Web 上，并且开发出了 HotJava 的第一个版本。Sun 公司于 1995 年正式以 Java 这个名字推出。当时由于国际互联网的飞速发展，这种功能强大而且新颖的编程语言在全球信息网舞台中成了一颗闪耀的明星。

1.1.2 Java 的定义

Java 是一种简单、面向对象、分布式、健壮、安全、可移植、性能优异、多线程、动态的语言。通常所说的 Java 有 3 层含义：首先是指一种编程语言，同时又是一种开发环境和一种应用环境。

Java 的开发环境有几种不同的版本，如 Sun 公司的 Java Development Kit，简称 JDK。

后来微软公司推出了支持 Java 规范的 Microsoft Visual J++ 的 Java 开发环境, 简称 VJ++。不过, 自从 C# 诞生后, Microsoft 就没有在 VJ++ 方面继续发展了。

1.1.3 Java 的特点

1. 平台无关性

平台无关性是指 Java 能够运行在不同的平台。Java 引进虚拟机原理, 并使用虚拟机实现不同平台的 Java 接口。使用 Java 编写的程序能在世界范围内共享。Java 的数据类型与机器无关, Java 虚拟机 (Java Virtual Machine) 是建立在硬件和操作系统之上, 实现 Java 二进制代码的执行功能, 提供基于不同平台的相同接口, 以屏蔽不同平台之间的差异。

2. 安全性

Java 的编程方法类似于 C++, 学习过 C++ 的读者能很快上手 Java。Java 舍弃了 C++ 的指针, 不支持对存储器地址的直接操作。程序运行时, 内存由操作系统分配, 这样可以避免病毒通过指针侵入系统。Java 对程序提供了安全管理器, 防止程序的非法访问。

3. 面向对象

Java 吸取了 C++ 面向对象的概念, 将数据封装于类中, 利用类的优点, 实现了程序的简洁性和可维护性。类的封装性、继承性等有关对象的特性, 使程序代码只需一次编译, 然后通过上述特性可以反复利用。程序员只需把主要精力放在类和接口的设计和應用上。Java 提供了众多一般对象的类, 通过继承即可使用父类的方法。在 Java 中, 类的继承关系是单一的而非多重的, 一个子类只有一个父类, 子类的父类又可以有一个父类。Java 提供的 Object 类及其子类的继承关系如同一棵倒立的树形, 根类为 Object 类。Object 类功能强大, 程序员经常会使用到它及其派生的子类。

4. 分布式

Java 建立在扩展 TCP/IP 网络平台上。库函数提供了用 HTTP 和 FTP 协议传送和接收信息的方法, 这使得程序员使用网络上的文件和使用本机的文件一样容易。

5. 健壮性

Java 致力于检查程序在编译和运行时的错误。类型检查可以帮助程序员检查出许多开发早期出现的错误。Java 本身并不操作内存, 这样可以减少内存出错的可能性。Java 还实现了真数组, 避免了覆盖数据的可能。这些功能特征大大提高了开发 Java 应用程序的周期。Java 检测包括: Null 指针检测、数组边界检测、异常出口检测、Byte code 校验等。

1.1.4 Java 与 C/C++ 语言

Java 提供了一个功能强大语言的所有功能, 但几乎没有一点含混特征。Java 去掉了 C++ 语言的许多功能, 这使得 Java 的语言功能很精炼, 同时 Java 增加了一些很有用的功

能,如自动收集碎片功能等。

Java 去掉了 C 和 C++ 中的指针运算、结构、Typedef 功能、#define 功能、释放内存功能等,这使得出错的概率降低了 50%,而且 Java 很小,整个解释器只需 215K 的 RAM。

面向对象:Java 实现了 C++ 的基本面向对象技术并有一些增强,为了语言简单,删除了一些功能。Java 处理数据方式和用对象接口处理对象数据方式一样。

1.1.5 Java 与 Internet

早期的万维网仅仅可以传送文本和图片。Java 的出现实现了动态的页面效果,这是一次伟大的变革。

Java 并不是仅仅为 Internet、万维网而设计的,它也可以用来编写独立的应用程序。Java 是一种面向对象语言,它类似于 C++ 语言,所以已熟练掌握 C++ 语言的编程人员,再学习 Java 语言就显得很容易。实际上有两种 Java 程序:一种 Java 应用程序是一个完整的程序,如 Web 浏览器。另一种 Java 小应用程序是运行于 Web 浏览器中的一个程序。

Java 程序和它的浏览器 HotJava,提供了可让读者的浏览器运行程序的方法。读者能从自己的浏览器里直接播放声音,还可以播放页面里的动画,此外,Java 还能告诉读者的浏览器怎样处理新的类型文件。当读者能在 2400baud 线上传输视频图像时,HotJava 将能显示这些视频。

当今 Internet 的一大发展趋势是电子商务,而 Internet 的安全问题是必须解决的问题,通常大的部门应设置防火墙,阻止非法侵入。

电子商务是当今的热门话题,然而传统的编程语言难以胜任电子商务系统。电子商务要求程序代码具有基本的要求:安全、可靠,同时还要求能与运行于不同平台机器的全世界客户开展业务。Java 以其强安全性、平台无关性、硬件结构无关性、语言简洁和面向对象的特性,在网络编程语言中占据无可比拟的优势,成为开发电子商务系统的首选语言。

1.2 J2EE 框架

随着软件系统的规模和复杂性的增加,软件体系结构的选择成为很重要的因素,三层客户/服务器体系结构为企业资源规划的整合提供了良好的框架。

J2EE (Java 2 Enterprise Edition, Java2 企业版)为中间件领域思想的统一发挥了很大的作用。比如,J2EE 为分布式事务管理、目录服务和消息服务提供了一套标准的编程接口。J2EE 的基础——Java2 标准版 (J2SE),成功地为 Java 提供了一套访问关系数据库的标准。

分布式企业应用软件结构复杂,涉及多种技术,这对设计开发人员提出了很高的要求。在这种情况下,运用框架,也就是运用可复用的设计方案进行软件的设计开发十分必要。框架适合在复杂的 J2EE 开发中使用,它可以为开发者提供一个简单易用模型的程序结构。在好的框架下,开发者只需要写少量的应用程序代码,而不需要直接接触底层的 API;经过良好设计的框架可以为程序提供清晰的结构并且提高程序的内聚性;清

晰的结构使得其他开发人员可以更易于加入项目；一个易于使用的框架可以通过一些例子和文档为用户提供最佳实践，采用成熟的框架的代码比程序员自己编写的代码更容易测试。

1.2.1 MVC 框架

MVC (Model-View-Controller, 模型-视图-控制器) 框架已被证明是一种成功的软件设计模式。MVC 模式是 Xerox PARC 公司在 20 世纪 80 年代为编程语言 Smalltalk-80 所设计的一种软件设计模式，至今仍被广泛使用，并被推荐为 Sun 公司 J2EE 平台的设计框架。MVC 框架可以将业务逻辑 (JavaBean 和 EJB 组件)、控制器逻辑 (Servlet/Struts 动作)、表示层 (JSP、XML/XSLT) 清晰地分离开来。MVC 将程序代码简单地划分为下面 3 个部分，分别对应三层体系系统结构的客户显示层、业务逻辑层、数据层，其结构图如图 1-1 所示。

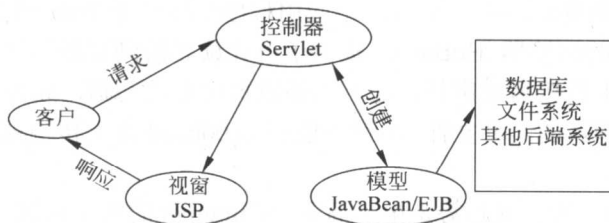


图 1-1 MVC 模式框架

- 视图：也叫客户显示层，即负责用户界面显示的代码，显示由控制器提供的模型对象，通常通过 JSP 及标记库来实现，有时也使用 XML 和 XSLT 来实现。
- 控制器：也叫业务逻辑层，即负责应用程序流程的代码，接收用户输入的信息，并调用业务逻辑去创建和更新模型对象，通常使用 Java Servlet 或像 Struts 控制器这样的类来实现。
- 模型：也叫数据库层，由它来与数据库打交道，提供要显示的数据，提供控制器和视图之间交互的规则。通常使用 EJB 或者普通的 Java 对象来实现。

目前基于 MVC 的 J2EE 框架有很多种，比较流行的有 Struts 和 Spring Web MVC 框架。

1.2.2 Struts 框架

Struts 由一组相互协作的类、Servlet 以及 JSP TagLib 组成。基于 Struts 构架的 web 应用程序基本上符合 JSP Model2 的设计标准，可以说是 MVC 设计模式的一种变化类型。Struts 是一个 Web Framework，而不仅仅是一些标记库的组合。但 Struts 也包含了丰富的标记库和独立于该框架工作的实用程序类。

Struts 有自己的控制器，同时整合了其他的一些技术去实现模型层和视图层。在模型层，Struts 可以很容易的与数据访问技术相结合，包括 EJB、JDBC 和 Object Relation Bridge。在视图层，Struts 能够与 JSP、Velocity Templates、XSL 等这些表示层组件相结合。

1.2.3 Spring MVC 框架

Spring Web MVC, 同 Struts 类似, 是基于 MVC 的 Web 框架。在 Spring Web MVC 使能应用中, 能够直接使用 Spring 的 IOC 和 AOP 的功能。借助于它提供的 DispatcherServlet 控制器 (类似于 Struts 中的 `org.apache.struts.action.ActionServlet` 控制器), 能够统一分发 Web 请求。在整个 Spring Web MVC 中, DispatcherServlet 是最为重要的组件之一, 它可以直接在 `Web.xml` 中配置使用。Spring 工作过程如下:

(1) 一旦客户 http 请求到来, DispatcherServlet 将负责分发它。DispatcherServlet 可以认为是 Spring 提供的前端控制器。所有的客户请求都需要经过它的统一分发。

(2) 在 DispatcherServlet 将请求分发给 Spring Controller 之前, 需要借助于 Spring 提供的 HandlerMapping 定位到具体的 Controller。

(3) Spring Controller 将处理来自 DispatcherServlet 的请求。Controller 相当于 Struts 中的 Action, 可以接收 `HttpServletRequest` 和 `HttpServletResponse`。Spring Controller 需要为用户处理请求, 同时实现线程安全和重用性。

(4) 一旦 Controller 处理完用户的请求, 则返回 ModelAndView 对象给 DispatcherServlet 前端控制器。ModelAndView 包含了模型和视图。

(5) DispatcherServlet 返回的视图可以是视图的逻辑名, 此时需要借助于 Spring 的视图解析器 (ViewResolver) 在 Web 应用中查找 View 对象, 从而将结果返回给用户。

1.3 开发工具

1.3.1 JDK 介绍

要运行 Java 程序, 必须安装 JDK (Java development kit), JDK 是整个 Java 的核心, 其中包括 Java 编译器、JVM (Java virtual machine)、大量的 Java 工具以及 Java 基础 API, 可以从 <http://Java.sun.com> 下载 JDK。下面介绍 JDK 的几个重要的命令。

- Java 执行工具, 是启动 JVM (Java 虚拟机) 并执行 class (BYTE CODE, 字节码) 文件的命令。
- Javac 编译器, 由 .java 文件生成 .class 文件。
- Jar Java 压缩打包工具。
- Javadoc 文档生成器。

最后就是 JDK Documentation, 这是 JDK 的联机帮助文档, 是最有用和最重要的学习参考文档。

1.3.2 Eclipse 介绍

Eclipse 是替代 IBM Visual Age for Java (以下简称 IVJ) 的下一代 IDE 开发环境, 但它未来的目标不仅仅是成为专门开发 Java 程序的 IDE 环境。根据 Eclipse 的体系结构,