



精通

*Java* 中间件编程

李华飏 李水根 郭英奎 等编著



中国水利水电出版社  
www.waterpub.com.cn

# 精通 Java 中间件编程

李华飏 李水根 郭英奎 等编著

## 内 容 提 要

本书使用丰富的案例介绍了使用 Java 技术进行中间件编程，它包括如下方面的内容：JSP、Java Servlet、JDBC 数据库开发、使用 JavaMail 开发邮件应用程序、RMI（远程方法调用）、Enterprise JavaBeans 开发、使用 CORBA 以及 Java IDL 进行开发、JNDI 编程、Java XML 编程以及 Java 开发 Web 服务等。

本书的内容均是目前的热点和读者所关注的问题，也包括对很多人来说是悬而未决的难题。书中的许多案例甚至可以不加修改就应用到开发实践中。

本书适合专业的 Java 程序员阅读，也可以作为正在进行 Java 开发的各类程序员的必备参考书。即使是不擅长这一领域的开发人员，通过详细的实例讲述，也可以循序渐进地掌握本书的内容。

### 图书在版编目（CIP）数据

精通 Java 中间件编程/李华飏等编著. —北京：中国水利水电出版社，2003  
ISBN 7-5084-1514-0

I. 精… II. 李… III. Java 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字（2003）第 038571 号

书 名	精通 Java 中间件编程
作 者	李华飏 李水根 郭英奎 等 编著
出版、发行	中国水利水电出版社（北京市三里河路 6 号 100044） 网址：www.waterpub.com.cn E-mail：mchannel@public3.bta.net.cn（万水） sale@waterpub.com.cn
经 售	电话：（010）63202266（总机）、68331835（营销中心）、82562819（万水） 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	787×1092 毫米 16 开本 22.875 印张 514 千字
版 次	2003 年 7 月第一版 2003 年 7 月北京第一次印刷
印 数	0001—4000 册
定 价	34.00 元

凡购买我社图书，如有缺页、倒页、脱页的，本社发行部负责调换

版权所有·侵权必究

## 序

互联网技术的蓬勃发展使得越来越多的企业应用从传统的 C/S（客户/服务器）结构向三层架构转化。中间件最早被定义为是介于应用程序和操作系统之间的软件，它代表一类产品。IT 环境的复杂性使得企业在信息化建设的过程中，需要构建一个能够提供门户、集成、运行和管理、安全、开发、部署等众多功能于一体的应用基础平台，中间件在完成这一任务方面发挥了巨大的作用。而从用户角度看，企业所构建的平台越复杂，就越需要一个具有统一性、简便性和扩展性的方案。

一般说来，中间件产品主要应用了以下几类技术：

- **数据访问技术**：提供直接访问不同数据源的手段，而不必改变应用层的程序，比如大家很熟悉的 ODBC、JDBC、JDO、ADO 这些数据标准接口。
- **基于消息的中间件技术（MOM）**：MOM 提供一个异步消息传送机制。MOM 中间件在所有需要集成的应用系统内，都要安装 MOM 的 API 调用程序，以把消息递交给中间件进行处理。显然，MOM 的优点是：发送和接收方不必在线等待（松耦合）。其缺点是：会发生过载情况。当然随着产品的不断改进，在负载平衡方面会得到进一步改善。提供 MOM 产品的主要有 IBM MQseries、微软的 MSMQ。随着 JMS（Java Message Service）的兴起，MOM 产品的应用也将变得更为广泛。相对于下面讲到的应用服务器，它比较“单纯”——专为整合服务。
- **远程过程调用（RPC, Remote Procedure Call）**：远程过程调用也同 MOM 一样利用客户/服务模式，不同点是 RPC 是同步在线的，即要等待接收方的回复才继续自己的工作。这减少了出错概率，当然等待使速度变慢。
- **交易处理监控器：TP 监控器（Transaction Processing Monitors）**作为一个中间件，是现代应用服务器的雏形。它主要用于监控和管理客户端及多个后台应用之间的交易状况，考虑负载平衡，将客户的请求映射到各个应用。
- **对象请求代理（Object Request Brokers, ORB）**：ORB 是专门替应用程序转交功能组件的“中介”，是业务功能整合层次的整合中间件，它是基于 COM、CORBA 及 Java RMI 等标准组件。
- **应用服务器**：应用服务器是一个软件开发平台，除可以开发独立的应用系统外，服务器也集成了各种整合技术。此外，很多厂商还专门开发了许多特制的适配器或连接器。因此，利用应用服务器做整合平台，也是目前很多企业考虑的选项之一。
- **Portal Server**：Portal Server 往往是提供一种集成的门户服务。通过可扩展的门户框架集成了对企业信息标准访问途经，可以将内部和合作伙伴与他们所需要

的信息连接起来。

SUN 公司的 Sun ONE (Open Net Environment, 开放网络环境) 是主流的 Web 技术体系之一。SUN ONE 以 Java 技术为核心, 包括 J2SE/J2EE/J2ME, 并基于一系列开放和流行标准、技术及协议。SUN ONE 的开放性、可移植性、跨平台和扩展性在进行中间件编程方面得天独厚, 具备无可比拟的优势。

本书介绍了基于 Java 技术的中间件编程, 它包括如下方面的内容: JSP、Java Servlet、JDBC 数据库开发、使用 JavaMail 开发邮件应用程序、RMI (远程方法调用)、Enterprise JavaBeans 开发、使用 CORBA 以及 Java IDL 进行开发、JNDI 编程、Java XML 编程以及 Java 开发 Web 服务等。

本书详细介绍了如何搭建开发和运行环境, 在讲解理论的同时提供了精彩的案例, 读者按照本书的实例, 可以举一反三, 融会贯通, 理解中间件编程的精髓。本书的内容均是目前的热点和读者所关注的问题, 也包括对很多人来说是悬而未决的难题。书中的许多案例甚至可以不加修改就应用到开发实践中。

本书适合专业的 Java 程序员阅读, 也可以作为正在进行 Java 开发的各类程序员的必备参考书。即使是不擅长这一领域的开发人员, 通过详细的实例讲述, 也可以循序渐进地掌握本书的内容。有关本书的技术讨论, 可以发邮件到 [washing\\_lhb@sina.com](mailto:washing_lhb@sina.com)。

本书由李华飏、李水根和郭英奎等编写。在编写过程中, 彭小梅、黎晓冬和吴德明参与了部分工作。IBM 中国有限公司的褚洪峰和中国国际电子商务中心的王福水提供了许多技术上的支持。顾有松教授和中国科学院周剑平博士对本书的结构提供了许多有益的意见。最后, 特别感谢北京万水电子信息有限公司孙春亮总编的大力鼓励和支持。

编者

2003 年 4 月

# 目 录

序

<b>第 1 章 准备上手</b>	<b>1</b>
1.1 为什么选择 Java .....	1
1.2 开发环境的搭建 .....	2
1.3 运行环境的搭建 .....	4
1.3.1 Java 运行环境的搭建 .....	4
1.3.2 Tomcat 的安装 .....	6
1.3.3 WebLogic 的安装 .....	9
1.4 小结 .....	11
<b>第 2 章 JSP 编程范例</b> .....	<b>12</b>
2.1 简单的 JSP 范例：显示一句话 .....	12
2.1.1 实例说明 .....	12
2.1.2 代码分析 .....	13
2.1.3 运行结果 .....	14
2.2 使用表单实例 .....	15
2.2.1 实例说明 .....	15
2.2.2 代码分析 .....	15
2.2.3 运行结果 .....	17
2.3 JSP 处理 cookie .....	17
2.3.1 实例说明 .....	17
2.3.2 代码分析 .....	18
2.3.3 运行结果 .....	19
2.4 SESSION 的管理范例：购物车 .....	20
2.4.1 实例说明 .....	20
2.4.2 编程思路 .....	21
2.4.3 代码分析 .....	21
2.4.4 运行结果 .....	23
2.5 连接数据库并分页显示记录范例 .....	23
2.5.1 实例说明 .....	23
2.5.2 准备工作 .....	24
2.5.3 编程思路 .....	27
2.5.4 代码分析 .....	27

2.5.5	运行结果.....	31
2.6	JSP 实现文件上传范例.....	31
2.6.1	实例说明.....	31
2.6.2	编程思路.....	32
2.6.3	代码分析.....	34
2.6.4	运行结果.....	38
2.7	小结.....	39
<b>第 3 章</b>	<b>Java Servlet 编程范例.....</b>	<b>40</b>
3.1	简单的 Servlet 例子.....	41
3.1.1	实例说明.....	41
3.1.2	代码分析.....	41
3.1.3	运行结果.....	43
3.2	Servlet 与 JSP 之间的通信.....	44
3.2.1	实例说明.....	44
3.2.2	编程思路.....	44
3.2.3	代码分析.....	45
3.2.4	运行结果.....	46
3.3	控制输出流.....	47
3.3.1	实例说明.....	47
3.3.2	编程思路.....	47
3.3.3	代码分析.....	47
3.3.4	运行结果.....	48
3.4	用 Servlet 管理广告条显示.....	49
3.4.1	实例说明.....	49
3.4.2	编程思路.....	49
3.4.3	准备工作.....	49
3.4.4	代码分析.....	50
3.4.5	运行结果.....	60
3.5	小结.....	61
<b>第 4 章</b>	<b>JDBC 数据库编程范例.....</b>	<b>62</b>
4.1	JDBC 简介.....	62
4.1.1	获得连接.....	62
4.1.2	发送 SQL 语句.....	62
4.1.3	返回结果.....	63
4.2	Java 数据库连接范例.....	63
4.2.1	实例说明.....	63
4.2.2	准备工作.....	64

4.2.3	代码分析.....	65
4.2.4	运行结果.....	68
4.3	JavaBean 封装数据库操作范例 .....	68
4.3.1	实例说明.....	68
4.3.2	代码分析.....	68
4.3.3	运行结果.....	71
4.4	数据库连接池.....	71
4.4.1	实例说明.....	71
4.4.2	编程思路.....	72
4.4.3	代码分析.....	72
4.4.4	运行结果.....	75
4.5	Java 开发存储过程.....	76
4.5.1	实例说明.....	76
4.5.2	代码分析.....	77
4.5.3	生成调用.....	79
4.6	使用 Java 开发触发器.....	80
4.6.1	编写代码.....	81
4.6.2	生成调用.....	81
4.7	使用 SQLJ .....	82
4.7.1	什么是 SQLJ.....	83
4.7.2	准备工作.....	85
4.7.3	代码分析.....	85
4.7.4	运行结果.....	87
4.7.5	SQLJ 的要点.....	87
4.8	小结 .....	88
<b>第 5 章</b>	<b>使用 Javamail .....</b>	<b>89</b>
5.1	用 Javamail 发送简单邮件 .....	89
5.1.1	实例说明.....	89
5.1.2	准备工作.....	90
5.1.3	编程思路.....	91
5.1.4	代码分析.....	91
5.1.5	运行结果.....	94
5.2	用 Javamail 发送 HTML 邮件.....	95
5.2.1	编程思路.....	95
5.2.2	代码分析.....	97
5.2.3	运行结果.....	101
5.3	用 Javamail 发送需要 SMTP 认证的邮件.....	102



5.3.1	实例说明.....	102
5.3.2	编程思路.....	103
5.3.3	代码分析.....	103
5.3.4	运行结果.....	107
5.4	用 Javamail 发送带附件的邮件.....	107
5.4.1	实例说明.....	107
5.4.2	编程思路.....	108
5.4.3	代码分析.....	108
5.4.4	运行结果.....	111
5.5	发送电子邮件综合演练.....	113
5.5.1	实例说明.....	113
5.5.2	编程思路.....	113
5.5.3	代码分析.....	113
5.5.4	运行结果.....	127
5.6	用 Javamail 接收邮件列表.....	128
5.6.1	实例说明.....	128
5.6.2	编程思路.....	128
5.6.3	代码分析.....	128
5.6.4	运行结果.....	133
5.7	用 Javamail 接收单封邮件.....	133
5.7.1	实例说明.....	133
5.7.2	编程思路.....	134
5.7.3	代码分析.....	134
5.7.4	运行结果.....	137
5.8	小结.....	137
<b>第 6 章</b>	<b>RMI 编程范例.....</b>	<b>139</b>
6.1	介绍 RMI.....	139
6.2	一个 RMI 会话系统.....	140
6.2.1	构建服务器程序.....	140
6.2.2	构建客户程序.....	142
6.2.3	注册对象并启动服务器和客户程序.....	143
6.3	带有回调的 RMI 会话.....	144
6.3.1	服务器程序.....	144
6.3.2	客户程序.....	146
6.4	远程对象激活.....	147
6.4.1	创建远程接口.....	148
6.4.2	实现远程接口.....	149

6.4.3	注册激活对象程序.....	149
6.4.4	客户程序.....	151
6.4.5	运行程序.....	151
6.5	在 IIOP 上运行 RMI .....	152
6.5.1	服务器端程序.....	152
6.5.2	客户端程序.....	154
6.5.3	运行程序.....	155
6.6	小结 .....	155
<b>第 7 章</b>	<b>EJB 编程范例.....</b>	<b>156</b>
7.1	了解 EJB .....	156
7.1.1	Enterprise Bean 是什么 .....	156
7.1.2	Enterprise Bean 的优点 .....	156
7.1.3	什么时候使用 Enterprise Bean .....	157
7.1.4	Enterprise Bean 的类型 .....	157
7.1.5	EJB 对象 .....	158
7.1.6	RMI 和 EJB 对象 .....	158
7.1.7	远程接口.....	158
7.1.8	Home 接口.....	159
7.1.9	Home 对象.....	159
7.2	Session Bean 概述 .....	160
7.2.1	Session Bean 是什么 .....	160
7.2.2	有状态 Session Bean .....	160
7.2.3	无状态会话 Bean .....	161
7.2.4	SessionBean 接口 .....	161
7.3	无状态 Session Bean 开发示例 .....	163
7.3.1	主接口 (StatelessHelloWorldHome) .....	164
7.3.2	远程接口 (StatelessHelloWorld) .....	164
7.3.3	Bean 实现类 (StatelessHelloWorldImp) .....	165
7.3.4	部署 EJB.....	166
7.3.5	生成访问无状态会话 Bean 的客户端程序.....	170
7.4	有状态 Session Bean 开发示例 .....	171
7.4.1	StatefulFundManagerEJB 的主接口 .....	172
7.4.2	StatefulFundManagerEJB 的远程接口 .....	173
7.4.3	StatefulFundManagerEJB 的实现类 .....	173
7.4.4	部署 EJB .....	175
7.4.5	生成访问有状态 Session Bean 的客户端程序 .....	175
7.5	Entity Bean 开发示例.....	178

7.5.1	Entity bean 和 Session Bean 的比较 .....	179
7.5.2	容器管理的持久性 .....	179
7.5.3	本地接口 .....	180
7.5.4	远程接口 .....	181
7.5.5	Bean 实现类 .....	182
7.5.6	回调方法 .....	187
7.5.7	Bean 管理的持久性 .....	190
7.6	Java 消息服务和消息驱动 Bean .....	197
7.6.1	消息概述 .....	197
7.6.2	消息驱动 Bean .....	197
7.6.3	EJB 和 JMS .....	197
7.6.4	MDB 体系结构 .....	199
7.6.5	JMS 消息接口 .....	200
7.6.6	MDB 客户程序接口 .....	202
7.6.7	点对点消息队列模式 .....	202
7.6.8	发布-订阅消息模式 .....	203
7.6.9	消息驱动 Bean 应用实例 .....	204
7.7	基于 Web 的 EJB 应用程序示例 .....	210
7.7.1	收集需求 .....	210
7.7.2	层的设计 .....	211
7.7.3	生成实体 Bean .....	212
7.7.4	生成会话 Bean .....	225
7.7.5	生成 Web 接口 .....	239
7.7.6	部署应用程序 .....	246
7.8	小结 .....	247
<b>第 8 章</b>	<b>CORBA 以及 JAVA IDL 编程范例 .....</b>	<b>249</b>
8.1	介绍 CORBA .....	249
8.1.1	对象管理组织 (OMG) 简介 .....	249
8.1.2	CORBA 主要版本的发展历程 .....	249
8.1.3	CORBA 体系结构概述 .....	250
8.1.4	CORBA 的主要应用方向及中间件产品介绍 .....	251
8.2	介绍 IDL .....	251
8.2.1	OMG IDL 文件概述 .....	251
8.2.2	JavaIDL 介绍 .....	252
8.3	开发 CORBA 应用的服务器程序 .....	252
8.4	开发 CORBA 应用的客户机程序 .....	255
8.5	编写 CORBA 客户机 APPLET .....	256

8.6	使用 CORBA 范例: Java 和 C++混合编程.....	257
8.6.1	编写 SysProp.idl.....	258
8.6.2	编写 Java 的服务器程序.....	258
8.6.3	编写 Java 的客户机.....	260
8.6.4	编写 C++的 IOR 客户机.....	261
8.6.5	运行程序.....	263
8.7	利用动态调用方式实现分布式应用.....	264
8.7.1	客户端动态调用接口 (DII).....	264
8.7.2	服务对象动态骨架接口 (DSI).....	265
8.7.3	程序举例.....	266
8.8	小结.....	268
<b>第 9 章</b>	<b>JNDI 编程范例.....</b>	<b>269</b>
9.1	介绍 JNDI.....	269
9.2	JNDI 架构.....	272
9.3	利用 JNDI 在网络上搜索资源.....	273
9.4	用 JNDI 进行查找实例.....	275
9.5	小结.....	279
<b>第 10 章</b>	<b>使用 JAVA 进行 XML 编程.....</b>	<b>280</b>
10.1	XML 简介.....	280
10.1.1	XML 与 HTML 的比较.....	280
10.1.2	XML 的优缺点.....	281
10.1.3	XML 的注释.....	283
10.2	DTD 和 Schema.....	283
10.2.1	DTD 简介.....	283
10.2.2	Schema 简介.....	287
10.2.3	Schema 的文件结构.....	288
10.2.4	名域和 Schema 的结合.....	291
10.2.5	使用 Java 解析 XML 文件.....	291
10.3	XML 的样式单.....	292
10.3.1	使用 CSS 样式表.....	292
10.3.2	使用 XSL 样式表.....	295
10.3.3	使用 XSL 的例子.....	296
10.4	DOM 和 SAX.....	302
10.4.1	使用 DOM.....	304
10.4.2	使用 SAX.....	309
10.5	在 JSP 中使用 XML.....	312
10.5.1	实例说明.....	312

10.5.2	编程思路.....	312
10.5.3	代码分析.....	312
10.5.4	运行结果.....	316
10.6	小结.....	317
<b>第 11 章</b>	<b>Java 开发 Web Service.....</b>	<b>318</b>
11.1	什么是 Web Service.....	318
11.2	一个简单的 SOAP 程序.....	319
11.2.1	实例说明.....	319
11.2.2	准备工作.....	320
11.2.3	编写代码.....	321
11.2.4	部署服务.....	321
11.2.5	程序调用.....	323
11.3	SOAP 的信息结构.....	326
11.3.1	SOAP 封套 (Envelope).....	326
11.3.2	SOAP 信息头 (Header).....	327
11.3.3	SOAP 信息体.....	330
11.3.4	SOAP 错误.....	330
11.4	WSDL 语言和 UDDI.....	331
11.4.1	WSDL 服务接口实例.....	337
11.4.2	根据 WSDL 服务接口创建的 UDDI tModel.....	338
11.4.3	WSDL 服务实现示例.....	340
11.4.4	根据 WSDL 服务实现创建的 UDDI 商业服务.....	341
11.4.5	查找 WSDL 服务接口.....	342
11.4.6	查找 WSDL 服务实现描述.....	342
11.4.7	查找 UDDI bindingTemplate.....	342
11.4.8	UDDI bindingTemplate 示例.....	343
11.5	JSP 调用 WEB SERVICE 范例.....	343
11.5.1	实例说明.....	344
11.5.2	代码分析.....	345
11.5.3	运行结果.....	352
11.6	小结.....	352

# 第 1 章 准备上手

## 1.1 为什么选择 Java

自从 1995 年 Java 语言问世以来，它带来的是一场革命，目前 Java 已经成为技术人员的一种时尚。为什么在众多的编程语言中，我们对 Java 情有独钟呢？首先让我们看看现在开发企业用户的 ERP、CRM、电子商务软件，大多都由传统的 C/S (Client/Server) 结构向 B/S (Browser/Server) 结构转型。其中 B/S 结构主要有下面两个优点：

- 易于维护：维护工作总是软件开发部门最大的心病，一个成功的软件的维护成本几乎占总成本的一半以上。而使用 B/S 结构的客户端几乎不需要维护，这样就大大地降低了维护成本。
- 客户端与平台无关性：由于客户端是浏览器，而现在的 OS (操作系统) 都有适合自己的浏览器，而这些浏览器采用的语言标准是统一的，所以开发人员不需要过多地关心客户端环境。

目前，在 B/S 结构中流行的技术主要有 ASP、PHP、JSP 和少数用户仍在使用的 ISAPI、CGI 等技术。后面两种技术由于上手困难、程序代码难写难懂以及调试修改困难，已经很少有人使用了。下面主要介绍一下 ASP、PHP、JSP 这 3 种技术。

ASP 和 ASP.NET 是一种完全基于微软平台的简便易学的技术，但是这种简便性只是限于简单的应用程序。一旦你的使用范围超出它的效能，就不得不使用 COM 组件来扩充 ASP 功能，而 COM 的开发难度就不只是不简便了，而是相当复杂。

PHP 是现在流行的字眼“自由软件”和“开放源代码”的产物，它和 MySQL 数据库完美结合，一样都是简便易学。但是 PHP 由于其过多地考虑了易用性，其代码尤其是变量使用易于产生混乱。如果用它开发大型项目，对程序员的要求将会比较高。同时，PHP 做扩展应用的难度一点也不比 ASP 简单，而是更为复杂。同时，PHP 的运行速度并不能让并发访问量大的用户满意。虽然 Zend 的出现基本上解决了这个问题，但是它并不是免费的。大多数 PHP 用户是使用 Linux、MySQL 平台，在这样一个环境下要向 Zend 付费，感觉总是怪怪的。虽然 PHP 也有 Windows 版本，但还很少有企业级应用会选择在 Windows 平台下使用 PHP。

JSP 是 Sun 公司提出来的解决方案，它是真正的跨平台、实现动态功能的一种技术。JSP、JavaBeans、Servlet 和 EJB (Enterprise Java Bean) 能使你简单方便地构建安全高速的企业级应用系统，而 JDBC (Java 数据库连接, Java Database Connectivity) 提供了各种数据库的统一接口。Java RMI (远程方法调用, Remote Method Invocation) 使 Java 对象的方法可以方便地被远程 Java 程序来调用，以用于开发分布式应用程序。如果调用程序不是 Java 程序

(当然是 Java 程序也可以), Java IDL (接口定义语言 Interface Definition Language) 可以通过 CORBA (普通对象请求代理体系结构, Common Object Request Broker Architecture) 来调用这个 Java 方法。因此, 从发展的眼光来看, JSP 的前途是无可限量的。

Java 在服务器端的应用并不是只局限于 JSP 和 Servlet, 应该说 JSP 和 Servlet 只是 Java 服务器端编程中很好的外部表现形式。使用它们可以方便快捷地创建界面级应用, 而内部逻辑使用 JavaBeans、EJB 等方式完成。另外, 还可以通过使用 Java 来创建包括中间件的开发等各种服务。

## 1.2 开发环境的搭建

要开发一个 JSP 程序, 就必须选择好 JSP 的开发工具。自从 JSP 的技术规范发布以后, 市场上出现了不少的 JSP 开发工具及支持 JSP 的应用程序服务器。下面我们将介绍一些实践中比较常用的开发工具。

现在的开发工具很多, 开发 Java 程序的如 Borland 公司的 Jbuilder, 其他的有 Web 服务器厂商开发的编程工具, 像 BEA 的 Weblogic Workshop、Allaire Jrun Studio, IBM 的 IBM WebSphere Studio 和 VisualAge For Java 等。对于初学者来说, 使用这些工具并没有什么很大的好处, 而且使用工具生成的一些代码有时更让我们费解, 因此建议还是使用一些文本编辑器。现在流行的能识别 Java、JSP 语法的文本编辑器有 EditPlus 和 UltraEdit。Web 的开发离不开界面的设计, 而 HTML 的开发工具大概就只有 Dreamweaver 一枝独秀了。

Dreamweaver MX 的界面如图 1-1 所示, 它的下载地址为<http://www.macromedia.com/>。而 EditPlus 2.0 的界面如图 1-2 所示, 它的下载地址为<http://www.editplus.com>。

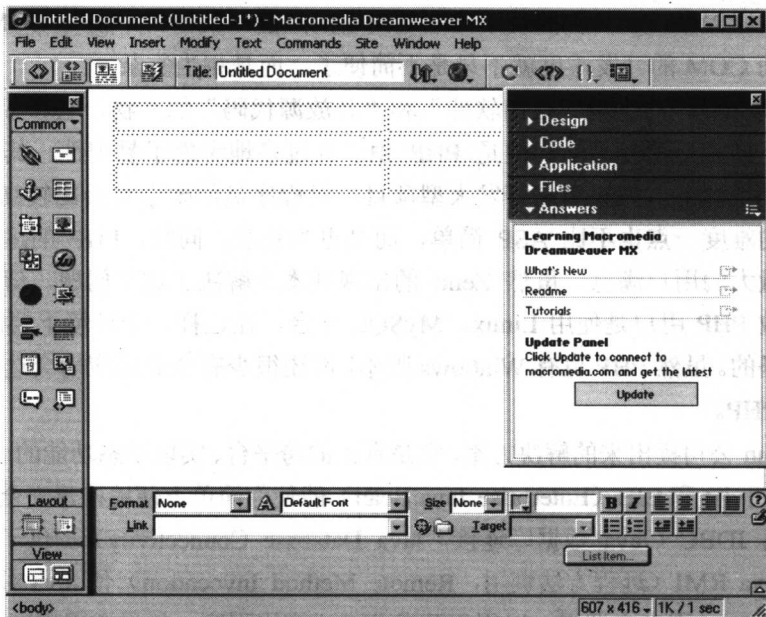


图 1-1 Dreamweaver MX 的界面

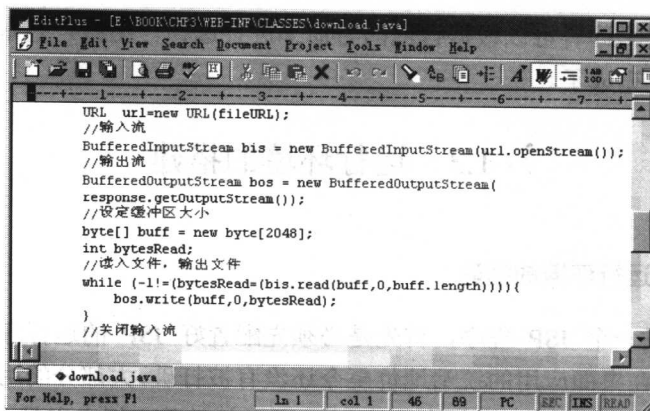


图 1-2 EditPlus 的界面

Dreamweaver 和 EditPlus 都是 Windows 下的应用程序。直接执行它们的安装程序，按照安装向导的指引就可以顺利地完成任务。

JBuilder (最新版本是 7，界面如图 1-3 所示)，它是一种纯 Java 的产品，其中一个最大的优点是提供了一个开发 Java 程序的集成环境，而且功能非常强大。JBuilder 对 EJB 具有良好的支持，而且可以很方便的和主流的 J2EE 服务器相结合，这些后边的章节会给大家详细介绍。但是，由于这个工具是用纯 Java 语言编写的，所以运行起来特别耗内存，并不是一般的机器都可以流畅运行的。JBuilder 对 JSP 的支持不是很理想，如果你只想做 JSP、Servlet 的开发，建议还是使用文本编辑器比较简便。

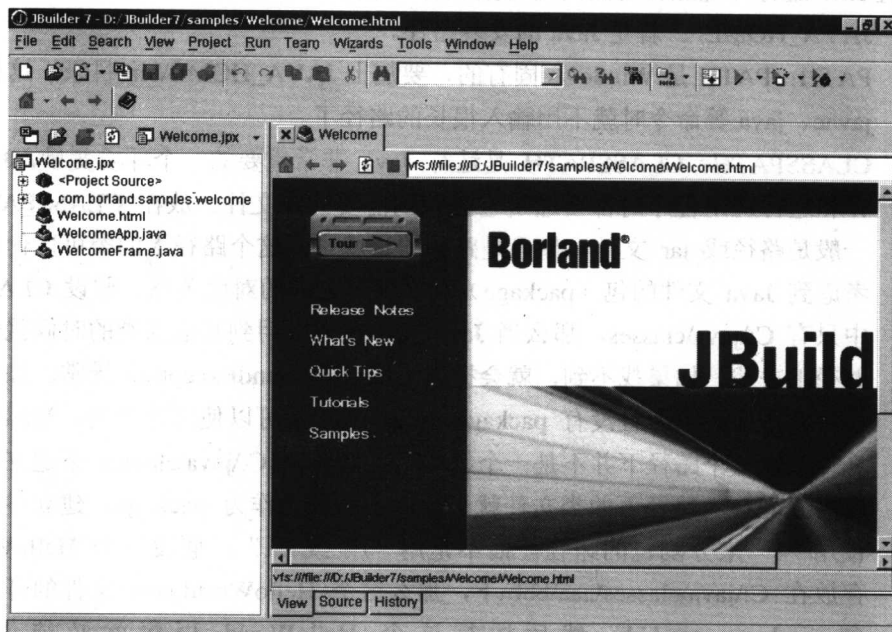


图 1-3 JBuilder 7 的界面

如果使用 JBuilder 或 VisualAge for Java 等集成开发环境 (IDE) 进行开发的话，直接安装该工具即可。因为编辑和编译运行都是集成在工具内部的，因而使用起来会更加方便。



究竟使用哪种方式，读者可以根据自己的习惯来进行选择。至此，基本的 Java 程序的编辑工具环境就搭建完成了。

## 1.3 运行环境的搭建

### 1.3.1 Java 运行环境的搭建

我们要想开发一个 JSP 程序，首先是必须先配置好 JSP 的运行环境。在众多的操作系统中，Windows 在桌面应用的统治地位至今还没有被打破，大部分的用户都是在 Windows 环境下开发程序的。即便是不运行在 Windows 平台的软件，好多也是在 Windows 下开发，然后再上传到 Linux、UNIX 服务器上的。下面，我们就针对在 Windows、Linux 操作系统下配置 JSP 运行环境的问题做详细说明。

要编译和调试运行 Java 程序，首先需要安装 JDK，可以从 Sun 公司的站点 [java.sun.com](http://java.sun.com) 下载 JDK，最新稳定版本是 1.4.1。当然需要根据使用的平台的不同来下载不同的版本。

在 Windows 平台下，下载 `j2sdk-1_4_1-windows-i586.exe` 后运行安装程序，选择安装路径后将会自动完成安装。

然后是环境变量的配置。在不同的操作系统下配置环境变量的方法有些不同，但本质都是一样。它的作用是让应用程序可以方便快捷地找到所需要的路径。

配置 Java 运行环境主要用到的环境变量有 3 个：

- **JAVA\_HOME**：就是 Java 的安装路径。
- **PATH**：PATH 是 Windows 固有的，要加上 `JAVA_HOME\bin` 目录，这样在运行 `javac`、`java` 等命令时就不用输入很长的路径了。
- **CLASSPATH**：CLASSPATH 是运行 Java 非常重要的一个环境变量，Java 在编译和运行应用程序时都要通过它去找到需要的类文件。放在 CLASSPATH 里的一般是路径或 jar 文件，如果是路径的话就是说这个路径下的类可以使用。还要考虑到 Java 文件的包 (package) 和文件夹之间的对应关系。假设 CLASSPATH 中只有 `C:\java\classes`，那么当 Java 文件在需要用到其他的类的时候就会到这个路径下去找。如果找不到，就会抛出 `ClassNotFoundException` 异常。如果在该路径下有需要的类而且没有 package，java 文件就可以使用这个类。当然把许多的文件放在一个路径下并不是一个好主意。需要在 `C:\java\classes` 下建立新的路径的时候，在新路径下的类文件就必须以该路径名作为 package。建立多级目录时使用 “.” 来分割包的路径，而不是用 “/” 或 “\”。假设一个 `HelloWorld.java` 存放在 `C:\java\classes\aaa\bbb\` 下，那么这个 `HelloWorld.java` 文件的开头必须包含 package `aaa.bbb`；然后运行这个 `HelloWorld` 程序就必须使用 `java aaa.bbb>HelloWorld` 命令。CLASSPATH 还有一个很重要的元素就是 “.”。没有人喜欢总是要到 CLASSPATH 包含的路径下去建立文件，当你建立一个正确的 `HelloWorld.java` 且编译成功后却不能运行，或者与本文件在同一路径下编译程序。