

精

通

# Java 核心技术

刘晓华 等编著  
飞思科技产品研发中心 监制



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

精通系列

# 精通 Java 核心技术

刘晓华等 编著

飞思科技产品研发中心 监制



B1283058

电子工业出版社

**Publishing House of Electronics Industry**

北京·BEIJING

## 内 容 简 介

本书力图用深刻的理论和丰富的实例揭示 Java 企业版中的最新核心技术,全面扫除 Java 领域的学习障碍。全书分为两篇,共 21 章,包括 GUI 程序设计、线程、本地方法调用、远程方法调用、命名与目录服务、JavaBeans 技术、JSP 技术、网络编程、EJB 技术、事务处理、基于 JMS 的消息处理、JavaMail 技术、基于 JAAS 的安全技术、Web 服务技术等专业开发者最关心的 Java 技术内容,使读者对整个 J2EE 架构有一个全面而透彻的认识。书中的例程对实际的应用开发有非常强的借鉴意义。飞思在线 <http://www.fecit.com.cn> 提供全书示例代码下载。

本书适合于专业和准专业的 Java 程序员阅读,也可作为正在转向 Java 开发的各类程序员的必备 Java 参考书。

未经许可,不得以任何方式复制或抄袭本书的部分或全部内容。  
版权所有,侵权必究。

### 图书在版编目(CIP)数据

精通 Java 核心技术 / 刘晓华等编著. —北京: 电子工业出版社, 2003.8

(精通系列)

ISBN 7-5053-8892-4

I. 精... 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 058517 号

责任编辑: 郭 晶 杨 鸽

印 刷: 北京市增富印刷有限责任公司

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京海淀区万寿路 173 信箱 邮编: 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 33.5 字数: 857.6 千字

版 次: 2003 年 8 月第 1 版 2003 年 8 月第 1 次印刷

印 数: 6 000 册 定价: 49.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系调换。联系电话:(010) 68279077

# 前 言

## 关于本丛书

“精通”系列是电子工业出版社经典的技术精品丛书，一直受到广大读者，特别是计算机专业技术人员的关注。在这些专业人士的支持和鼓励下，“精通”系列已经成为一个标杆，反映出目前国内外最新计算机技术的动态和发展方向。

“精通”系列中的每部著作完全是作者的呕心沥血之作，代表了作者在该领域的最高成就，集成了作者多年的理论和实践经验，凸显了作者为计算机技术的发展做总结和展望的写作初衷。丛书的作者由著名高校的学科带头人、国际上知名的教授学者、权威的计算机专业人士和业界的集大成者组成。他们的知识结构、理论和实践体系有着突出的卓越之处：

- 站在技术的最前沿。
- 有最深刻的理论基础。
- 实践环境具有广泛的代表性和权威性。
- 结论的指导价值。

因此，这样雄厚的写作班子保证了本系列丛书的高层次、高质量和高品质，也足以满足国内读者的高品位、高需求和高要求。除了作者之外，审校者同样实力超群，他们从理论的角度、读者需求的角度、技术成熟度的角度等不同的侧面，为作者提出了大量的参考意见和修改建议，使得每部著作的结构更坚实，体系更完整，理论更完善，针对性更强。

电子工业出版社计算机研发部本着服务于读者、服务于科技的精神，在选题上精益求精，综合考虑和平衡了目前技术上的热点、未来发展的重点及潜在读者需求的卖点等多方面因素，精心推出此“精通”系列，并将不断进行补充。

## 关于本书

随着电子商务的迅猛发展，社会各个部门之间越来越需要依赖于企业级应用程序来共同协作。开发企业级应用程序的关键在于要有一个功能强大而又方便灵活的体系结构。J2EE 为这种需求提供了一种良好的机制，其核心技术搭建起了 Java 企业应用的坚实平台。综观国内的技术发展，广大的 Java 程序开发人员及正在转向 Java 体系进行开发的技术人员虽然已面临这一令人激动和鼓舞的转型期，却苦于没有足够的相关资料和文献，尤其对国内的最新 Java 技术动态和技术现状知之甚少。而图书市场上 Java 的书籍尽管汗牛充栋，但精品罕见，能反映出 J2EE 框架全貌的书籍更是奇缺。

本书力图用深刻的理论和丰富的实例揭示 Java 企业版中的最新核心技术，全面扫除 Java 领域的学习障碍。全书分为两篇，共 21 章，包括 GUI 程序设计、线程、本地方法调用 (JNI)、远程方法调用 (RMI)、命名与目录服务 (JNDI)、JavaBeans 技术、JSP 技术、

网络编程、EJB 技术、事务处理、基于 JMS 的消息处理、JavaMail 技术、基于 JAAS 的安全技术、Web 服务技术等专业开发者最关心的 Java 技术内容，使读者对整个 J2EE 架构有一个全面而透彻的认识。

本书中包含了大量的例程，这些例程对于实际的应用开发有着非常强的借鉴意义。书中的例程主要分为两类。一是完整的可以直接运行的程序。对于这类例程，我们给出了程序源代码、输出结果和程序注解。读者可以在 JCreator、JBuilder 等集成开发环境中编译和运行这些程序。另一类例程是用来说明某个具体问题的子程序或代码段。对于这类例程，我们给出了源代码和程序注解。要运行这些子程序或代码段，必须将它们放入具体的主程序和运行环境中。飞思在线 <http://www.fecit.com.cn> 提供书中示例代码的下载。

本书具有如下特点：

- 丰富的实例，将深奥的技术和简明易懂的编程思想紧密结合。
- 深入剖析 J2EE 整体架构，分析讲解开发企业级应用程序的各个层次。
- 全面介绍 J2EE 核心技术及它们在 J2EE 体系结构中所处的地位。

本书适合于专业和准专业的 Java 程序员阅读，也是正在转向进行 Java 开发的各类程序员的必备 Java 参考书。

本书由飞思科技产品研发中心策划，刘晓华主笔。李海平、蒋恒青、黄凯、李华、孙建设、丁越、由杨、邢玮、陈志雄、韩其昌、罗玉奎参与了本书的写作；吴锋、王树彦、李胜贤、张亦君、魏宏负责收集相关资料；赵文杰、胡刚、崔洪政编写了本书中的部分示例；徐炜、何晓亮对书中的代码进行了测试；范丽洁出色的文字润色工作也使本书增色不少。特别感谢廖彬、陈祥勇、区涛，他们对本选题提出了许多建设性意见。

由于本书涉及的内容十分丰富，加之篇幅、时间所限，书中不足之处，敬请读者批评指正。如果在学习本书的过程中有问题，请发邮件给我们。

我们的联系方式：

咨询电话：(010) 68134545 68131648

答疑邮件：[support@fecit.com.cn](mailto:support@fecit.com.cn)

网 址：<http://www.fecit.com.cn> <http://www.fecit.net>

答 疑：<http://www.fecit.com.cn> 的“问题解答”专区

下 载：<http://www.fecit.com.cn> 的“下载专区”

通用网址：计算机图书、FECIT、飞思教育、飞思科技、飞思

飞思科技产品研发中心

# 目 录

## 第 1 篇 基础篇

第 1 章 J2EE 概述.....	3
1.1 J2EE 的产生背景及其优点.....	3
1.2 J2EE 的体系结构.....	4
1.3 J2EE 技术概览.....	6
1.4 小结.....	8
第 2 章 Java 语言基础.....	9
2.1 标识符和关键字.....	9
2.2 数据类型和变量.....	10
2.3 运算符.....	12
2.3.1 算术运算符.....	12
2.3.2 赋值运算符.....	12
2.3.3 关系运算符.....	13
2.3.4 逻辑运算符.....	13
2.3.5 位运算符.....	14
2.3.6 移位运算符.....	14
2.3.7 条件运算符.....	15
2.3.8 运算符的优先级.....	16
2.4 控制语句.....	16
2.4.1 选择语句.....	16
2.4.2 循环语句.....	19
2.4.3 break 和 continue 语句.....	20
2.5 类和对象.....	21
2.6 专题讨论: 转型.....	24
2.6.1 向上转型.....	24
2.6.2 向下转型.....	26
2.6.3 混合转型.....	27
2.7 专题讨论: 类的动态加载.....	28
2.8 小结.....	30
第 3 章 GUI 程序设计.....	31
3.1 事件处理模型.....	31
3.2 常用组件.....	32
3.2.1 JOptionPane.....	33
3.2.2 JFrame.....	34

3.2.3 JLabel.....	35
3.2.4 JTextField 和 JPasswordField.....	37
3.2.5 JButton.....	39
3.2.6 JCheckBox 和 JRadioButton.....	42
3.2.7 JTextArea.....	47
3.2.8 JMenu.....	48
3.3 键盘和鼠标事件.....	51
3.3.1 实例: 键盘事件.....	51
3.3.2 实例: 鼠标事件.....	53
3.3.3 实例: Adapter 类的使用.....	56
3.4 布局管理器.....	58
3.4.1 实例: FlowLayout 布局.....	59
3.4.2 实例: BorderLayout 布局.....	61
3.4.3 实例: CardLayout 布局.....	65
3.4.4 实例: GridLayout 布局.....	68
3.5 Graphics 和 Java2D.....	69
3.5.1 实例: Graphics 绘图.....	69
3.5.2 实例: Java2D 绘图.....	71
3.6 常用高级 Swing 组件的使用.....	76
3.6.1 JTree 的使用.....	76
3.6.2 JTable 的使用.....	78
3.7 小结.....	80
第 2 篇 技术篇	
第 4 章 异常处理.....	83
4.1 何谓异常.....	83
4.2 使用 try 和 catch 捕获异常.....	84
4.2.1 实例: 异常的捕获.....	85
4.3 使用 throw 和 throws 引发异常.....	86
4.3.1 实例: throw 语句的 使用.....	86
4.3.2 实例: throws 语句的 使用.....	87

4.4	finally 关键字.....	89
4.4.1	实例: finally 关键字的用法.....	89
4.5	getMessage 和 printStackTrace 方法.....	90
4.5.1	实例: 输出异常信息.....	91
4.6	创建自己的异常类.....	92
4.6.1	实例: 用自定义异常类控制端口访问.....	92
4.7	小结.....	94
<b>第 5 章</b>	<b>流和序列化.....</b>	<b>95</b>
5.1	字节流.....	95
5.1.1	实例: 使用字节流进行 I/O 操作.....	97
5.2	字符流.....	99
5.2.1	实例: 使用 FileReader 类进行文件操作.....	100
5.3	随机访问文件.....	102
5.3.1	使用 RandomAccessFile 写文件.....	102
5.3.2	使用 RandomAccessFile 读文件.....	103
5.4	序列化.....	104
5.4.1	实例: 序列化的应用.....	105
5.5	综合实例: 典型的 IO 使用.....	106
5.6	Java NIO 的使用.....	112
5.7	小结.....	115
<b>第 6 章</b>	<b>接口和抽象类.....</b>	<b>117</b>
6.1	接口.....	117
6.1.1	接口的定义.....	117
6.1.2	接口的实现.....	118
6.1.3	接口的扩展.....	120
6.1.4	实例: 接口的使用.....	120
6.2	抽象类.....	122
6.2.1	抽象类的概念.....	122
6.2.2	实例: 使用抽象类计算图形面积.....	123
6.3	小结.....	126

<b>第 7 章</b>	<b>线程.....</b>	<b>127</b>
7.1	何谓线程.....	127
7.2	线程的生命周期.....	128
7.3	Thread 类和 Runnable 接口.....	129
7.3.1	实例: 通过继承 Thread 类实现线程体.....	130
7.3.2	实例: 通过实现 Runnable 接口实现线程体.....	132
7.4	线程优先级与线程的调度.....	134
7.4.1	实例: 优先级的设置对程序运行的影响.....	135
7.5	线程同步.....	137
7.5.1	实例: 用 synchronized 关键字实现线程同步.....	138
7.6	线程间通信.....	140
7.6.1	实例: 用 wait()和 notify()方法实现线程间通信.....	140
7.7	小结.....	142
<b>第 8 章</b>	<b>本地方法调用.....</b>	<b>143</b>
8.1	JNI 概述.....	143
8.2	数据类型的对应.....	144
8.2.1	Java 字符串与 C 字符串.....	144
8.2.2	Java 数组与 C 数组.....	145
8.3	Java 与 C/C++代码的互相调用.....	146
8.3.1	实例: 从 Java 程序调用 C/C++代码.....	146
8.3.2	实例: 从 C/C++程序调用 Java 代码.....	150
8.4	本地方法中的线程和同步.....	156
8.5	小结.....	157
<b>第 9 章</b>	<b>远程方法调用.....</b>	<b>159</b>
9.1	RMI 概述.....	159
9.2	如何创建 RMI 应用程序.....	161
9.3	实例: 用 RMI 构建一个简单的分布式系统.....	162
9.3.1	定义远程接口.....	163
9.3.2	实现远程接口和服务端.....	164
9.3.3	编写客户端程序.....	166

9.3.4	运行程序 .....	167
9.4	RMI 的一些高级话题 .....	168
9.5	小结 .....	170
<b>第 10 章</b>	<b>命名与目录服务 .....</b>	<b>171</b>
10.1	JNDI 的实现原理 .....	171
10.1.1	命名与目录服务 .....	171
10.1.2	JNDI 的体系结构 .....	172
10.2	JNDI 与 LDAP .....	175
10.3	使用 JNDI 访问命名与目录 服务 .....	177
10.3.1	实例: 访问命名服务 .....	177
10.3.2	实例: 访问目录服务 .....	179
10.4	小结 .....	182
<b>第 11 章</b>	<b>JavaBeans 技术 .....</b>	<b>183</b>
11.1	何谓 Bean .....	183
11.2	Bean 的优点 .....	184
11.3	实例: 编写一个简单的 Bean .....	184
11.3.1	编写 Bean 要遵循的规则 .....	185
11.3.2	创建一个简单的 Bean .....	187
11.4	Bean 的属性 .....	194
11.4.1	单值属性 .....	195
11.4.2	关联属性 .....	195
11.4.3	受限属性 .....	196
11.4.4	索引属性 .....	196
11.5	属性编辑器与定制器 .....	196
11.6	java.beans 包 .....	197
11.6.1	BeanInfo 接口 .....	198
11.6.2	SimpleBeanInfo 类 .....	198
11.6.3	FeatureDescription 类 .....	199
11.6.4	MethodDescriptor 类 .....	200
11.6.5	ParameterDescriptor 类 .....	200
11.6.6	PropertyDescriptor 类 .....	200
11.6.7	IndexedPropertyDescriptor 类 .....	201
11.6.8	BeanDescriptor 类 .....	202
11.6.9	EventSetDescriptor 类 .....	202
11.6.10	IntroSpector 类 .....	203
11.7	小结 .....	204

<b>第 12 章</b>	<b>Java 网络编程 .....</b>	<b>205</b>
12.1	Java 网络类和接口 .....	205
12.2	InetAddress 类 .....	205
12.2.1	创建 InetAddress 对象的 方法 .....	206
12.2.2	实例: 查询 IP 地址的 版本 .....	207
12.3	URL 和 URLConnection 类 .....	209
12.3.1	实例: URL 对象的创建及 使用 .....	211
12.3.2	实例: 使用 URLConnection 从 Web 服务器读取文件 .....	212
12.4	TCP/IP 服务器与客户端 套接字 .....	213
12.4.1	客户端套接字 .....	214
12.4.2	服务器套接字 .....	216
12.4.3	实例: C/S 环境下的套接字 应用程序 .....	218
12.5	Datagram 套接字 .....	224
12.5.1	UDP 套接字 .....	224
12.5.2	实例: 利用 DatagramSocket 查询端口占用情况 .....	226
12.5.3	实例: 利用数据报通信的 C/S 程序 .....	228
12.5.4	组播套接字 .....	230
12.5.5	实例: 组播套接字 C/S 程序 .....	231
12.6	使用 Java NIO 提高服务端程序的 性能 .....	236
12.7	小结 .....	241
<b>第 13 章</b>	<b>JDBC 数据访问接口 .....</b>	<b>243</b>
13.1	JDBC 的体系结构 .....	243
13.2	JDBC 的作用及重要接口 .....	244
13.2.1	DriverManager .....	245
13.2.2	Connection .....	245
13.2.3	Statement .....	246
13.2.4	ResultSet .....	251
13.3	连接数据库 .....	252



13.3.1	创建指定数据库的 URL	252	15.1.4	JSP 与 XML	306
13.3.2	加载驱动程序	253	15.1.5	JSP 与 ASP、PHP	307
13.3.3	实例: 建立数据库连接	253	15.2	JSP 语法	308
13.4	实例: 数据库的增删查改	254	15.2.1	JSP 语法概述	308
13.5	数据库事务	259	15.2.2	JSP 指令	309
13.6	高级数据类型	261	15.2.3	JSP 动作标识	313
13.7	数据源在 JDBC 中的应用	262	15.2.4	脚本元素	322
13.7.1	实例: 注册数据源	263	15.3	JSP 内置对象	324
13.7.2	实例: 查找数据源	264	15.3.1	request 对象	324
13.7.3	实例: 数据源的重新 绑定	266	15.3.2	实例: 使用 request 对象 获取表单参数	327
13.7.4	实例: 删除数据源	267	15.3.3	response 对象	329
13.8	小结	268	15.3.4	实例: URL 重定向和定时 刷新	329
<b>第 14 章</b>	<b>Java Servlets 技术</b>	<b>269</b>	15.3.5	out 对象	330
14.1	Java Servlet 概述	269	15.3.6	session 对象	332
14.1.1	Servlet 的功能	269	15.3.7	pageContext 对象	333
14.1.2	Java Servlet 与现有网络 技术的差别	270	15.3.8	application 对象	333
14.1.3	Servlet 的生命周期	270	15.3.9	config 对象	334
14.2	编写 HTTP Servlet 程序	272	15.3.10	page 对象	334
14.2.1	实例: 一个简单的 Servlet 程序	276	15.4	JavaBean 在 JSP 中的应用	334
14.3	HTTP POST/GET 请求	279	15.5	标签库的开发	336
14.3.1	实例: Servlet 处理 POST 请求	283	15.5.1	简单标签	337
14.4	会话跟踪	287	15.5.2	带有主体和属性的标签	339
14.4.1	实例: 会话跟踪 Servlet 程序	288	15.6	在 JSP 页面的安全	344
14.5	Cookie 的应用	290	15.6.1	配置 Web 服务器的 SSL	344
14.5.1	实例: 使用 Cookie 的 Servlet 程序	291	15.6.2	配置 realms	345
14.6	Filter Servlet	294	15.7	小结	347
14.7	Listener Servlet	299	<b>第 16 章</b>	<b>事务处理</b>	<b>349</b>
14.8	小结	302	16.1	事务的 ACID 特征	349
<b>第 15 章</b>	<b>Java Server Pages 技术</b>	<b>303</b>	16.2	J2EE 事务服务	351
15.1	JSP 在 Web 程序中的应用	303	16.3	JTA 事务	353
15.1.1	B/S 应用开发技术	303	16.3.1	UserTransaction 接口	353
15.1.2	JSP 技术	304	16.3.2	Status 接口	354
15.1.3	JSP 应用	305	16.3.3	JTA 定义的事务属性	355
			16.4	JTS	356
			16.5	小结	358
			<b>第 17 章</b>	<b>基于 JMS 的消息处理</b>	<b>359</b>
			17.1	JMS 概述	359

17.2	JMS API 编程模型 .....	361	18.4.4	在客户端使用 EJB .....	426
17.2.1	Administered Objects .....	361	18.5	消息驱动 Bean .....	430
17.2.2	Destinations .....	362	18.6	小结 .....	433
17.2.3	Connections .....	362	<b>第 19 章</b>	<b>JavaMail 技术 .....</b>	<b>435</b>
17.2.4	Sessions .....	363	19.1	E-mail 协议概述 .....	435
17.2.5	Message Producers .....	363	19.1.1	SMTP 协议 .....	435
17.2.6	Message Consumers .....	364	19.1.2	POP3 协议 .....	436
17.2.7	Message Listeners .....	365	19.1.3	IMAP 协议 .....	436
17.2.8	Message Selectors .....	365	19.1.4	MIME 协议 .....	436
17.2.9	Message .....	365	19.2	邮件消息模型 .....	436
17.3	JMS Point-to-Point (PTP) 模型 编程 .....	367	19.2.1	简单消息模型 .....	437
17.4	JMS Publish/Subscribe (Pub/Sub) 模型编程 .....	382	19.2.2	Multipart 消息模型 .....	437
17.5	小结 .....	389	19.3	JavaMail 核心类 .....	438
<b>第 18 章</b>	<b>EJB 技术 .....</b>	<b>391</b>	19.3.1	Session .....	438
18.1	何谓 EJB .....	391	19.3.2	Message .....	438
18.1.1	EJB 的设计目标 .....	391	19.3.3	Address .....	439
18.1.2	EJB 容器和服务器 .....	392	19.3.4	Authenticator .....	440
18.1.3	EJB 的体系结构 .....	393	19.3.5	Transport .....	440
18.1.4	EJB 的重要特性 .....	395	19.3.6	Store 和 Folder .....	441
18.1.5	EJB 中的角色 .....	395	19.4	发送邮件 .....	441
18.1.6	EJB 与 JavaBean 的 区别 .....	396	19.5	获取邮件 .....	447
18.2	EJB 的开发过程 .....	397	19.6	邮件的回复和转发 .....	449
18.2.1	Home 接口 .....	397	19.7	小结 .....	450
18.2.2	远程接口 .....	398	<b>第 20 章</b>	<b>基于 JAAS 的安全技术 .....</b>	<b>451</b>
18.2.3	Bean 类 .....	398	20.1	Java 的安全性 .....	451
18.2.4	配置描述器 .....	398	20.2	JAAS 概览 .....	453
18.3	会话 Bean .....	405	20.3	JAAS 核心类 .....	454
18.3.1	有状态会话 Bean .....	406	20.3.1	Subject 和 Principal .....	455
18.3.2	无状态会话 Bean .....	410	20.3.2	LoginContext .....	455
18.3.3	会话 Bean 的特点及 适用场合 .....	414	20.3.3	LoginModule .....	456
18.4	实体 Bean .....	415	20.3.4	CallbackHandler 和 Callback .....	457
18.4.1	实体 Bean 的开发 .....	416	20.4	认证 .....	458
18.4.2	实体 Bean 的关系 .....	419	20.4.1	实例: 使用 JAAS 进行 认证 .....	458
18.4.3	EJB 查询语言 .....	424	20.5	授权 .....	463
			20.5.1	实例: 声明性授权 .....	464
			20.5.2	实例: 程序性授权 .....	465

20.6 小结.....	466	21.5.4 SOAP 编码规则.....	488
<b>第 21 章 Web 服务技术.....</b>	<b>467</b>	21.5.5 SOAP RPC 表示.....	490
21.1 Web 服务体系结构.....	467	21.6 Web 服务开发.....	492
21.2 XML 解析——JAXP.....	469	21.6.1 Servlet 作为 Web 服务的端点.....	492
21.3 UDDI.....	476	21.6.2 EJB 作为 Web 服务的 端点.....	498
21.3.1 商业实体信息.....	478	21.7 小结.....	502
21.3.2 商业服务信息.....	478	附录 A 主题词表.....	503
21.3.3 绑定模板信息.....	478	附录 B J2EE 兼容应用服务器.....	505
21.3.4 技术规范信息.....	479	附录 C 参考资料.....	526
21.4 WSDL.....	480		
21.5 SOAP.....	483		
21.5.1 常用术语.....	484		
21.5.2 SOAP 消息.....	485		
21.5.3 SOAP Fault.....	487		

# 第 1 篇 基 础 篇

本篇主要讲述了 J2EE 的基本架构、Java 语言的基础知识和 GUI 程序设计。在第 1 章中，介绍了 J2EE 的产生背景、主要优点及其分层体系结构，并对 J2EE 中的核心技术进行了概括性的描述。在第 2 章中简要地介绍了 Java 语言的基础知识，包括标识符、关键字、数据类型、运算符、控制语句等语言要素，并简单地介绍了类和对象的概念。在第 3 章中，着重介绍了 GUI 程序的设计，包括事件处理、常用组件、布局管理器及图形绘制等内容。

115534/04



# 第 1 章 J2EE 概述

Java 2 企业版本 (The Java 2 Enterprise Edition, 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。它提供了一个统一的开发平台, 不仅降低了开发多层应用的费用和复杂性, 而且加快了应用的分布式部署, 增强了安全机制, 提高了系统的性能。利用 J2EE 体系结构, 企业应用开发人员可以把一些通用的、繁琐的服务端任务交给中间件供应商来完成, 而把精力集中在商业逻辑的设计和应用的表示上。

本章主要介绍如下内容:

- J2EE 的产生背景及其优点
- J2EE 的体系结构
- J2EE 技术概览

## 1.1 J2EE 的产生背景及其优点

随着电子商务与信息技术的迅猛发展, 社会各个部门之间越来越需要依赖于企业级应用程序共同协作。例如, 企业需要借助网络进行采购和分销, 和其他企业进行在线交易; 电子商务网站需要连接银行的系统以完成在线支付和结算; 政府部门需要将政务处理逐渐转移到网络上进行……企业级应用程序正在不断地渗入社会生产和生活的各个领域。它将对社会经济的模式及其发展产生深刻的影响, 也必将在很大程度上影响和改变人们的生活方式。

与此同时, 应用程序开发人员正在感受到前所未有的巨大压力。因为他们必须用更少的资金和资源来更快地设计、建立和生产具有高可用性、高可靠性及可扩展性的企业应用程序。而且, 电子商务的发展对传统的 Web 技术提出了强有力的挑战。由于电子商务的内部逻辑复杂, 安全性要求苛刻, 商务形式发展变化快, 这就要求 Web 技术提供足够的复杂度和灵活性以适应电子商务的需求。

所有这些需求和挑战, 使得人们不得不借助于一个功能强大而又方便灵活的体系结构来完成企业级应用程序的设计、开发、装配及部署。J2EE 就是在这种需求下应运而生的。它提供了一个基于组件的方法和多层的分布式应用模型、一致化的安全模型, 以及灵活的事务控制。利用 J2EE, 您不仅能以更快的速度向市场推出创造性的客户解决方案, 而且您的平台独立的、基于组件的 J2EE 解决方案不会被束缚在任何一个厂商的产品和 API 上。

目前, J2EE 得到了企业应用开发人员的广泛青睐。它已经成为电子商务的主要开发平台, 并且正在日益完善。J2EE 之所以能够从众多技术中脱颖而出, 是因为它有着许多突出的优点:

- J2EE 架构能够在已有的企业信息系统的基礎上开发新的系统, 从而可以充分利用

用户的原有投资。这是因为 J2EE 拥有业界的广泛支持和一些重要的企业计算领域供应商的参与，而且基于 J2EE 平台的产品对软硬件的要求不高，因此原有的操作系统和硬件也可以继续使用。

- J2EE 平台是建立在 Java 语言基础上的。Java 是一种纯 OO（面向对象）语言，具有丰富的数据类型和强大的功能，能完成几乎任何复杂的功能——这是一般的 Web CGI 编程语言所无可比拟的。面向对象的设计方法，不但可以设计庞大而复杂的系统，还可以使 Web 应用程序具有良好的扩充性和维护性，能够方便地实现国际化和本地化的功能，因此深受 Web 开发人员的喜爱。
- J2EE 具有很好的平台无关性，支持异构环境。Java 是一个跨平台的语言，在任何平台上，只要有 Java 虚拟机（Java Virtual Machine, JVM），就能在不同平台上执行同一个 Java 程序。因此，用 J2EE 开发的应用程序可以非常方便地部署在不同的硬件和操作系统中。J2EE 标准的平台无关性使得任何符合 J2EE 标准的应用服务器之间可以共用标准的组件，而且程序的移植也变得更加轻松简单。
- J2EE 提供了标准的系统框架和服务。标准化的框架结构是以分布式的多层应用体系为基础的。在系统的任意层面中可以增加新的功能，而不影响原有的系统。J2EE 平台提供了很多系统服务，而且这些服务只要通过简单的配置就可获得。
- J2EE 可以在很大程度上提高开发效率。因为 J2EE 允许企业应用开发人员把一些通用的、繁琐的服务端任务交给中间件供应商来完成，而把精力集中在商业逻辑上。而且，J2EE 的构架非常适合团体开发的模式。J2EE 标准的结构自然将应用分成表现层、企业逻辑层和数据层，这样可以使企业中的页面设计人员、业务逻辑开发人员和系统分析员明确分工、各司其职，从而提高整个团队的开发效率。

## 1.2 J2EE 的体系结构

在一个多层的分布式应用模型中，应用逻辑通常被根据功能划分成若干个组件。J2EE 应用就是由一个个的组件构成的。组件是一种具有独立功能的软件单元，它可以在 J2EE 服务器中执行，也可以调用别的组件的功能。J2EE 规范定义了以下几种类型的组件：

Application client components 和 Applets、Java Servlets 及 JavaServer Pages (JSP)、Enterprise JavaBeans components (EJB)。这些不同的组件可以安装在同一个服务器或不同的服务器上。

J2EE 的体系结构可以分为 4 层，即客户层、Web 层、业务层和企业信息系统层 (EIS)，如图 1-1 所示。不同的层包含了不同的组件。

下面我们就对各层逐一进行详细的分析和阐述。

### ● 客户层

用户对一个企业级应用程序的感受与客户层的行为密切相关。客户层负责将用户的请求发送给服务器，并将服务器的返回的结果呈现给用户。

J2EE 平台支持多种类型的用户，它的客户端可以是 Web 客户、Applets 或客户端应用程序，客户运行的硬件可以是功能强大的台式机，也可以是小巧的便携设备。Web 客户通常运行在浏览器中，使用浏览器的服务显示 Web 层提供的内容。在这些客户中，用户界面

在服务器端由 Web 层生成并使用 HTML 通信。Web 客户简单方便，但不能做复杂的操作。Applet 是一个小的客户端程序，它需要浏览器的 Java 虚拟机来执行，可以把一部分的业务逻辑用 Applet 实现。客户端应用程序提供了更加友好的图形界面的功能，它可以直接调用业务逻辑层中的 EJB，当然，也可以打开一个 HTTP 链接，与 Web 层的 Servlet 通信。

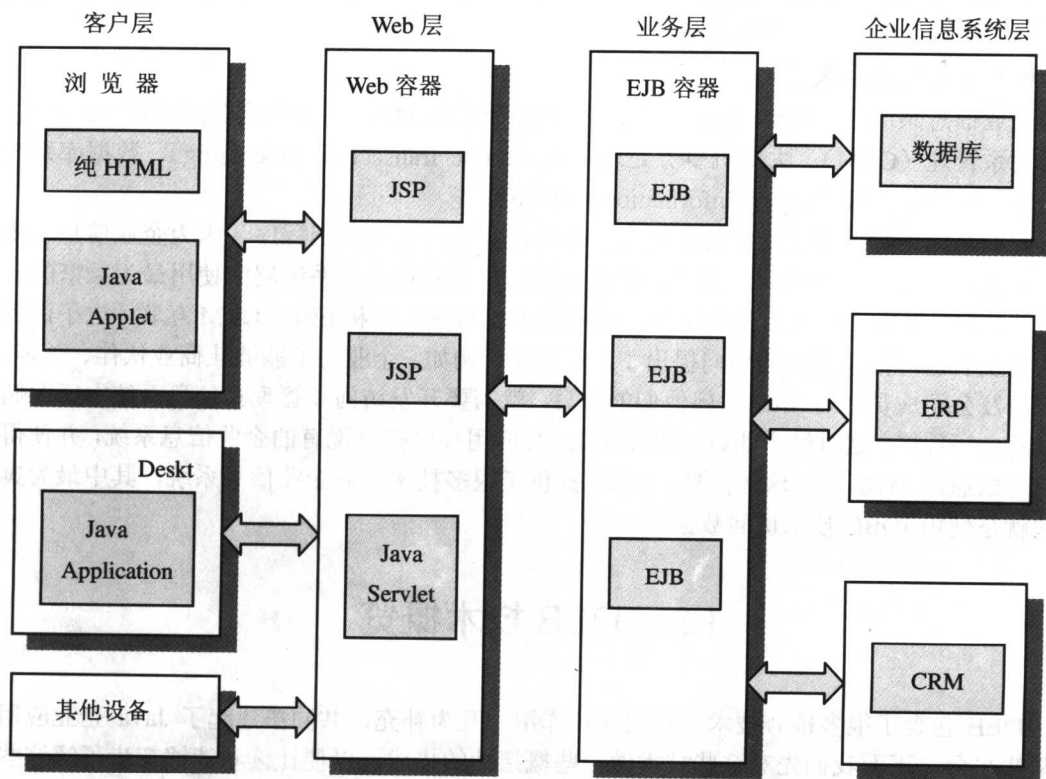


图 1-1 J2EE 的体系结构

J2EE 应用程序可以是基于 Web 的，也可以不是基于 Web 的。在一个基于 Web 的 J2EE 应用程序中，用户的浏览器在客户层运行，并从一个 Web 服务器上下载 Web 层中的静态 HTML 页面或由 JSP 或 Servlets 生成的动态 HTML 页面。在一个不基于 Web 的 J2EE 应用程序中，一个独立的客户程序，并不运行在一个 HTML 页面中，而是运行在其他一些基于网络的系统中，在不经 Web 层的情况下访问 Enterprise Beans。

#### ● Web 层

这一层是为基于 Web 的应用程序服务的。Web 层组件可以是 Servlet 或 JSP。Servlet 和 JSP 以一种可移植、跨平台的方式支持动态网页的生成。它们的主要作用是响应 Web 客户端的请求，接受并检验用户输入的数据，把数据传给业务层的商务逻辑组件，并把业务层处理的结果返回给用户。

#### ● 业务层

业务层经常被称做 Enterprise JavaBeans (EJB) 层。EJB 层容纳与应用程序相关的商务



逻辑和诸如事务管理、并发控制 and 安全性这样的系统级服务。EJB 技术提供分布式的组件模型，使得应用程序开发者可以解决诸如银行、零售、金融等特定业务领域的商务需求。一个 EJB 从客户程序接收数据，对数据进行相应的处理，再将数据发送到企业信息系统层存储。另一方面，EJB 也负责从存储中检索数据，并将数据返回给客户程序。

运行在业务层的 EJB 依赖容器为诸如事务、生命期、状态管理、多线程及资源存储池提供通常都是非常复杂的系统级代码。

- 企业信息系统层

企业信息系统层 (EIS) 包括了各种企业基础设施系统，例如企业资源计划 (ERP)、客户关系管理 (CRM)、大型机事务处理 (mainframe transaction processing)、数据库系统及其他遗留信息系统 (legacy information systems) 等。

企业级应用程序要求能够访问运行在企业信息系统上的应用程序，因为企业信息系统提供了企业的信息基础结构。通常，数据库被认为是企业信息系统层中使用最为频繁的一个系统，但是随着商业逻辑的复杂化，越来越多的系统需要和 ERP、CRM 互联。电子商务模式的发展也对信息系统的访问提出了新的要求。例如，企业为了能够让商业伙伴、顾客、供应商及公司成员通过 Web 访问他们的信息，就需要开发访问和管理其信息系统中所存储信息的应用程序。这时候企业就可以使用 J2EE 应用程序扩展现有的企业信息系统，并使得用户可以通过 Web 访问这些信息。J2EE 提供了很多技术访问企业信息系统，其中最为典型的就是使用 JDBC 技术访问数据库。

## 1.3 J2EE 技术概览

J2EE 包含了很多核心技术，它们互相作用，互为补充，共同搭建起了 Java 企业应用的坚实平台。下面我们先对这些技术做一些概括性的描述，以便让读者能够初步了解这些技术在 J2EE 平台上的作用及其地位。

- RMI

RMI (Remote Method Invocation) 提供了一种在不同主机上的 Java 虚拟机 (Java Virtual Machine, JVM) 之间进行通信的方式。RMI 能够调用远程主机上的对象，并且就像这个对象在本地虚拟机上一样使用它。RMI 也可以动态地加载类和安全管理器，在网络上安全地传输 Java 类。

- JNDI

JNDI (Java Naming and Directory Interface) 是 J2EE 中用来给对象命名的技术。这里所说的对象包括 Web 组件、EJB 组件、数据库、文件系统、机器等。J2EE 提供的命名和目录服务可以将这些名字和具体的对象绑定在一起，然后应用程序就可以通过这些名字定位这些对象，从而访问用户信息、机器信息和各种服务。

- JDBC

JDBC (Java Database Connection) 是 J2EE 中用来访问数据库的技术。利用 JDBC API 可以在 J2EE 平台和数据库之间建立连接。在 EJB、JSP、Servlets 中都可以使用 JDBC 对数据库进行各种操作，比如查询、修改、存储、管理事务，等等。