

Java

网络编程精解

Helping readers to master Java network programming and distributed computing.

精通Java套接字编程

构建多线程的服务器及非阻塞的服务器

探索流行的分布式软件架构：RMI、CORBA和SOAP

掌握常用的客户端API：JDBC API和JavaMail API

深刻领悟面向对象思想

巩固UML建模技术和常用的Java设计模式



孙卫琴

飞思科技产品研发中心 编著 监制



电子工业出版社

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

Java 开发专家

Java

网络编程精解

孙卫琴

飞思科技产品研发中心

编著

监制



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

· 内 容 简 介 ·

本书结合大量的典型实例,详细介绍了用 Java 来编写网络应用程序的技术。本书的范例都基于最新的 JDK 1.5 版本,书中内容包括:Java 网络编程的基础知识、套接字编程、非阻塞通信、创建 HTTP 服务器与客户程序、数据报通信、对象的序列化与反序列化、Java 反射机制、RMI 框架、JDBC API、JavaMail API、MVC 设计模式、安全网络通信、CORBA 和 Web 服务。另外,本书还涵盖了 Sun 公司的 SCJD (Sun Certified Java Developer) 认证的考试要点。

书中范例源文件请到 <http://www.fecit.com.cn> 的“下载专区”下载。

阅读本书,读者不仅可以掌握网络编程的实用技术,还可以进一步提高按照面向对象的思想来设计和编写 Java 软件的能力。本书适用于所有 Java 编程人员,包括 Java 初学者及资深 Java 开发人员。本书还可作为高校的 Java 教材,以及企业的 Java 培训教材,也可作为 Sun 公司 SCJD 认证的辅导教材。

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

图书在版编目(CIP)数据

Java 网络编程精解 / 孙卫琴编著. — 北京: 电子工业出版社, 2007.3

(Java 开发专家)

ISBN 978-7-121-03900-3

I. J… II. 孙… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 024148 号

责任编辑: 宋兆武 徐 磊

印 刷: 北京智力达印刷有限公司

装 订: 北京中新伟业印刷有限公司

出版发行: 电子工业出版社

北京海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×1092 1/16 印张: 31.25 字数: 800 千字

印 次: 2007 年 3 月第 1 次印刷

印 数: 7 000 册 定价: 55.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系电话: (010) 68279077; 邮购电话: (010) 88254888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

“开发专家之 Sun ONE”全新提升为“Java 开发专家”系列
——源自精品 成就理想

出版说明

★ 从“开发专家之 Sun ONE”到“Java 开发专家”

“开发专家之 Sun ONE”系列丛书从诞生之日至今，已经四岁了。在这个系列里面，我们一直努力体现着这么一个理念：用一种较为敏锐的视角来跟踪 IT 技术的发展轨迹，并把可能为广大程序员所希望获得的知识，用图书出版的方式奉献给大家。

在这个系列中，我们陆续出版了约 30 种图书，有《Java 与模式》、《JSP 应用开发详解（第 2 版）》、《精通 EJB（第 3 版）》、《Tomcat 与 Java Web 应用开发详解》、《精通 Struts：基于 MVC 的 Java Web 设计与开发》、《JBoss 管理与开发核心技术（第 3 版）》、《精通 Spring》、《精通 Hibernate：Java 对象持久化技术详解》等一大批读者朋友耳熟能详的作品。很多作品都是在国内没有同类图书的情况下出版的。在这几年的出版工作中，我们时刻感受着市场的风险，也时刻收获着无数读者给我们的认可。

在这个系列中，凝聚了大量资深技术专家的心血。有大家都熟知的阎宏、刘晓华、孙卫琴、罗时飞等，还有一些正在不断腾跃的开发高手。这些非常优秀的国内原创作者们一直都在支持着“开发专家之 Sun ONE”系列的出版工作，在这里，我们要向他们说声：谢谢。

桃李不言，下自成蹊。由于这些年“开发专家之 Sun ONE”在“两个效益”中的杰出表现，电子工业出版社授予这个系列“最佳品牌奖”。

时代不断前进，技术不断变革。为了顺应 Java 领域的技术发展态势，为了赋予这个经典的图书系列更强的生命力，我们将“开发专家之 Sun ONE”升级为“Java 开发专家”。我们将继承原有的出版理念，紧密跟踪技术热点和发展趋势，会聚更多优秀作者，全力奉献更经典的作品。

★ 规划你的 Java 开发之路

喜马拉雅山脉的最高峰不断地在温室效应中降低，而 Java 世界的颠峰永远都在技术人员的追求中不断升高。每个人都有不同的路，每个人都有不同的行路方式，不过，往往“到了山顶才发现，错误的路和正确的路就差那么几步！”

身处 Java 洪流中的程序员最累（不过大家都说 Java 程序员薪水最高，呵呵），我们简单整理了一下 Java 领域的相关技术、工具、架构，如下图所示。这个框图中的每一个英文单词（或缩写）都可以写成一本书。Java 领域还有一个特点，那就是商业产品和开源产品层出不穷，潮流不断。相比于其他领域，如 .NET，Java 开发更是体现了这句谚语：条条大路通罗马。

罗马只有一个，大路却有多条。看上去，似乎到罗马很容易，反正路多嘛。不过，路多却容易迷失方向。当你在 Java 领域中摸爬滚打几年后，发现自己在无数条道路上走了很久，却不知道罗马何日才能到达，甚至连罗马的方向都不知道，这时你肯定会很失落。

很遗憾，在这个简短的出版说明文章里面，我们无法告诉你每一条连贯的、不费周折的通往罗马的道路该如何走。或许，通过“Java 开发专家”系列中的某本书，你可以找到属于你的正确道路。在一般情况下，我们不会就某一项很窄的话题来单独写一本书，我们还是希望通过我们的一些专业和智慧，尽力把一些相关技术整合起来，用较为简明的方式表达出来，最后由你来选择。

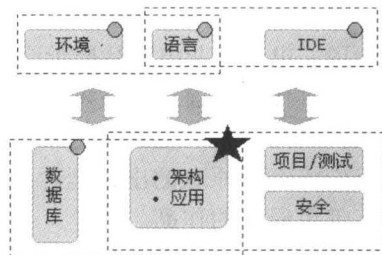
这里有句话与大家共勉：少走弯路，就是捷径！



★ “Java 开发专家”的奉献

犹如在上面那个框图中展现的那样，我们希望在各个层面、各个方向上都能给读者奉献出优秀的图书作品，全面体现技术与应用的结合。从宏观上看，我们会从语言、IDE、环境、数据库、架构与应用、安全、项目与测试等方面进行选择，选出一些读者迫切需要的技术来先行规划。

“Java 开发专家”虽然新蓓初绽，但因其源自盛放的“开发专家之 Sun ONE”系列而根基稳健，两个系列会有一段很长的并行时间，我们会用一种优化的方式来保证读者的顺利选择。无论哪一个系列，必定都有大家喜欢的图书。



在技术上，有着持久化的方法，在学习上，也需要有持久化的精神。

从“开发专家之 Sun ONE”到“Java 开发专家”，希望可以带给你持久化的动力。

联系方式

咨询电话：(010) 68134545 88254160

电子邮件：support@fecit.com.cn

服务网址：http://www.fecit.com.cn http://www.fecit.net

通用网址：计算机图书、飞思、飞思教育、飞思科技、FECIT

Java 语言是第一个完全融入网络的语言，它之所以适合编写网络应用程序，归功于它的以下优势。

(1) Java 语言与生俱来就是与平台无关的。Java 程序能够运行在不同的平台上，运行在不同平台上的 Java 程序能够方便地进行网络通信。

(2) Java 语言具有完善的安全机制，可以对程序进行权限检查，这对网络程序至关重要。

(3) JDK 类库提供了丰富的网络类库，如套接字 API、JavaMail API 和 JDBC API 等，大大简化了网络程序的开发过程。

本书将展示如何利用 Java 网络类库来快速便捷地创建网络应用程序，致力于完成以下任务：

- 实现访问 HTTP 服务器的客户程序
- 实现 HTTP 服务器
- 实现多线程的服务器，以及非阻塞的服务器
- 解析并展示 HTML 页面
- 通过 JDBC API 访问数据库
- 通过 JavaMail API 接收和发送电子邮件
- 利用 RMI 框架实现分布式的软件系统
- 进行安全的网络通信，对数据加密，验证身份，保证数据的完整性
- 利用 CORBA 和 Web 服务实现分布式的软件系统

本书的组织结构和主要内容

本书结合大量典型的实例，详细介绍了用 Java 来编写网络应用程序的技术。本书内容包括：Java 网络编程的基础知识、套接字编程、非阻塞通信、创建 HTTP 服务器与客户程序、数据报通信、对象的序列化与反序列化、Java 反射机制、RMI 框架、JDBC API、JavaMail API、MVC 设计模式、安全网络通信、CORBA 和 Web 服务。如图 P-1 展示了本书各个章之间的顺序渐进关系。

从图 P-1 中可以看出，套接字 (Socket) 是 Java 网络编程的基础，第 2 章和第 3 章分别详细介绍了 Socket 与 ServerSocket 的用法。本书第 1 章介绍了分层的网络体系结构，Java 网络程序位于最上层——应用层，并且通过套接字访问底层网络，也可以说，套接字为应用层封装了底层网络传输数据的细节。Java 网络程序都采用客户/服务器模式，客户端发出获得特定服务的请求，服务器接收请求，执行客户端所请求的操作，然后向客户端发回响应。在介绍服务器端编程时，探讨了服务器端实现并发响应多个客户请求的两种方式：一种方式是运用线程池 (第 3 章)，还有一种方式是采用非阻塞通信 (第 4 章)。在介绍客户端编程时，介绍了 JDK 提供的一种通用的客户端协议处理框架 (第 6 章)。

利用 Java 网络 API，可以实现基于各种应用层协议 (如 HTTP 协议和 FTP 协议) 的服务器程序与客户程序，本书侧重介绍了 HTTP 服务器 (第 5 章) 与 HTTP 客户程序 (第 7 章) 的实现方法，HTTP 客户程序也称为浏览器。

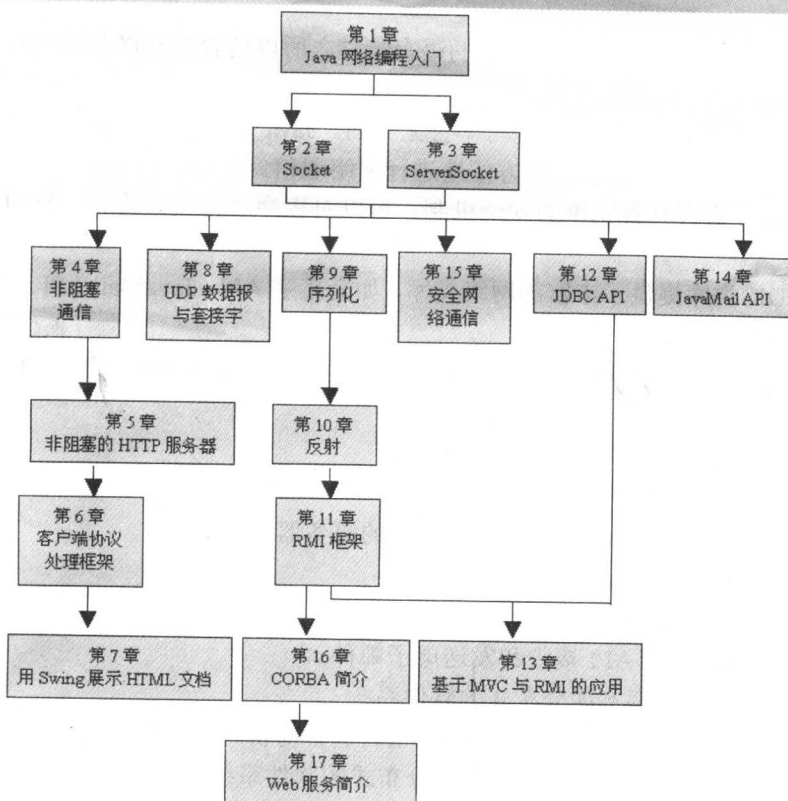


图 P-1 本书各章之间的顺序渐进关系

本书还介绍了 3 种分布式的软件架构：RMI（第 11 章）、CORBA（第 16 章）和 Web 服务（第 17 章）。这些分布式架构主要解决的问题是，如何让客户端调用服务器端的远程对象的方法。RMI 是 JDK 自带的，它要求客户端与服务器端都是 Java 程序，而 CORBA 和 Web 服务允许用任意编程语言编写的客户程序与服务器程序相互通信。本书重点介绍了 RMI 框架的用法。RMI 框架在其实现中封装了用套接字通信的细节，此外，RMI 框架的实现会把客户端的方法调用请求信息序列化为字节序列，并把它发送给服务器端，然后在服务器端再通过反序列化把字节序列还原为方法调用请求。RMI 框架还运用了动态代理机制，为客户端提供了远程对象的代理。客户端实际上直接访问的是远程对象的代理。为了帮助读者理解 RMI 框架的实现原理，本书第 9 章和第 10 章分别介绍了 Java 序列化及反射机制。在介绍反射机制时，介绍了动态代理。

本书还介绍了两个常用的客户端的网络 API：JDBC API（第 12 章）和 JavaMail API（第 14 章），这两个 API 分别用于访问数据库服务器和邮件服务器，在它们的实现中都封装了用套接字与服务器通信的细节。Java 客户程序可以通过 JDBC API 来访问各种数据服务器，还可以通过 JavaMail API 来访问各种邮件服务器。

本书第 13 章介绍了一个运用了 MVC 设计模式和 RMI 框架的综合应用。MVC 设计模式把实际的软件应用分为视图、控制器和模型 3 个层次，每个层次相对独立。本书的范例把模型作为远程对象放置到 RMI 的服务器端，把视图和控制器放置到 RMI 的客户端。

本书第 15 章介绍了 SSLSocket, 它支持 SSL(Server Socket Layer)协议和 TLS(Transport Layer Security) 协议。运用 SSLSocket, 可以实现安全的网络通信, 网络上传输的是被加密的数据, 并且通信两端还能验证对方的身份。

本书在介绍以上技术时, 采用 UML 建模语言中的类框图及时序图来展示对象模型, 以及类与类之间的协作关系。此外, 本书还把一些常见的设计模式, 如静态代理模式、动态代理模式和 MVC 设计模式等运用到实际例子中。阅读本书, 读者不仅可以掌握 Java 网络编程的实用技术, 还可以进一步提高按照面向对象的思想来设计和编写 Java 软件的能力。

这本书是否适合您

阅读本书, 要求读者已经具备了 Java 编程的基础知识。对于还不熟悉 Java 语言的读者, 建议先阅读作者的另一本书《Java 面向对象编程》(电子工业出版社出版), 本书是它的姊妹篇。《Java 面向对象编程》自 2006 年 7 月份出版后, 受到了广大 IT 读者的欢迎。本书围绕着网络编程, 进一步介绍了 Java 语言的一些高级特性, 如套接字编程、非阻塞通信、数据报通信、对象的序列化与反序列化、Java 反射机制、RMI 框架、JDBC API 和 JavaMail API 等, 这些特性是作为一个高级 Java 开发人员必须掌握的。深入了解这些高级特性, 有助于开发人员熟练地从头开发分布式的软件系统, 或者轻松地学习和掌握现有的, 如 Java EE 等分布式的软件架构。

本书一方面由浅入深组织内容, 迎合 Java 网络编程初学者的需求, 一方面与实际项目紧密结合, 介绍了线程池、非阻塞通信和动态代理等高级话题。本书不仅可作为 Java 开发人员的参考手册, 还可以作为高校的 Java 教材, 以及企业培训教材。

致谢

本书在编写过程中得到了 Sun 公司在技术上的大力支持, 飞思科技产品研发中心负责监制工作, 复旦软件学院的 MSE 师生们为本书的编写提供了有益的帮助, JavaThinker 网站 (www.javathinker.org) 的许多热心网友及作者的亲友, 包括 axman、李红军、余泉、王庆华、gogolong、李洪成、曹文伟和刘巧云等为本书提供了宝贵的建议, 在此表示衷心的感谢!

尽管我们尽了最大努力, 但本书难免会有不妥之处, 欢迎各界专家和读者朋友批评指正。以下网址会征集并公布本书的勘误信息:

<http://www.javathinker.org/bbs/topic.jsp?db=7&topic=4058>



第 1 章	Java 网络编程入门	1
1.1	进程之间的通信	1
1.2	计算机网络的概念	3
1.3	OSI 参考模型	5
1.4	TCP/IP 参考模型和 TCP/IP 协议	8
1.4.1	IP 协议	11
1.4.2	TCP 协议及端口	14
1.4.3	RFC 简介	15
1.4.4	客户/服务器通信模式	16
1.5	用 Java 编写客户/服务器 程序	17
1.5.1	创建 EchoServer	18
1.5.2	创建 EchoClient	20
1.6	小结	22
1.7	练习题	23
第 2 章	Socket 用法详解	25
2.1	构造 Socket	25
2.1.1	设定等待建立连接的 超时时间	26
2.1.2	设定服务器的地址	26
2.1.3	设定客户端的地址	27
2.1.4	客户连接服务器时 可能抛出的异常	27
2.2	获取 Socket 的信息	30
2.3	关闭 Socket	32
2.4	半关闭 Socket	33
2.5	设置 Socket 的选项	38
2.5.1	TCP_NODELAY 选项	38
2.5.2	SO_REUSEADDR 选项	38
2.5.3	SO_TIMEOUT 选项	39
2.5.4	SO_LINGER 选项	42
2.5.5	SO_RCVBUF 选项	44
2.5.6	SO_SNDBUF 选项	45
2.5.7	SO_KEEPALIVE 选项	45
2.5.8	OOBINLINE 选项	45
2.5.9	服务类型选项	45

2.5.10	设定连接时间、延迟和 带宽的相对重要性	46
2.6	发送邮件的 SMTP 客户程序 ..	47
2.7	小结	51
2.8	练习题	52
第 3 章	ServerSocket 用法详解	55
3.1	构造 ServerSocket	55
3.1.1	绑定端口	55
3.1.2	设定客户连接请求 队列的长度	56
3.1.3	设定绑定的 IP 地址	58
3.1.4	默认构造方法的作用	58
3.2	接收和关闭与客户的连接	59
3.3	关闭 ServerSocket	60
3.4	获取 ServerSocket 的信息	60
3.5	ServerSocket 选项	62
3.5.1	SO_TIMEOUT 选项	62
3.5.2	SO_REUSEADDR 选项	63
3.5.3	SO_RCVBUF 选项	64
3.5.4	设定连接时间、延迟和 带宽的相对重要性	64
3.6	创建多线程的服务器	65
3.6.1	为每个客户分配一个线程	65
3.6.2	创建线程池	67
3.6.3	使用 JDK 类库提供的 线程池	72
3.6.4	使用线程池的注意事项	74
3.7	关闭服务器	76
3.8	小结	80
3.9	练习题	81
第 4 章	非阻塞通信	83
4.1	线程阻塞的概念	83
4.1.1	线程阻塞的原因	83
4.1.2	服务器程序用多线程 处理阻塞通信的局限	84
4.1.3	非阻塞通信的基本思想	85
4.2	java.nio 包中的主要类	87

Contents

4.2.1 缓冲区 Buffer.....	88	5.2.6 代表 HTTP 请求的 Request 类	143
4.2.2 字符编码 Charset.....	90	5.2.7 代表 HTTP 响应的 Response 类	145
4.2.3 通道 Channel.....	90	5.2.8 代表响应正文的 Content 接口及其实现类	147
4.2.4 SelectableChannel 类	92	5.2.9 运行 HTTP 服务器	149
4.2.5 ServerSocketChannel 类	93	5.3 小结	150
4.2.6 SocketChannel 类	93	5.4 练习题	151
4.2.7 Selector 类	96	第 6 章 客户端协议处理框架.....	153
4.2.8 SelectionKey 类	97	6.1 客户端协议处理框架的 主要类	153
4.3 服务器编程范例	100	6.2 在客户程序中运用 协议处理框架	154
4.3.1 创建阻塞的 EchoServer.....	100	6.2.1 URL 类的用法	154
4.3.2 创建非阻塞的 EchoServer.....	103	6.2.2 URLConnection 类的用法	156
4.3.3 在 EchoServer 中混合用 阻塞模式与非阻塞模式	110	6.3 实现协议处理框架.....	160
4.4 客户端编程范例	114	6.3.1 创建 EchoURLConnection 类	161
4.4.1 创建阻塞的 EchoClient	114	6.3.2 创建 EchoURLStreamHandler 及工厂类	162
4.4.2 创建非阻塞的 EchoClient	116	6.3.3 创建 EchoContentHandler 类及工厂类	163
4.4.3 创建非阻塞的 PingClient	120	6.3.4 在 EchoClient 类中运用 ECHO 协议处理框架	165
4.5 小结	126	6.4 小结	166
4.6 练习题	127	6.5 练习题	167
第 5 章 创建非阻塞的 HTTP 服务器 ...	129	第 7 章 用 Swing 组件展示 HTML 文档.....	169
5.1 HTTP 协议简介	129	7.1 在按钮等组件上展示 HTML 文档	170
5.1.1 HTTP 请求格式	129	7.2 用 JEditorPane 组件 创建简单的浏览器	171
5.1.2 HTTP 响应格式	132	7.3 小结	179
5.1.3 测试 HTTP 请求	133	7.4 练习题	179
5.2 创建非阻塞的 HTTP 服务器	137		
5.2.1 服务器主程序: HttpServer 类	137		
5.2.2 具有自动增长的缓冲区的 ChannelIO 类	138		
5.2.3 负责处理各种事件的 Handler 接口	140		
5.2.4 负责处理接收连接就绪 事件的 AcceptHandler 类	140		
5.2.5 负责接收 HTTP 请求和 发送 HTTP 响应的 RequestHandler 类	141		

第 8 章 基于 UDP 的

数据报和套接字 181

8.1 UDP 协议简介 181

8.2 DatagramPacket 类 184

8.2.1 选择数据报的大小 185

8.2.2 读取和设置 DatagramPacket
的属性 185

8.2.3 数据格式的转换 186

8.2.4 重用 DatagramPacket 187

8.3 DatagramSocket 类 189

8.3.1 构造 DatagramSocket 189

8.3.2 接收和发送数据报 190

8.3.3 管理连接 190

8.3.4 关闭 DatagramSocket 191

8.3.5 DatagramSocket 的选项 191

8.3.6 IP 服务类型选项 193

8.4 DatagramChannel 类 193

8.4.1 创建 DatagramChannel 194

8.4.2 管理连接 194

8.4.3 用 send() 方法发送数据报 194

8.4.4 用 receive() 方法接
收数据报 195

8.4.5 用 write() 方法发送数据报 198

8.4.6 用 read() 方法接收数据报 199

8.5 组播 Socket 202

8.5.1 MulticastSocket 类 205

8.5.2 组播 Socket 的范例 207

8.6 小结 209

8.7 练习题 210

第 9 章 对象的序列化与反序列化 213

9.1 JDK 类库中的序列化 API 213

9.2 实现 Serializable 接口 218

9.2.1 序列化对象图 220

9.2.2 控制序列化的行为 222

9.2.3 readResolve() 方法在
单例类中的运用 229

9.3 实现 Externalizable 接口 231

9.4 可序列化类的不同

版本的序列化兼容性 233

9.5 小结 235

9.6 练习题 236

第 10 章 Java 语言的反射机制 239

10.1 Java Reflection API 简介 239

10.2 在远程方法调用中运用
反射机制 244

10.3 代理模式 248

10.3.1 静态代理类 248

10.3.2 动态代理类 250

10.3.3 在远程方法调用中
运用代理类 253

10.4 小结 258

10.5 练习题 259

第 11 章 RMI 框架 261

11.1 RMI 的基本原理 262

11.2 创建第一个 RMI 应用 264

11.2.1 创建远程接口 264

11.2.2 创建远程类 265

11.2.3 创建服务器程序 267

11.2.4 创建客户程序 269

11.2.5 运行 RMI 应用 270

11.3 远程对象工厂设计模式 272

11.4 远程方法中的参数与
返回值传递 277

11.5 回调客户端的远程对象 281

11.6 远程对象的并发访问 286

11.7 分布式垃圾收集 289

11.8 远程对象的 equals()、
hashCode() 和 clone() 方法 294

11.9 使用安全管理器 294

11.10 RMI 应用的部署及类的
动态加载 295

11.11 远程激活 297

11.12 小结 303

11.13 练习题 304

第 12 章 通过 JDBC API

访问数据库.....	305
12.1 JDBC 的实现原理	306
12.2 安装和配置 MySQL 数据库.....	308
12.3 JDBC API 简介	310
12.4 JDBC API 的基本用法	314
12.4.1 处理字符编码的转换	317
12.4.2 把连接数据库的各种属性放在配置文件中	318
12.4.3 管理 Connection、Statement 和 ResultSet 对象的生命周期.....	321
12.4.4 执行 SQL 脚本文件.....	326
12.4.5 处理 SQLException	328
12.4.6 输出 JDBC 日志	329
12.4.7 获得新插入记录的主键值.....	329
12.4.8 设置批量抓取属性	330
12.4.9 检测驱动器使用的 JDBC 版本	330
12.4.10 元数据	331
12.5 可滚动及可更新的结果集	333
12.6 行集	339
12.7 调用存储过程	346
12.8 处理 Blob 和 Clob 类型数据.....	347
12.9 控制事务	351
12.9.1 事务的概念	351
12.9.2 声明事务边界的概念	353
12.9.3 在 mysql.exe 程序中声明事务	354
12.9.4 通过 JDBC API 声明事务边界	356
12.9.5 保存点	357
12.9.6 批量更新	358
12.9.7 设置事务隔离级别	360
12.10 数据库连接池	362
12.10.1 创建连接池	363

12.10.2 DataSource 数据源.....	369
-----------------------------	-----

12.11 小结.....	371
12.12 练习题.....	372

第 13 章 基于 MVC 和 RMI 的

分布式应用	375
13.1 MVC 设计模式简介.....	375
13.2 store 应用简介	377
13.3 创建视图.....	381
13.4 创建控制器.....	389
13.5 创建模型.....	390
13.6 创建独立应用.....	394
13.7 创建分布式应用.....	395
13.8 小结.....	398
13.9 练习题.....	398

第 14 章 通过 JavaMail API

收发邮件	401
14.1 E-mail 协议简介	401
14.1.1 SMTP 简单邮件传输协议	401
14.1.2 POP3 邮局协议	402
14.1.3 接收邮件的新协议 IMAP	402
14.1.4 MIME 简介	403
14.2 JavaMail API 简介	403
14.3 建立 JavaMail 应用程序的开发环境	405
14.3.1 获得 JavaMail API 的类库	405
14.3.2 安装和配置邮件服务器.....	406
14.4 创建 JavaMail 应用程序	408
14.5 身份验证.....	412
14.6 URLName 类	416
14.7 创建和读取复杂电子邮件	418
14.7.1 邮件地址	419
14.7.2 邮件头部	420
14.7.3 邮件标记	421

14.7.4 邮件正文	422
14.8 操纵邮件夹	427
14.9 小结	432
14.10 练习题	433
第 15 章 安全网络通信	435
15.1 SSL 简介	435
15.1.1 加密通信	436
15.1.2 安全证书	436
15.1.3 SSL 握手	437
15.1.4 创建自我签名的 安全证书	438
15.2 JSSE 简介	439
15.2.1 KeyStore、KeyManager 与 TrustManager 类	442
15.2.2 SSLContext 类	443
15.2.3 SSLServerSocketFactory 类	444
15.2.4 SSLSocketFactory 类	444
15.2.5 SSLSocket 类	444
15.2.6 SSLServerSocket 类	447
15.2.7 SSLEngine 类	448
15.3 创建基于 SSL 的安全 服务器和安全客户	453
15.4 小结	457
15.5 练习题	457
第 16 章 CORBA 简介	459
16.1 创建 IDL 接口	460
16.2 创建 IDL 接口的实现类	460
16.3 创建服务器程序	461
16.4 创建客户程序	462

16.5 运行 CORBA 程序	463
16.6 小结	464
16.7 练习题	465
第 17 章 Web 服务简介	467
17.1 SOAP 简介	467
17.2 建立 Apache AXIS 环境	469
17.3 在 Tomcat 上发布 Apache-AXIS Web 应用	470
17.4 创建 SOAP 服务	471
17.4.1 创建提供 SOAP 服务的 Java 类	471
17.4.2 创建 SOAP 服务的 发布描述符文件	471
17.5 管理 SOAP 服务	472
17.5.1 发布 SOAP 服务	472
17.5.2 删除 SOAP 服务	473
17.6 创建和运行 SOAP 客户程序	473
17.7 发布 JWS 服务	476
17.8 小结	476
17.9 练习题	477
附录 A 本书范例的运行方法	479
A.1 本书所用软件的下载地址	479
A.2 部分软件的安装	479
A.2.1 安装 JDK	480
A.2.2 安装 ANT	480
A.2.3 安装 Tomcat	481
A.3 编译源程序	481
A.4 运行客户/服务器程序	482

第 1 章 Java 网络编程入门

所有上过网的人都熟悉这样的过程：打开浏览器程序，输入一个 URL 地址，这个地址指向的网页就会从远程 Web 服务器传到客户端，然后在浏览器中显示出来。网络编程最基础的任务就是开发像浏览器这样的客户程序，以及像 Web 服务器这样的服务器程序，并且使两者能有条不紊地交换数据。

张三给李四邮寄一封信，张三不必亲自把信送到李四家里，送信的任务由邮政网络来完成。张三只需提供李四的地址，邮政网络就会准确地把这封信送达目标地址。同样，服务器程序与客户程序只需关心发送什么样的数据给对方，而不必考虑如何把这些数据传输给对方，传输数据的任务由计算机网络来完成，如图 1-1 所示。

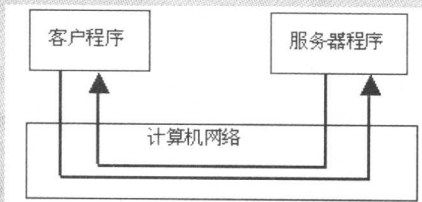


图 1-1 计算机网络负责传输通信数据

由此可见，网络应用程序建立在计算机网络的基础上。本章介绍计算机网络的一些基本概念，重点介绍了网络的分层思想和 TCP/IP 协议。最后以 EchoServer 和 EchoClient 为例，介绍如何利用套接字来创建一个简单的 Java 服务器程序和 Java 客户程序。

1.1 进程之间的通信

进程是指运行中的程序，进程的任务就是执行程序中的代码。如例程 1-1 所示的 EchoPlayer 类是一个独立的 Java 程序，它可以在任意一台安装了 JDK 的主机上运行。EchoPlayer 类不断读取用户从控制台输入的任意字符串 XXX，然后输出 echo:XXX。如果用户输入的字符串为“bye”，就结束程序。例程 1-1 为 EchoPlayer 类的源代码。

例程 1-1 EchoPlayer.java

```
import java.io.*;

public class EchoPlayer {
    public String echo(String msg) {
        return "echo:"+msg;
    }

    public void talk()throws IOException {
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        String msg=null;
        while((msg=br.readLine())!=null){
```

```
System.out.println(echo(msg));
if(msg.equals("bye"))           //当用户输入"bye", 结束程序
    break;
}
}

public static void main(String arg[])throws IOException{
    new EchoPlayer().talk();
}
}
```

运行“java EchoPlayer”命令，就启动了 EchoPlayer 进程，该进程执行 EchoPlayer 类的 main()方法。如图 1-2 所示演示了 EchoPlayer 进程的运行过程，它从本地控制台中获得标准输入流和标准输出流。本地控制台为用户提供了基于命令行的用户界面，用户通过控制台与 EchoPlayer 进程交互。

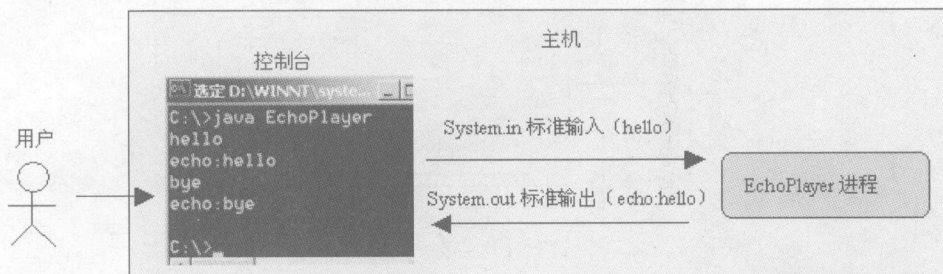


图 1-2 EchoPlayer 进程的运行过程

Tips

确切地说，运行“java EchoPlayer”命令，就启动了一个 JVM（Java Virtual Machine, Java 虚拟机）进程，该进程执行 EchoPlayer 类的 main()方法。本书为了叙述的方便，把运行 EchoPlayer 类的 main()方法的 JVM 进程直接称为 EchoPlayer 进程。

EchoPlayer 类的 echo(String msg)方法负责生成响应结果。如果需要把生成响应结果的功能（即 echo(String msg)方法）移动到一个远程主机上，那么上面的 EchoPlayer 类无法满足这一需求。在这种情况下，要创建两个程序：客户程序 EchoClient 和服务程序 EchoServer。EchoClient 程序有两个作用：

- 与用户交互，从本地控制台获得标准输入流和标准输出流；
- 与远程的 EchoServer 通信，向 EchoServer 发送用户输入的字符串，接收 EchoServer 返回的响应结果，再把响应结果写到标准输出流。

EchoServer 程序负责接收 EchoClient 发送的字符串，然后把响应结果发送给 EchoClient。如图 1-3 所示演示了 EchoClient 与 EchoServer 的通信过程。客户机和远程服务器是通过网络连接的两台主机。客户机上运行 EchoClient 进程，远程服务器上运行 EchoServer 进程。

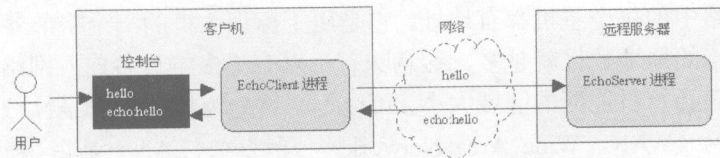


图 1-3 EchoClient 与 EchoServer 的通信过程

张三给李四打电话，两者顺利通话的前提条件是他们各自的电话机都连接到了电话网络上。张三和李四只需关注他们谈话的具体内容，而不必考虑如何把自己的语音传输到对方的电话机上。传输语音信息的任务是由电话网络来完成的。

同样，两个进程顺利通信的前提条件是它们所在的主机都连接到了计算机网络上。EchoClient 与 EchoServer 只需关注它们通信的具体内容，例如，EchoClient 发送信息“hello”，那么 EchoServer 返回信息“echo:hello”。EchoClient 和 EchoServer 都无须考虑如何把信息传输给对方。传输信息的任务是由计算机网络来完成的。

Java 开发人员的任务是编写 EchoClient 和 EchoServer 程序，接下来在两台安装了 JDK 的主机上分别运行它们，两个进程就会有有条不紊地通信。

由于进程之间的通信建立在计算机网络的基础上，Java 开发人员有必要对计算机网络有基本的了解，这有助于更容易地掌握 Java 网络编程技术。

1.2 计算机网络的概念

计算机网络是现代通信技术与计算机技术相结合的产物。所谓计算机网络，是指把分布在不同地理区域的计算机用通信线路互连起来的一个具有强大功能的网络系统。在计算机网络上，众多计算机可以方便地互相通信，共享硬件、软件和数据信息等资源。通俗地说，计算机网络就是通过电缆、电话线或无线通信设施等互连的计算机的集合。

网络中每台机器称为节点（node）。大多数节点是计算机，此外，打印机、路由器、网桥、网关和哑终端等也是节点。在本书中，用节点一词指代网络中的任意一个设备，用主机指代网络中的计算机节点。

如图 1-4 所示，人与人之间通过某种语言来交流，网络中的主机之间也通过“语言”来交流，这种语言称为网络协议，这是对网络协议的通俗解释，在本章第 1.3 节，还会更深入地介绍网络协议的概念。

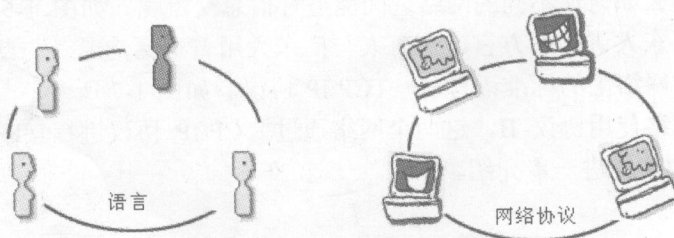


图 1-4 网络协议是网络中主机之间通信的语言

网络中的每个主机都有地址，它是用于标识主机的一个字节序列。字节序列越长，可以表示的地址数目就越多，这意味着可以有更多的设备连入网络。

按照计算机连网的区域大小，可以把网络分为局域网（LAN，Local Area Network）和广域网（WAN，Wide Area Network）。局域网（LAN）是指在一个较小地理范围内的各种计算机互联在一起的通信网络，可以包含一个或多个子网，通常局限在几千米的范围之内。例如，在一个房间、一座大楼或是在一个校园内的网络可称为局域网。广域网（WAN）连接地理范围较大，常常是一个国家或是一个洲，其目的是为了分布较远的各局域网互联。

到 Internet 海洋去冲浪，如今已成为一种时尚。Internet 是指国际互联网，也叫做因特网，是全球范围内的广域网，为世界上不同类型的计算机交换数据提供了通信媒介。

Internet 目前已经覆盖了 233 个国家和地区，连接着千千万万个子网，Internet 的用户超过了 10 亿。如图 1-5 所示为 www.internetworldstats.com 网站于 2006 年 12 月 1 日发布的 Internet 用户统计图。Internet 上汇聚了成千上万的信息资源，成为世界上信息资源最丰富的公共计算机网络。

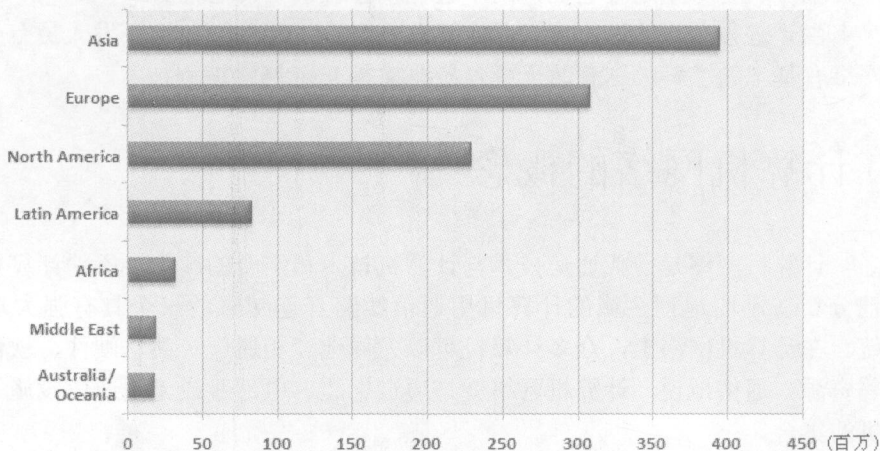


图 1-5 全球范围内的 Internet 用户

Internet 提供的服务包括 WWW（World-Wide Web）服务、电子邮件（E-mail）服务、文件传输（FTP）服务和远程登录（Telnet）服务等，全球用户可以通过这些服务，来获取 Internet 上的信息，或者开展各种业务。

Internet 是由许多小的网络互联成的国际性大网络，在各个小网络内部使用不同的协议，那么如何使不同的网络之间能进行信息交流呢？如图 1-6 所示，上海人讲上海方言，广东人讲广东方言，上海人与广东人用普通话沟通。与此相似，不同网络之间的互联靠网络上的标准语言——TCP/IP 协议。如图 1-7 所示，一个网络使用协议 A，另一个网络使用协议 B，这两个网络通过 TCP/IP 协议进行互联。本章第 1.4 节会对 TCP/IP 协议作进一步介绍。