

Java 网络应用编程入门

计算机网络技术系列教材

● 陈强 主编 孙建华 副主编 ●

Java 网络应用编程入门

Java 网络应用编程入门

朱喜福 郭逢昌 赵 玺 编

网络应用编程入门

Java 网络应用编程入门

各应用编程入门

网络应用编程入门

Java 网络应用编程入门

用编程入门

各应用编程入门

Java 网络应用编程入门

 人民邮电出版社
POSTS & TELECOM PRESS

计算机网络技术系列教材

Java 网络应用编程入门

朱喜福 郭逢昌 赵 玺 编

人 民 邮 电 出 版 社

图书在版编目 (CIP) 数据

Java 网络应用编程入门 / 朱喜福, 郭逢昌, 赵玺编. —北京: 人民邮电出版社, 2005.10
(计算机网络技术系列教材)

ISBN 7-115-13989-X

I. J... II. ①朱...②郭...③赵... III. JAVA 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 106928 号

内 容 提 要

本书是为 Java 学习者了解 Java 编程的基础内容后, 进一步比较全面地学习 Java 网络应用编程而编写的入门教材。全书内容分为 4 章。第 1 章讲述 Java 的网络编程基础, 包括 Java 网络编程的基本概念, URL 类和 URLConnection 类的使用, 套接字 (Socket) 通信和 UDP 数据报通信及其应用。第 2 章使用 MySQL 讲述 Java 的 JDBC 数据库应用, 包括 SQL 语言的基本知识, 通过 JDBC 来操作 SQL 语句, 在 Java 语言中操作数据库及数据库的应用。第 3 章讲述 JSP 及 JavaBean 网站编程, 包括 Tomcat 介绍及其安装, JSP 语法对象, JSP 隐含对象及其主要 API 等, 并通过综合实例讲述基于 Web 的电子商务网站框架和实现。第 4 章讲述 J2ME 开发简单的手机应用程序, 包括 J2ME 开发环境及配置, 高、低级用户界面编程和事件响应, 记录管理系统 (RMS) 应用, 短信交互和 HTTP 通信, 简单的手机游戏编程等。

本书内容详尽, 取舍和安排恰当, 讲解通俗易懂, 循序渐进, 实例丰富, 并注重培养学生解决实际问题的能力。本书每章都安排了大量有针对性的上机编程练习题, 便于教师教学和检验学生的学习效果, 使学生对所学内容得到巩固和加强。本书可作为高等院校 Java 网络应用编程入门课程的教材和教学参考书, 对于具有一定的 Java 编程基础, 并希望进一步熟悉 Java 网络编程、数据库编程、商务网站开发和手机编程的读者, 是一本理想的入门参考书。

计算机网络技术系列教材

Java 网络应用编程入门

-
- ◆ 编 朱喜福 郭逢昌 赵 玺
责任编辑 潘春燕
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京隆昌伟业印刷有限公司印刷
新华书店总店北京发行所经销
 - ◆ 开本: 787×1092 1/16
印张: 15
字数: 359 千字 2005 年 10 月第 1 版
印数: 3 001—5 000 册 2006 年 2 月北京第 2 次印刷

ISBN 7-115-13989-X/TP · 4961

定价: 22.00 元

读者服务热线: (010) 67170985 印装质量热线: (010) 67129223

丛书前言

计算机网络技术是近年来兴起的、发展相当迅速的计算机新技术。21 世纪的今天不会使用网络就等于不会使用计算机。IBM 公司早在 20 世纪 80 年代就提出：网络就是计算机。因此，必须十分重视计算机网络的普及与推广应用，使更多的人能够利用网上的资源提高工作水平和效率。

为适应社会的需要和计算机网络技术的发展，全国高等院校的各个专业都开设了有关计算机网络技术的课程。特别是近年来高等职业教育的发展，急需以计算机网络应用为主的实用教材，使学生在学计算机网络时，适当减少那些枯燥难懂的理论，取而代之的是建网、管网、上网的实际操作和网络的应用开发技术。根据这一发展趋势，我们组织了一批学术水平高、教学经验丰富、实践能力强的老师编写了这套计算机网络技术系列教材。

本套教材共 11 本。为了便于教学，本套教材每本均配有实训内容，且每一章都有内容提要 and 习题。

本套教材在编写过程中力求做到：网络理论以必需、够用为度，注重网络实用技术及实际应用的介绍，并以实际需要的技术、操作和使用技巧为主体，使学生在学计算机网络理论的同时，掌握相关的实际操作和应用技巧。本套教材突出了内容新、讲述方法浅显、重应用和技术的特点，适合高等职业学校、成人高校以及本科院校设立的二级职业技术学院学生和自学计算机网络应用和开发的人员使用。

本套教材已经被选为北京市高等教育自学考试高职自考“网络技术应用与服务”专业的指定教材。

编者的话

由于计算机网络的普及和基于网络应用的信息技术的迅猛发展,对计算机网络应用人才的需求也在不断增加。在网络程序设计应用领域,Java 面向对象编程技术正在逐步成为计算机网络应用开发的主流。电子商务、远程教学、网络游戏等都在纷纷使用 Java 技术,Java 手机编程和基于 Java 技术的各种芯片的应用等也在日常生活中随处可见。从当前的网络编程应用的发展来看,Java 网络编程技术有比较广阔的应用前景,许多与网络技术应用相关的行业对 Java 程序员的需求也在不断增加。相比其他的编程语言,学习和掌握 Java 更为容易。因此,全国各地高校与计算机相关的很多专业,尤其是计算机网络应用技术和其他相关专业的高职高专,更是将 Java 语言作为学习和掌握面向对象编程技术的必修课,相继开设了 Java 程序设计或与 Java 技术相关的课程,并将重点放在培养掌握应用技术为主的中等层次人才上,以适应当前社会对信息技术和网络应用人才的需要。

在掌握了有关 Java 编程的基础知识,熟悉基本的 Java 类库后,教师和学生都希望对基于 Web 应用的 Java 编程和基于 J2ME 的手机应用编程有一定的了解,以便进一步熟悉与 Java 相关的网络编程应用,为就业打下较好的技术基础。但由于学生的知识基础、学习能力和水平参差不齐,也造成了教学过程中传授知识和培养应用能力的难度。为此,本书作者根据多年从事 Java 教学所积累的教学经验和实际开发中的一些体会,并根据高职高专学生的学习兴趣、学习特点、知识基础和能力水平等,选择了一些比较适合高职高专相关专业 Java 网络编程教学的内容作为教材的主要内容。本书内容力求新颖,结构合理、概念清晰,通俗易懂和有较强的实用性。例题的选择既考虑加深对知识的理解和掌握,又考虑到学生的学习兴趣 and 编程应用,并进行详细的讲解和分析。每章实训练习题的编写具有针对性,以帮助学生巩固和拓展所学知识,提高实际编程的能力。

本书内容包括基于 C/S 的 Java 网络编程,包含后台数据库、基于 Web 应用的 Java 网络编程,基于 J2ME 的手机应用编程等。第 1 章讲述 Java 网络编程的基本概念,使用 URL 类和 URLConnection 类及方法访问 Web 服务器的客户应用程序,基于 TCP/IP 的套接字(Socket)通信应用编程,UDP 数据报通信及其应用编程。第 2 章使用 MySQL 讲述 SQL 语言的基本知识,通过 JDBC 来操作 SQL 语句,运用 Java 操作数据库和实现 JDBC 数据库应用。第 3 章讲述 JSP 及 JavaBean 网站编程,对 Tomcat 的安装和配置、JSP 编程应用的相关知识和主要 API 等进行了比较详细的说明,并通过综合实例讲述了基于 Web 的电子商务网站框架和实现。第 4 章讲述在 J2ME 下开发简单的手机应用程序,详细说明了 J2ME 开发环境的配置,高、低级用户界面编程和各种事件响应,记录管理系统(RMS)应用,短信交互和 HTTP 通信等。

本书可作为人民邮电出版社出版的《Java 程序设计》(ISBN 7-115-13005-1/TP.4399,朱喜福等编写)教材的后续教材或实训教材,根据教学课时的多少选择部分或全部内容进行讲

解和实训，为学生掌握 Java 网络应用编程提供很好的入门，为进一步学习与 Java 相关的网络应用编程技术打下比较扎实的基础。参加本书编写的人员还有张志纯、乐娟。

编 者

2005 年 7 月

目 录

第 1 章 Java 网络编程基础	1
1.1 Java 网络编程的基本概念	1
1.1.1 域名和 IP 地址	1
1.1.2 服务器和客户机	3
1.1.3 端口号	3
1.2 URL 类和 URLConnection 类	4
1.2.1 小程序访问 WWW 服务器	4
1.2.2 小型浏览器	6
1.2.3 通过 URLConnection 连接到 WWW	10
1.3 套接字 (Socket) 通信	12
1.3.1 TCP 客户机/服务器 (C/S) 通信简介	12
1.3.2 多客户机/服务器通信	17
1.4 UDP 数据报通信	23
1.4.1 UDP 数据报通信	23
1.4.2 简单的 UDP 客户机/服务器 (C/S) 通信	24
1.5 综合实例——简单的 C/S 考试系统	29
1.5.1 C/S 考试系统的服务器端程序的设计	29
1.5.2 C/S 考试系统的客户端程序的设计	30
1.5.3 C/S 考试系统的运行和套接字通信的实现说明	31
1.5.4 C/S 考试系统的服务器端程序实现	34
1.5.5 C/S 考试系统的客户端程序实现	45
练习题	57
第 2 章 JDBC 数据库应用	58
2.1 关系数据库管理系统	58
2.2 SQL 基础	60
2.2.1 SQL 数据类型	60
2.2.2 数据定义言语 (DDL)	61
2.2.3 数据处理语言 (DML)	63
2.3 Java JDBC	66
2.3.1 什么是 JDBC	66
2.3.2 JDBC 如何工作	67
2.3.3 JDBC 驱动	67
2.3.4 连接池	68

2.3.5 事务操作	68
2.4 JDBC 应用	68
2.4.1 常用接口及类	69
2.4.2 建立数据库连接	69
2.4.3 关闭数据库连接	70
2.4.4 添加新数据	71
2.4.5 查询数据	72
2.4.6 修改数据	73
2.4.7 删除数据	74
2.4.8 JDBC 操作流程	74
2.5 综合实例	76
2.5.1 Category 商品类别类	77
2.5.2 CategoryManager 类别管理类	79
2.5.3 ConnectionManager 数据库连接管理类	81
2.5.4 DBCategory 商品类别数据库操作类	84
2.5.5 Console 功能演示主控类	88
练习题	93
第 3 章 JSP 及 JavaBean 网站编程	95
3.1 JavaServerPages 技术	95
3.1.1 网络请求与响应	95
3.1.2 什么是 JSP	95
3.1.3 JSP 的优势	96
3.1.4 JSP 与 Servlet	97
3.1.5 JSP 的执行过程	98
3.2 运行 JSP	99
3.2.1 Tomcat 介绍及其安装	99
3.2.2 部署目录结构	101
3.2.3 服务器配置	102
3.3 JSP 语法对象	104
3.3.1 页面指令	105
3.3.2 include 指令	106
3.3.3 声明	106
3.3.4 表达式	107
3.3.5 Scriptlet	107
3.3.6 JSP 注释	108
3.3.7 <jsp:useBean>操作	109
3.3.8 <jsp:setProperty>操作	109
3.3.9 <jsp:getProperty>操作	109

3.3.10	<jsp:param>操作	110
3.3.11	<jsp:include>操作	110
3.4	JSP 隐含对象及其主要 API	110
3.4.1	request	110
3.4.2	response	110
3.4.3	session	110
3.4.4	application	111
3.4.5	out	111
3.5	综合实例——电子商务网站框架	111
3.5.1	JavaBean 及其 Servlet	112
3.5.2	JSP 页面	135
	练习题	157
第 4 章	J2ME 开发手机应用程序入门	158
4.1	J2ME 开发环境	158
4.1.1	J2ME 概述	158
4.1.2	J2ME 开发环境配置	158
4.1.3	J2ME 应用程序编译及运行	160
4.2	高、低级用户界面编程和事件响应	164
4.2.1	高、低级用户界面编程和事件响应概述	164
4.2.2	高级用户界面	165
4.2.3	低级用户界面	181
4.2.4	高级事件响应	187
4.2.5	低级事件响应	188
4.3	记录管理系统 (RMS) 应用	191
4.3.1	记录管理系统 (RMS)	191
4.3.2	记录的存储	196
4.3.3	记录的遍历	200
4.3.4	记录的监听	203
4.4	短信交互和 HTTP 通信	207
4.4.1	短信交互	207
4.4.2	HTTP 通信	217
4.5	综合实例——猜数字游戏	222
	练习题	229

第 1 章 Java 网络编程基础

1.1 Java 网络编程的基本概念

Java 成功应用的一个重要领域是网络。Java 语言以其独立于平台、面向对象、分布式、多线程、完善的安全机制和不断增强的网络编程应用等特色，与因特网密切结合，使它在全球取得了巨大的成功，成为现代信息系统建设中的良好开发平台和运行环境。

Java 的网络类库支持多种 Internet 协议，包括 Telnet、FTP 和 HTTP (WWW) 等。TCP/IP 代表传输控制协议/网际协议，它们合在一起定义了因特网上的工作方式，TCP 可以将网络上的任何资源当作流来看待，IP 标识因特网上的计算机。

Java 的网络功能集成在几个包中，java.net 包及其子包提供了大量的类和接口，实现了很多基于网络编程的功能：提供了基于流的通信，使应用程序可以把 Internet 作为代码、数据及输入的来源；提供了基于数据包的通信，用于传输单独的信息包，如音频和视频等；处理 WWW 页面内容，对 HTML 和 HTTP 的支持等。通过使用 java.net 包及其子包中的相关类和接口，很容易编写实现网络应用的各种 C/S 或 B/S 通信程序。

1.1.1 域名和 IP 地址

为了辨别和确定网络中应用程序所要连接的计算机，因特网通过 IP 地址惟一标识网络中的每台计算机。IP 地址是用句点分隔的四组数字，每一组数字由 8 位二进制位组成。因此 IP 地址的每一组数字都不能超过 255，例如 202.98.32.111。另外一种大家最熟悉的标识网络中应用程序所要连接的计算机形式是 DNS（域名系统）。例如，假定域名是 success.com，域内有一台名为 xyz 的计算机，那么它的域名就可以是 xyz.success.com，这些域名通常收集到一个万维网（WWW）地址里。

Java 网络包 java.net 中的 InetAddress 类是 Java 中的 IP 地址封装类。InetAddress 类中没有公共的构造函数，但是它提供了 3 个静态方法得到 IP 地址。当调用这些方法时，若找不到主机则会抛出异常 UnknownHostException，因此其方法调用语句通常放在 try 块中，并通过 catch 捕捉异常。

(1) public static InetAddress getByName(String hostname)

将需要寻找的主机名作为参数，并使用 DNS 查找主机的 IP 地址。

(2) public static InetAddress[] getAllByName(String hostname)

返回一个包含与该主机名相对应的所有 IP 地址的数组。

(3) public static InetAddress getLocalHost()

当在一台计算机上执行 getLocalHost() 时，此方法将返回该计算机的 InetAddress。

值得注意的是，IP 版本 4（简称 Ipv4）规定的 IP 地址都是 32 位，目前由于 IPv4 地址已近耗尽，所以 IPv6 地址正逐渐代替 Ipv4 地址，Ipv6 地址是 128 位无符号整数。Java 网络包

java.net 中提供了 Inet4Address 类和 Inet6Address 类对他们进行处理, 这两个类都是 InetAddress 的子类。

例 1.1 编写应用程序, 根据域名自动到 DNS 上查找 “www.263.net” 的 IP 地址并显示。

(1) 源程序: GetIP.java

```
1:  import java.net.InetAddress;
2:  import java.net.UnknownHostException;
3:  public class GetIP {
4:      public static void main(String args[]) {
5:          try {
6:              InetAddress ad=InetAddress.getByName("www.263.net");
7:              //InetAddress ad=InetAddress.getLocalHost();
8:              System.out.println(ad);
9:          }
10:         catch(UnknownHostException e) {
11:             System.out.println(e.toString());
12:         }
13:     }
14: }
```

(2) 程序运行结果

如果运行该程序的计算机连接在因特网上, 并能访问 www.263.net, 则输出如图 1-1 (a) 所示的结果。如果没有网络连接, 那么可按下法在本机上模拟演示: 去掉第 7 行的 “//”, 并注释掉第 6 行。运行程序后显示本机的 InetAddress, 如图 1-1 (b) 所示。

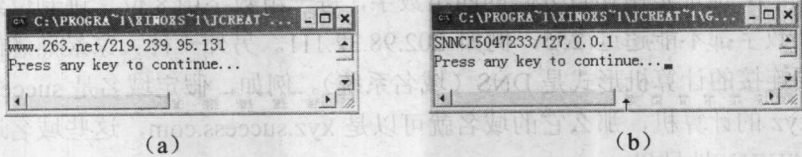


图 1-1 例 1.1 程序的运行结果

(3) 程序解释

程序第 6 行使用 InetAddress 类的静态方法 getByName() 得到网络中主机名为 “www.263.net” 的计算机的 IP 地址。由于使用方法 getByName() 得到 IP 地址时, 可能找不到主机而抛出异常 UnKnownHostException, 因此, 应把调用该方法的语句放在 try 块中, 并捕捉异常 UnKnownHostException。

使用 InetAddress 类的以下两个方法, 可将主机名或 IP 地址作为字符串返回。

- (1) public String getHostName(), 将主机名作为字符串返回。
- (2) public String getHostAddress(), 将 IP 地址作为字符串返回。

例 1.2 编写应用程序, 得到 IP 地址 “127.0.0.1” (本机) 所代表的主机名。

(1) 源程序: GetHostName.java

```
1:  import java.net.*;
```

```

2:  public class GetHostName {
3:      public static void main(String[] args) {
4:          try {
5:              InetAddress name=InetAddress.getByName("127.0.0.1");
6:              System.out.println("主机名:"+name.getHostName());
7:          }
8:          catch(UnknownHostException e){
9:              System.out.println(e.toString());
10:         }
11:     }
12: }

```

(2) 程序运行结果

编译并运行程序后输出如图 1-2 所示的结果。

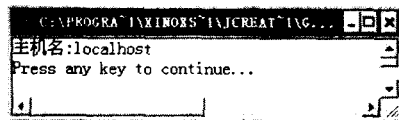


图 1-2 例 1.2 的运行结果

1.1.2 服务器和客户机

网络最基本的连接形式就是让两台计算机连接在一起，并相互通信。服务器的主要任务是侦听客户机建立连接的请求，而客户机的任务是试着与一台服务器建立连接。一旦连接好，就变成了一种双向通信。无论对于服务器端还是客户机，连接则成了一个 I/O 数据流对象，这时可以如同读、写一个普通的文件那样对待网络中连接上的两台计算机。

使用 `java.net` 包中提供的类，容易编程实现客户机/服务器模型（C/S 模型）的网络通信。C/S 结构通过消息传递机制，由客户端发出请求给服务器，服务器进行相应处理后将结果传送回客户端。

很多时候，基于因特网上的许多 Java 应用环境实际上是浏览器/服务器模型（B/S 模型，简称 Web 模型）。Java 的 Web 模型是三层结构，服务器端被分解成两部分：一部分是应用服务器（Web 服务器），另一部分是数据库服务器；客户端则是浏览器。本章 1.2 节只对 Java 中的 URL 连接做一些简单介绍。

针对分布式计算环境，Java 通过其网络类库提供了良好的支持。`java.net` 包中的 URL 类（Uniform Resource Locator）可用来创建一个 URL 对象，利用此对象可打开并访问 WWW 上的资源，其访问方式与访问本地文件系统几乎完全相同。Java 的客户机/服务器模式可以把运算从服务器分散到客户端，从而提高整个系统的执行效率，增加动态可扩展性。

1.1.3 端口号

“端口”并不是计算机的一个物理上存在的标识，而是一种软件抽象，客户端程序通过计算机的 IP 地址同服务器连接，通过指定端口使应用程序同所需要的服务通信。通常，每个服

务都同一台特定服务器上特定的端口编号关联在一起, 客户程序必须事先知道自己(或指定)所需的服务的端口号。请求一个特定的端口, 便相当于请求与那个端口关联的服务。在一台服务器中, 往往运行着多个服务程序, 设置一个客户机或者服务器的时候, 必须选择一个无论客户机还是服务器都认可连接的端口。

由 IP 表示的每台计算机包含了“端口”。端口号可以是 0~65535 之间的任意一个整数。一些操作系统将 1024 以下的端口保留给系统服务使用(如 WWW 服务, 电子邮件等), 因此, 用户程序中, 绑定可连接的端口时不应使用这些端口号。

1.2 URL 类和 URLConnection 类

1.2.1 小程序访问 WWW 服务器

URL 是统一资源定位器的简称, 它表示因特网上某一资源的地址。通过 URL 可以访问因特网和 WWW。一个 URL 包含两部分内容: 协议名和资源名, 中间用冒号隔开。

例如: `http://www.sun.com`

`http://www.263.net`

为了表示 URL, `java.net` 包中提供了 URL 类, 该类中包含许多方法用来创建 URL 对象。下面的两个构造方法可用来创建 URL 对象。

(1) `public URL(String str) throws MalformedURLException`

参数 `str` 是用字符串表示的 URL 地址。

(2) `public URL(String pro, String host, int port, String file) throws MalformedURLException`

参数 `pro` 是协议名, `host` 是主机名, `port` 是端口号, `file` 是访问的文件。

由于这两个构造方法头中声明抛出异常 `MalformedURLException`, 调用它们的语句通常放在 `try` 块中, 并通过 `catch` 块捕捉异常。

例 1.3 编写小程序, 通过 Applet 打开一个新的 IE 浏览器窗口, 访问指定的网站。程序中用 `JList` 对象 `namelist` 显示字符串, 如: “新浪”、“搜狐”、“网易”、“雅虎”等, 并通过内部类 `MyJList` 实现列表选择事件 (`ListSelectionEvent`) 的监听处理。通过 `HashMap` 对象 `sites` 存放键/值对应的映射, 其中键是 `namelist` 中的字符串, 值是它们对应的 URL 网址。使用 `AppletContext` 类的 `showDocument()` 方法在新建的浏览器窗口中显示所选择的网站的内容。

(1) 源程序: `BrowsePage.java`

```
1: import java.awt.*;
2: import javax.swing.*;
3: import javax.swing.event.*;
4: import java.util.HashMap;
5: import java.net.*;
6: import java.applet.AppletContext;
7:
8: public class BrowsePage extends JApplet {
9:
```

```

10:     HashMap sites;//声明一个 HashMap 对象, 用来存放键/值映射
11:     JList namelist; //声明一个列表对象
12:     String   name[]={"新浪","搜狐","网易","雅虎"}; //列表中的各项
13:     URL      url[]=new URL[4]; //创建存放 URL 对象的数组, 有 4 个 URL 对象
14:
15:     public void init() {
16:         try{//给数组元素赋值
17:             url[0]=new URL("http://www.sina.com.cn");
18:             url[1]=new URL("http://www.sohu.com.cn");
19:             url[2]=new URL("http://www.netease.com.cn");
20:             url[3]=new URL("http://www.yahoo.com.cn");
21:         }
22:         catch(MalformedURLException e){
23:             System.out.println(e.toString());
24:         }
25:
26:         sites = new HashMap();
27:         for(int i=0;i<name.length;i++){
28:             sites.put(name[i], url[i]);
29:         }
30:         Container container = getContentPane();
31:         container.add( new JLabel( "请选择要浏览的网站" ),
32:                        BorderLayout.NORTH );
33:         namelist = new JList( name);
34:         container.add( new JScrollPane(namelist), BorderLayout.CENTER );
35:         namelist.addListSelectionListener(new MyJlist());
36:     }
37:
38:     class MyJlist implements ListSelectionListener{
39:         public void valueChanged( ListSelectionEvent event ) {
40:             Object object =namelist.getSelectedValue();
41:             URL   context = ( URL ) sites.get( object );
42:             AppletContext browser = getAppletContext();
43:             browser.showDocument( context,"_blank");
44:         }
45:     }
46: }

```

与程序字节码文件 BrowsePage.class 在同一目录下网页文件: page1.htm。

<html>

```
<title>page1</title>
<body>
    <applet code = "BrowsePage.class" width = "200" height = "100">
    </applet>
</body>
</html>
```

(2) 程序运行结果

编译源程序 BrowsePage.java, 用 IE 打开网页 page1.htm, 显示界面如图 1-3 (a) 所示。若选中“新浪”, 则打开新浪网首页, 如图 1-3 (b) 所示。

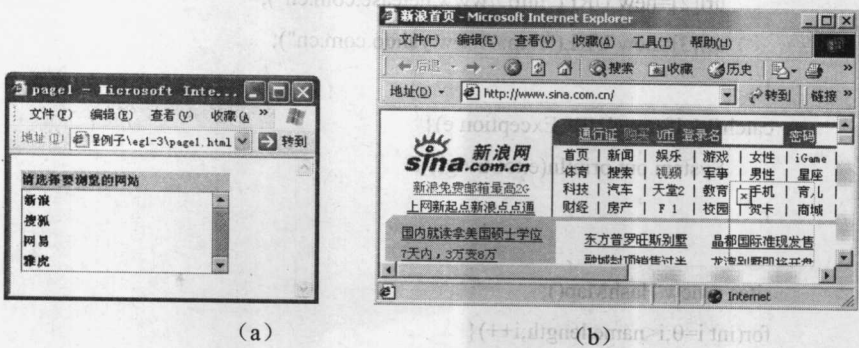


图 1-3 例 1.3 的运行结果

(3) 程序解释

程序第 16~24 行创建多个 URL 对象 (存放在数组中), 它们是列表 namelist 中各字符串所对应的网址。

程序第 26 行创建 HashMap 对象 sites, 第 27~29 行将键/值对应的映射增加到 sites 中, 其中键是字符串数组 name 的各元素, 值是 URL 数组 url 中的各个网址。

程序第 33~34 行创建 JList 对象 namelist, 并添加到面板容器 (缺省布局是 BorderLayout) 的中间, 第 35 行对 namelist 增加列表选择事件 (ListSelectionEvent) 监听, 列表选择事件的处理是由内部类 MyJlist 实现的。

程序第 38~45 行定义实现了 ListSelectionListener 接口的内部类 MyJlist。第 39~44 行实现 ListSelectionListener 接口中的方法 valueChanged(ListSelectionEvent event), 其中第 40 行得到列表中所选择的值并赋给 object, 第 41 行得到哈希映射中键为 object 的对应值, 它是一个 URL 对象, 第 42 行调用小程序类的方法 getAppletContext(), 该方法返回一个 AppletContext 对象, 第 43 行调用 AppletContext 的方法 showDocument() 在新建的浏览器窗口中显示所选择的网站的内容。

1.2.2 小型浏览器

Java 扩展包 javax.swing 中提供了 JEditorPane 类, 创建该类的对象后, 通过使用该类的 setPage() 方法可显示 URL 所指定的网页内容。在 JEditorPane 对象不能编辑的情况下, 它能生成并响应超链接事件 (HyperlinkEvent)。

① 显示网页内容的方法 `setPage()`

```
public void setPage(String url) throws IOException
```

方法功能是显示参数 `url` 指定的网页内容。请注意，由于网页中可能添加了其他脚本语言的脚本，有时并不能完全正确的显示。

② 监听并响应超链接事件的 `HyperlinkListener` 接口

通过 `HyperlinkListener` 接口中的方法：

```
public void hyperlinkUpdate(HyperlinkEvent e)
```

可响应超链接事件 (`HyperlinkEvent`)，但只有在 `JEditorPane` 不能编辑的情况下才生成超链接事件 (`HyperlinkEvent`)。

下面通过例 1.4 中自定义的具有 IE 功能的小型浏览器来说明它们的用法。

例 1.4 编写一个具有 IE 功能的小型浏览器。

(1) 源程序：GetHostName.java

```
1:  import java.awt.*;
2:  import java.awt.event.*;
3:  import javax.swing.*;
4:  import javax.swing.event.*;
5:  import java.io.*;
6:  import java.net.*;
7:  public class MyBrowser extends JFrame {
8:      JLabel    jLabel;
9:      JTextField tf;
10:     JEditorPane content;
11:
12:     public MyBrowser() {
13:         super("我的浏览器");
14:         Container con = getContentPane();
15:
16:         jLabel=new JLabel("输入连接的网址");
17:         tf = new JTextField("");
18:         tf.addActionListener(new MyFieldDealing());
19:
20:         JPanel panel=new JPanel();
21:         panel.setLayout(new GridLayout(2,1));
22:         panel.add(jLabel);
23:         panel.add(tf);
24:         con.add(panel, BorderLayout.NORTH);
25:
26:         content = new JEditorPane();
27:         content.setEditable( false );
```



```
28:         content.addHyperlinkListener(new MyHyperListener());
29:         con.add( new JScrollPane(content), BorderLayout.CENTER);
30:         setSize( 400, 300 );
31:         setVisible( true );
32:     }
33:
34:     void getThePage( String site ) {
35:         try {
36:             content.setPage(site); //在 content 中显示网页
37:         }
38:         catch ( IOException ex ) {
39:             JOptionPane.showMessageDialog(this, "连接错误", "错误提示",
40:                                     JOptionPane.ERROR_MESSAGE);
41:         }
42:     }
43:
44:     class MyFieldDealing implements ActionListener{//处理文本框回车事件
45:         public void actionPerformed( ActionEvent e ){
46:             String str=tf.getText(); //得到网址
47:             getThePage(str); //调用自定义方法显示指定网址的网页内容
48:         }
49:     }
50:
51:     class MyHyperListener implements HyperlinkListener{//处理超链接事件
52:         public void hyperlinkUpdate( HyperlinkEvent e ) {
53:             if ( e.getEventType() ==HyperlinkEvent.EventType.ACTIVATED ){
54:                 String str= e.getURL().toString(); //得到超链接的网页地址
55:                 getThePage(str); //显示超链接指向的网页内容
56:             }
57:         }
58:     }
59:
60:     public static void main( String args[] ) {
61:         MyBrowser mb= new MyBrowser();
62:         mb.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
63:     }
64: }
```

(2) 程序运行结果

程序运行后,若计算机连接在因特网中,在文本框中输入新浪网的网址: <http://www.sina.com/>,