

高等学校计算机科学与技术教材

# Java 网络高级编程

殷兆麟 张永平 姜淑娟 编著

清华大学出版社

北京交通大学出版社

·北京·

## 内 容 简 介

本书第1~3章介绍了网络进程通信的主要技术。其中,第1章介绍网络进程通信的基础类(InetAddress、URL、Socket等);第2章介绍Java远程方法调用RMI;第3章介绍Java与CORBA不同体系的组件间的通信技术。第4~6章主要介绍Java有关的Web技术。其中,第4章介绍Javabeans技术;第5章介绍Servlet技术;第6章介绍JSP技术。第7~9章主要介绍Java有关的EJB及相关的技术。其中,第7章介绍Java XML技术;第8章介绍异步消息服务技术JMS和Java Mail技术;第9章介绍利用WebLogic为服务器的EJB技术。第10章介绍Java安全体系结构、密钥、消息摘要、数字签名、访问控制、安全套接口SSL、主体身份认证JAAS和公共密钥构架PKI技术。第11章简明地介绍移动代理编程技术。

本书可作为高校计算机应用、网络信息、电子商务技术等专业高年级学生及研究生的教材,也可以作为Java网络编程爱好者的自学参考书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

(本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。)

### 图书在版编目(CIP)数据

Java网络高级编程/殷兆麟,张永平,姜淑娟编著. —北京:清华大学出版社;北京交通大学出版社,2005.5

(高等学校计算机科学与技术教材)

ISBN 7-81082-503-8

I. J... II. ①殷... ②张... ③姜... III. Java语言—程序设计—高等学校—教材  
IV. TP312

中国版本图书馆CIP数据核字(2005)第019260号

责任编辑:谭文芳

出版者:清华大学出版社 邮编:100084 电话:010-62776969

北京交通大学出版社 邮编:100044 电话:010-51686414

印刷者:北京东光印刷厂

发行者:新华书店总店北京发行所

开本:185×260 印张:25.25 字数:640千字

版次:2005年5月第1版 2005年5月第1次印刷

书号:ISBN 7-81082-503-8/TP·185

印数:1~5000册 定价:34.00元

---

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。

投诉电话:010-51686043, 51686008; 传真:010-62225406; E-mail: press@center.bjtu.edu.cn。

# 前 言

计算机应用类的专业课程以 Java 技术为主线已得到众多的 IT 教育专家的共识。“网页设计技术基础”、“Java 语言程序设计”、“Java 网络高级编程”、“面向对象的软件工程——UML 及其软件系统建模”及“Web Service 技术”组成的系列化、模块化、立体化专业教材就是服务于这一教学目的的。该系列课程涵盖了当代开发电子商务、电子政务、企业 ERP、建立和维护网站及系统信息集成的主要技术，避免了按应用设置课程而引起的课程内容前后不衔接、或专业知识重复讲授等缺点。学生专业技术训练从低年级网页设计技术开始，有利于学生在校学习期间积累系统开发的经验，适应 IT 人才市场需要。

“Java 网络高级编程”是 Java 技术教学的第二阶段，该课程的先导课程为：网页设计技术基础、Java 语言程序设计、数据库和计算机网络。教学实践表明，“Java 网络高级编程”有利于提高学生综合使用语言、网页制作、数据库、网络等众多课程知识解决问题的能力。

本教材突出两个工程示例教学：一是网上聊天系统，它分别用 Socket 技术、RMI 技术、RMI-IIOP 技术实现；二是网上书店系统，它分别重点应用 JSP 技术、EJB 技术实现。教材强调工程应用。

本教材的编写得到江苏省新世纪高等教育教学改革工程“网络技术与信息处理专业人才培养规格、模式、方案及其实践的研究”项目的资助，得到江苏省教委研究生百门优秀课程建设基金的资助，得到江苏省教委研究生 30 门公开课程建设资金的资助，配套课件得到中国矿业大学教改基金资助，在此表示衷心感谢。

衷心感谢东南大学教育部网络重点实验室、南京大学国家软件新技术重点实验室分别资助的“代理平台的研究与开发”和“代理规范的融合与代理平台的安全”科研项目，本教材从一个侧面体现了这些项目带来的成果。

衷心感谢我国著名人工智能科学家史忠植向我们提供 MAGE 代理平台。

教材编写过程中引用了 JBuilder, WebLogic, EJBCA 等产品的部分资料，在此向有关公司、作者表示衷心的感谢。

教材编写中，张爱娟、孙晋非、戴磊、孙进成、谢宜才、王竹晓、李宁、朱红、邵安贤、张凝、张丽丽、杨林森等帮助编写示例、调试程序、制作课件，做了大量工作，在此深表谢意。最后，感谢汪玉华女士所做的大量文字编辑工作。

本教材配套的课件可以从北京交通大学出版社网站 <http://press.bjtu.edu.cn> 和中国矿业大学 <http://www.cumt.edu.cn> 主页进入“网络教学”自由下载。如果使用中出现的问题，可以与作者殷兆麟联系：zhlyin@cumt.edu.cn。

由于时间仓促和作者水平有限，书中难免有不妥和错误之处，恳请广大读者指正。

殷兆麟

2005 年 2 月于徐州

## 内 容 简 介

本书第1~3章介绍了网络进程通信的主要技术。其中,第1章介绍网络进程通信的基础类(InetAddress、URL、Socket等);第2章介绍Java远程方法调用RMI;第3章介绍Java与CORBA不同体系的组件间的通信技术。第4~6章主要介绍Java有关的Web技术。其中,第4章介绍Javabeans技术;第5章介绍Servlet技术;第6章介绍JSP技术。第7~9章主要介绍Java有关的EJB及相关的技术。其中,第7章介绍Java XML技术;第8章介绍异步消息服务技术JMS和Java Mail技术;第9章介绍利用WebLogic为服务器的EJB技术。第10章介绍Java安全体系结构、密钥、消息摘要、数字签名、访问控制、安全套接口SSL、主体身份认证JAAS和公共密钥构架PKI技术。第11章简明地介绍移动代理编程技术。

本书可作为高校计算机应用、网络信息、电子商务技术等专业高年级学生及研究生的教材,也可以作为Java网络编程爱好者的自学参考书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

### 图书在版编目(CIP)数据

Java网络高级编程/殷兆麟,张永平,姜淑娟编著. —北京:清华大学出版社,北京交通大学出版社,2005.5

(高等学校计算机科学与技术教材)

ISBN 7-81082-503-8

I. J... II. ①殷... ②张... ③姜... III. Java语言—程序设计—高等学校—教材  
IV. TP312

中国版本图书馆CIP数据核字(2005)第019260号

责任编辑:谭文芳

出版者:清华大学出版社 邮编:100084 电话:010-62776969 <http://www.tup.com.cn>

北京交通大学出版社 邮编:100044 电话:010-51686414 <http://press.bjtu.edu.cn>

印刷者:北京东光印刷厂

发行者:新华书店总店北京发行所

开本:185×260 印张:25 字数:640千字

版次:2005年5月第1版 2005年5月第1次印刷

书号:ISBN 7-81082-503-8/TP·185

印数:1~5000册 定价:34.00元

---

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。

投诉电话:010-51686043, 51686008; 传真:010-62225406; E-mail: [press@center.bjtu.edu.cn](mailto:press@center.bjtu.edu.cn)

# 第 1 章 Java 基本网络类

## 本章导读

本章介绍 Java 基本的网络类（InetAddress、URL、URLConnection 和 Socket）的使用，以及如何利用 Socket 通信机制实现客户 – 服务器方式的一对一的通信，实现浏览器 – 服务器方式的网络聊天。

## 1.1 Java 网络进程通信概述

网络编程一般指利用不同层次的通信协议提供的接口实现网络进程安全通信和应用的编程。

网络进程就是网点机（连入网络的计算机）上运行的程序。这里不讲两个网点机之间的通信是因为一个网点机上可能同时运行多个程序，也就是说，多个进程并行运行。所以，两个网点机之间的通信的表述是不确切的。网络进程在通信协议中用端口（port）标识，它驻留的网点机用其 IP 地址或域名来标识。

网络进程同步通信，是指通信的发起者向接收者发出服务请求后，必须得到对方回应，才继续运行。如果通信的发起者向接收者发出服务请求后，可以继续运行，需要时再获取接收者的响应，这叫异步通信。

网络进程通信的实现是一个复杂的问题，它涉及网点计算机硬件、计算机操作系统、通信双方使用语言、网络通信协议等。网络协议工程就是利用软件工程的方法（分层次、分模块等技术）来解决复杂网络进程通信的问题。网络进程往往按照应用的需要，采用不同层次的协议进行通信。

现代计算机操作系统，包括个人计算机 Windows 操作系统，都具有不同程度支持网络通信管理的能力。实现互联网 TCP/IP 协议成为现代操作系统功能的一部分。

## 1.2 Java 网络基本类

### 1.2.1 InetAddress 类

IP 地址是 TCP/IP 网络中的每台计算机都有的唯一地址。在 Java 中，InetAddress 类用来描述 Internet 地址的类。

#### 1. Java 获取本机的 IP 地址

例 1-1 利用 Java InetAddress 类获取本机的 IP 地址。

```
package chapter1 ;
import java.net.* ;
import java.io.* ;
public class GetLocalHost {
    public static void main(String [ ]arg)
    {   InetAddress myIp = null ;
        // InetAddress 类不用构造函数
        try {myIp = InetAddress.getLocalHost();

            }
            catch (UnknownHostException e){}
        System.out.println(myIp);
    }
}
```

## 2. Java 根据域名获得 IP 地址

例 1-2 根据域名自动到 DNS 上查找 IP 地址。

```
package chapter1 ;
import java.net.* ;
import java.io.* ;
public class GetIP
{   public static void main(String args[ ])
    {   InetAddress cumt = null ;
        try {
            cumt = InetAddress.getByName("www.cumt.edu.cn");
        }
        catch (UnknownHostException e)
        {
        }
        System.out.println(cumt);
    }
}
```

可以看到，与 DNS 服务器连接只用了一条语句。

### 1.2.2 URL 类

#### 1. URL 类简介

统一资源定位地址 (Uniform Resource Locator, URL) 规范了 WWW 资源网络定位地址的表示方法。WWW 资源包括 Web 页、文本文件、图形文件、声频片段等。

URL 类定义了 WWW 资源的特征及读其内容的方法。其基本表示格式如下：

```
protocol ://hostname :/resourcename# anchor
```

其中：protocol 表示使用的协议，它可以是 HTTP，FTP，NEWS，Telnet 等；

hostname 表示主机名，指定域名服务器（DNS）能访问到的 WWW 服务的计算机，如 www.sun.com；

port 表示端口号，是可选的，表示所连的端口号，如默认，将连接到协议默认的端口；

resourcename 表示资源名，是主机上能访问到的目录或文件；

anchor 表示标记，也是可选的，它指定文件内的有特定标记的位置。

下面是几个合法的 URL 例子：

```
http://www.ncsa.uiuc.edu/demoweb/url-primer.htm
```

```
http://www.ncsa.uiuc.edu:8080/demoweb/url-primer.htm
```

```
ftp://local/demo/information#myinfo
```

```
file://local/demo/readme.txt
```

上面第 2 个 URL 把标准 Web 服务器端口 80 指定为 8080 端口；第 3 个 URL 加上符号“#”，用于指定在文件 information 中标记为 myinfo 的部分。

## 2. URL 类对象

URL 类的构造方法有如下四种。

### (1) URL (URL absoluteURL)

利用绝对 URL 地址，创建一个 URL 对象。

参数 absoluteURL 指绝对 URL 地址。例如：

```
URL myUni = URL("http://www.cumt.edu.cn/");
```

### (2) URL (URL url, String relativeURL)

利用已建立的 URL 对象和相对 URL 地址，创建一个 URL 对象。

参数 url 指 URL 对象，relativeURL 指相对 URL 地址。例如：

```
URL myUni = URL("http://www.cumt.edu.cn/");
```

```
URL mydoc = URL(myUni, "mydoc.html") // mydoc.html 是网站 http://www.cumt.edu.cn/ 的一个文件
```

### (3) URL (String protocol, String host, String resourcename)

利用 protocol, host, resourcename 创建一个 URL 对象。

参数 protocol 指协议，host 指主机，resourcenameurl 指资源名。例如：

```
URL myUni = URL("http"; "www.cumt.edu.cn"; "/mydoc.html");
```

这种方法与下列方法是等价的：

```
URL myUni = URL("http://www.cumt.edu.cn/mydoc.html");
```

### (4) URL (String protocol, String host, int port, String resourcename)

利用 protocol, host, port, resourcename 创建一个 URL 对象。

参数 protocol 指协议，host 指主机，port 指端口号，resourcenameurl 指资源名。例如：

```
URL myUni = URL("http"; "www.cumt.edu.cn"; 80; "/mydoc.html");
```

这种方法与下列方法是等价的：

此为试读，需要完整PDF请访问：[www.ertongbook.com](http://www.ertongbook.com)

```
URL myUni = URL("http://www.cumt.edu.cn %0A/mydoc.html");
```

URL 类的构造方法中的参数，如果无效就会抛出 `MalformedURLException` 异常。一般情况下需要捕获并处理这个异常。

### 3. URL 类获取其特征的主要方法

URL 类提供的方法主要包括对 URL 类对象特征（如协议名、主机名、文件名、端口号和标记）的查询和对 URL 类对象的读操作。

- ✎ `String getProtocol ()`; 返回 URL 的协议名。
- ✎ `String getHost ()`; 返回 URL 的主机名。
- ✎ `Int getPort ()`; 返回 URL 的端口号，如果没有设置端口号返回值为 -1。
- ✎ `String getFile ()`; 返回 URL 的文件名及路径。
- ✎ `String getRef ()`; 返回 URL 的标记。

并不是任何一个 URL 地址都包括这些组成部分。任意给出一个 URL 的字符串描述，只要创建一个 URL 对象，就可以调用它的特征获取方法来得到所需的信息。

### 4. 用 URL 获取网上 HTML 文件

用 URL 获取网上 HTML 文件有三个步骤：

#### (1) 构造 URL 的对象

```
URL url = new URL("http://www.cumt.edu.cn");
```

#### (2) 将 `DataInputStream` 类对象与 url 的 `openStream ()` 流对象绑定

```
DataInputStream din = new DataInputStream(url.openStream())
```

#### (3) 利用 `din` 读取 HTML 文件

图 1-1 描述了用 URL 获取网上 HTML 文件涉及的几个类。

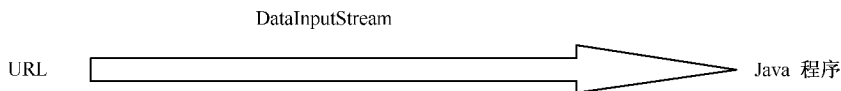


图 1-1 从 URL 获取网络上资源

#### 例 1-3 用 URL 获取网络上资源的 HTML 文件。

```
package chapter1 ;
import java.net. * ;
import java.io. * ;
public class ReadURL{
static public void main(String args[ ])
{
try {
//根据参数 args[0 ]构造一个绝对的 URL 对象
URL url = new URL(args[0 ]);
//通过 URL 对象打开一个数据输入流
DataInputStream din = new DataInputStream(url.openStream());
```

```
String inputLine ;
//从输入流读入数据
while((inputLine = din.readLine()) != null)
{
//将数据信息显示到系统标准输出上
System.out.println(inputLine) ;}
//关闭输入流
din.close();
}
catch(MalformedURLException me){}
catch(IOException ioe){}
}
}
}
```

运行下列 dos 命令：

```
java ReadURL http://www.cumt.edu.cn
```

运行结果就可以看到中国矿业大学主页的 HTML 文件。

## 5. 用 URL 获取文本和图像

用 URL 可以方便地获取文本和图像。文本数据源可以是网上或者本机上的任何文本文件。如果要用 URL 来获取图像数据，就不能再使用 `openStream()` 方法，而是要使用方法 `getImage(URL)`。这个方法会立即生成一个 `Image` 对象，并返回程序对象的引用，但这并不意味着图像文件的数据已经读到了内存之中，而是系统与此同时产生一个线程去读取图像文件的数据。因此，就可能存在程序已经执行到了 `getImage()` 后面的语句部分而系统还正在读图像文件数据的情形。

例 1-4 用 URL 来获取文本文件 (.txt) 和图像文件 (.jpeg, .gif)。

```
// package chapter1 ;
import java.net. * ;
import java.awt. * ;
import java.io. * ;
//GetDataByURL 类继承了 Frame 类 ,生成一个带标题的窗口
public class GetDataByURL extends Frame{
MenuBar menuBar ;
boolean drawImage = false ;
DataInputStream dataInputStream ;
int i = 0 ;
String line _ str ;
boolean first = true ;
Font font ;
//类 GetDataByURL 的构造方法
public GetDataByURL () {
```

```
//生成一个菜单条
menuBar = new MenuBar();
setMenuBar(menuBar);
//为菜单条第一个菜单取名"display"
Menu display = new Menu("display");
menuBar.add(display);
//生成"display"菜单下的两个菜单项
MenuItem beauty_display = new MenuItem("displaybeauty");
MenuItem text_display = new MenuItem("display text");
display.add(beauty_display);
display.add(text_display);
//设置背景颜色和文本的字体
setBackground(Color.white);
font = new Font("System", Font.BOLD, 20);
//设置带有菜单的窗口的标题
setTitle("sample :use URL get data");
setSize(400, 300);
//显示窗口
show();
}
//处理窗口中的菜单事件
public boolean action(Event evt, Object what) {
    if (evt.target instanceof MenuItem) {
        String message = (String) what;
        if (message == "display beauty") {
            drawImage = true;
            doDrawImage();
        }
        else {
            drawImage = false;
            first = true;
            if (message == "display text")
                doWrite("file ://d ://plbackup/tt.txt");
        }
    }
}
return true;
}
//处理窗体事件
public boolean handleEvent(Event evt)
{
    switch (evt.id)
    {
        case Event.WINDOW_DESTROY : dispose();
            System.exit(0);
```

```
        default :
            return super.handleEvent(evt) ;
    }
}
static public void main(String args[ ])
{
    new GetDataByURL() ;
}
public void paint(Graphics g) {
if(drawImage)
{
    try
    {
        //生成一个 URL 对象 ,它指向本机上的一个类型为 .jpeg 的图形文件
        URL image _ URL = new URL("file ://D :/plbackup /zy4.jpeg") ;
        Toolkit object _ Toolkit = Toolkit.getDefaultToolkit() ;
        Image object _ Image = object _ Toolkit.getImage(image _ URL) ;
        g.setColor(Color.white) ;
        g.fillRect(0 ,0 ,300 ,400) ;
        g.drawImage(object _ Image 40 ,80 ,160 ,200 ,this) ;
    }
}
catch(MalformedURLException e){}
}
else
{
    if(first)
    {
        first = false ;
        g.setColor(Color.white) ;
        g.fillRect(0 ,0 ,400 ,300) ;
        g.setFont(font) ;
    }
    if(line _ str != null)
        g.drawString(line _ str ,10 ,i * 20) ;
    i++ ;
}
}
//画图像函数
private void doDrawImage()
{
    drawImage = true ;
    repaint() ;
}
}
```

此为试读,需要完整版PDF帮助网: [www.ertongbook.com](http://www.ertongbook.com)

```
private void doWrite(String url_str)
{
try
{
    //用参数 url_str 生成一个绝对的 URL ,它指向本机上的一个文本文件
    URL url = new URL(url_str);
    dataInputStream = new DataInputStream(url.openStream());
    try
    {
        i = 1;
        line_str = dataInputStream.readLine();
        while(line_str != null){
            paint(getGraphics());
            line_str = dataInputStream.readLine();
        }
    }
    catch(IOException e){}
    dataInputStream.close();
}
catch(MalformedURLException e1){}
catch(IOException e2){}
}
}
```

### 1.2.3 URLConnection () 类

URL 类仅提供读取地址为 URL 的 Web 服务器内容的方法。如果除了读取其内容外，还要向 URL 对象发送服务请求及参数，那么必须要使用 URLConnection () 类。利用 URL 类提供的 openConnection () 方法可以建立一个 URLConnection 类对象。然后，可以使用这个 URLConnection 类对象绑定的数据输入流读 URL 的内容，使用这个 URLConnection 类对象绑定的数据输出流发送服务请求及参数。

URLConnection 类是一个以 http 为中心的类，同时支持 get 和 post 两种方法，默认情况下为 post 方法。

URL 和 URLConnection 的区别在于前者代表一个资源的位置，后者代表一种连接。

#### 1. 读取 URL 对象内容

例 1-5 利用 URLConnection 类获取网络上的 HTML 文件。

```
// package chapter1 ;
import java.net. * ;
import java.io. * ;
public class ReadURLConnection{
static public void main(String args[ ])
}
```

```
{
try
{
//根据绝对 URL 地址构造 URL 对象
URL cumtURL=new URL("http://www.cumt.edu.cn/");
//利用 cumtURL 建立一个 URLConnection 对象
URLConnection cumtConnection=cumtURL.openConnection();
//获取 cumtConnection 对象的数据输入流
DataInputStream din=new
        DataInputStream(cumtConnection.getInputStream());
String inputLine;
//从输入流 din 读入一行数据
while((inputLine=din.readLine())!=null)
{
//将数据信息显示到系统标准输出上
System.out.println(inputLine);
}
//关闭输入流
din.close();
}
catch(MalformedURLException me){}
catch(IOException ioe){}
}
}
```

从读取 URL 内容的功能看，它与例 1-3 与例 1-4 程序很相似。

## 2. 与 URL 对象交互

利用 URLConnection 类对象向 URL 对象发送服务请求及参数，这里以 Java 程序与服务器的 CGI (Common Gateway Interface, 公共网关接口) 交互为例，具体实现步骤如下。

(1) 创建 URL (包括 CGI 文件名) 对象。

(2) 打开一个到该 URL 的连接，建立相应的 URLConnection 对象。

(3) 从 URLConnection 对象获取其绑定的输出流。这个输出流就是连接到服务器上 CGI 的标准输入流 (stdin)。

(4) 向这个输出流中写数据，并关闭这个输出流。

(5) 从 URLConnection 对象获取其绑定的输入流。这个输入流就是连接到服务器上 CGI 的标准输出流 (stdout); 从这个输入流中读服务结果。关闭这个输入流。

例 1-6 通过 URLConnection 对象运行 java.sun.com 上一个叫 backwards 的 CGI, backwards 将其反转后写到标准输出。这个 CGI 需要下面这种格式的输入：“string = string \_ to \_ reverse”，这里 string \_ to \_ reverse 参数就是要反转的字符串。CGI backwards 可执行文件从标准输入读取字符串，将其转换成大写字符串写到标准输出。如图 1-2 所示。CGI backward 程序的编写这里不介绍。

例 1-6 利用 URLConnection 类向 CGI 发送请求消息。

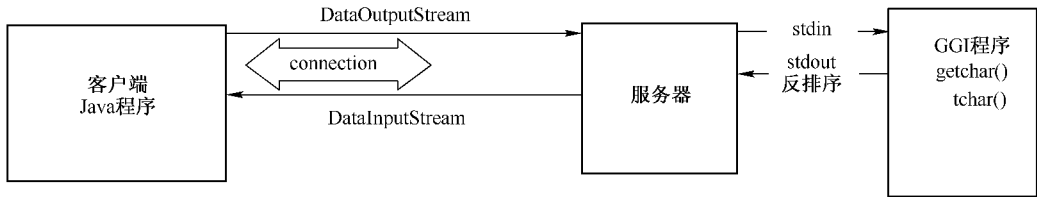


图 1-2 例 1-6 的 Java 程序与 Web CGI 交互

```

// package chapter1 ;
import java.net. * ;
import java.io. * ;
public class RequestURLConnection{
static public void main(String args[ ])
{
try
{
String string_ to _ reverse = URLEncoder.encode(args[0 ]);
//根据绝对 URL 地址构造 URL 对象
URL exampleURL = new URL("http :// java . sun . com / cgi - bin / backwards" );
//利用 exampleURL 建立一个 URLConnection 对象
URLConnection exampleConnection = exampleURL.openConnection();
//获取 exampleConnection 对象的数据输入流
PrintStream dout = new PrintStream (
new DataOutputStream(exampleConnection .getOutputStream()));
//向从输出流 dout 输出数据(要反序的字符串)
dout.println(" string = " + string_ to _ reverse)
//关闭输出流
dout.close();
DataInputStream din = new
DataInputStream(exampleConnection .getInputStream());
String inputLine ;
//从输入流 din 读入一行数据
while((inputLine = din.readLine()) != null)
{
//将数据信息显示到系统标准输出上
System.out.println(inputLine);
}
//关闭输入流
din.close();
}
catch(MalformedURLException me) //构造 URL 对象异常 ,显示 URL 异常信息
{System.err.println("MalformedURLException" + me) ;
}
catch(IOException ioe) //IO 异常处理

```

```

{ System.err.println("IOException" + ioe );
}
}
}
}
}

```

运行 `java RequestURLConnection abcd` 命令，可以看到下列结果：

```

abcd
reversed is :
dcba

```

### 1.2.4 套接口类 (Socket、ServerSocket) 概述

套接口是网络协议传输层提供的接口。套接口是进程之间通信的抽象连接点。它封装了端口、主机地址、传输层通信协议三方面内涵，如图 1-3 所示。两个网络进程采用套接口方式通信时，两进程扮演的角色不同，它们使用的套接口也不同。主动请求服务的一方叫客户，它使用客户建立的套接口 `Socket`，通过它主动与对方连接；服务器等待接收客户服务请求，提供服务，返回结果，它使用服务器套接口 `ServerSocket`。

用户目前可以使用的两种套接口，即流套接口和数据报套接口。流套接口提供了双向的、有序的、无重复并且无记录边界的数据流服务，TCP 是一种流套接口协议。数据报套接口支持双向的数据流，但并不保证是可靠、有序、无重复的，也就是说，一个以数据报套接口接收的信息的进程有可能发现信息重复了，或者和发出的顺序不同。数据报套接口的一个重要特点是它保留了记录边界。UDP 是数据报套接口协议。

每一进程运行在该主机的一个对外开放的端口上。主机地址用其 IP 地址或域名地址表示，如图 1-3 所示。一个端口与使用该端口的进程是对应的。端口常以整数编号，作为服务者可选择一个端口号，网络的客户以服务者主机和服务者选择的端口号与服务者连接。一些端口号是网络服务约定的，如 80 是 WWW 服务的默认端口号。

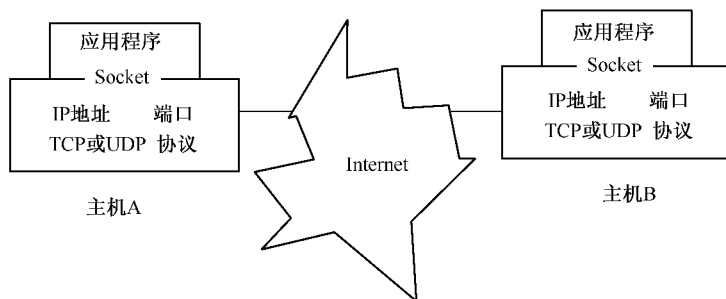


图 1-3 基于 Socket 的点对点通信

#### 1. 套接口类 (Socket)

##### (1) Socket 的构造方法

Socket 的构造方法有以下几种。

✎ `Socket (String host , int port) throws UnknownHostException , IOException`

构造一个流套接口（即 Socket 对象），利用 TCP 协议将其连接至特定主机 host 的特定端口 port 上。

参数 host 指主机名，port 指端口号。

↪ Socket (String host, int port, boolean stream)

构造一个套接口，而此套接口是流套接口（或 TCP 协议）还是数据报（datagram 或 UDP 协议）套接口，则是由最后一个参数 stream 决定的。通信对方由主机名 host 和端口 port 来确定。

参数 host 指主机名，port 指端口号，stream 指用于决定生成的套接口是流套接口还是数据报套接口。

↪ Socket (InetAddress address, int port)

构造一个流套接口，并把它连接到特定 IP 地址 address 的主机和特定端口 port 上。

参数 address 指特定的 IP 地址，port 指端口。

↪ Socket (InetAddress address, int port, boolean stream)

构造一个套接口，此套接口是流套接口还是数据报（datagram）套接口，则是由最后一个参数 stream 决定的。通信对方由其主机 IP 地址 address 和端口 port 确定。

参数 address 指主机 IP 地址，port 指端口号，stream 指用于决定构造的套接口是流套接口还是数据报套接口。

## (2) 套接口类的主要方法

套接口类的主要方法如下。

↪ InetAddress getInetAddress ()

返回该套接口所连接的 IP 地址。

↪ int getPort ()

返回该套接口（Socket）所连接的端口。

↪ synchronized void close () throws IOException

关闭套接口。

↪ InputStream getInputStream () throws IOException

获得与套接口绑定的数据输入流对象。

↪ OutputStream getOutputStream () throws IOException

获得与套接口绑定的数据输出流对象。

## 2. 服务器套接口类（ServerSocket）

### (1) 服务器套接口类的构造方法

服务器套接口类有两种构造方法。

↪ ServerSocket (int port) throws IOException

在指定的端口上构造一个服务器套接口，即构造一个 ServerSocket 实体对象。

参数 port 指端口号。

↪ ServerSocket (int port, int count)

构造一个服务器套接口，即构造一个 ServerSocket 实体对象，并且该对象是与指定的当地端口绑定。此外，可以对它进行监听。用户也可以通过将 port 设为 0 来将该对象与一个匿名端口相连接。

参数 port 指端口号，count 指容许等待对该 ServerSocket 对象的连接的个数。

## (2) 服务器套接口类主要方法

服务器套接口类提供的主要方法如下。

↳ Socket accept () throws IOException

服务器等待连接，该方法将阻塞当前服务线程，直到连接到来。该方法返回一个新的套接口对象，通过该对象，与客户套接口进行通信。

↳ void close () throws IOException

关闭套接口。

## 1.3 使用 TCP 协议的 Socket 网络编程

### 1. 基于客户-服务器方式的一对一的 Socket 通信

TCP 是一种可靠的、基于连接的网络协议。网络上的两进程采用 Socket 方式通信时，客户与服务器都必须建立各自的 Socket，其中服务器构建 ServerSocket 对象，并侦听来自网络的客户连接请求，另一方作为客户，通过 Socket 向网络上的服务器发送连接请求。图 1-4 描述基于连接的客户-服务器程序流程图。

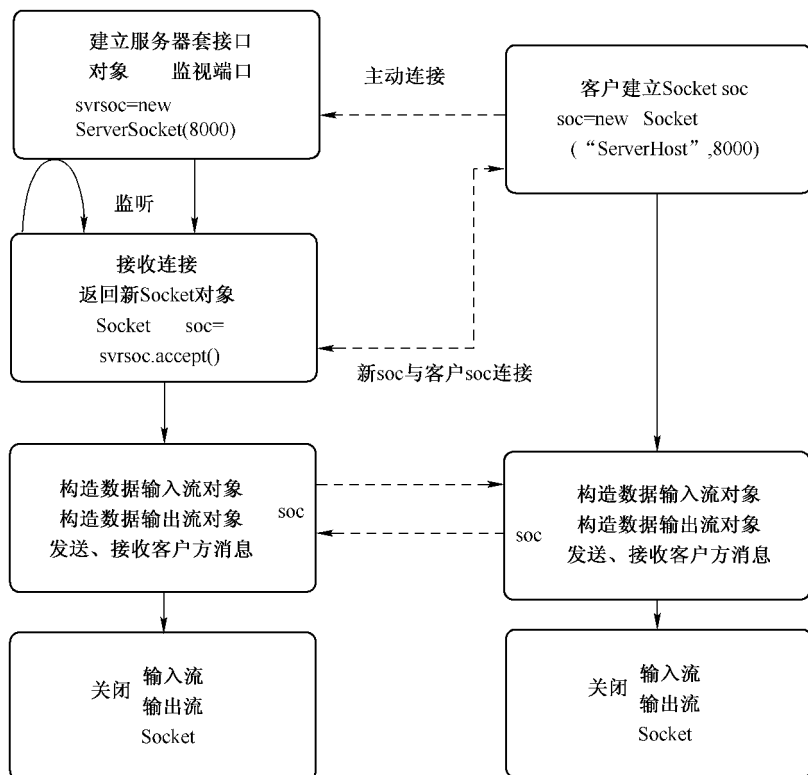


图 1-4 客户-服务器的通信流程图

例 1-7 利用 Socket 实现服务器和客户一对一的通信。

图 1-5 是服务器类 TcpServer 不完全规范的 UML 模型简图。它描述类 TcpServer 继承 Object 类（可默认）及其它依赖的类，如 ServerSocket、Socket、DataInputStream、InputStream、