



完全基于互联网丛书



随书赠光盘

本书所用技术:

Java、Perl、PHP、ASP

SQL、MySQL



高谈阔论 在线聊天室与BBS的设计开发



编著



机械工业出版社
China Machine Press

完全基于互联网丛书

高谈阔论——在线聊天室与 BBS 的设计开发

摆渡人工作室 编著



机械工业出版社

随着网络的日益普及，聊天室和论坛越来越成为众多网友每天必去之所，几乎每个大型网站都要建立自己的聊天室，在许多功能丰富的个人网站中，也建立了很有个性的聊天室和论坛。

本书是《完全基于互联网丛书》的第 12 本，通过 8 个完整实例页面，手把手地去教读者如何设计和实现聊天室与 BBS 论坛。

本书分别使用 Java Applet、ASP、PHP 和 Perl 制作聊天室，使用 ASP、PHP、Perl 和 FreeBSD 制作 BBS 论坛，使读者真正成为聊天室和论坛开发高手。

本书适合网络爱好者和网络开发人员使用。

JS433/09

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

责任编辑：王听讲 王英杰 封面设计：艾迪 艾藤

责任印制：路琳

中国建筑工业出版社密云印刷厂印刷·新华书店北京发行所发行

2001 年 1 月第 1 版第 1 次印刷

787mm×1092mm 1/16·16.75 印张·413 千字

0 001—6000 册

定价：31.00 元（1CD，含配套书）

ISBN 7-900043-90-X/TP·86

E-mail: sbs@mail.machineinfo.gov.cn

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

本社购书热线电话（010）68993821、68326677-2527

前 言

从 20 世纪 60 年代网络雏形的发展到目前全球化 Internet 的广泛应用, 互联网正逐渐改变我们的生活方式。我国自 20 世纪 90 年代初才开始接入 Internet, 而仅仅过了 10 年左右的时间, 我们周围的生活就显现出了“Internet 化”——从网上资讯浏览、网上购物到网上聊天, 我国网民人数正以日逾万人的速度增长着。

近几年, WWW(World Wide Web)的应用日益成为 Internet 发展中的重点, 不但各种类型的网站应运而生, 而且 WWW 的设计技术也层出不穷。大型 ICP 为了吸引网民的注意力, 不断扩大麾下的个人主页空间, 这为网络爱好者提供了一个可以展示个人风采的大舞台。

因为 WWW 具有易于浏览、可操作性强、使用环境简单、开发容易的特点, 我们可以预见, 未来 WWW 的创意、设计和编程人员将越来越多, 人才需求越来越大, 应用范围也会越来越广, 传统的应用编程也向 WWW 靠拢。

完全基于互联网丛书 (Fully Based on Internet, 简称 FBI) 主要就是针对 Internet 的发展倾向, 面向广大读者, 从 WWW 的创意、设计和编程的各个方面进行说明。FBI 丛书的主要特点可以用“一个中心, 两个基本点”来概括。

♣ 一个中心——完全基于互联网的发展技术为中心。从书中包括目前几乎所有互联网所涉及到的方方面面, 从网站的整体规划到最细微方面的设计, 囊括了 WWW 领域内时兴的应用点, 如典型的计数器、BBS、聊天室、数据查询与搜索、Web 数据库、各种网页特技、安全控制、电子商务等。换句话说, “网上有什么, FBI 中就讲述什么”。

♣ 两个基本点——便于阅读和便于交流

(1) 按效果 (功能) 划分, 方便了读者阅读 在 FBI 丛书的每一本书中都会向读者讲解几种相关软件的应用技术, 便于读者阅读和应用。为了实现某一种效果 (功能), 往往会涉及到很多种应用工具软件, 如在制作网页时, 不但要熟悉 FrontPage、Dreamweaver、Flash、Fireworks、Photoshop 等软件的不同特点, 而且还要学会各种编程技术, 如 Java、ASP (JavaScript/VBScript)、PHP、Perl、SQL/MySQL 等, 这样才能使您制作的网页更加精彩。在同一本书中您可以看到各种不同的软件和语言为了实现同一个效果 (功能) 协同作用。而采用更有针对性的页面形式编排, 融技术于具体应用, 读者只要按照书中的步骤就可以完成预定的效果 (功能)。这是目前市面上仅有的按照效果 (功能) 划分并以网页形式编排的系列图书。

(2) 完善的售后服务体系, 便于读者与作者之间的交流 为了方便广大读者与本丛书作者的相互沟通, 我们特为本套丛书建立了专用网站。请记住我们的网址 <http://www.buydoren.com>, 欢迎您指正 FBI 丛书中的问题, 或者就您感兴趣的问题和我们探讨。我们也将定期对新出现的计算机技术在网进行实例讲解。

我们都是 Internet 的爱好者, 相信通过我们的共同努力, 可以为互联网在我国的普及尽一点微薄之力。

摆渡人工作室

www.buydoren.com

关于本书

本书是《完全基于互联网丛书》的第 12 本，主要介绍如何设计和实现聊天室与 BBS 论坛。

目前，丰富多彩的聊天室和 BBS 论坛层出不穷，为专业或行业信息的发布和人与人之间的交流提供了更方便的途径。聊天室和 BBS 的功能一般都比较复杂，因此实现起来也比较困难。通常都采用 Web 编程语言和后台数据库系统来实现，常用的编程语言有 Java、Perl、PHP、ASP 等，后台数据库在 NT 下一般采用 Microsoft SQL 系统，在 Unix 或 Linux 下一般采用 MySQL 系统。一般开发人员根据实际对聊天室或 BBS 的要求程度，例如界面环境、功能模块、要达到的效果等，选取适当的编程语言和数据库系统来进行设计。对于聊天室来讲 Java 可能是最好的，但对于 BBS 来讲，ASP 可能是最容易实现的。

本书分别提供了聊天室和 BBS 论坛的四种不同实现。这些例子都有各自的特点，界面都不一样，功能上也有差异，所使用编程语言也不相同，不同读者可以选取希望使用的语言。本书的主要目的就是通过有限的实例，让读者明白聊天室和 BBS 设计时需要考虑的一些问题，另外还可以充分掌握 Web 数据库系统的应用。本书中的几个例子都不是特别完整，因此读者在使用时还需要在某些方面进行扩充和完善。

本书涉及的主要技术有 Java Applet、ASP、PHP、Perl 和 FreeBSD 等。

本书适于设计和开发聊天室与 BBS 论坛的广大爱好者阅读。

关于我们——摆渡人工作室

“摆渡人工作室”的前身是在两年前成立的“清汉计算机工作室”。

两年多来，我们和机械工业出版社通力合作，精心策划、编辑、出版了《计算机开发与制作实例丛书》（以下简称《实例》）。这套丛书共计 26 本，覆盖了从 Microsoft Office 到应用程序和数据库开发，以及网页制作和多媒体等多方面实用技术和技巧，以其独特的魅力在同类计算机图书中发展成了一个大家族。

我们立足清华大学，广泛了解并研究计算机发展方向，认真进行计算机图书的策划、组织和编辑工作。

我们的努力得到了应有的回报！2000 年我们的实例丛书获得了巨大的成功，得到了读者和出版社的广泛好评。其中《FrontPage 2000 网页制作实例》和《Visual Basic 6.0 数据库开发实例》一年内加印了 4 次，累计销量达到 16000 册，丛书平均销量也达到了近 10000 册。

在实例丛书的具体运作中我们体验了创业的艰辛，成功的喜悦，在成功后我们开始在更深的层次上思考读者真正需要什么。

我们开始思考两个问题：

◆ 我们何不推出更具针对性的实例呢？

我们发现实例丛书中的实例虽然是我们精心设计的，但很多却不是“现存的”，也就是说这些实例不是在网上存在的，而是我们为了讲解某些知识专门设计的。这对需要解决实际问题的读者来说还需一个消化吸收的过程。

◆ 何不建立一个更完善的图书售后服务体系呢？

有些读者购买了我们的实例丛书后，针对书中的某些问题或打电话或发 Email 和我们探讨，我们为读者对我们的厚爱感到由衷的高兴，但同时也感到交流的不顺畅阻碍了我们和读者进一步的探讨。读者买到图书，正是学习的开始，我们的工作应该延伸到读者的学习过程中，而不是售出书就万事大吉！

带着这些急需解决的问题，我们考虑了现在最便利的联络方式——互联网，在对现在全球互联网技术现状和未来发展走势的精心调查研究下，我们隆重推出了《完全基于互联网丛书》（Fully Based on Internet，简称 FBI）。这套丛书继承了《计算机开发和制作实例丛书》的特点，以页面的形式讲解当今互联网业涉及的几乎所有的技术。从申请域名建立网站，到实现人机交互和建立自己的电子商务网站等全过程。读者只需要按照我们的讲解，按部就班地学习丛书中介绍的各种技术，不仅可以建立展现自我风采的个人网站，还可以成功地建立完善的商业网站。同时这套丛书还具有“一个中心+两个基本点”的特征。关于这方面的内容您可以从本丛书的前言中看到。相信在您认真阅读了这套丛书后，会对这些特点有更深刻的理解。



随着 FBI 丛书的发展,我们相信一定会与您成为好朋友,因为我们和您一样都是爱好新鲜事物的年轻人,充满着对未来的憧憬。我们努力、创造,我们自信、把握,我们相信机会在自己的手中。

您有什么好的建议或者对我们的丛书有什么意见,请访问我们的网站:

www.buydoren.com。

关注生活, 关注我们的图书!

附录: 关于《实例》丛书

《实例》丛书的特点:

(1) 通过一个个精心设计的实例,用户可以轻松地逐步掌握一种软件。

(2) 书中每一个实例的主题对应该软件的一个主要功能,通过所有实例的学习,您可以全面掌握该软件的精髓。

(3) 虽然《实例》丛书采用一个个相对独立的实例,但我们在丛书的策划中考虑到知识的连贯性和统一性,丛书中的“归纳和注释”可以帮助读者举一反三地掌握书中介绍的内容。使用附光盘的书籍更可以使您的学习和开发如虎添翼。

(4) 随书光盘中还包含和该书籍相关的软件及网络资源,使您在阅读该书的同时还可以涉猎其他相近的领域。

《实例》丛书书目:

1. Word 2000 中文版实例
2. Excel 2000 中文版实例
3. PowerPoint 2000 中文版实例
4. Access 2000 中文版实例
5. Delphi 5.0 开发实例 (附 CD)
6. VBScript 开发实例 (附 CD)
7. ASP 开发实例 (附 CD)
8. C++ Builder 网络开发实例 (附 CD)
9. Flash 5.0 网页动画制作实例 (附 CD)
10. Photoshop 6.0 图像制作实例 (附 CD)
11. FrontPage 2000 网页制作实例 (附 CD)
12. CorelDRAW 9.0 图画制作实例 (附 CD)
13. Photoshop 5.5 图像制作实例 (附 CD)
14. 3D Studio MAX R3.0 动画制作实例 (附 CD)
15. Authorware 5.0 制作实例 (附 CD)
16. Director 7.0 多媒体制作实例 (附 CD)
17. Visual Basic 6.0 数据库开发实例 (附 CD)
18. Visual Basic 6.0 多媒体开发实例 (附 CD)
19. Visual Basic 6.0 网络开发实例 (附 CD)
20. Visual FoxPro 6.0 开发实例 (附 CD)
21. Flash 4.0 动画、网页制作实例 (附 CD)
22. AutoCAD 2000 中文版综合应用实例 (附 CD)
23. Visual C++ 6.0 数据库与网络开发实例 (附 CD)
24. Visual C++ 6.0 多媒体开发实例 (附 CD)
25. Dreamweaver 3.0 网页制作实例 (附 CD)
26. Fireworks 3.0 网页素材制作实例 (附 CD)

目 录

前言

关于本书

关于我们——摆渡人工作室

页面 1	用 Java Applet 制作聊天室	1
页面 2	用 ASP 制作 BBS 论坛	13
页面 3	用 ASP 制作聊天室	61
页面 4	用 PHP 制作聊天室	91
页面 5	用 PHP/MySQL 制作 BBS 论坛	134
页面 6	用 Perl 制作聊天室	187
页面 7	用 Perl 制作 BBS 论坛	198
页面 8	架设 FireBird BBS 论坛	233



<http://www.yellowmount.com/chat/chat.html>



页面效果和技术概要

在 Internet 日益广泛应用的今天,网络生活已经成为很多人生活中不可缺少的一部分,通过网络媒体进行相互交流已经是相当普遍的现象,而在各种交流方式中,网上聊天似乎是最激动人心的事。

基于 Web 聊天室(Chat Room)是 Internet 上最成功的应用之一,尽管存在更好的实现网络聊天的方法,但 Web 浏览器的广泛普及和易于使用使 Web 聊天室占据了显要的位置。本页面制作的聊天室如图 1-1 所示。

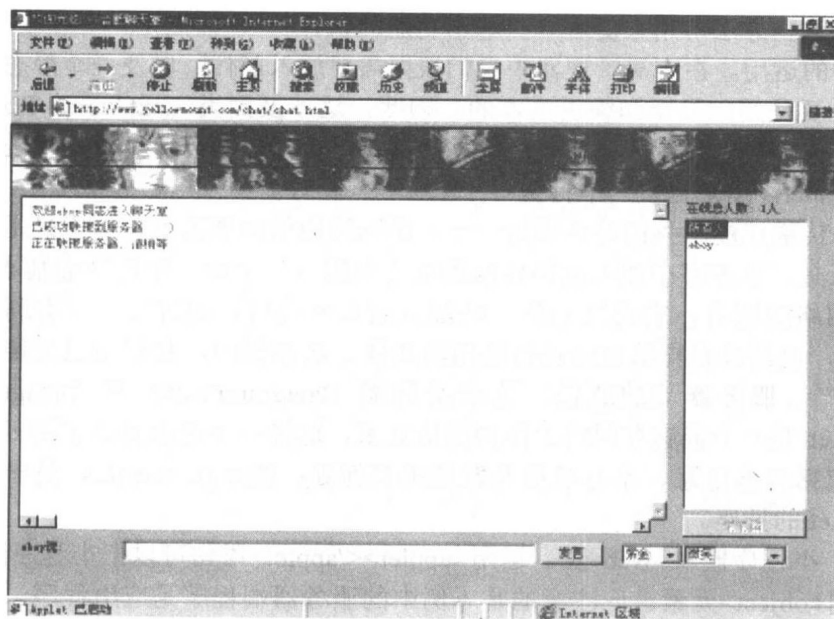


图 1-1 “合肥聊天室”

一般 Web 聊天室的基本流程为:

- ① 输入访问聊天室的 URL。
- ② 当链接成功,出现登录界面,要求用户输入聊天代号和密码。如果你是未注册用户,要求先注册。

④ 登录成功，进入聊天室。一般的聊天室都有可供用户输入的文本框和显示信息的文本区域，当聊天者输入信息时，可以在显示区域显示出来。

本页面中，要告诉大家的不是聊天的技巧，而是要如何制作一个聊天室。

Web 聊天室的制作有很多的技术可供我们使用，比较流行的是编制通用网关接口 (CGI)，但是这比较复杂。在这里，要向大家介绍的是用 Java Applet 制作聊天室的页面。

Java 聊天室实现简便并且功能强大，有较强的安全性，在网上的实际应用也比较普遍。限于篇幅，在这里仅给出聊天室的一个基本实例，即文本信息的输入、传送及显示。实现这些功能对于以面向对象为特征的 Java 语言来说，有许多标准的类库及方法可供调用。本例我们将用一个常见的例子来说明如何实现 Java 聊天室，以客户和服务器两端的 Java 小程序的形式出现。

本例 Java 聊天室中涉及的技术要点：

④ 布局管理器。Java 中的布局管理器 (Layout Manager) 可以方便地实现聊天室的框架。BorderLayout 类可用来设计和管理用户接口中的一些部件，当在一个布局上添加一个部件时，通过 North、South、East、West 和 Center 定义；类 Frame 是一个加入了一些功能的窗口，就其表面特征，一个 Frame 有边界和标题，就其功能可在 Frame 中加入一个菜单。

④ 容器 (Container)。布局管理器需要放在一个容器中，类 Container 是一种基本类，可容纳诸如 Frame、Panel 等其他容器类。在例子中，用到类 Container 中的方法 SetLayout 来设置布局管理器。注意 Container 类继承于 Component 类，Component 类为部件提供相匹配的方法，这些方法可用来促进一些基本功能的执行。

④ 部件的运用。在布局管理器中我们根据需要放入部件，以及一个静态的标签，要放入的部件主要是用来显示和编辑文本的。因此，这涉及到类 TextArea、TextField 的应用，前者是生成一个可显示和编辑且可滚动的文本区域，后者可对一行文本进行显示和编辑，还有类 Label，用来生成一个标签。

④ 网络传输功能。我们是在编写一个可供网络通信的聊天室，而不只是一个输入和显示文本的工具，在程序中加入网络传输的功能是编写一个聊天室程序的基本特征。Java 是一种基于网络的语言，它通过创建一些接口或与一些接口的链接，或者利用 URL 和 Internet 地址，使其具有与其他信息源通信的功能。在本例中，我们用报文传送的简单方式来实现客户、服务器端的通信，这涉及到类 DatagramPacket 和 DatagramSocket。DatagramPacket 是一个能够在网络上传输的信息束，这样一个对象封装了许多关于此信息束的信息，包括内含信息、本身长度及发送来自何处；DatagramSocket 是一个用于传输 DatagramPacket 的插座。

④ Java 小程序嵌入 Web 页。运用 <applet></applet> 语句可以把小程序嵌入 HTML 中，也可以用 <object> 元素。应注意的是本例中的服务器端程序是应用程序，不需要嵌入而直接运行。

综上，涉及到的要点有：Java Applet 的编写方法、基本结构及运行机制，C/S 结构，Java 布局管理器及相关部件的运用，Java 的网络通信编程方法及相关类的使用，Java 小程序嵌入 Web 页。



实现步骤和技术详解



打开文本编辑器

打开文本编辑器如记事本、写字板、Word 或 WPS 等，在编辑窗口中输入如下的 Java Applet 源程序。

当然也可以使用 VJ++、JBuilder 等 Java 快速开发工具，它们需要先建立一个工程 (Project)，接下来在其工具提供的程序编辑器中输入下面的程序或修改相应的方法。



编写客户端小程序

在编写客户端小程序的开始，照例引入要用到的相关类，这与 C/C++ 要引入头文件类似，不过在 Java 中要用 import 语句而不是 include。

根据前面的分析，要引入的类包括基本的 Applet, awt 包中的类 Container、BorderLayout、TextArea、TextField、Panel、Label 等，net 包中的 DatagramPacket 和 DatagramSocket，还有基本输入输出类。

```
import java.applet.*;
import java.awt.*;
import java.net.*;
import java.io.*;
```

定义一个扩展于 Applet 的类 javaChatClient。

```
public final class javaChatClient extends Applet
```

类成员定义，包括标签、面板、文本区、信息束等对象，这些对象应定义成私有，只供本类使用。

```
private Label label1, label2;
private Panel panel1, panel2;
private TextField textfield;
private TextArea textarea;
private DatagramPacket sendPacket, receivePacket;
private DatagramSocket sendSocket, receiveSocket;
```

方法 init 在小程序装载入浏览器后即开始执行，一般重载此方法，可实现成员变量的初始化，这里也不例外。

```
public void init()
{
    //设置背景色
    setBackground(Color.orange);
    //设置一个布局管理器
    setLayout(new BorderLayout());
    //建立一个新的面板容器
    panel1=new Panel();
    //在容器中放置布局管理器
    panel1.setLayout(new BorderLayout());
    //文本显示区域的标签
    label1=new Label("聊天记录:");
    //创建一个文本显示区域的页面
    textarea=new TextArea(10,20);
    //在文本区域显示一行文字
    textarea.setText("服务者:\t 欢迎光临! \n");
    //将标签添加到布局管理器中
    panel1.add("North",label1);
    //将文本域置于布局管理器中
    panel1.add("Center",textarea);
    add("North",panel1);
    //创建另一个容器
    panel2=new Panel();
    panel2.setLayout(new BorderLayout());
    //输入框区域的标签
    label2=new Label("发言:");
    panel2.add("Center",label2);
    textfield=new TextField(20);
    textfield.setText("");
    panel2.add("South",textfield);
    add("Center",panel2);
    show();
}
```

重载方法 start, 调用自定义方法 waitForPackets。

```
public void start()
{
    waitForPackets();
}
```

方法 `waitForPackets` 用来监听来自于服务端的数据报文，当获得时，在文本显示区域显示出来。

```
public void waitForPackets()
{
    try{
        //定义发送和接收到的数据套接字
        sendSocket=new DatagramSocket();
        receiveSocket=new DatagramSocket(5001);
    }
    catch(SocketException se) //捕获异常
    {}
    while(true)
    {
        try{
            //定义存储数组的数据报文
            byte array[]=new byte[100];
            //定义一个报文页面对象并获取数据报文
            receivePacket=new DatagramPacket(array,array.length);
            receiveSocket.receive(receivePacket);
            //将接收到的报文显示出来
            textarea.appendText("\n 服务者:\t");

            byte data[]=receivePacket.getData();
            String received=new String(data,0);
            textarea.appendText(received);
        }
        //捕获异常并处理
        catch(IOException ioe)
        {
            textarea.appendText("网络出错! \n");
            textarea.appendText("问题在于 "+ioe.toString()+"，等待再次连接! \n");
        }
    }
}
```

重载方法 `action`，处理数据报文发送。方法 `action` 属于类 `Commonet`，当其动作发生时，`action` 方法被 `handleEvent()` 所调用，进行事件处理。如果事件得到成功控制，则返回一个真值，否则抛出一个异常，需要捕获异常并处理。

```
public boolean action(Event e,Object o)
{
```

```

try{
    textarea.appendText("\n 客户:");
    String s=o.toString();
    textarea.appendText(s);
    byte data[]=new byte[100];
    s.getBytes(0,s.length(),data,0);
    //数据报文发送, "Host IP Address"由用户根据实际情况定义
    sendPacket=new DatagramPacket(data,s.length(),
                                   InetAddress.getByName("Host IP Address"),5000);
    sendSocket.send(sendPacket);
}
//捕获异常并处理
catch(IOException ioe)
{
    textarea.appendText("网络出错! \n");
    textarea.appendText("问题在于 "+ioe.toString()+"等待再次连接! \n");
}
return true;
}
}

```



编写服务器端程序

服务器端程序与客户端类似, 主要也是使用布局管理器, 定义一个显示和编辑文本的区域以及一个输入框, 也要完成数据通信功能, 即可以接收和显示来自于客户的信息和请求并作相应的处理。

服务器端程序与客户端程序不同的是, 客户端是 Java Applet, 而服务器端继承了 Frame, 并定义一个应用程序 (Application) 的类来调用它, 使之运行。

由于与前面的程序有很大的相似性, 我们在此仅列出程序, 而不作详细解释, 读者可参考前面的程序注释。

```

import java.awt.*;
import java.io.*;
import java.net.*;
import java.applet.Applet;

public class javaChatServer extends Frame
{
    private Label label1,label2;
    private Panel panel1,panel2;

```

```
private TextField textfield;  
private TextArea textarea;  
private String name,name1;  
private DatagramPacket sendPacket,receivePacket;  
private DatagramSocket sendSocket,receiveSocket;
```

```
public javaChatServer()  
{  
    super("聊天室服务台: ");  
    panel1=new Panel();  
    panel1.setLayout(new BorderLayout());  
    label1=new Label("聊天记录:");  
    textarea=new TextArea(10,20);  
    textarea.setText("");  
    panel1.add("North",label1);  
    panel1.add("Center",textarea);  
    add("North",panel1);  
    panel2=new Panel();  
    panel2.setLayout(new BorderLayout());  
    label2=new Label("发言:");  
    panel2.add("Center",label2);  
    textfield=new TextField(20);  
    textfield.setText("");  
    panel2.add("South",textfield);  
    add("Center",panel2);  
    resize(450,260);  
    show();  
    try  
    {  
        sendSocket=new DatagramSocket();  
        receiveSocket=new DatagramSocket(5000);  
    }  
    catch(SocketException se)  
    {  
        se.printStackTrace();  
        System.exit(1);  
    }  
}
```

```
public void waitForPackets()
```

```
{
    while(true)
    {
        try
        {
            byte array[]=new byte[100];
            receivePacket=new DatagramPacket(array,array.length);
            receiveSocket.receive(receivePacket);
            name=receivePacket.getAddress().toString();

            if (name!=name)
            {
                textarea.appendText("\n 来自主机: "+name+"\n 端口: "
                                   +receivePacket.getPort());
            }
            textarea.appendText("\n 客户:\t");
            byte data[]=receivePacket.getData();
            String received=new String(data,0);
            textarea.appendText(received);
            name1=name;
        }
        catch(IOException ioe)
        {
            textarea.appendText("\nProblem lies at "+ioe.toString());
        }
    }
}

public boolean handleEvent(Event e)
{
    if(e.id==Event.WINDOW_DESTROY)
    {
        hide();
        dispose();
        System.exit(0);
    }
    return super.handleEvent(e);
}

public boolean action(Event e,Object o)
```



```

{
    try
    {
        textarea.appendText("\n 服务者:");
        String s=o.toString();
        textarea.appendText(s);
        byte data[]=new byte[100];
        s.getBytes(0,s.length(),data,0);
        sendPacket=new DatagramPacket(data,s.length(),
            InetAddress.getByName(name),5001);
        sendSocket.send(sendPacket);
    }
    catch(IOException ioe)
    {
        textarea.appendText("问题可能存在于"+ioe.toString());
    }
    return true;
}
}

```

类 ChatServerRun,构造一个应用类,调用服务器类 javaChatServer,使之运行。

```

import javaChatServer;
class ChatServerRun extends javaChatServer
{
    public static void main(String args[])
    {
        javaChatServer cs=new javaChatServer();
        cs.waitForPackets();
    }
}

```



注意问题:

数据通信方式

程序中使用的是简单的报文传输的信息通信的方式,这是一个简单的不安全的通信方式,在实际的运用中,我们并不是使用这种传输方式。