Java面向对象程序设计
-网络程序设计

本课件部分内容摘自耿祥义的《Java大学实用教程》第4版

回顾:ArrayList、HashMap和流

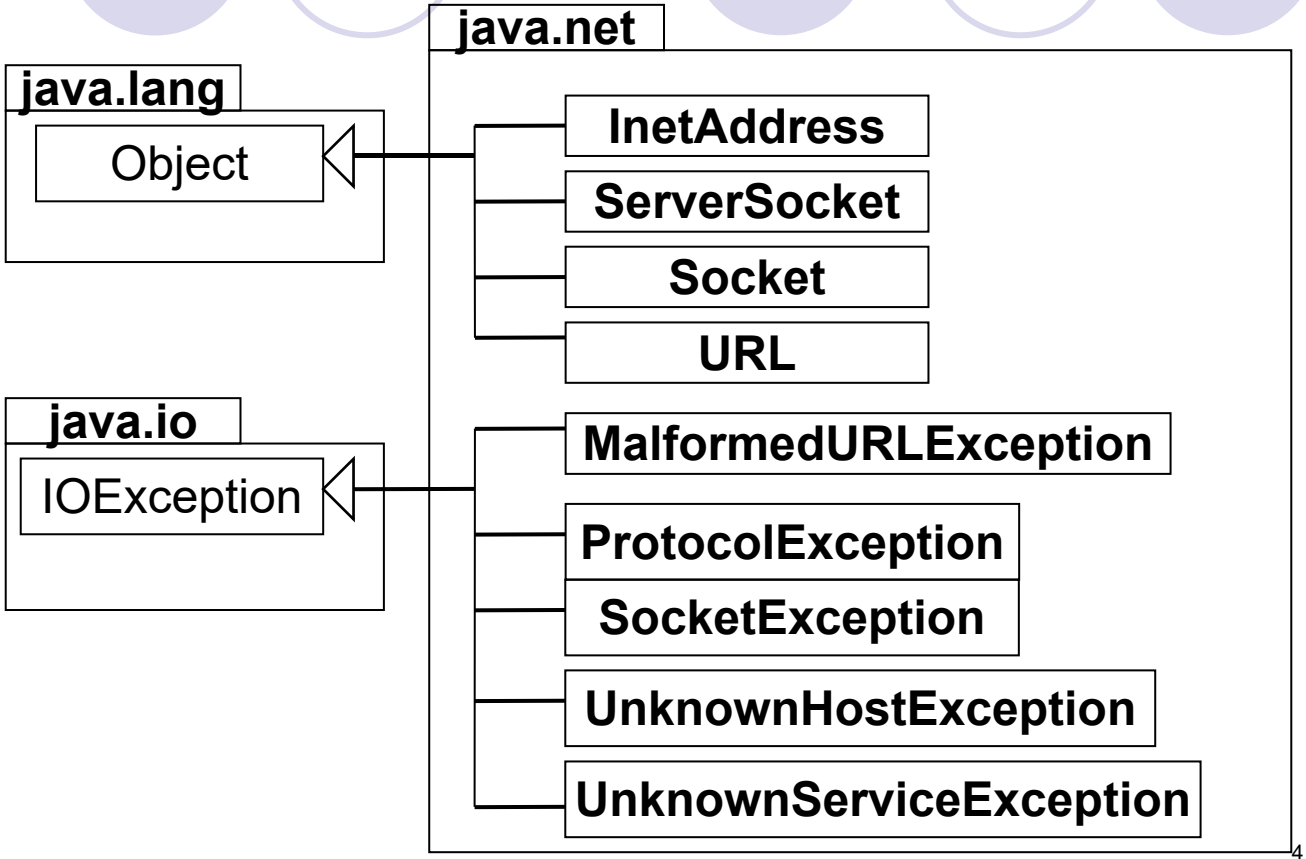
- ArrayList类可以实现数组所有的功能，在程序运行时可以根据需要自动调整大小。
- HashMap使用哈希表技术实现最快的对象检索，用键对象来检索值对象。
- 要在Java程序中永久地保存数据或要访问已存储的数据，必须有一个流。
  - 输入流：从数据源读取数据。
  - 输出流：将数据写入数据源。

如何利用网络资源

java的主要能力之一就是提供对Internet和客户端/服务器编程的支持。

- Java.net包提供了一套强大且易于使用的类来支持网络编程。
- 网络由许多协议组合而成，大部分协议由通用计算机上运行的软件实现。java封装了这些协议问题。
  - URL类包含一些方法获取Internet上特定URL相关联的资源
  - Socket和ServerSocket类使我们能跟一个Internet主机直接连接
  - DatagramPacket和DatagramSocket类支持更低层的互联应用程序开发

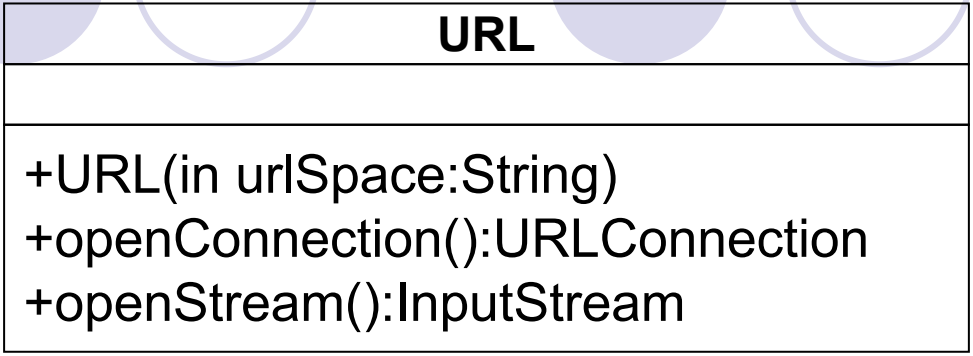
java网络常用类


java.net
java.lang
Object
InetAddress
ServerSocket
Socket
URL
java.io
IOException
MalformedURLException
ProtocolException
SocketException
UnknownHostException
UnknownServiceException

URL统一资源定位器[1]

- 如果 IP 地址 唯一 标识 了 Internet 上 的 计算机，则 URL 标识 了 计算机 上 的 资源。
- URL（Uniform Resource Locator，统一 资源 定位地址）充当 一个 指针，指向 Web 上 的 Web 页、二进制 文件 以及 其他 信息 对象。
- 一个 URL通常包含如下信息：
  - ① http 服务 使用 的 协议（HTTP）；
  - ② dlrin.edu.cn 存储资源的计算机的域名地址；
  - ③hotlink.html 资源。

URL类



- 创建一个URL对象，如果无效抛出MalformedURLException异常。

```
try {
  URL  url=new
URL("http://www.prehall.com:80/moreli/index.html");
}catch (MalformedException e) {
    System.out.println("Malformed URL:"+url.toString());}
```

网页小程序applet

- 按如下格式编辑一个html网页，调用applet程序class文件即可。

```
<html>
<applet code=MyApplet width=300 height=300>
</applet>
</html>
```

- applet对资源的利用严格受限！不能使用宿主机存储使得它不能保存下载的文件，且只能从源主机上下载文件。

读取URL中的资源[2]

- URL 对象 调用 openStream() 方法 可以 返回 一个 输入 流InputStream，该 输入 流 指向 URL 对象 所 包含 的 资源。通过 该 输入 流 可以 将 服务器 上 的 资源 信息 读入 到 客户 机。

- 【例 1】在 一个 文本 框 中 输入 网址，然后 单击“ 确定” 按钮 读取 服务器 上 的 资源（ 效果 如 图 11- 1 所示）。由于 网络 速度 或 其他 因素，URL 资源 的 读取 可能 会 引起 堵塞，因此 程序 需 在 一个 线程 中 读取 URL 资源，以免 堵塞 主 线程。

读取URL资源程序示例[3](1)

```
import javax. swing.*;
import java. awt.*;
import java. awt. event.*;
import java. net.*;
import java. io.*;
public class NetWin extends JFrame implements
ActionListener, Runnable{
    JButton button; URL url; JTextField text; JTextArea area;
    byte b[]= new byte[118];
    Thread thread;
```

读取URL资源程序示例(2)

```
NetWin(){
    text= new JTextField(20); area= new JTextArea(12,12);
    button= new JButton("确定");
    button.addActionListener(this);
    thread= new Thread(this);  JPanel p = new JPanel( );
    p.add(new JLabel("输入网址:")); p.add(text);
    p.add(button);
 super.add(new JScrollPane(area),BorderLayout.CENTER);
    super.add(p,BorderLayout.NORTH);
    super.setBounds(60,60,360,300);
    super.setVisible(true);
super.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

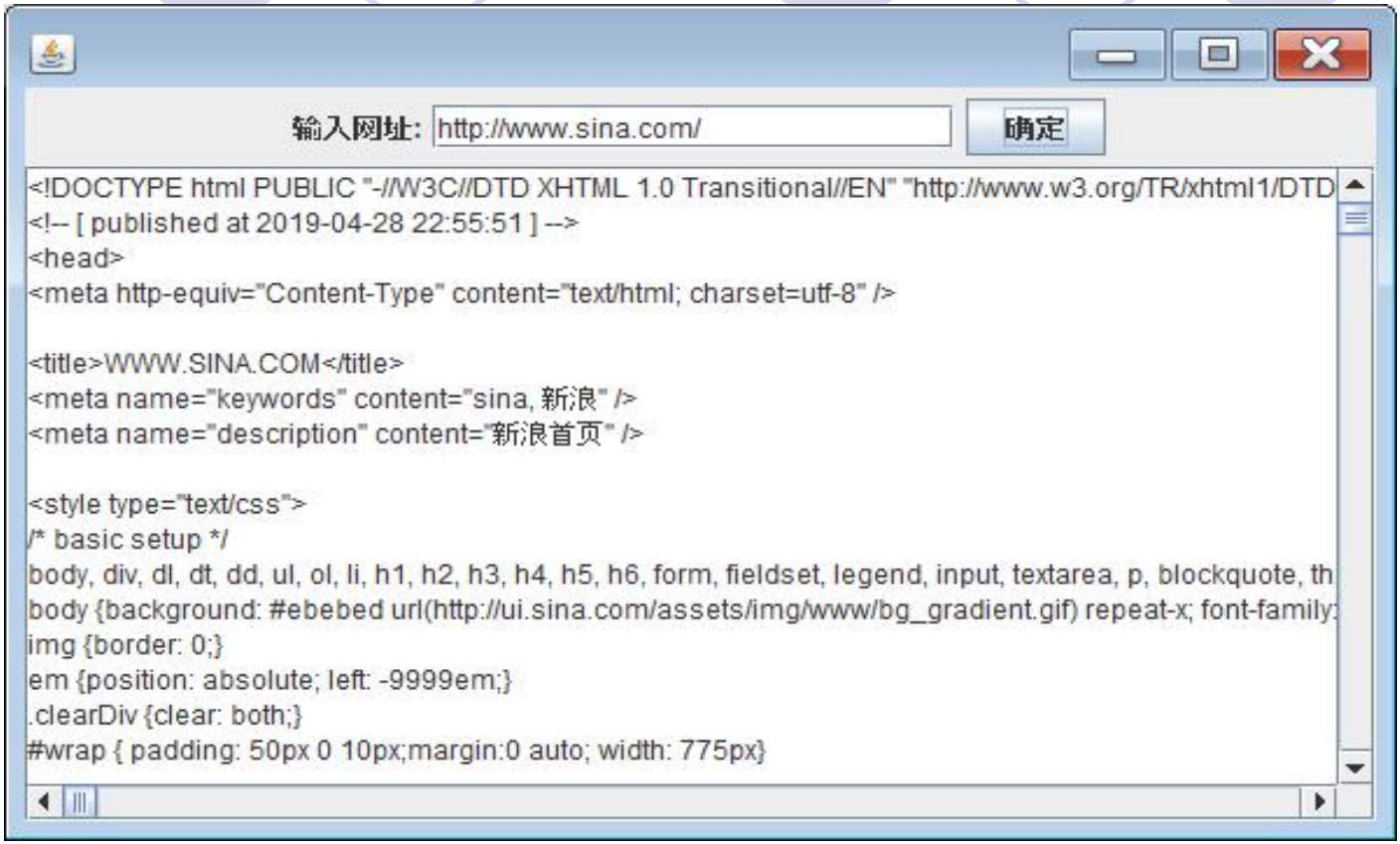
读取URL资源程序示例(3)

```
public void actionPerformed( ActionEvent e){
    if(!( thread.isAlive()))
        thread= new Thread( this);
        try{ thread.start(); }catch( Exception ee){ }
}
public void run(){
    try { int n = -1;  area.setText( null);
        url= new URL( text.getText().trim());
        InputStream in= url.openStream();
        while(( n = in.read(b)) != -1){
            String s= new String( b, 0, n);
            area.append(s);  }
```

读取URL资源程序示例(4)

```
    } catch( MalformedURLException e1){
        text.setText(""+ e1);
    } catch( IOException e1){
        text.setText(""+ e1);
    }
}
public static void main(String[] args) {
    new NetWin();
}
```

读取URL中的资源程序效果



Java应用程序读取URL图片音频

```
//java应用程序中下载和显示图片
import java.io.*;import java.net.*;

  private Image imgRef;
  private AudioClip audio1;
  try {
     imgRef = Toolkit.getDefaultToolkit().getImage(url);
//使用工具Toolkit类下载图片
     audio1 = Toolkit.getDefaultToolkit().getAudioClip(url);
 //使用工具Toolkit类下载音频文件
     audio1.play();
     repaint();  //刷新窗口，显示图片和声音
  }catch (IOException e) {
    System.out.println(e.getMessage());//下载资源会抛出
异常
  }
```

下载URL资源程序示例(1)

```
import java.io.*;
import java.net.*;
import java.util.*;
/* < p > Description: 将指定的HTTP网络资源在本地以文件形式存放 < /p > */
public class HttpGet {
    public final static boolean DEBUG = true; //调试用
    private static int BUFFER_SIZE = 8096; //缓冲区大小
    private ArrayList<String> vDownLoad = new
ArrayList<String>(100); //URL列表
    private ArrayList<String> vFileList = new
ArrayList<String>(500); //下载后的保存文件名列表

    public HttpGet() {}
```

下载URL资源程序示例(2)

```
/* 清除下载列表*/
public void resetList() {
    vDownLoad.clear();
    vFileList.clear();
}
/* *增加下载列表项
     * @param url String
     * @param filename String
     */
public void addItem(String url, String filename) {
    vDownLoad.add(url);
    vFileList.add(filename);
}
```

下载URL资源程序示例(3)

```
/**根据列表下载资源*/
public void downLoadByList() {
    String url = null;   String filename = null;
    for (int i = 0; i < vDownLoad.size(); i++) {
        url = vDownLoad.get(i); filename = vFileList.get(i);
        try {
            saveToFile(url,filename);
        } catch (IOException err) {
            if (DEBUG) {
                System.out.println("资源[" + url + "]下载失败!!!");
            }
        }
    }
    if (DEBUG) {
        System.out.println("下载完成!!!");
    }
}
```

下载URL资源程序示例(4)

```
    /**
     * 将HTTP资源另存为文件
     * @param destUrl String
     * @param fileName String
     * @throws Exception
     */
    public void saveToFile(String destUrl, String fileName)
throws IOException {
        FileOutputStream fos = null;
        BufferedInputStream bis = null;
        HttpURLConnection httpUrl = null;
        URL url = null;
        byte[] buf = new byte[BUFFER_SIZE];
        int size = 0;
```

下载URL资源程序示例(5)

```
//建立链接
        url = new URL(destUrl);
        httpUrl = (HttpURLConnection) url.openConnection();
//连接指定的资源
        httpUrl.connect();
//获取网络输入流
        bis = new BufferedInputStream(httpUrl.getInputStream());
//建立文件
        fos = new FileOutputStream(fileName);
        if (this.DEBUG)
            System.out.println("正在获取链接[" + destUrl + "]的内
容...\n将其保存为文件[" +fileName + "]");
```

openStream方法和openConnection方法

- 其实底层的实现，openStream（）方法的实现也是调用了 openConnection生成一个URLConnection 对象，然后再通过这个对象调用的 getInputStream（）方法的。
- 《Java网络编程》说“如果希望与服务器直接通信，应当使用这个方法”。
- 当你确定 URL 指向的绝对是文本且编码格式为ASCII时，使用 openStream（）方法比较方便，除此之外，就用 openConnection（）方法吧。--作者：conleyfree 来源：CSDN

下载URL资源程序示例(6)

```
//保存文件
        while ((size = bis.read(buf)) != -1)
            fos.write(buf, 0, size);
        fos.close();
        bis.close();
        httpUrl.disconnect();
    }
/*带用户密码的资源保存*/
      public void saveToFile2(String destUrl, String fileName)
throws IOException {
            FileOutputStream fos = null;
            BufferedInputStream bis = null;
            HttpURLConnection httpUrl = null;
            URL url = null;
            byte[] buf = new byte[BUFFER_SIZE];
            int size = 0;
```

下载URL资源程序示例(7)

```
//建立链接
        url = new URL(destUrl);
        httpUrl = (HttpURLConnection) url.openConnection();
//String authString = "username" + ":" + "password";
         String authString = "50301" + ":" + "88888888";
        String auth = "Basic " + new
sun.misc.BASE64Encoder().encode(authString.getBytes());
        httpUrl.setRequestProperty("Proxy-Authorization", auth);
 //连接指定的资源
        httpUrl.connect();
//获取网络输入流
        bis = new BufferedInputStream(httpUrl.getInputStream());
//建立文件
        fos = new FileOutputStream(fileName);
```

下载URL资源程序示例(8)

```
        if (this.DEBUG)
            System.out.println("正在获取链接[" + destUrl + "]的
内容...\n将其保存为文件[" +fileName + "]");
//保存文件
        while ((size = bis.read(buf)) != -1)
            fos.write(buf, 0, size);
        fos.close();
        bis.close();
        httpUrl.disconnect();
    }
```

下载URL资源程序示例(9)

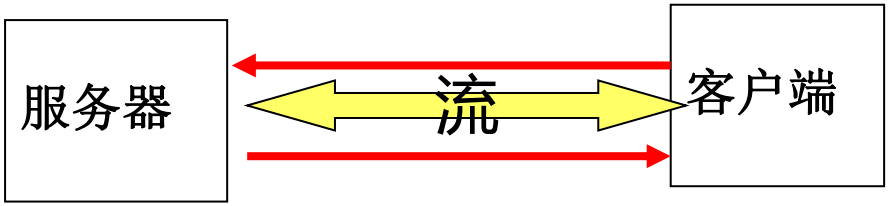
```
public void setProxyServer(String proxy, String proxyPort) {
//设置代理服务器
    System.getProperties().put("proxySet", "true");
    System.getProperties().put("proxyHost", proxy);
    System.getProperties().put("proxyPort", proxyPort);
}

public void setAuthenticator(String uid, String pwd) {
    //Authenticator.setDefault(new Authenticator());
}
```

下载URL资源程序示例(10)

```
    /**
     * 主方法(用于测试)
     * @param argv String[]
     */
    public static void main(String argv[]) {
        HttpGet oInstance = new HttpGet();
        try {
//增加下载列表,此处用户可以自己来增加下载列表
oInstance.addItem("https://www.zhihu.com/question/19709300",
"./info01.html");//
            //开始下载
            oInstance.downLoadByList();
        } catch (Exception err) {
            System.out.println(err.getMessage());
        }
    }
}
```

2、客户端/服务器通信程序

- 通过URL获取的免费开放资源是有限的，重要信息多以服务的方式提供。
- 轻而易举便能搭接网络，是java最强大的威力之一。网络通信方法和文件读写极为相似，只不过这个"文件"位于网络中某台机器，且这台机器有权决定你所请求的信息如何处理。
- 客户端/服务器通信框架：与对端建立网络通信socket；连接输入输出流到对端；读写数据；关闭socket。



Socket套接字[4]

- IP 地址 标识 Internet 上 的 计算机， 端口 号 标识 正在 计算 机上 运行 的 进程 （程序） 。 端 口号 与 IP 地址 的 组合 得出 一个 网络 套 接 字。
- 端 口号 被 规定 为 一个 16 位 的 整数 0 ~ 65535。 其中， 0 ~ 1023 被 预先 定义 的 服务 通信 占用 （ 如 telnet 占用 端口 23， http 占用 端口 80 等） 。
- 当 两个 程序 需要 通信 时， 它们 可以 通过 使用 Socket 类 建立 套 接 字 对象 并 连接 在一 起。

Socket类的使用

- Socket(套接字)是两个程序通过网络通信的一个简单信道。利用协议支持客户端和服务器之间的双工通信。
- 1、服务器程序根据协议在某个端口创建一个ServerSocket,等待客户端的请求。
- 2、客户端创建一个Socket，尝试和已知的服务器端口发出连接请求。
- 3、服务器程序收到并认可客户端的Socket，连接建立。服务器程序为Socket创建输入输出流，开始互传数据。
- 4、客户端为Socket创建输入输出流，开始交换信息。
- 5、客户端关闭Socket，服务器继续等待下一个请求。

建立 服务器套接字[5]

- 当 建立 服务器 套 接 字 时 可能发生 IOException 异常。

```
try{
  ServerSocket wait= new ServerSocket( 1880);
} catch( IOException e){ }
```

- 接收 客户 的 套 接 字 也可能 发生 IOException 异常。

```
try{ Socket socketAtServer= wait.accept();
} catch( IOException e){}
```

建立客户端套接字[6]

- 客户端程序使用 Socket 类 创建 对象

Socket(String host, int port)

try{ Socket socketAtClient= new Socket(" http://192.168.0.78"，1880);

} catch(IOException e){ }

- 客户 机 建立 socketAtClient 对象 的 过程 就是 向 服务器 发出 套 接 字 连接 请求，如果 服务器 相应 的 端口 上有 ServerSocket 对象 正在 使用 accept() 方法 等待 客户 机，那么 双方 的 套 接 字 对象 socketAtClient 和 socketAtServer 就都 诞生 了。

建立流连接[7]

- 客户机和服务器的套接字对象诞生后，还必须进行输入/输出流的连接。

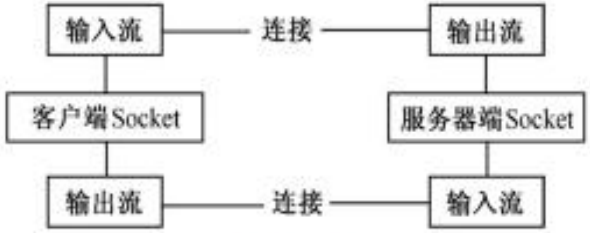


图11-3　套接字连接示意

- Socket 对象使用方法 getOutputStream() 获得输出流，使用方法 getInputStream() 获得输入流。

关闭套接字[8]

- 套 接 字 调用 close() 方法 可以 关闭 双方 的 套 接 字 连接;

- 如果只有一方 关闭 连接， 就会 导致 对方 发生 IOException 异常。

网络版小棍游戏（1）

```
//服务器端程序,创建线程子类，覆盖其run方法
import java.net.*;
import java.io.*;
public class NimServer extends Thread {
  private ServerSocket server;
  private Socket socket;
  public NimServer(int port, int backlog) {
    try { server=new ServerSocket(port,backlog);
    }catch (IOException e) {e.printStackTrace(); }
  }
  public void run() {
    try { while (true) {
        socket =server.accept();
        providService(socket);
        socket.close();
        }//while
    }catch (IOException e) {e.printStackTrace(); }
  }//run()
```

线程类，并发运行

网络版小棍游戏（2）

```
//服务器端程序,创建服务并启动服务providService
 public static void main(String args[]) {
    NimServer server=new NimServer(10001,5);
    server.start();
  }//main()
  private void providService(Socket socket) {
    String str="";
    int userTakes =0;
    OneRowNim nim=new OneRowNim();
    try { writeToSocket(socket,"Hi Nim!"+nim.report()+"\n");
          str = readFromSocket(socket);
          userTakes=Integer.parseInt(str);
          nim.takeSticks(userTakes);
          if (!nim.gameOver()) writeToSocket(socket,
nim.report()+"\n");
    }catch (IOException e) {e.printStackTrace(); }
  }//providService()
```

网络版小棍游戏（3）

```
//服务器端或客户端程序,输入输出流操作
 private void writeToSocket(Socket sock,String str) throws
IOException {
    OutputStream oStream=sock.getOutputStream();
    for (int k=0;k<str.length();k++)
       oStream.write(str.charAt(k));
  }//writeToSocket()

  private String readFromSocket(Socket socket) throws
IOException {
    InputStream iStream = socket.getInputStream();
    String str="";  char c;
    while ( (c=(char)iStream.read()) !='\n')
        str=str+c;
    return str;
  }//readFromSocket()
```

网络版小棍游戏（4）

```
//客户端程序,类似但简单
public class NimClient extends Thread {
  private Socket socket;
  private KeyboardReader kb=new KeyboardReader();

  public NimClient(String url,int port) {
    try { socket=new Socket(url,port);
    }catch (IOException e) {e.printStackTrace(); System.exit(1); }
  }
  public void run() {
    try { requestService(socket);
        socket.close();
    }catch (IOException e) {e.printStackTrace(); }
  }//run()

  public static void main(String args[]) {
    NimClient client=new NimClient("java.trincoll.edu",10001);
    client.start();
  }//main()
```

线程类，并发运行

网络版小棍游戏（5）

```
//客户端程序,请求服务requestService
private void requestService(Socket socket) throws
IOException {
  String servStr = readFromSocket(socket);
  kb.prompt("Nim Server:"+servStr+"\n");
  String userStr="";
  do {userStr = kb.getKeyboardInput();
        writeToSocket(socket,userStr+"\n");
        servStr = readFromSocket(socket);
        kb.prompt("Nim Server:"+servStr+"\n");
  }while servStr.indexOf("GameOver!"==-1);
}
```

OneRowNim小棍游戏（6）

```
public class OneRowNim {
    private int nSticks=7;
    private int player=1;
    OneRowNim(int sticks){
        nSticks=sticks;
    }
    public boolean takeSticks(int num)
    {   if (num<1) return false;//error
        else if (num>3) return false;   //error
        else { nSticks=nSticks-num;  //valid move
               player=3-player;
               return true;
         }
    }
```

OneRowNim小棍游戏（7）

```
    public boolean gameOver()
    {  return (nSticks<=0);
    }
    public void report()
    {   String s=new String("Number of sticks left:"+
getSticks());
    }//每步骤的完整提示
    public int getSticks()
    {  return nSticks;
    }//信息隐藏，尽量少函数访问私有变量

}
```