# 实验 8 网络编程

## 实验目的

（1） 掌握 Socket 通信

（2） 掌握 UDP 数据报通信

（3） 掌握网络编程中的多线程应用

在 Eclipse 下以自己的学号创建 java 项目，每道题目建独立的包，如 work1，work2，work3…… 作业提交时将整个项目和实验报告压缩后提交。

## 任务一：掌握 Socket 类的使用

在两台机器上分别运行以下服务器端程序和客户端程序，截图观察程序的多次运行结果，回答问题。

```
// 服务器端程序
public class Server {
    public static void main(String[] args) {
        ServerSocket sSocket = null;
        Socket  socket = null;
        try {
            sSocket = new ServerSocket(2018);                    //s1
        } catch (IOException e) {
            System.out.println("端口已被占用！");
        }
        try {
            socket = sSocket.accept();                           //s2
            System.out.println(socket.getInetAddress());         //s3
            System.out.println(socket.getPort());                //s4
            System.out.println(socket.getLocalSocketAddress()); //s5
            System.out.println(socket.getLocalPort());           //s6
            socket.close();
        } catch (IOException e) {
            System.out.println("连接客户端异常。");
        }
    }
}
```

```
//客户端程序
public class Client {
    public static void main(String[] args) {
        Socket  socket = new Socket();                           //c1
        InetSocketAddress isa = new InetSocketAddress("[redacted]",2018);
        try {
            socket.connect(isa);                                 //c2
            System.out.println(socket.getInetAddress());         //c3
            System.out.println(socket.getPort());                //c4
            System.out.println(socket.getLocalSocketAddress()); //c5
            System.out.println(socket.getLocalPort());           //c6
            socket.close();
        } catch (IOException e) {
            System.out.println("连接服务器异常。");
        }
    }
}
```

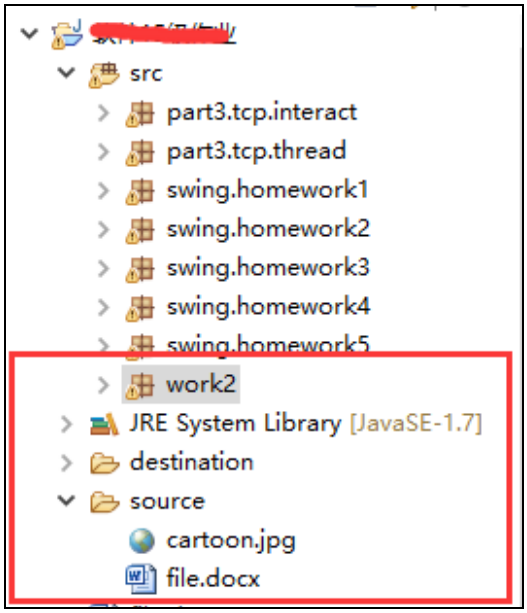
实际服务器端主机IP

对比运行截图，说明服务器端程序和客户端程序添加注释的每行代码作用。

## 任务二：使用 TCP 协议实现文件传送

在当前项目的根目录新建两个文件夹：“source”和“destination”，在 source 中放置两个文件：“cartoon.jpg”以及“file.docx”。这两个文件见实习附件。实现一个服务器端程序 Server，服务器端可以通过文件流直接读取文件内容。实现一个客户端程序 Client，客户端程序运行后，提示用户输入文件名。如，用户在客户端输入“file.docx”并回车，则程序通过 TCP 协议读取服务器端的数据，将该文件传送一个副本保存在“destination”目录中。

项目的目录以及客户端的执行过程如下图所示。

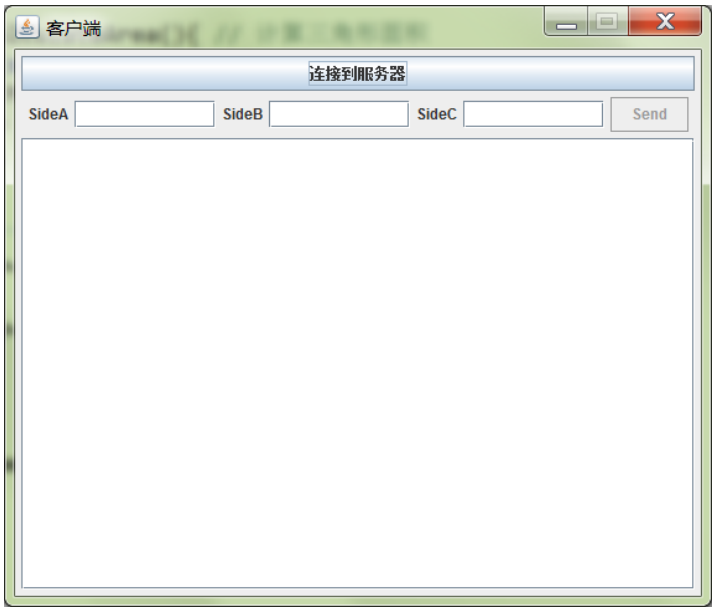



## 任务三：对象流在网络中的应用

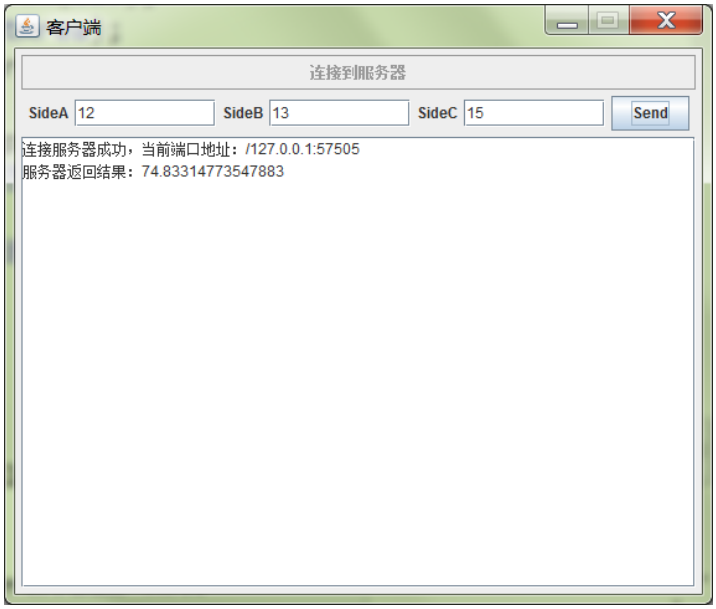
现有一个 Triangle（三角形）类：

```
public class Triangle implements Serializable{
    double sideA;
    double sideB;
    double sideC;
    String area;
    public Triangle(double a,double b,double c){
        this.sideA = a;
        this.sideB = b;
        this.sideC = c;
    }

    public boolean isLegal(){ //判断三角形三条边是否合法

    public String calculateArea(){ // 计算三角形面积
        double p = (sideA + sideB + sideC) / 2.0;
        area = "" + Math.sqrt(p * (p - sideA) * (p - sideB)
                * (p - sideC));
        return area;
    }
}
```

- 思考：为什么 Triangle 需要实现 Serializable 接口？
- 服务器用于接收客户端传来的 Triangle 对象，计算三角形面积，将计算结果传给客户端。
- 连接服务器之前：



- 点击“连接服务器”实现连接，点击“send”提交计算请求：




- 提示：在“连接到服务器”按钮中，使用 Socket 的 connect()方法进行连接；接收和发送数据时，上层流使用对象流。
- 基本要求：使用 ServerSocket 类和 Socket 类实现单线程对象的发送与接收。实现计算结果的发送与接收。服务器端可以不使用界面。

## 选做内容：

- 在服务器中实现多线程，使得一个服务器可以同时和多个客户端通信。

## 任务四：使用 UDP 数据报进行聊天

用 UDP 数据报传输数据，实现两个终端的消息通信

使用 Java 命令分别运行服务器端和客户端

基本要求：实现两个终端 UDPA 和 UDPB，两个终端可以互发字符串消息。

若某一端（如 A）发送了一个“TIME”字符串，则另一端（如 B）将其视作一个命令，将目前的系统时间立即发送过去。

- 提示：接收到数据报文之后进行判断，str.equals(“TIME”)，如果为真，则通过 java.util.Date 类中的 toString()获得时间字符串。