

# 计算机网络实验二

## 基于 java 的网络选课模拟程序

班级：电气 15

学号：01041129

姓名：杨菁

指导教师：薛涛

## 1 实验目的

- (1) 熟悉和掌握网络编程的基本方法和步骤；
- (2) 进一步理解 client/server 交互模式；
- (3) 加深学生对于网络协议概念的理解以及协议的设计和实现方法。

## 2 实验任务

使用任意网络编程语言（Java、C、VB、Delphi 等）编写网络选课模拟程序，它由 client 和 server 两部分组成，client 和 server 之间的通信基于 TCP 协议，并且实现 NCSP 应用层协议。

## 3 程序介绍

运行环境：Windows9x/NT/2000/XP

开发环境：Windows XP

JCreator

程序简介：

**编译：** 直接在开发环境 JCreator 中编译具体的运行顺序是：先运行服务器端程序，然后运行客户端程

序。

## 4 设计思路:

该模拟选课程序主要实现了以下功能,该程序也是基于以下功能设计的:

### SETCOURSE 指令:

当接收到 client 发送的 GETCOURSE 协议消息,返回该用户已经选择的所有课程名字;

### GETCOURSE 指令:

当接收到 client 发送的 SETCOURSE 协议消息,记录用户名和课程名,以便 client 使用 GETCOURSE 命令查询;

### SHUTDOWN 指令:

当接收到 client 发送的 SHUTDOWN 协议消息,检查用户是否有此权限(即要求用户输入特权用户名和密码),如果有则结束 server 程序。

### 设计思路如下:

该模拟选课程序遵从 client/server 交互模式。在该模型中,服

服务器应用是一种专门用来提供某一种服务的程序，它需要先运行，等待通信，而客户应用主动地启动通信，请求服务器的服务。服务器程序既可以接收信息也能发送信息，还可以成为客户。

TCP/UDP 赋予每个服务一个唯一的协议端口号。服务器程序通过协议端口号来指定它所提供的服务，然后被动地等待通信。客户在发送连接请求时，必须说明服务器程序运行主机的 IP 地址和协议端口号来指定它所希望的服务。服务器端计算机通过此端口号将收到的请求转向正确的服务器程序。

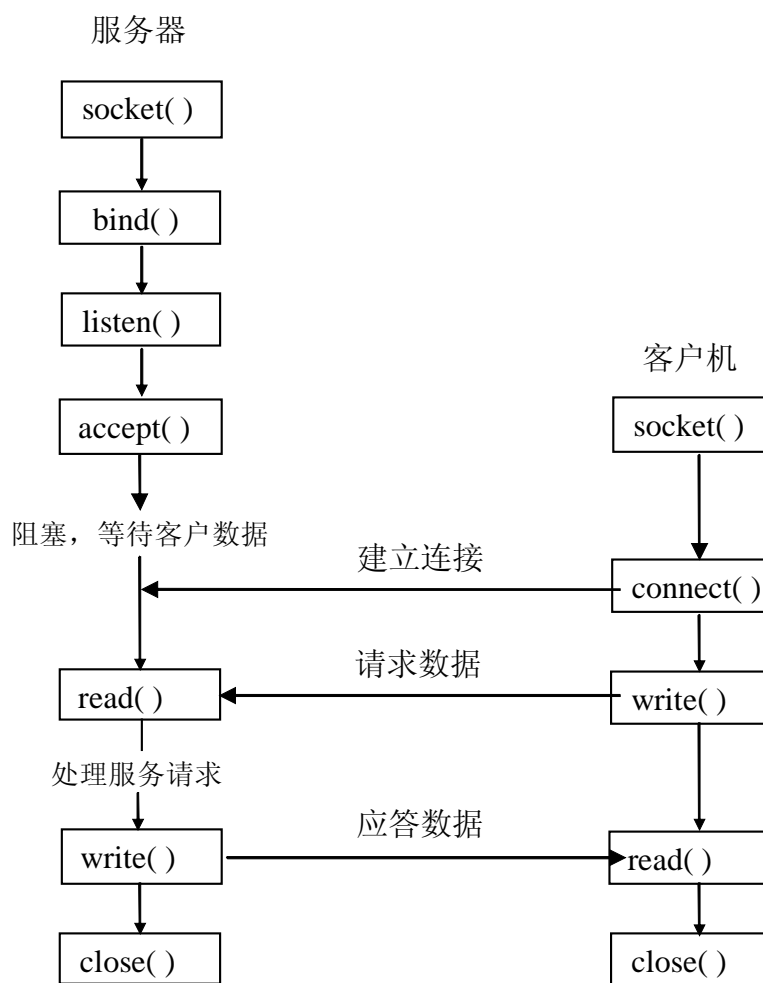
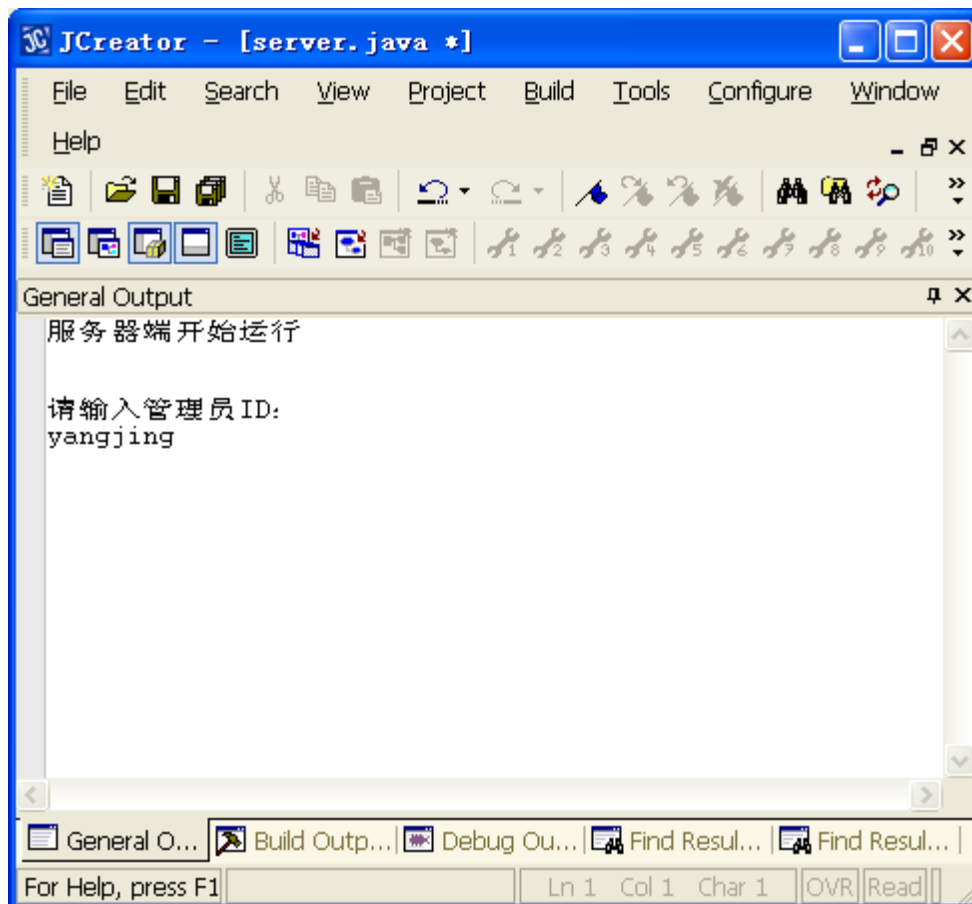


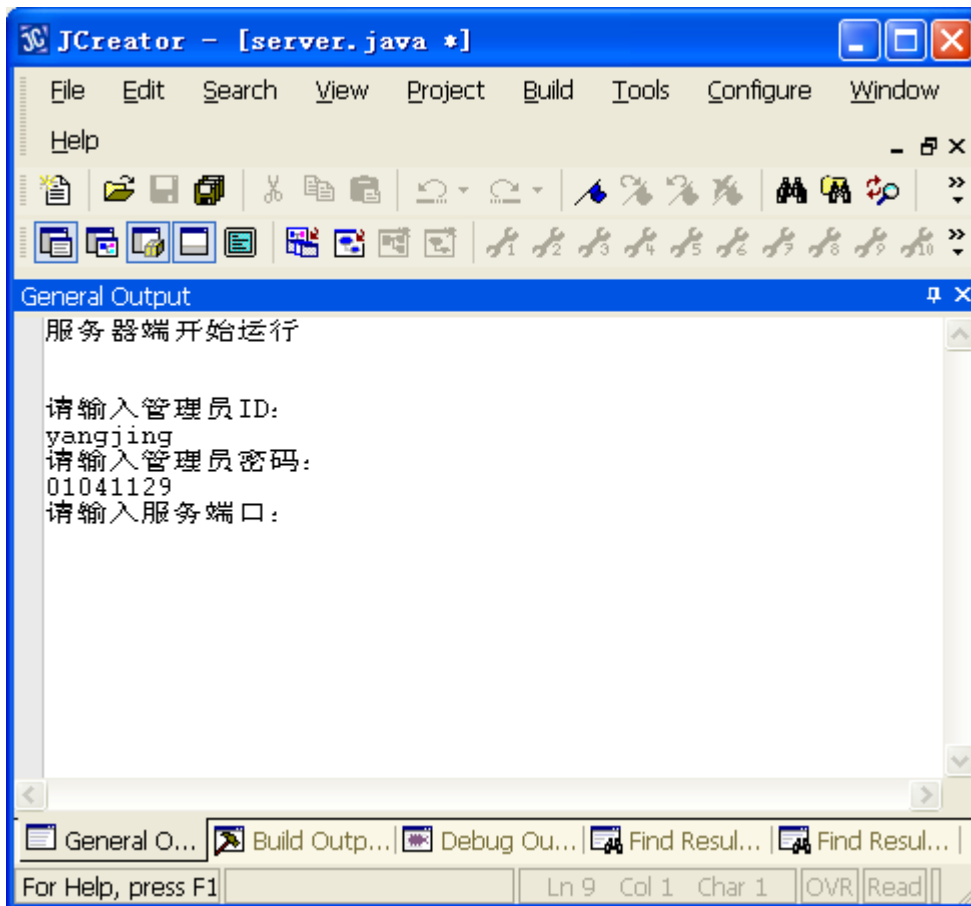
图 1 面向连接套接口应用程序时序图

## 5 具体实验过程如下:

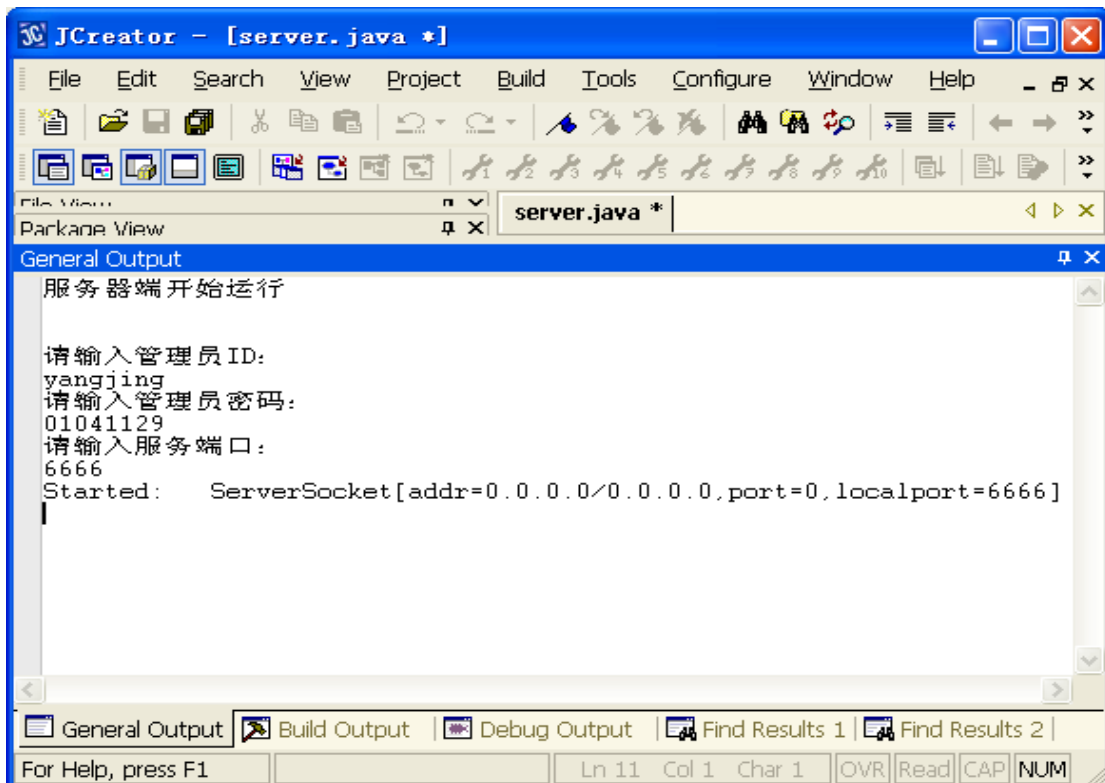
1、服务器端启动，输入管理员的 ID:



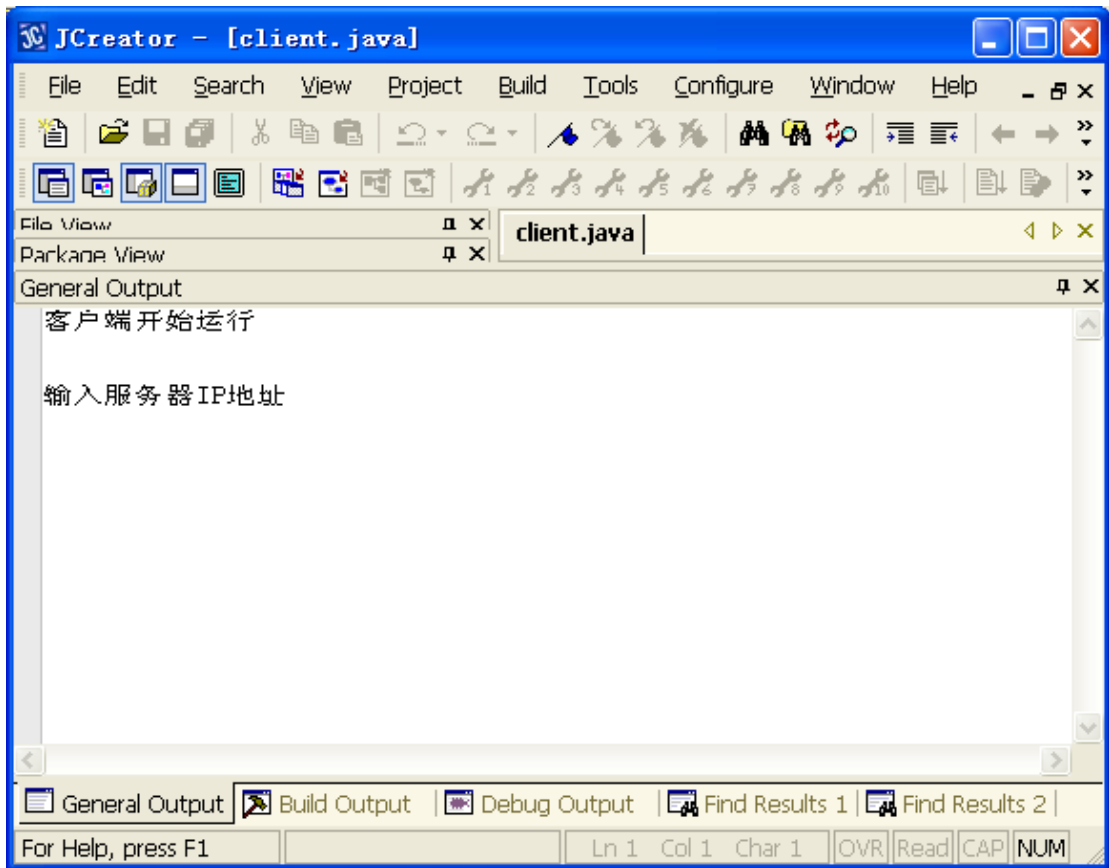
2. 输入管理员的密码:



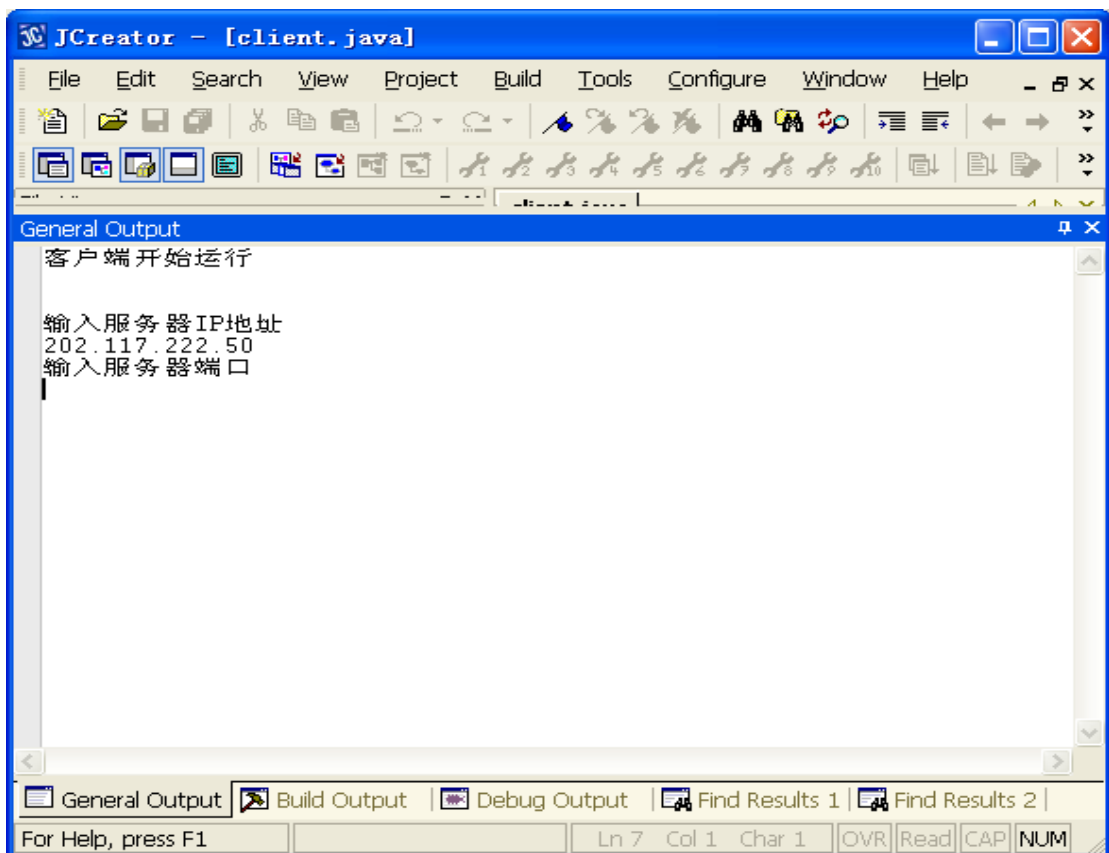
### 3、输入服务器的端口号:



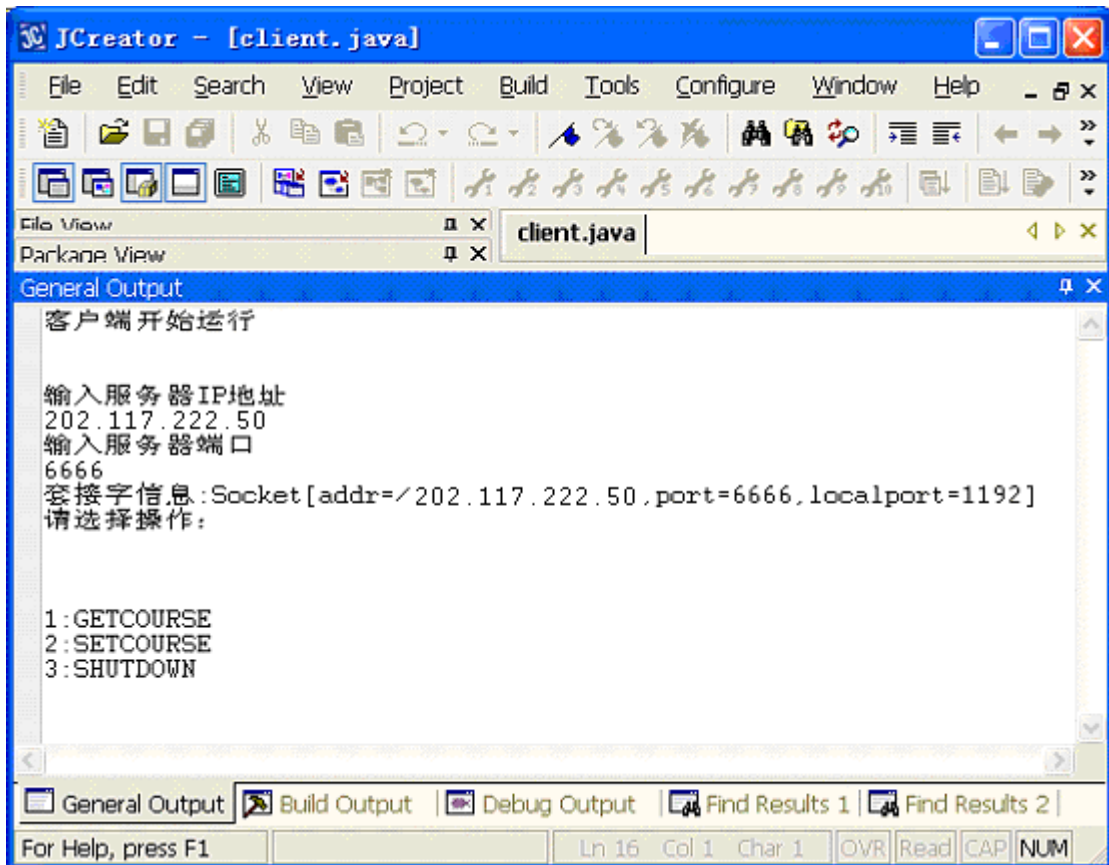
### 4、服务器端已开启，启动客户端:



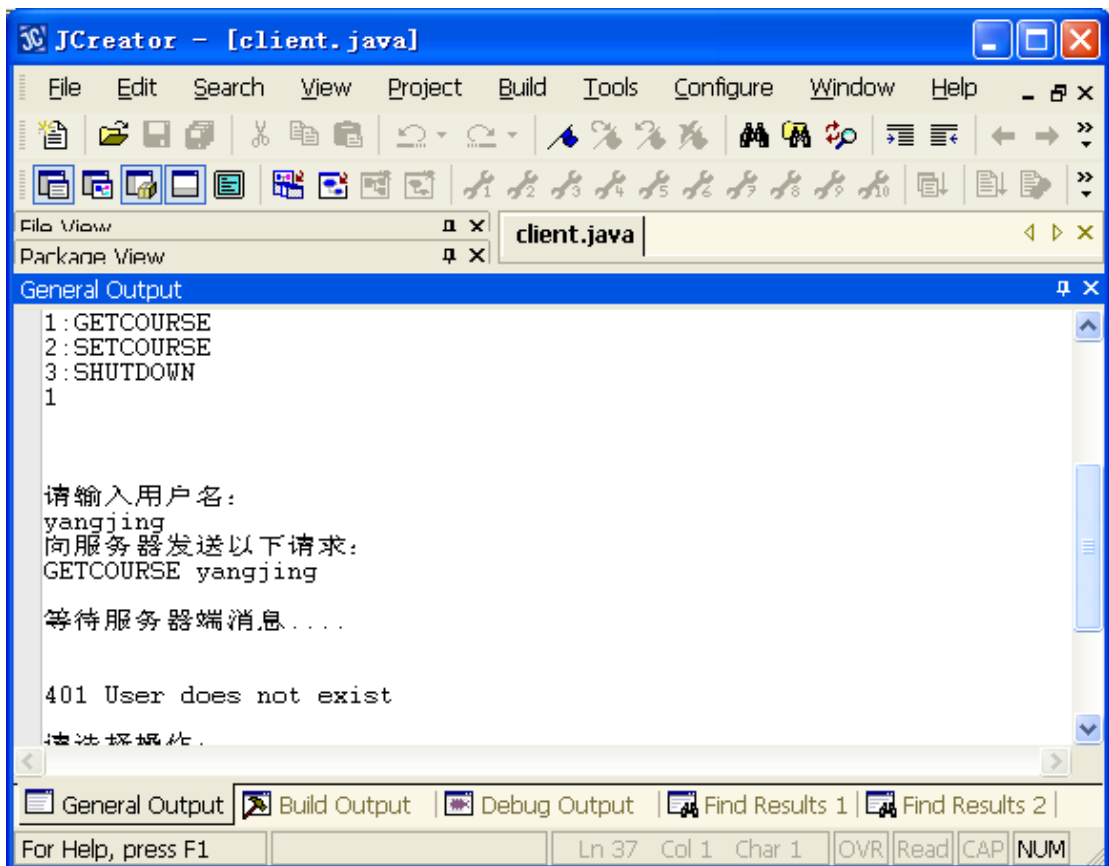
##### 5、输入服务器的 IP 地址



##### 6、输入服务器的端口号:



7、进入选课的操作见面，选1并输入一个不存在的用户名，服务器拒绝服务



服务器段显示如下:



```
JCreator - [server.java *]
File Edit Search View Project Build Tools Configure Window Help
File View server.java *
General Output
请输入管理员ID:
yangjing
请输入管理员密码:
01041129
请输入服务端口:
6666
Started: ServerSocket[addr=0.0.0.0/0.0.0.0,port=0,localport=6666]
Socket: Socket[addr=/202.117.222.50,port=1192,localport=6666]
等待客户端通信

GETCOURSE yangjing

401 User does not exist
等待客户端通信

General Output Build Output Debug Output Find Results 1 Find Results 2
For Help, press F1 Ln 1 Col 1 Char 1 OVR Read CAP NUM
```

8、选 2 并输入用户名:

```
JCreator - [client.java]
File Edit Search View Project Build Tools Configure Window Help
File View client.java
Parkane View
General Output
2: SETCOURSE
3: SHUTDOWN
2

请输入姓名:
yangjing
向服务器发送请求:

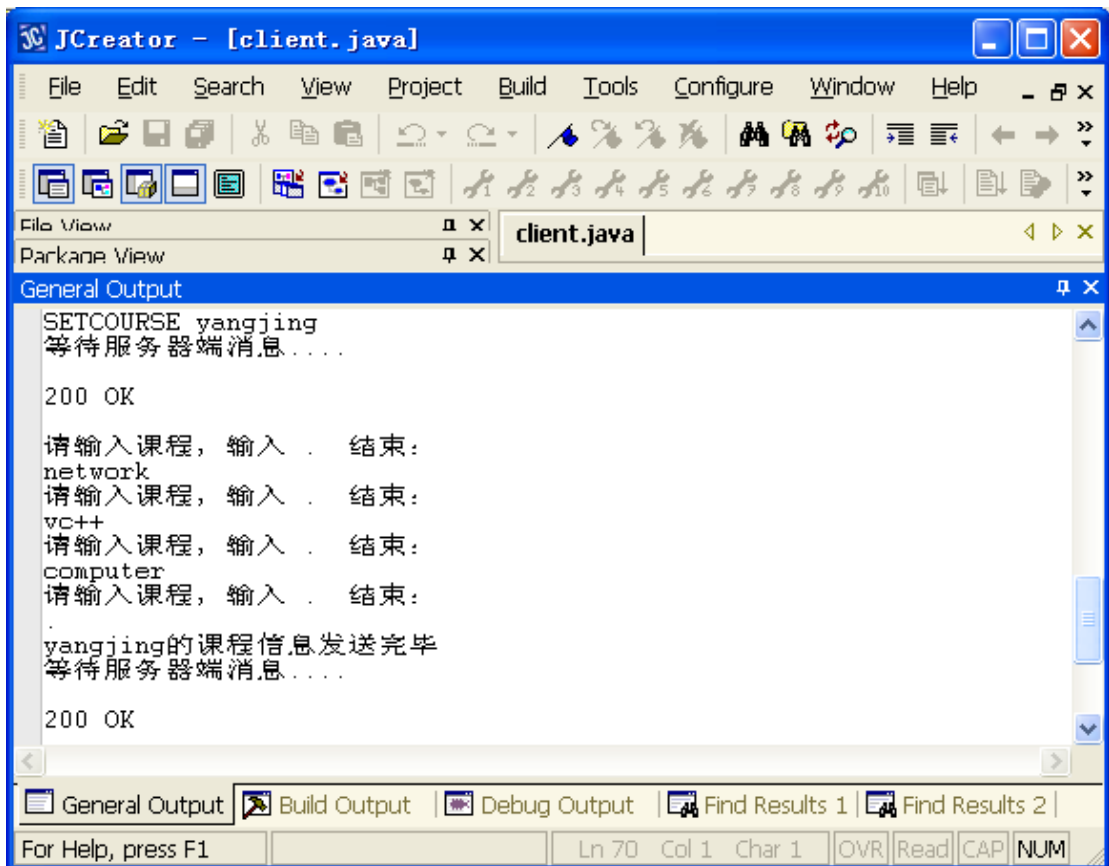
SETCOURSE yangjing
等待服务器端消息....

200 OK

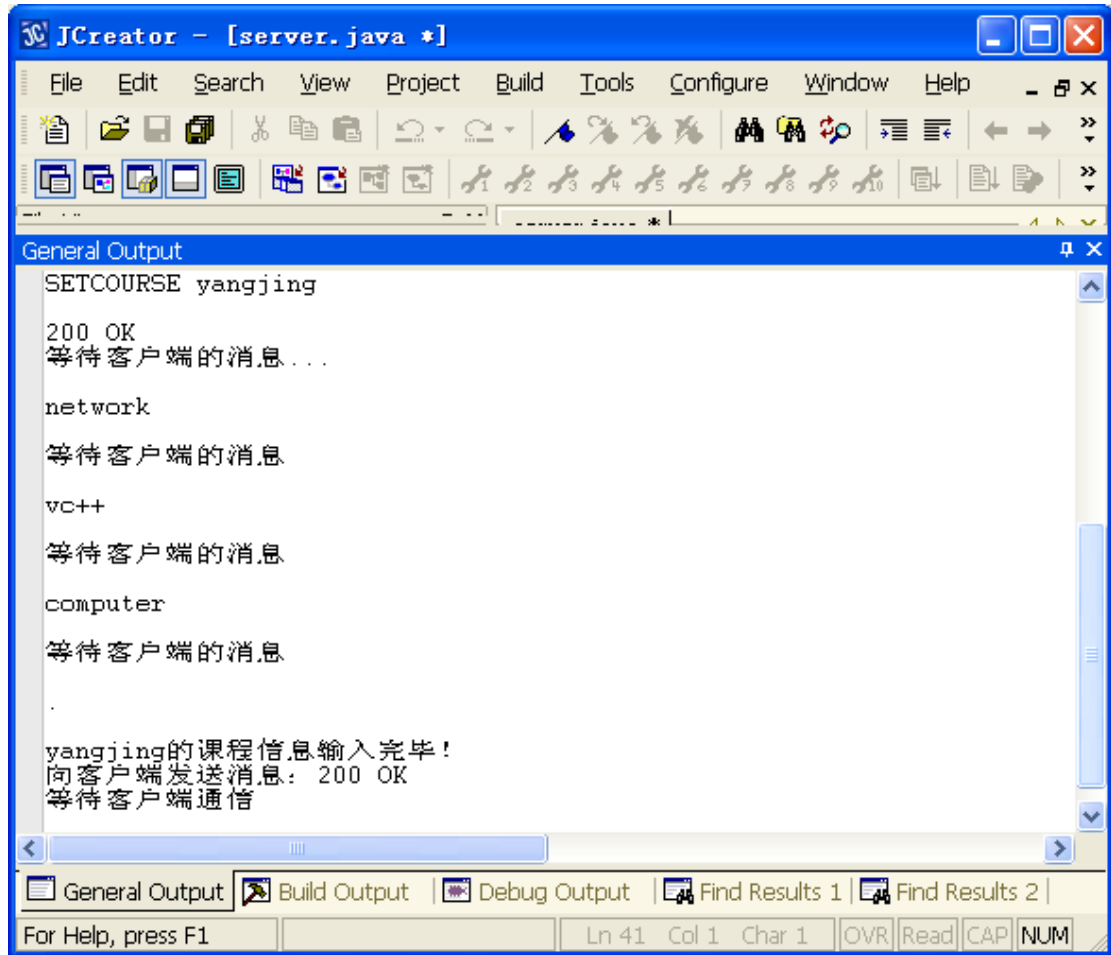
请输入课程, 输入 . 结束:

General Output Build Output Debug Output Find Results 1 Find Results 2
For Help, press F1 Ln 51 Col 1 Char 1 OVR Read CAP NUM
```

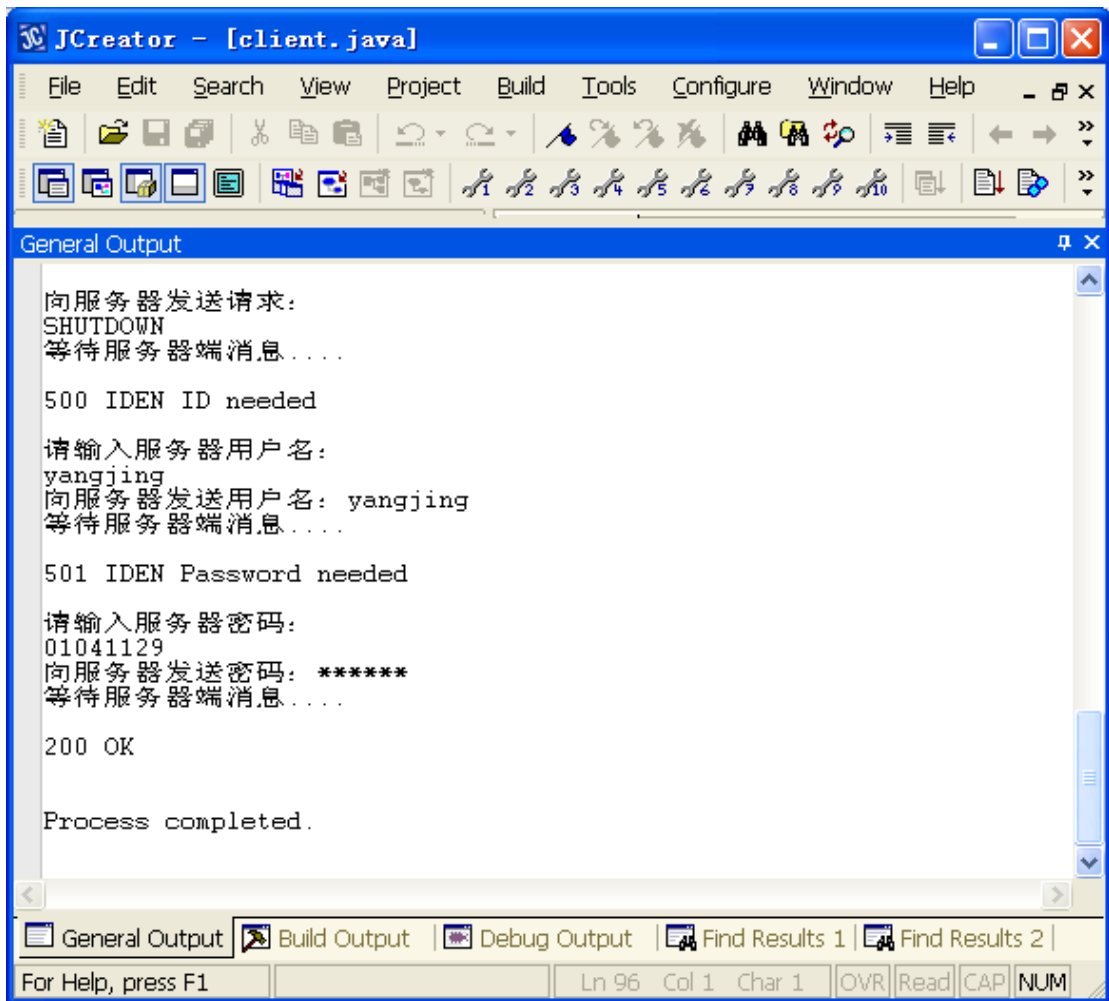
9、输入一系列课程，进行网上选课（可以选多门课），用.结束选课：



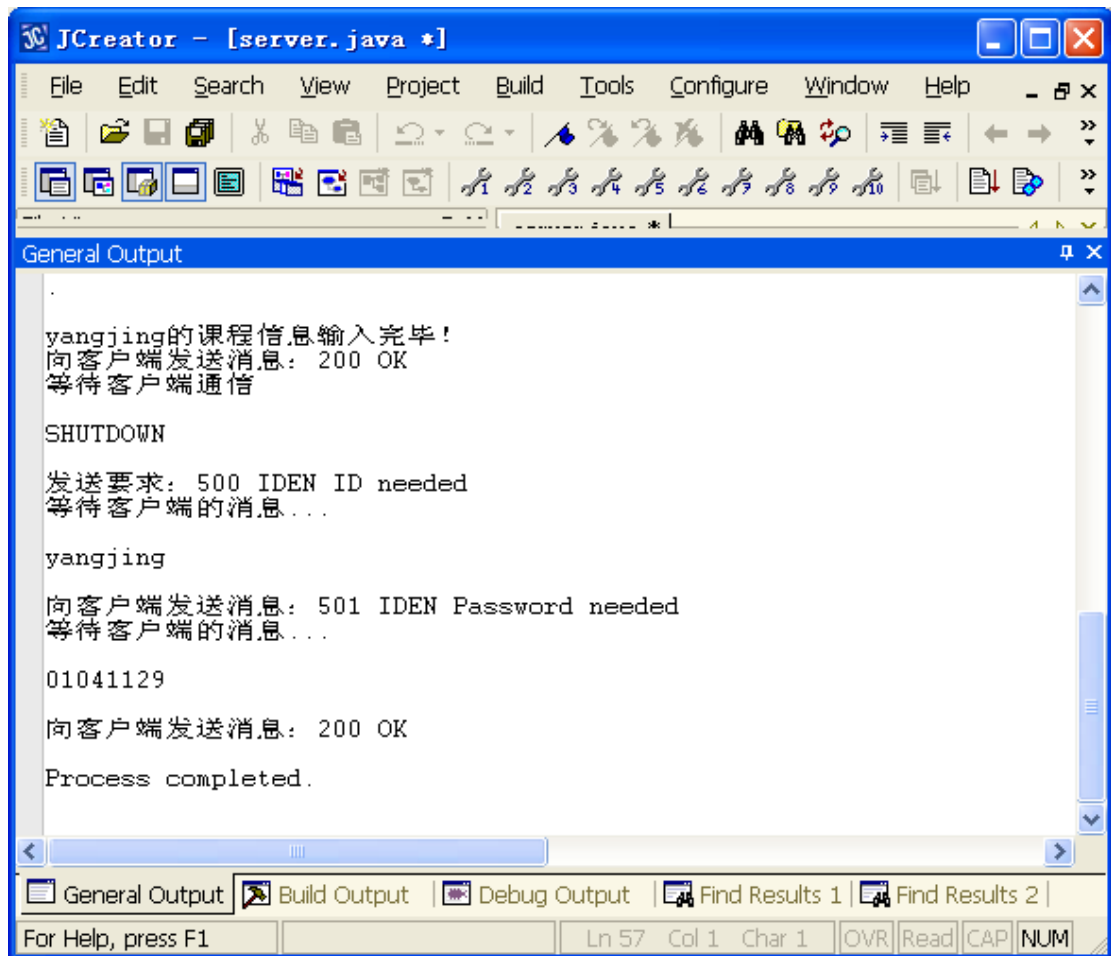
10、服务器段显示如下：



11、选3，在客户端输入服务器管理员的用户名和密码，关闭服务器：



14、服务器段显示如下：



6 源代码如下：

```
server:  
//server  
import java.net.*;  
import java.io.*;  
import java.util.*;  
public class server  
{  
    public static Vector Log;  
  
    private static int Count = 0;
```

```

//记录学生记录数
public static void main(String args[])
{
    Log = new Vector();
    //记录选课信息
    String lin;
    String lef;
    //定义字符串变量
    int port ;
    //定义端口号
    String iden;
    String pass;
    //定义服务的用户名及其密码
    boolean quit=false;
    //控制标志
    try
    {
        InputStreamReader spuou= new InputStreamReader(System.in);
        BufferedReader spuin = new BufferedReader(spuou);
        //用户键盘输入流
        System.out.println("服务器端开始运行");
        System.out.println("\n");
        System.out.println("请输入管理员 ID: ");
        iden = spuin.readLine();
        System.out.println("请输入管理员密码: ");
        pass = spuin.readLine();
        System.out.println("请输入服务端口: ");
        String inputPort;
        inputPort = spuin.readLine();
        port = Integer.parseInt(inputPort);
        //输入建立服务必需的信息: 用户名, 密码, 端口
        ServerSocket server = new ServerSocket(port);
        System.out.println("Started: " + server);
        Socket socket = server.accept();
        System.out.println("Socket: " + socket);
        //建立 Socket 连接
        InputStream fIn = socket.getInputStream();
        OutputStream fOut = socket.getOutputStream();
        InputStreamReader uou = new InputStreamReader(fIn);
        BufferedReader in = new BufferedReader(uou);
        PrintStream out = new PrintStream(fOut);
        //取得 Socket 连接的输入输出流
        while(true)//进入与客户端的通信
        {

```

```

System.out.println("等待客户端通信");
    lin = in.readLine();
    System.out.println("\n"+lin+"\n");
    //取得客户端消息
    int p=ServRes(lin);
    //分离命令
    switch (p)
    {
        case 1://GETCOURSE
        {
            StringTokenizer min=new StringTokenizer(lin," ");
            min.nextToken();
            String sname = min.nextToken();
            //取得用户名
            if(!exitUser(sname))//用户名不存在
            {
                lef = "401 User does not exist";
                System.out.println(lef);
                out.println(lef);
            }//if
            else//用户存在
            {
                lef = "200 OK";
                System.out.println("发送要求: "+lef);
                out.println(lef);
                //发送课程
                for(int k= 0; k <  Log.size(); k++)
                {

                    if(((NCLog)Log.elementAt(k)).sname.equals(sname))
                    {
                        int h = 0;

                        while(((NCLog)Log.elementAt(k)).scourse[h]!=""&&
                            ((NCLog)Log.elementAt
(k)).scourse[h]!=null)
                            {
                                lef=
                                    ((NCLog)Log.elementAt
(k)).scourse[h];
                                h++;
                                System.out.println("向客户端发送消
息
                                    : "+lef);
                                out.println(lef);

```

```

        }
        out.println(".");//发送 . 表示推出
    }//if
} //for
} //else
break;
} //case 1
case 2://SETCOURSE
{
    StringTokenizer min=new StringTokenizer(lin," ");
    min.nextToken();
    String sname = min.nextToken();
    //取得用户名
    if(exitUser(sname))//用户存在
    {
        lef = "301 User exists";
        System.out.println(lef);
        out.println(lef);
    } //if
    else//不存在用户
    {
        lef = "200 OK";
        System.out.println(lef);
        out.println(lef);
        //添加用户，接受课程
        Log.addElement(new NCLog(sname));
        Count++;
        System.out.println("等待客户端的消息...");
        lin = in.readLine();
        System.out.println("\n"+lin+"\n");
        int n=0;
        while(!lin.equals("."))//记录课程信息，为 . 就退出
        {
            for(int m= 0; m < Log.size(); m++)
            {
                if(((NCLog)Log.elementAt(m)).sname.equals
                (sname))
                {
                    ((NCLog)Log.elementAt(m)).scourse
                    [n]=lin;
                    n++;
                }
            }
        }
    }
}

```



消息

毕! ");

"+lef);

```
System.out.println("等待客户端的
...");
lin = in.readLine();
System.out.println("\n"+lin+"\n");
} //if
} //for
}
System.out.println(sname+"的课程信息输入完
lef = "200 OK";
System.out.println("向客户端发送消息: "+lef);
out.println(lef);
//成功记录信息
} //else
break;
} //case 2
case 3://SHUTDOWN
{
lef = "500 IDEN ID needed";
System.out.println("发送要求: "+lef);
out.println(lef);
//索取用户名
System.out.println("等待客户端的消息...");
lin = in.readLine();
System.out.println("\n"+lin+"\n");
//取得用户名
if(lin.equals(iden))//如果校验用户名成功
{
lef = "501 IDEN Password needed";
System.out.println("向客户端发送消息: "+lef);
out.println(lef);
//索取密码
System.out.println("等待客户端的消息...");
lin = in.readLine();
System.out.println("\n"+lin+"\n");
//取得密码
if(lin.equals(pass))//如果校验密码成功
{
lef = "200 OK";
System.out.println("向客户端发送消息:
out.println(lef);
```

```

        quit = true;
        //改变运行标志以正常退出
    }
    else//如果校验密码错误
    {
        lef = "301 Permission deny";
        System.out.println("向客户端发送要求:
"+lef);

        out.println(lef);
        //发送拒绝消息
    }
} //if
else//如果校验用户名错误
{
    lef = "301 Permission deny";
    System.out.println("向客户端发送要求: "+lef);
    out.println(lef);
    //发送拒绝消息
}
break;
}
default : //客户端请求无法解析
{
    lef = "300 Message format error";
    System.out.println("向客户端发送消息: "+lef);
    out.println(lef);
    //发送解析错误信息
    break ;
}
}
if(quit) break;
//判断是否退出
} //while
socket.close();
server.close();
//关闭 Socket
} //try
catch(Exception e)//异常处理
{
    System.out.println("程序发生异常: "+e);
}
//保存
}
//是否存在用户

```

```

public static boolean exitUser(String name)
{
    for(int i = 0; i < Log.size(); i++)
    {
        if(((NCLog)Log.elementAt(i)).sname.equals(name))
        {
            return true;
        }
    }
    return false;
}
//处理客户端发送过来的消息
public static int ServRes(String Req)
{
    StringTokenizer lin=new StringTokenizer(Req, " ");
    String firstStr = lin.nextToken();
    int i = 0;
    if(firstStr.equals("GETCOURSE"))i=1;
    else if(firstStr.equals("SETCOURSE"))i=2;
    else if(firstStr.equals("SHUTDOWN"))i=3;
    return i;
}
}
class NCLog //NCLog 类，用于记录选课信息
{
    public String sname;//用户名
    public String [] course;//课程数组
    public final static int CourseNum = 10;//最大选课数目
    NCLog(String name)//构造函数
    {
        course = new String[CourseNum];
        sname = name;
    }
}
}

```

### **client:**

```

//client
import java.net.*;
import java.io.*;
public class client
{

```

```

private final static int StudentNum = 10;//定义最大选课数目
public static void main(String[] args)
{
    String lin;
    //定义传递消息的字符串

    String host ;
    int port ;
    boolean quit=false;
    //定义标志位
    try
    {
        InputStreamReader spuou = new InputStreamReader(System.in);
        BufferedReader spuin = new BufferedReader(spuou);
        //用户键盘输入流
        System.out.println("客户端开始运行");
        System.out.println("\n");
        System.out.println("输入服务器 IP 地址");
        host = spuin.readLine();
        System.out.println("输入服务器端口");
        String inputPort = spuin.readLine();
        port = Integer.parseInt(inputPort);
        //输入建立连接必需的服务器地址和端口
        InetAddress addr = InetAddress.getByName(host);
        Socket socket = new Socket(addr,port);
        System.out.println("套接字信息:"+socket);
        //建立 Socket 连接
        InputStream fIn = socket.getInputStream();
        OutputStream fOut = socket.getOutputStream();
        InputStreamReader uou = new InputStreamReader(fIn);
        BufferedReader in = new BufferedReader(uou);
        PrintStream out = new PrintStream(fOut);
        //取得 Socket 连接的输入输出流
        while(true)//进入与服务器的通信
        {
            System.out.println("请选择操作： ");
            System.out.println("\n\n");
            System.out.println("1:GETCOURSE");
            System.out.println("2:SETCOURSE");
            System.out.println("3:SHUTDOWN ");
            int option= Integer.parseInt(spuin.readLine());
            System.out.println("\n\n");
            //发送请求
            String name;

```

```

//用户名
switch (option)
{
    case 1://GETCOURSE
    {
        System.out.println("请输入用户名: ");
        name = spuin.readLine();
        //输入用户名
        System.out.println("向服务器发送以下请求: ");
        lin = "GETCOURSE "+name;
        System.out.println(lin+"\n");
        out.println(lin);
        //发送 GETCOURSE request
        System.out.println("等待服务器端消息...."+"n");
        while(!lin.equals("."))//取得课程, 输出 . 表示结束
        {
            lin=in.readLine();
            if(lin.equals("."))//接受结束
            {
                System.out.println(name+"的课程信息接受完毕
"+"n");
                break;
            }
            System.out.println("\n"+lin+"\n");
            if(lin.equals("401 User does not exist")||lin.equals("300
Message
format error"))break;
            //显示出错信息
        }
        break;
    }
    case 2://SETCOURSE
    {
        System.out.println("请输入姓名: ");
        name = spuin.readLine();
        //用户输入用户名
        System.out.println("向服务器发送请求: "+"n");
        lin = "SETCOURSE "+name;
        System.out.println(lin);
        out.println(lin);
        //发送 SETCOURSE request
        System.out.println("等待服务器端消息....");
        lin=in.readLine();
        System.out.println("\n"+lin+"\n");
    }
}

```

```

//接受响应及显示
if(!lin.equals("301 User exists"))break;//错误信息
lin = "Blank";
int num = 0;//选课课程计数
while(!lin.equals("."))
{
    if(num<StudentNum-1)
    {
        num++;
        System.out.println("请输入课程，输入 . 结束：");

        lin = spuin.readLine();
        out.println(lin);
    }
    else
    {
        System.out.println("已经达到最大课程数");
        lin = ".";
        out.println(lin);
        break;
    }
}
System.out.println(name+"的课程信息发送完毕");
System.out.println("等待服务器端消息....");
lin=in.readLine();
System.out.println("\n"+lin+"\n");
break;
}
case 3://SHUTDOWN
{
    System.out.println("向服务器发送请求： ");
    lin = "SHUTDOWN";
    System.out.println(lin);
    out.println(lin);
    //发送 SHUTDOWN request
    System.out.println("等待服务器端消息....");
    lin=in.readLine();
    System.out.println("\n"+lin+"\n");
    if(!lin.equals("500 IDEN ID needed"))break;
    //解析错误则退出
    System.out.println("请输入服务器用户名： ");
    lin = spuin.readLine();
    //输入用户名
    System.out.println("向服务器发送用户名： "+lin);
}

```

```
out.println(lin);
//发送用户名
System.out.println("等待服务器端消息....");
lin=in.readLine();
System.out.println("\n"+lin+"\n");
//接受服务器回复显示
if(lin.equals("301 Permission deny"))//用户名错误
{
    System.out.println("无停止服务权限！");
    break;
}
else//用户名正确
{
    System.out.println("请输入服务器密码： ");
    lin = spuin.readLine();
    //输入密码
    System.out.println("向服务器发送密码： *****");
    out.println(lin);
    //发送密码
    System.out.println("等待服务器端消息....");
    lin=in.readLine();
    System.out.println("\n"+lin+"\n");
    //接受服务器回复
        if(lin.equals("301 Permission deny"))//密码错误
        {
            System.out.println("密码有误！");
            break;
        }
        else if (lin.equals("200 OK"))//密码正确
        {
            quit=true;//正常退出
        }
    }
    break;
}
default:
    break;
}
if(quit)//判断退出标志
{
    break;
}
}
}while
socket.close();//关闭通信
```

```
    }//try
    catch(Exception e)//异常处理
    {
        System.out.println("程序出现异常:"+e);
    }
}
}
```

## 7 测试数据

1、从客户端输入 GETCOURSE, 当服务端接收到 client 发送的 GETCOURSE 协议消息, 返回该用户已经选择的所有课程名字;

2、从客户端输入 SETCOURSE, 当接收到 client 发送的 SETCOURSE 协议消息, 记录用户名和课程名, 以便 client 使用 GETCOURSE 命令查询;

3、从客户端输入 SHUTDOWN, 当接收到 client 发送的 SHUTDOWN 协议消息, 检查用户是否有此权限, 如果有则结束 server 程序。

## 8 实验感受

完成这个程序后, 我最大的感受就是我对网络编程有了更深的了解。我对 java 有了更多的认识, 对它的掌握也很熟悉多了。除此之外, 我对网络的协议也有了很深的了解, 比如说传输层的协议 TCP, 它需要建立连接, 提供可靠行传输, 而且我对它的格式也有了很深的了解。也正是因为对它有了比较深的了解, 我在编程时候才能很好的应用它。还有, 我也更加清楚了 C/S 模式。在该模型中, 服务器应用是一种专门用来提供某一种服务的程序, 它需要先运行, 等待通信, 而客户应用主动地启动通信, 请求服务器的服务。另外, 我很特别的清楚了, 进程



之间的通信应该应用 IP 地址和端口号来确保正确的通信。总之，通过这个程序的编写，我真的学到了很好东西，我对书上的知识有了更深入的了解。而且，我也通过查阅资料和工具书，增强了我的 java 实际变成能力。尤其是当我程序的调试好程序后，看到自己的劳动成果时，看到自己的知识变成一个实用的程序时，我真的是十分满足和十分高兴。这次实验激发了我对网络编程的兴趣，增强了我对网络的认识，今后我会更加深入的学习网络知识，编更好的网络程序。