重庆大学大数据与软件学院

上机实验报告

上机实践项目　基于 TCP 的套接字编程

课程名称　计算机网络

姓名	贺凯	成绩	
学号	20221783	教师	**胡海波**
班级	软工 04 班	日期	2024/5/18

《计算机网络》上机实验报告

开课实验室： D1502 **2024 年 5 月 18 日**

姓　名	贺凯	年级、班级	2022 级软件工程 04 班	学号	**20221783**
上机（项目）名称		基于 TCP 的套接字编程		指导教师	胡海波
教师评语	教师签名： **年　月　日**				

一、上机目的

学习如何构建最简单的、基于 C/S 模式的通信程序

二、基本原理

IDEA, short for IntelliJ IDEA, is an integrated development environment for the Java programming language. It is widely recognized as the best Java development tool in the industry, especially for its exceptional features in intelligent code assistant, code auto-completion, refactoring, JavaEE support, various version control tools (such as git, svn), JUnit, CVS integration, code analysis, and innovative GUI design. IDEA is a product of JetBrains, a company headquartered in Prague, Czech Republic, known for its rigorous and professional team of developers, mainly from Eastern Europe. The flagship version of IDEA also supports HTML, CSS, PHP, MySQL, Python, and more. The free version only supports a few languages like Java and Kotlin.

MySQL is an open-source relational database management system developed by MySQL AB, a company based in Sweden and now maintained by Oracle Corporation. It supports multiple operating systems, including Linux, Windows, and macOS. MySQL operates on a client/server model and provides efficient, reliable, and stable data storage and management services. It is one of the most popular open-source relational databases currently available and is widely used in various fields such as web applications, enterprise applications, and mobile applications.

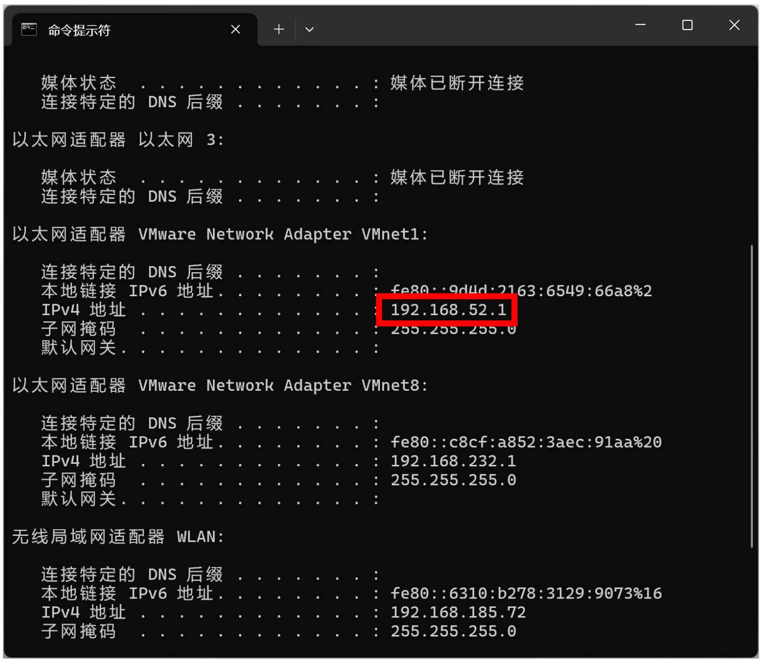
三、使用的软件、硬件

软件：IDEA 和 MySQL 服务

硬件：Windows-PC

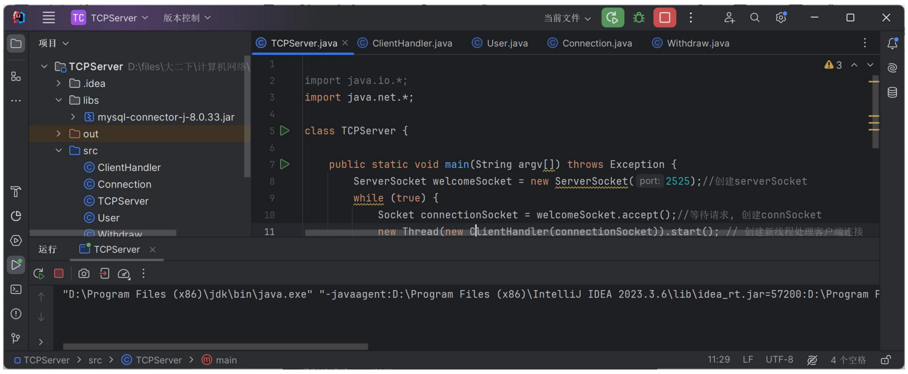
四、上机操作步骤

1. 在命令提示符使用’ipconfig’指令获取服务端 IP 地址



```
   媒体状态  . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :

以太网适配器 以太网 3:

   媒体状态  . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :

以太网适配器 VMware Network Adapter VMnet1:

   连接特定的 DNS 后缀 . . . . . . . :
   本地链接 IPv6 地址. . . . . . . . : fe80::9d4d:2163:6549:66a8%2
   IPv4 地址 . . . . . . . . . . . . : 192.168.52.1
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . :

以太网适配器 VMware Network Adapter VMnet8:

   连接特定的 DNS 后缀 . . . . . . . :
   本地链接 IPv6 地址. . . . . . . . : fe80::c8cf:a852:3aec:91aa%20
   IPv4 地址 . . . . . . . . . . . . : 192.168.232.1
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . :

无线局域网适配器 WLAN:

   连接特定的 DNS 后缀 . . . . . . . :
   本地链接 IPv6 地址. . . . . . . . : fe80::6310:b278:3129:9073%16
   IPv4 地址 . . . . . . . . . . . . : 192.168.185.72
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
```

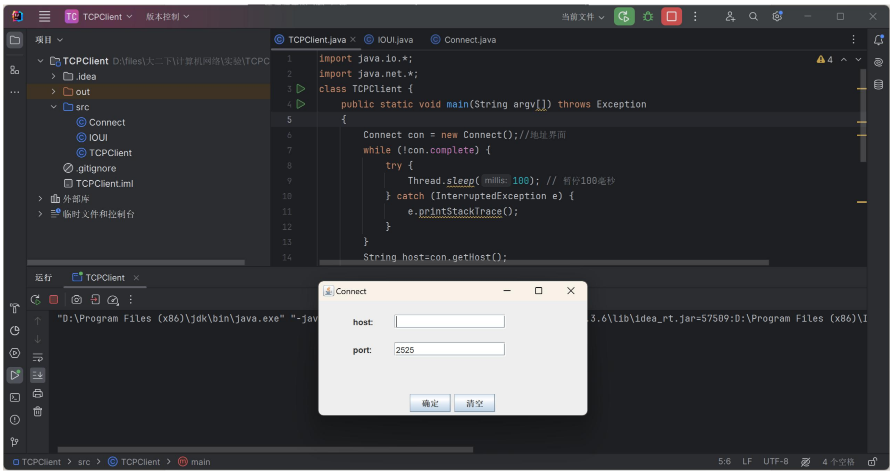
2. 运行服务端，开启服务



```java

import java.io.*;
import java.net.*;

class TCPServer {

    public static void main(String argv[]) throws Exception {
        ServerSocket welcomeSocket = new ServerSocket( port: 2525);//创建serverSocket
        while (true) {
            Socket connectionSocket = welcomeSocket.accept();//等待请求, 创建connSocket
            new Thread(new ClientHandler(connectionSocket)).start(); // 创建新线程处理客户端连接
```

```
"D:\Program Files (x86)\jdk\bin\java.exe" "-javaagent:D:\Program Files (x86)\IntelliJ IDEA 2023.3.6\lib\idea_rt.jar=57200:D:\Program F
```

3. 运行客户端，准备连接服务器



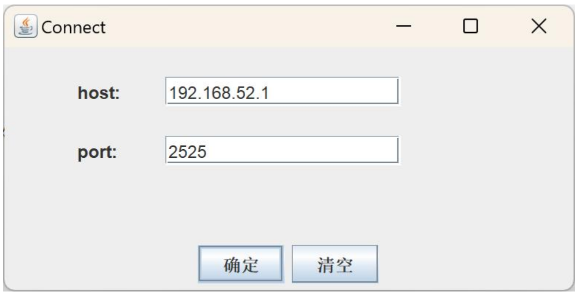
```java
import java.io.*;
import java.net.*;
class TCPClient {
    public static void main(String argv[]) throws Exception
    {
        Connect con = new Connect();//地址界面
        while (!con.complete) {
            try {
                Thread.sleep( millis: 100); // 暂停100毫秒
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        String host=con.getHost();
```

```
"D:\Program Files (x86)\jdk\bin\java.exe" "-jav
3.6\lib\idea_rt.jar=57509:D:\Program Files (x86)\I
```

host:

port: 2525

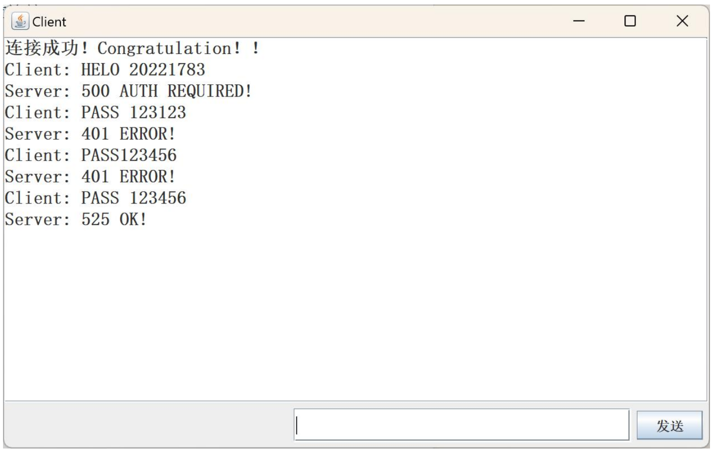
4. 输入获取到的服务端 IP 地址，点击确定或回车键建立连接



host: 192.168.52.1

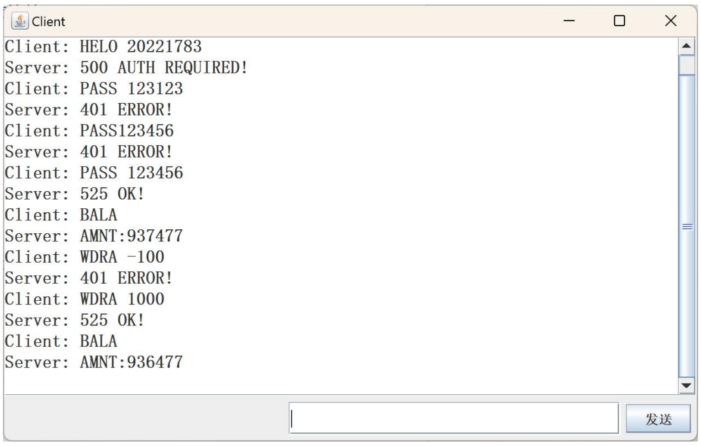
port: 2525

5. 输入正确的账号密码登录账户



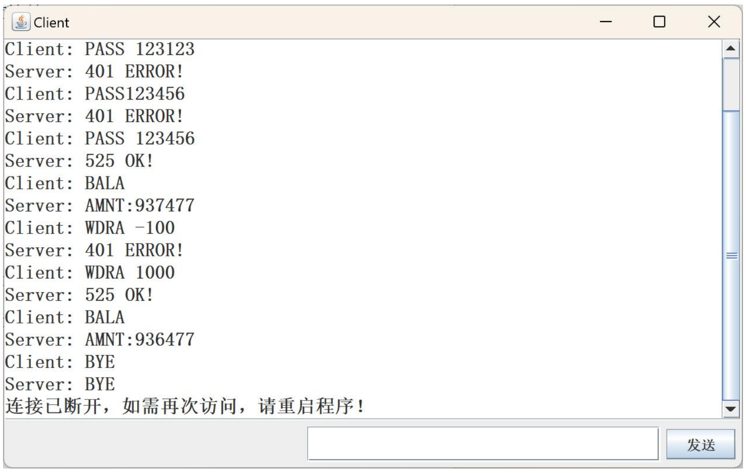
```
连接成功！Congratulation！！
Client: HELO 20221783
Server: 500 AUTH REQUIRED!
Client: PASS 123123
Server: 401 ERROR!
Client: PASS123456
Server: 401 ERROR!
Client: PASS 123456
Server: 525 OK!
```

6. 使用 BALA 查询余额并通过 WDRA 进行取钱操作



```
Client: HELO 20221783
Server: 500 AUTH REQUIRED!
Client: PASS 123123
Server: 401 ERROR!
Client: PASS123456
Server: 401 ERROR!
Client: PASS 123456
Server: 525 OK!
Client: BALA
Server: AMNT:937477
Client: WDRA -100
Server: 401 ERROR!
Client: WDRA 1000
Server: 525 OK!
Client: BALA
Server: AMNT:936477
```

7. 操作完毕使用 BYE 退出客户端

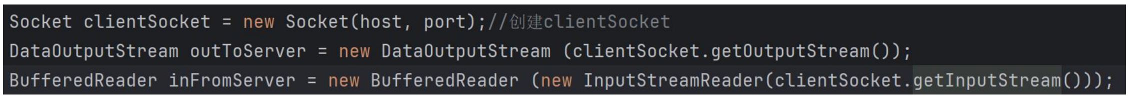


```
Client: PASS 123123
Server: 401 ERROR!
Client: PASS123456
Server: 401 ERROR!
Client: PASS 123456
Server: 525 OK!
Client: BALA
Server: AMNT:937477
Client: WDRA -100
Server: 401 ERROR!
Client: WDRA 1000
Server: 525 OK!
Client: BALA
Server: AMNT:936477
Client: BYE
Server: BYE
连接已断开，如需再次访问，请重启程序！
```

五、过程原始记录(数据、图表、计算等)

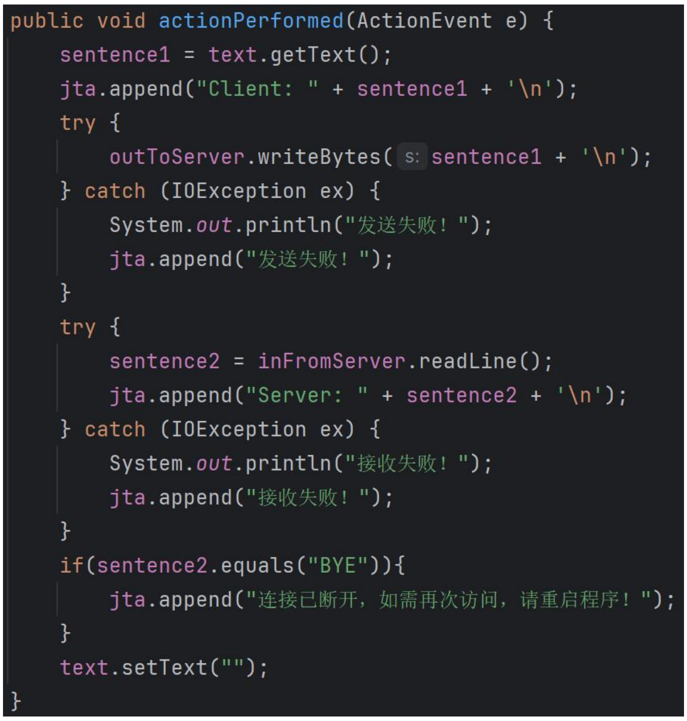
1. 关键代码（全部代码打包在文件夹）

(1) 客户端基本设置，包括输入输出流



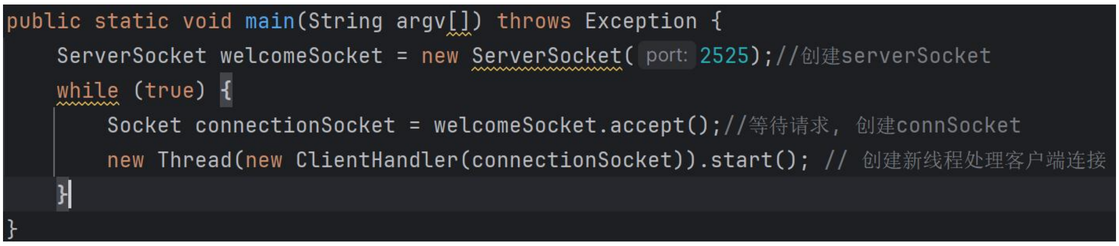
```
Socket clientSocket = new Socket(host, port);//创建clientSocket
DataOutputStream outToServer = new DataOutputStream (clientSocket.getOutputStream());
BufferedReader inFromServer = new BufferedReader (new InputStreamReader(clientSocket.getInputStream()));
```

(2) 客户端发送代码



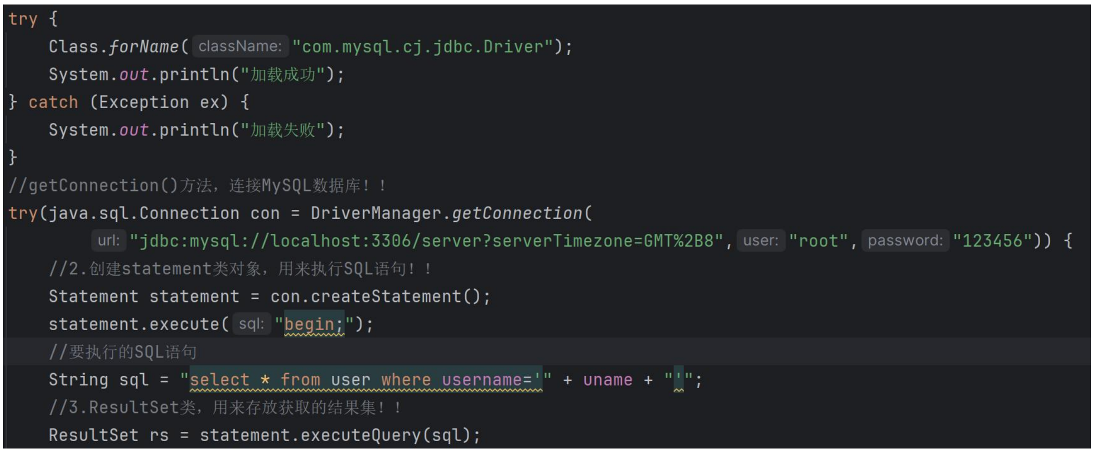
```
public void actionPerformed(ActionEvent e) {
    sentence1 = text.getText();
    jta.append("Client: " + sentence1 + '\n');
    try {
        outToServer.writeBytes( s: sentence1 + '\n');
    } catch (IOException ex) {
        System.out.println("发送失败！");
        jta.append("发送失败！");
    }
    try {
        sentence2 = inFromServer.readLine();
        jta.append("Server: " + sentence2 + '\n');
    } catch (IOException ex) {
        System.out.println("接收失败！");
        jta.append("接收失败！");
    }
    if(sentence2.equals("BYE")){
        jta.append("连接已断开，如需再次访问，请重启程序！");
    }
    text.setText("");
}
```

(3) 服务端基本设置，包括线程处理



```
public static void main(String argv[]) throws Exception {
    ServerSocket welcomeSocket = new ServerSocket( port: 2525);//创建serverSocket
    while (true) {
        Socket connectionSocket = welcomeSocket.accept();//等待请求，创建connSocket
        new Thread(new ClientHandler(connectionSocket)).start(); // 创建新线程处理客户端连接
    }
}
```

(4) 数据库部分操作



```
try {
    Class.forName( className: "com.mysql.cj.jdbc.Driver");
    System.out.println("加载成功");
} catch (Exception ex) {
    System.out.println("加载失败");
}
//getConnection()方法，连接MySQL数据库！！
try(java.sql.Connection con = DriverManager.getConnection(
        url: "jdbc:mysql://localhost:3306/server?serverTimezone=GMT%2B8", user: "root", password: "123456")) {
    //2.创建statement类对象，用来执行SQL语句！！
    Statement statement = con.createStatement();
    statement.execute( sql: "begin;");
    //要执行的SQL语句
    String sql = "select * from user where username='" + uname + "'";
    //3.ResultSet类，用来存放获取的结果集！！
    ResultSet rs = statement.executeQuery(sql);
```

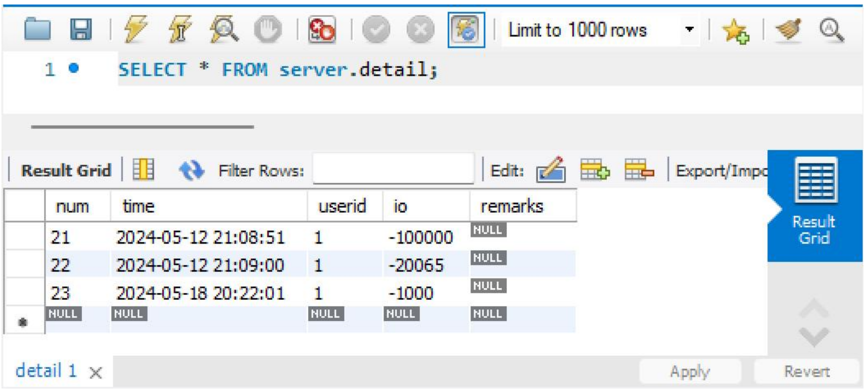
(5) 服务端中央处理代码

```java
switch (command) {
    case "HELO":
        if (parts.length>1){
            this.uname=parts[1];
            con=new Connection(uname);
            response = "500 AUTH REQUIRED!";
        }else {
            response="401 ERROR!";
        }
        break;
    case "PASS":
        int uid=con.validate();
        if(parts.length>1&&parts[1].equals(con.getUpass())){
            user=new User(uid,this.uname,con.getSum());
            isLogin=true;
            response="525 OK!";
        }else {
            response="401 ERROR!";
        }
        break;
    case "BALA":
        if(isLogin) {
            response = "AMNT:" + user.getSum();
        }
        break;
    case "WDRA":
        if (isLogin&&parts.length>1){
            int wdr=0;
            try {
                wdr = Integer.parseInt(parts[1]);
            } catch (NumberFormatException ex) {
                System.out.println("输入的不是有效的整数：" + parts[1]);
            }
            if (wdr>0&&wdr<user.getSum()){
                Withdraw add=new Withdraw(user.getUserId(), -wdr);
                if (add.complete) {
                    response = "525 OK!";
                }else response="401 ERROR!";
            }else response="401 ERROR!";
        }
        break;
    case "BYE":
        response = "BYE";
        outToClient.writeBytes(response + "\n"); // 发送 "BYE" 响应
        connectionSocket.close(); // 关闭连接
        break;
    default:
        response = "401 ERROR!";
        break;
}
```

六、结果及分析

客户端成功访问服务器端并进行了登录、查询、取钱等操作

服务端成功接收访问，识别相关 Messages 并在数据库留下取款记录



```
SELECT * FROM server.detail;
```

num	time	userid	io	remarks
21	2024-05-12 21:08:51	1	-100000	NULL
22	2024-05-12 21:09:00	1	-20065	NULL
23	2024-05-18 20:22:01	1	-1000	NULL
NULL	NULL	NULL	NULL	NULL

七、上机实验总结

在相同局域网下，用户与服务器双方可通过套接字进行交流，并完成一系列操作。