计算机网络课程作业二

ATM模拟编程作业报告

# 目的

通过作业二使同学们掌握所学TCP套接字编程方法，理解TCP套接字在服务器和客户端之间的应用层通信的流程和形式，巩固计算机网络的理论和知识。此外，通过作业的开展使同学们提高计算机网络程序设计和开发能力，熟悉所用编程语言的开发环境及调试过程，熟悉所用语言中的数据类型、数据结构、语法结构、运算方法等。基于作业一的内容进行程序设计开发，同学们也可以体会计算机网络通信协议(应用层)的制定、标准化等工作对实际应用的价值和意义。

# 作业内容

要求两位同学组队(参考石墨文档1-作业2分组)，基于模拟的应用层通信协议标准(参考石墨文档2-“RFC20212021”)完成ATM机客户端和服务器端的程序设计和开发。程序语言不限。

客户端具备GUI可以模拟ATM机的账户登录和存款取款等功能，同时应该包括错误信息的提示。服务器端应该需要通过后台数据库维护账户信息、具备记录日志的功能。

各组同学独立完成，不能抄袭。作业结束时，每组需提交作业报告，其中包含程序设计思路和软件源代码。

里程碑1：第2次实验课，完成组内程序设计开发和测试，尝试连接其他小组的服务器端。

里程碑2：第3次实验课，分析解决里程碑1中的问题，完成程序调试，通过测试脚本完成对其他所有小组服务器端的连接，记录结果。

里程碑3：第4次实验课，继续调试程序，最终提交文档、代码。

# 运行环境

系统环境：Windows、Mac

编译环境：IDEA、JDK 11

程序语言：Java

# 程序设计

## 4.1设计思路

//在本节介绍程序的流程、客户端和服务器端的交互流程图等

本程序建立了loginFrame和funcFrame两个窗口，以及客服端和服务端，

通过建立Socket来进行客户端与服务端的通信，然后再通过Swing的GUI界面来与用户交互，通过GUI界面获取用户输入的信息，然后传入客户端，客户端收到后，再发送给服务端，服务端对收到的数据进行处理，响应状态码及处理后的数据给客户端，客户端再将收到的信息显示给用户。



图4.1.1客户端与服务器端的通信

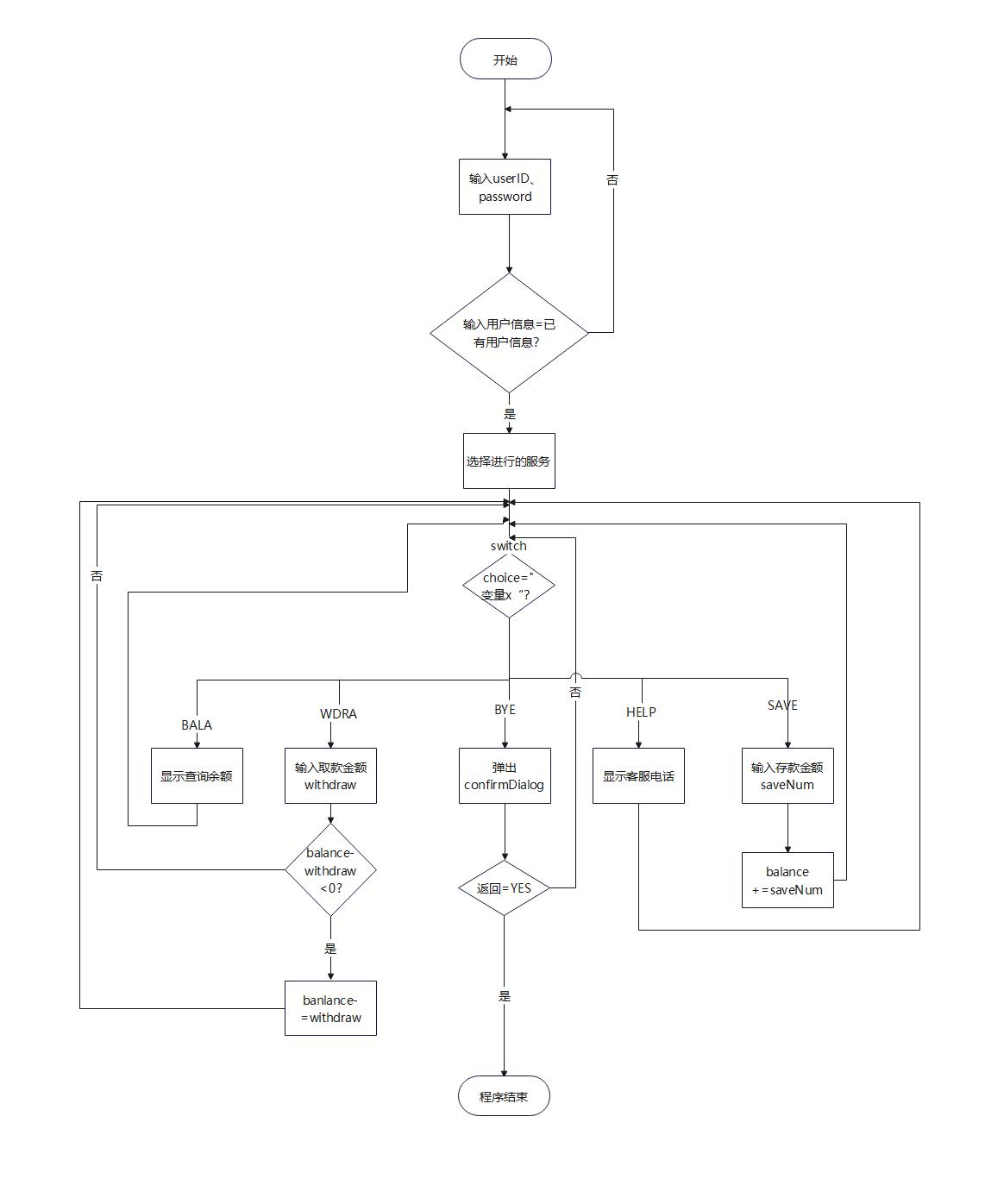
tu

图4.1.2 程序流程图

## 4.2服务器端

|  |
| --- |
| Server |
| package com.yt.atm;  import org.ietf.jgss.GSSContext;  import java.io.\*; import java.net.ServerSocket; import java.net.Socket; import java.util.Properties;  public class Server {   public static String *balance* = "50000";   public static void main(String[] args) throws IOException {  ServerSocket ss = new ServerSocket(2525);   *//把本地文件中正确的用户名和密码获取到* Properties prop = new Properties();  FileInputStream fis = new FileInputStream("servicedir\\userinfo.txt");  prop.load(fis);  fis.close();   *//只要来了一个客户端，就开一条线程处理* while (true) {  Socket socket = ss.accept();  System.*out*.println("有客户端来连接");  new Thread(new MyRunnable(socket, prop)).start();  }   }   }  class MyRunnable implements Runnable {  Socket socket;  Properties prop;   public MyRunnable(Socket socket, Properties prop) {  this.prop = prop;  this.socket = socket;  }   @Override  public void run() {  try {  BufferedReader br = new BufferedReader(new InputStreamReader(socket.getInputStream()));  *//用while包裹，保证能输错后还能继续输入* while (true) {  *//第一次读：读取用户输入的登录信息* String userInfo = br.readLine();  *//调用登录函数判断信息是否正确* judgeLoginInfo(userInfo);  }   } catch (IOException e) {  throw new RuntimeException(e);  }  }   public void judgeLoginInfo(String userInfo) throws IOException {  *//获取输出流* BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(socket.getOutputStream()));  *//拆分用户输入的信息* String[] userInfoArr = userInfo.split("&");  String usernameInput = userInfoArr[0];  String passwordInput = userInfoArr[1];  System.*out*.println("用户输入的用户名为：" + usernameInput);  System.*out*.println("用户输入的密码为：" + passwordInput);  if (prop.containsKey(usernameInput)) {  *//第一次回写：登录成功响应码码* bw.write("525");  bw.newLine();  bw.flush();  *//登录成功，进入功能选择界面* while (true) {  BufferedReader br = new BufferedReader(new InputStreamReader(socket.getInputStream()));  *//接受客户端转来的信息，并判断应该执行的操作* String choice = br.readLine();  switch (choice) {  case "BALA":  *//收到客户端传来的“BALA"信息，将余额传回客户端* bw.write(Server.*balance*);  bw.newLine();  bw.flush();  break;  case "WDRA":  *//收到客户端传来的”WDRA“信息，执行取款操作  //再接受客户端传来的取款金额* String withdrawNum = br.readLine();  *//将计算后的余额传回客户端* if ((Integer.*parseInt*(Server.*balance*) - Integer.*parseInt*(withdrawNum)) < 0) {  bw.write("401");  bw.newLine();  bw.flush();  bw.write(Server.*balance*);  bw.newLine();  bw.flush();  break;  } else {  bw.write("525");  bw.newLine();  bw.flush();  Server.*balance* = String.*valueOf*(Integer.*parseInt*(Server.*balance*) - Integer.*parseInt*(withdrawNum));  bw.write(Server.*balance*);  bw.newLine();  bw.flush();  break;  }    case "SAVE":  *//收到客户端传来的”SAVA“信息，执行取款操作  //再接受客户端传来的存款金额* String saveNum = br.readLine();  *//将计算后的余额传回客户端* Server.*balance* = String.*valueOf*(Integer.*parseInt*(Server.*balance*) + Integer.*parseInt*(saveNum));  bw.write(Server.*balance*);  bw.newLine();  bw.flush();  break;  case "BYE":  bw.write("BYE");  bw.newLine();  bw.flush();  break;   }  }  } else {  *//第一次回写：登录失败响应码* bw.write("401");  bw.newLine();  bw.flush();  }   } } |

## 4.3客户端

|  |
| --- |
| Client |
| package com.yt.atm;  import java.io.\*; import java.net.Socket;  public class Client {   public static void main(String[] args) throws IOException {   new MainFrame().init();   System.*out*.println("服务器已经连接成功");  }   public static String login(String userId, String password, Socket socket) throws IOException {  *//获取输出流* BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(socket.getOutputStream()));   *//拼接收到的用户名和密码* StringBuilder sb = new StringBuilder();  sb.append(userId).append("&").append(password);  *//第一次写：往服务器写用户名和密码* bw.write(sb.toString());  bw.newLine();  bw.flush();   *//第一次接受数据  //获取输入流* BufferedReader br = new BufferedReader(new InputStreamReader(socket.getInputStream()));  *//接收从服务端返回的值* return br.readLine();  } } |

|  |
| --- |
| FuncFrame |
| package com.yt.atm; import javax.swing.\*; import java.awt.\*; import java.io.\*; import java.net.Socket;  public class FuncFrame {   JFrame funcFrame=new JFrame("ATM");  public void init(Socket socket,Client client) throws IOException {  funcFrame.setBounds(800, 400, 440, 400);  funcFrame.setResizable(false);  JPanel root = new JPanel();  funcFrame.setContentPane(root);  root.setLayout(null);   *//获取窗口大小* int width = funcFrame.getWidth();  int height = funcFrame.getHeight();   *//为登录窗口添加背景图片* ImageIcon background = new ImageIcon("images\\1.jpg");  *// 创建一个JLabel对象，显示背景图片* JLabel label = new JLabel(background);  *// 设置标签的大小和位置为图片的大小和位置* label.setBounds(0, 0, 440, 600);  *// 将内容窗格转化为JPanel，否则不能用方法setOpaque()来使内容窗格透明* JPanel imagePanel = (JPanel) funcFrame.getContentPane();  imagePanel.setOpaque(false);  *// 将背景图片添加到分层窗格的最底层作为背景* funcFrame.getLayeredPane().add(label, new Integer(Integer.*MIN\_VALUE*));   *//添加提示标签* JLabel tipsLabel=new JLabel("请选择您要进行的业务");  root.add(tipsLabel);  tipsLabel.setBounds((int) (width \* 0.32), (int) (height \* 0.04), 200, 40);  tipsLabel.setFont(new Font("微软雅黑", Font.*PLAIN*, 15));   *//获取输出流* BufferedWriter bw= new BufferedWriter(new OutputStreamWriter(socket.getOutputStream()));   *//获取输入流* BufferedReader br = new BufferedReader(new InputStreamReader(socket.getInputStream()));    *//添加查询按钮* JButton balanceButton=new JButton("查询余额");  *//取消按钮焦点* balanceButton.setFocusPainted(false);  root.add(balanceButton);  balanceButton.setBounds((int) (width \* 0.27), (int) (height \* 0.18), 200, 40);  balanceButton.addActionListener((e)->{  try {  *//写入选择的功能，通过规定的字符传给客服端* bw.write("BALA");  bw.newLine();  bw.flush();   *//接收服务端传来的信息,并显示* JOptionPane.*showMessageDialog*(funcFrame,"您当前的余额为："+br.readLine());  } catch (IOException ex) {  throw new RuntimeException(ex);  }   });   *//添加取款按钮* JButton withdrawButton=new JButton("取款");  *//取消按钮焦点* withdrawButton.setFocusPainted(false);  root.add(withdrawButton);  withdrawButton.setBounds((int) (width \* 0.27), (int) (height \* 0.31), 200, 40);  withdrawButton.addActionListener((e)->{  String withdrawNum = JOptionPane.*showInputDialog*(funcFrame, "请输入您的取款金额：", "输入", JOptionPane.*PLAIN\_MESSAGE*);  try {  *//写入选择的功能，通过规定的字符传给客服端* bw.write("WDRA");  bw.newLine();  bw.flush();   *//再将取款金额传进去* bw.write(withdrawNum);  bw.newLine();  bw.flush();  *//接收服务端传来的取款后的余额信息,并显示* if("401".equals(br.readLine())){  JOptionPane.*showMessageDialog*(funcFrame,"取款失败，当前余额不足！\n 您当前的余额为："+br.readLine());  }else {  JOptionPane.*showMessageDialog*(funcFrame,"取款成功！\n 您当前的余额为："+br.readLine());   }  } catch (IOException ex) {  throw new RuntimeException(ex);  }   });   *//添加存款按钮* JButton saveButton=new JButton("存款");  *//取消按钮焦点* saveButton.setFocusPainted(false);  root.add(saveButton);  saveButton.setBounds((int) (width \* 0.27), (int) (height \* 0.44), 200, 40);  saveButton.addActionListener((e)->{  String saveNum = JOptionPane.*showInputDialog*(funcFrame, "请输入您的存款金额：", "输入", JOptionPane.*PLAIN\_MESSAGE*);  try {  *//写入选择的功能，通过规定的字符传给客服端* bw.write("SAVE");  bw.newLine();  bw.flush();   *//再将取款金额传进去* bw.write(saveNum);  bw.newLine();  bw.flush();   *//接收服务端传来的取款后的余额信息,并显示* JOptionPane.*showMessageDialog*(funcFrame,"存款成功！\n 您当前的余额为："+br.readLine());  } catch (IOException ex) {  throw new RuntimeException(ex);  }   });   *//添加帮助按钮* JButton helpButton=new JButton("帮助");  *//取消按钮焦点* helpButton.setFocusPainted(false);  root.add(helpButton);  helpButton.setBounds((int) (width \* 0.27), (int) (height \* 0.57), 200, 40);  helpButton.addActionListener((e)->{  JOptionPane.*showMessageDialog*(funcFrame, "请拨打在线客服：4006 700 700");  });   *//添加退出按钮* JButton exitButton=new JButton("退出");  *//取消按钮焦点* exitButton.setFocusPainted(false);  root.add(exitButton);  exitButton.setBounds((int) (width \* 0.27), (int) (height \* 0.70), 200, 40);  exitButton.addActionListener((e)->{  int result = JOptionPane.*showConfirmDialog*(funcFrame, "您确认要退出程序吗？", "提示", JOptionPane.*YES\_NO\_CANCEL\_OPTION*);  if (result == JOptionPane.*YES\_OPTION*) {  try {  *//用户点击确认后，发送”BYE“给服务端* bw.write("BYE");  bw.newLine();  bw.flush();  } catch (IOException ex) {  throw new RuntimeException(ex);  }  *//收到服务端的"BYE指令后退出程序* try {  if("BYE".equals(br.readLine())){  funcFrame.dispose();  }  } catch (IOException ex) {  throw new RuntimeException(ex);  }   }  });   funcFrame.setVisible(true);  }  } |

|  |
| --- |
| MainFrame |
| package com.yt.atm;  import javax.swing.\*; import java.awt.\*; import java.io.BufferedWriter; import java.io.IOException; import java.io.OutputStreamWriter; import java.net.Socket;  import static com.yt.atm.Client.*login*;  public class MainFrame {  *//1.创建Socket对象并连接服务端* Socket socket = new Socket("127.0.0.1", 2525);  *//创建客户端对象* Client client = new Client();    *//创建一个登录窗口  //利用多态创建对象* JFrame loginFrame = new JFrame("ATM");    *//为文本框前添加标签控件* JLabel userIdLabel = new JLabel("用户名");  JLabel passwordLabel = new JLabel("密码");  JLabel loginLabel = new JLabel("ATM");  *//添加文本框* JTextField userIdText = new JTextField(20);  JTextField passwordText = new JTextField(20);  *//添加登录按钮* JButton loginButton = new JButton("登录");  *//添加复选框控件* JCheckBox agreeCheckBox = new JCheckBox("同意用户协议");   public MainFrame() throws IOException {  }    *//组装窗口* public void init() {  loginFrame.setBounds(800, 400, 440, 330);  loginFrame.setResizable(false);   JPanel root = new JPanel();  loginFrame.setContentPane(root);  root.setLayout(null);   *//为登录窗口添加背景图片* ImageIcon background = new ImageIcon("images\\1.jpg");  *// 创建一个JLabel对象，显示背景图片* JLabel label = new JLabel(background);  *// 设置标签的大小和位置为图片的大小和位置* label.setBounds(0, 0, 440, 330);  *// 将内容窗格转化为JPanel，否则不能用方法setOpaque()来使内容窗格透明* JPanel imagePanel = (JPanel) loginFrame.getContentPane();  imagePanel.setOpaque(false);  *// 将背景图片添加到分层窗格的最底层作为背景* loginFrame.getLayeredPane().add(label, new Integer(Integer.*MIN\_VALUE*));    *//为文本框前添加标签控件* root.add(userIdLabel);  root.add(passwordLabel);  root.add(loginLabel);  userIdLabel.setFont(new Font("微软雅黑", Font.*PLAIN*, 16));  passwordLabel.setFont(new Font("微软雅黑", Font.*PLAIN*, 16));  loginLabel.setFont(new Font("微软雅黑", Font.*PLAIN*, 25));    *//添加文本框* root.add(userIdText);  root.add(passwordText);   *//添加登录按钮* root.add(loginButton);  *//取消按钮焦点* loginButton.setFocusPainted(false);  *//为按钮添加监听器* loginButton.addActionListener((e) -> {  *//判断输入的密码是否正确，如果正确的话，就跳转到新窗口* String judgePassword = judgePassword();  if ("525".equals(judgePassword)) {  JOptionPane.*showMessageDialog*(loginFrame, "登录成功");  try {  new FuncFrame().init(socket, client);  } catch (IOException ex) {  throw new RuntimeException(ex);  }  loginFrame.dispose();  } else if ("401".equals(judgePassword)) {  JOptionPane.*showMessageDialog*(loginFrame, "密码或用户名输入错误，请重试！");  }  });   *//添加复选框控件,并与登录按钮绑定* root.add(agreeCheckBox);  agreeCheckBox.setSelected(false);  loginButton.setEnabled(false);  agreeCheckBox.addActionListener((e) -> {  if (agreeCheckBox.isSelected()) {  loginButton.setEnabled(true);  } else {  loginButton.setEnabled(false);  }  });  agreeCheckBox.setOpaque(false);   int width = loginFrame.getWidth();  int height = loginFrame.getHeight();  loginLabel.setLocation((int) (width \* 0.43), (int) (height \* 0.18));  loginLabel.setSize(loginLabel.getPreferredSize());  userIdLabel.setLocation((int) (width \* 0.19), (int) (height \* 0.32));  userIdLabel.setSize(userIdLabel.getPreferredSize());  passwordLabel.setLocation((int) (width \* 0.21), (int) (height \* 0.42));  passwordLabel.setSize(passwordLabel.getPreferredSize());  userIdText.setLocation((int) (width \* 0.31), (int) (height \* 0.33));  userIdText.setSize(userIdText.getPreferredSize());  passwordText.setLocation((int) (width \* 0.31), (int) (height \* 0.43));  passwordText.setSize(passwordText.getPreferredSize());  agreeCheckBox.setLocation((int) (width \* 0.41), (int) (height \* 0.51));  agreeCheckBox.setSize(agreeCheckBox.getPreferredSize());  loginButton.setBounds((int) (width \* 0.22), (int) (height \* 0.61), 250, 25);   loginFrame.setVisible(true);  }   private String judgePassword() {  *//获取用户输入的值* String inputUserId = userIdText.getText();  String inputPassword = passwordText.getText();  try {  *//调用客户端login函数,并返回收到的状态码* return *login*(inputUserId, inputPassword, socket);  } catch (IOException e) {  throw new RuntimeException(e);  }  }  } |

# 测试运行结果

## 组内测试

第一次实验课进行的设定服务器IP、端口，利用ping进行连接测试：

图5.1.1设定IP、端口号

在测试连接的过程中，一开始始终无法连接成功，最终通过关闭两台电脑的防火墙才得以解决；

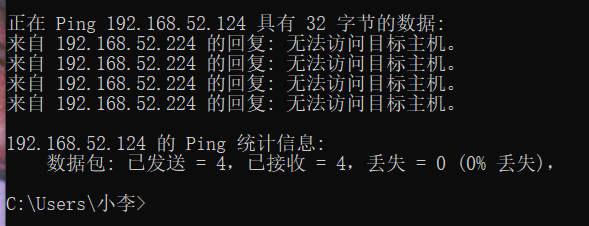


图5.1.2 第一次测试连接

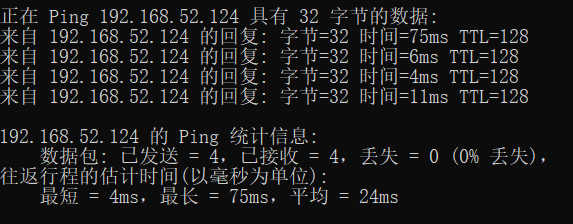
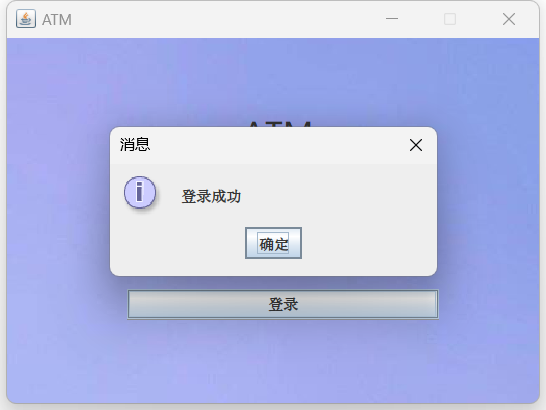
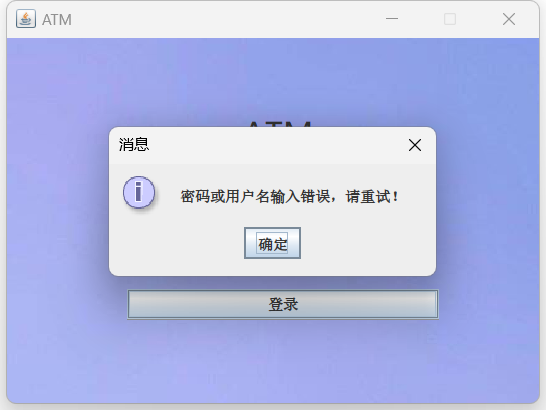


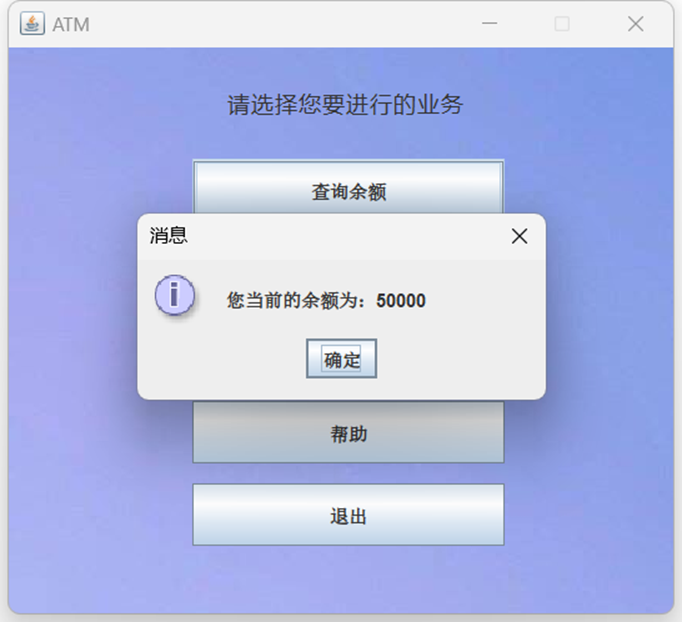
图5.1.3 成功连接服务器端

组内测试基本功能:

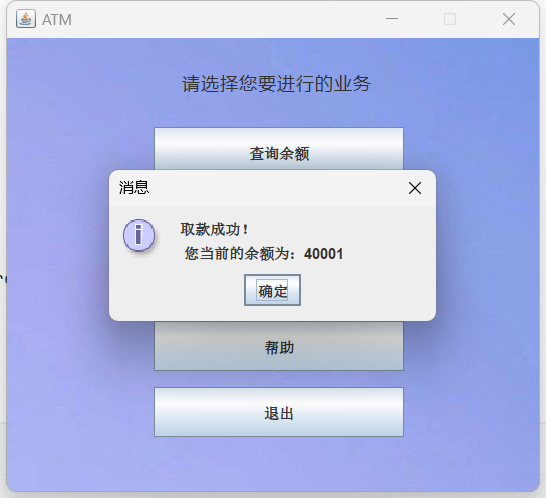
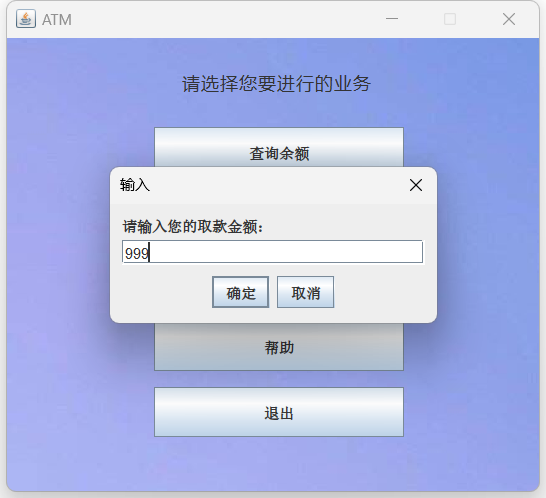
1. 登录
   1. 登录成功
   2. 登录失败

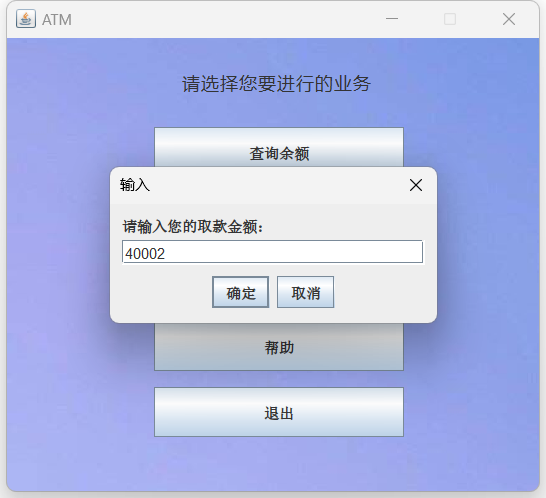
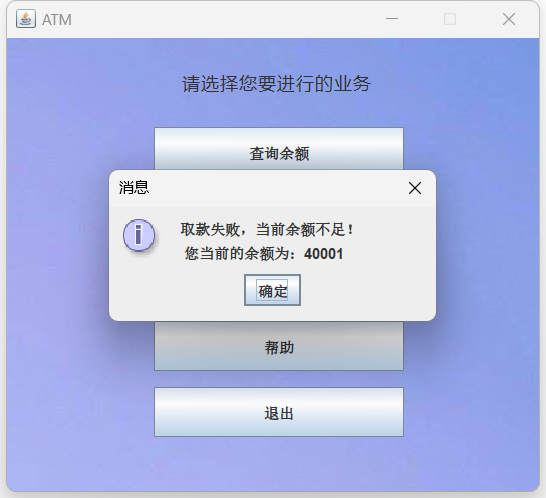


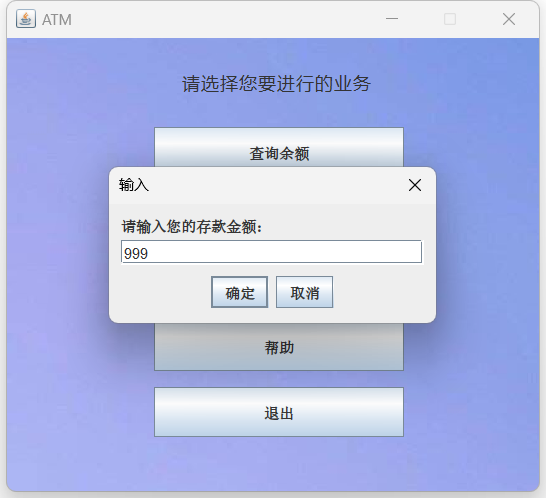
1. 查询余额

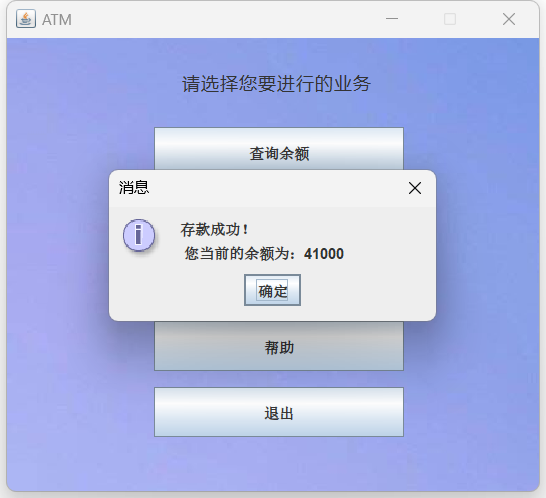


1. 取款
   1. 取款成功

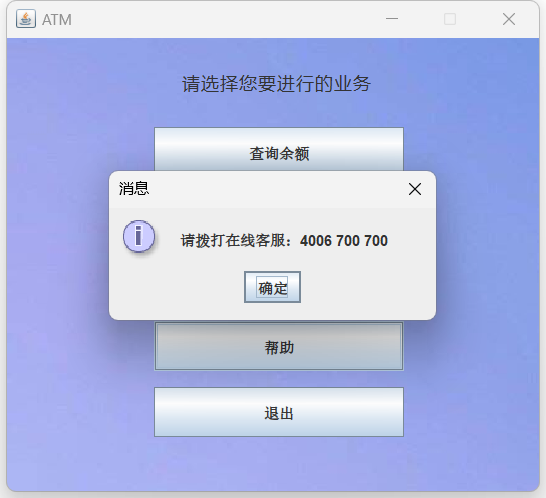


* 1. 取款失败

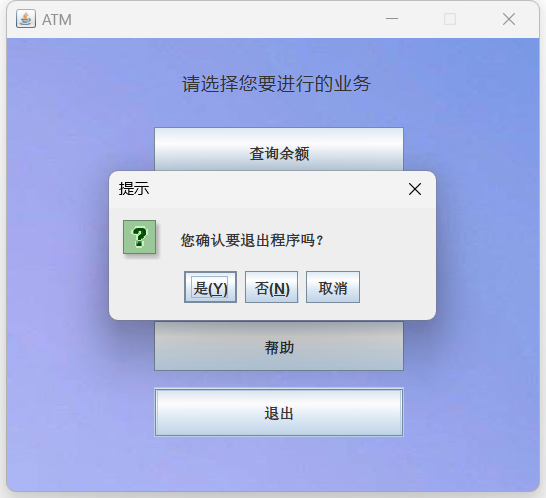
1. 存款



1. 帮助



1. 退出



## 组间测试

与别的小组相互测试的过程中，发现无法打开多个客户端程序，最后通过多线程处理进行解决，代码如下：

*//只要来了一个客户端，就开一条线程处理*while (true) {  
 Socket socket = ss.accept();  
 System.*out*.println("有客户端来连接");  
 new Thread(new MyRunnable(socket, prop)).start();  
}



图5.2.1 服务端显示别的小组的连接请求

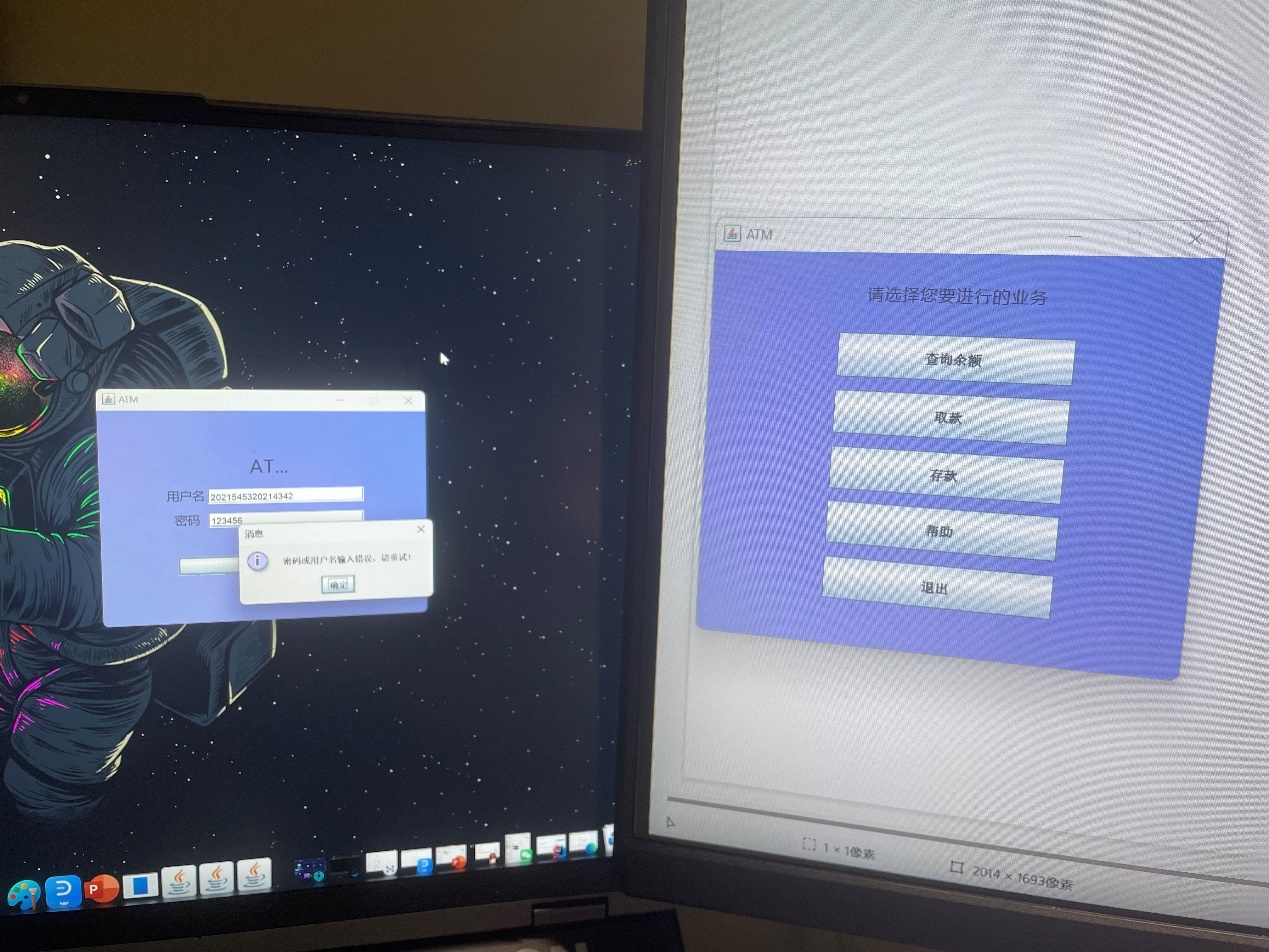


图5.2.2 组间测试多开客户端

# 总结

这个小程序用到了TCP socket网络编程的思想和Java Swing。在开发这个小程序的过程中，我们遇到了一些问题，例如：

* 如何建立TCP连接
* 如何发送和接收数据
* 如何处理多个客户端的请求
* 如何在Java Swing中创建GUI
* 如何读取文件里的内容

我们最后通过这些方法来进行解决：

* 使用Java Socket API建立TCP连接
* 使用Java IO类发送和接收数据
* 使用多线程处理多个客户端的请求
* 使用Java Swing创建GUI
* 使用FileInputStream获取文件内容

在开发这个小程序的过程中，我们对TCP socket网络编程、Java Swing、多线程编程有了更深入的了解。每做一个程序，对我们都是很大的提升，在做这个程序的初期，我们小组缺少的知识很多，我们先是通过学习网络编程的相关知识，对整个题目的有了初步认识，然后利用java实现了无界面的atm程序，在补充java Swing知识后，才将程序的GUI界面做出来，在这个过程中，我真真确确的感受到了团队的重要性，身为团队中的一员，一定要做好团队个成员之间的沟通，要彼此协作。充分发挥团队的力量，对于软件项目开发至关重要。同时，这次也暴露了自己在软件开发过程中相关理论、技术方面的不足，在以后的学习过程中要重视理论的学习，开发经验的积累，要多练习，多去尝试开发一些项目，查漏补缺，积累开发经验，并且要不断的去学习一些新技术。