# 订餐系统

1. **问题描述**

某企业为了方便员工用餐，为餐厅开发了一个**订餐系统（COS：Cafeteria Ordering System）**，企业员工可通过企业内联网使用该系统。

企业的任何员工都可以查看**菜单**和**今日特价**。

系统的顾客是注册到系统的员工，可以**订餐**（如果未**登录**，需先登录）、注册**工资支付**、预约规律的订餐，在特殊情况下可以覆盖预订。

餐厅员工是特殊顾客，可以进行备餐、生成付费请求和**请求送餐**，其中对于注册工资支付的顾客生成付费请求并发送给工资系统。

菜单管理员是餐厅特定员工，可以**管理菜单**。

送餐员可以打印送餐说明，记录送餐信息（如送餐时间）以及记录收费（对于没有注册工资支付的顾客，由送餐员收取现金后记录）**。**

顾客订餐过程如下：

1. 顾客请求查看菜单；

2. 系统显示菜单和今日特价；

3. 顾客选菜；

4. 系统显示订单和价格；

5. 顾客确认订单；

6. 系统显示可送餐时间；

7. 顾客指定送餐时间、地点和支付方式；

8. 系统确认接受订单，然后发送Email给顾客以确认订餐，同时发送相关订餐信息通知给餐厅员工。

二. **问题分析与模型设计**

1.系统的用户

系统的用户有企业员工、餐厅员工和送餐员，他们必须以相应的身份登录系统

1. 企业员工：系统只知道用户的注册用户名，用户的真实姓名不重要。
2. 餐厅员工：假设餐厅由若干个承包商户组成，每家商户中**有且仅有一名员工**负责订餐系统的操作
3. 送餐员：假设总是在线并无条件地接收来自商户的派送请求

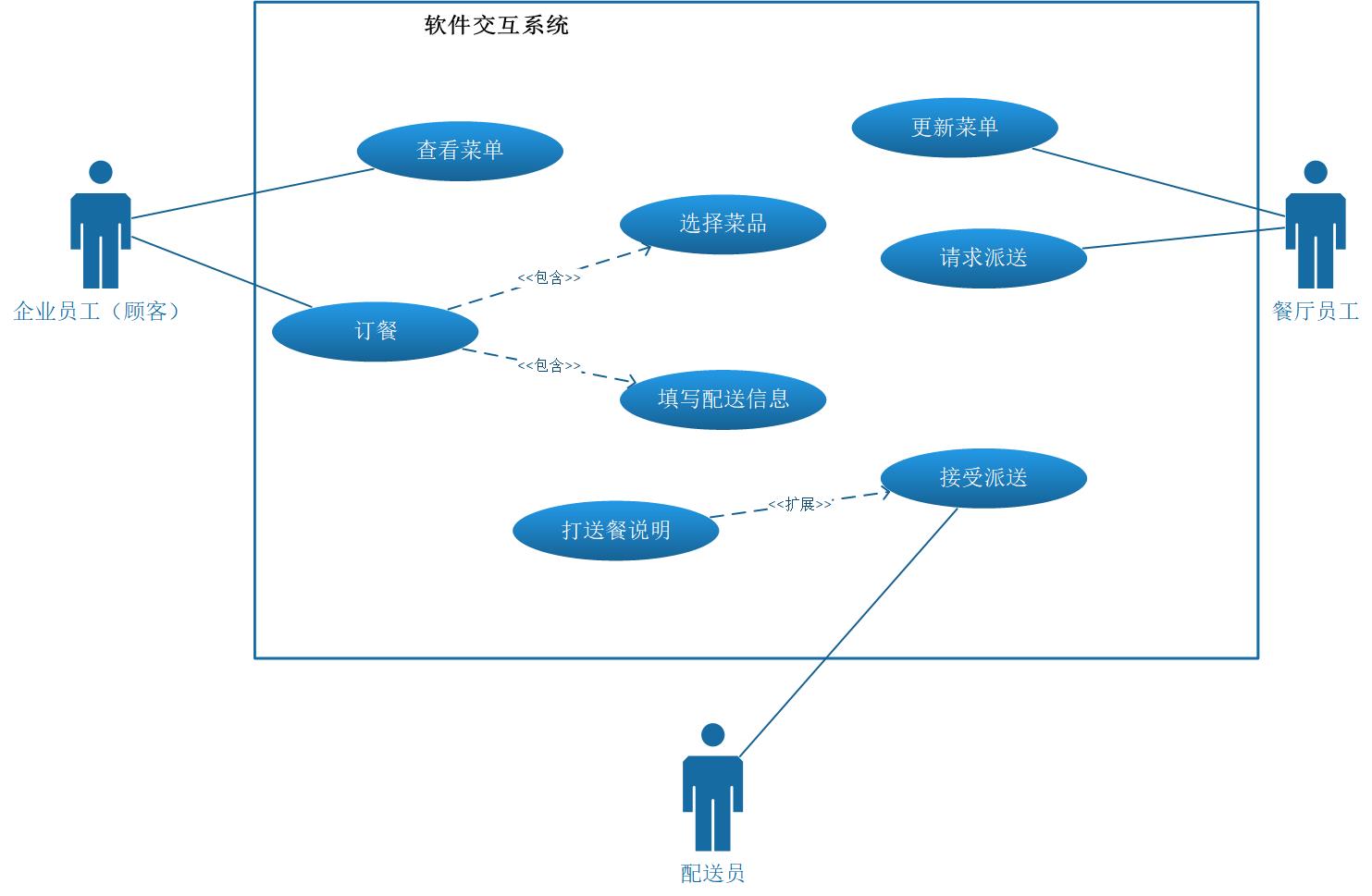
2. 系统的设计

a. 服务于企业员工（顾客）的：顾客可以选择心仪的商户，选定商户后，**普通菜单**和**特价菜单**中加载出菜品的信息，顾客可以选中某一菜品，并输入购买的数量，点击按钮便可以加入到购物车中，**值得注意的是，此处假设顾客每笔订单只包含一家商户的菜品。**选购完毕后，顾客填写配送地址、支付方式和备注（可选），点击按钮后便可提交订单给商户。

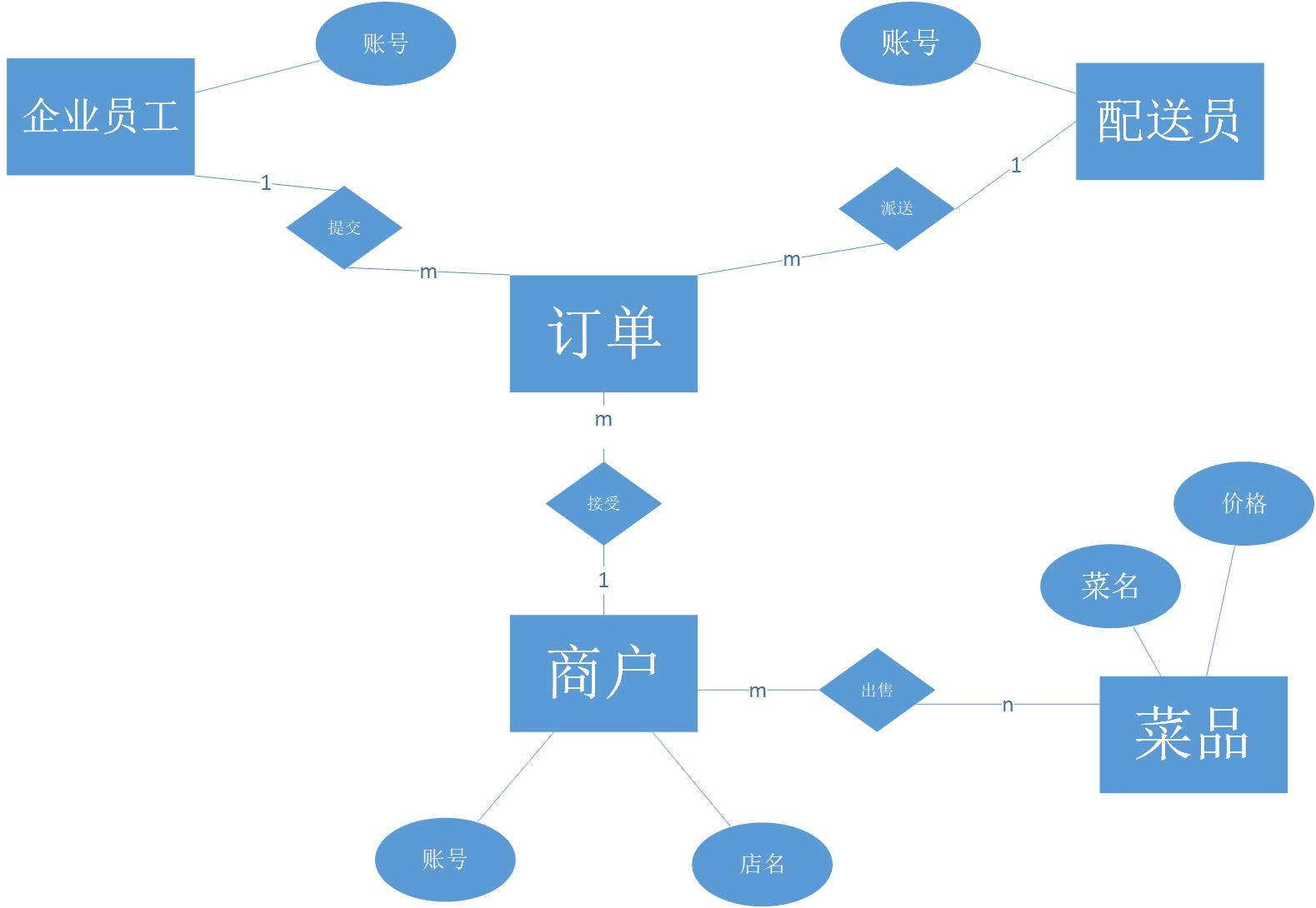
b. 服务于商户的：商户需要实时接收订单信息。员工可以根据经营的需要添加特色菜、增加菜品和删除菜品等常规菜单操作。备餐完毕后，员工将订单指派给派送员。

c. 服务于派送员的：派送员需要实时接收订单信息，取餐并完成配送后，将订单设置为成功送达状态。

3. 系统的模型（UML用例图）

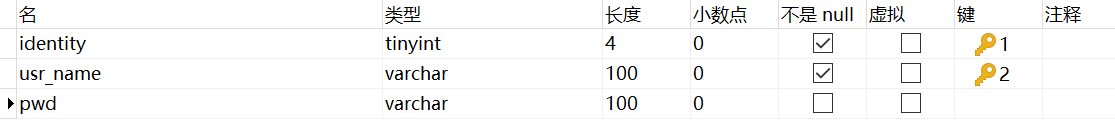


3. E-R图设计



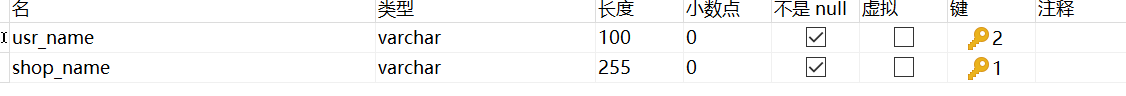
4.**关系表的设计**

a. login表



相同身份下，用户名不可重复，即**身份与用户名唯一确定一个个体**，故将（身份，用户名）设置为主键。

b. Usr\_shop表



usr\_name的身份必为餐厅员工。一个账号只服务于一家店面，一家店面也只需要一个账号用于运营，故usr\_name和shop\_name都是unique的，且将二者共同设置为主键。

c. dishes表



主键是（店铺名，菜名）

d.discount\_dishes表

表结构同上表

e. orders表



将订单编号no设置为自增主键。who字段记录配送员名，done字段标记订单是否完成配送

5. **类、方法、全局变量或静态变量的命名规则**

a. 类名的每个单词首字母大写

b. 方法名的单词小写，单词之间以下划线”\_”分隔

c．全局变量的每个单词大写

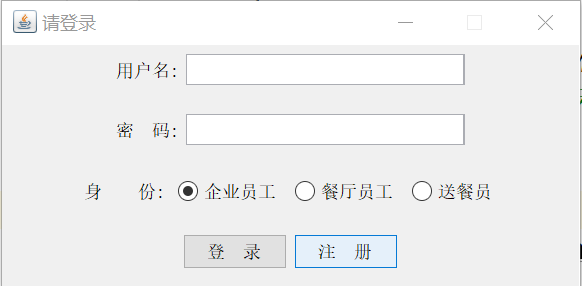
d. 局部变量的第一个单词小写，其后的单词首字母大写

三. **源程序清单**

|  |  |  |
| --- | --- | --- |
| **类型** | **类名** | **功能** |
| 入口类 | Login.java | 登录界面 |
| 重要类 | EStaff.java | 顾客界面 |
| CStaff.java | 商户界面 |
| PostMan.java | 派送员界面 |
| 辅助类 | Checker.java | 辅助登录 |
| ClientSend.java | 发送信息 |
| Server.java | 转发信息 |
| TableCellTextAreaRenderer.java | 调整表格单元格高度 |

四. **运行情况**

**1. 登录界面**



**2.顾客界面**

**a.Menu**

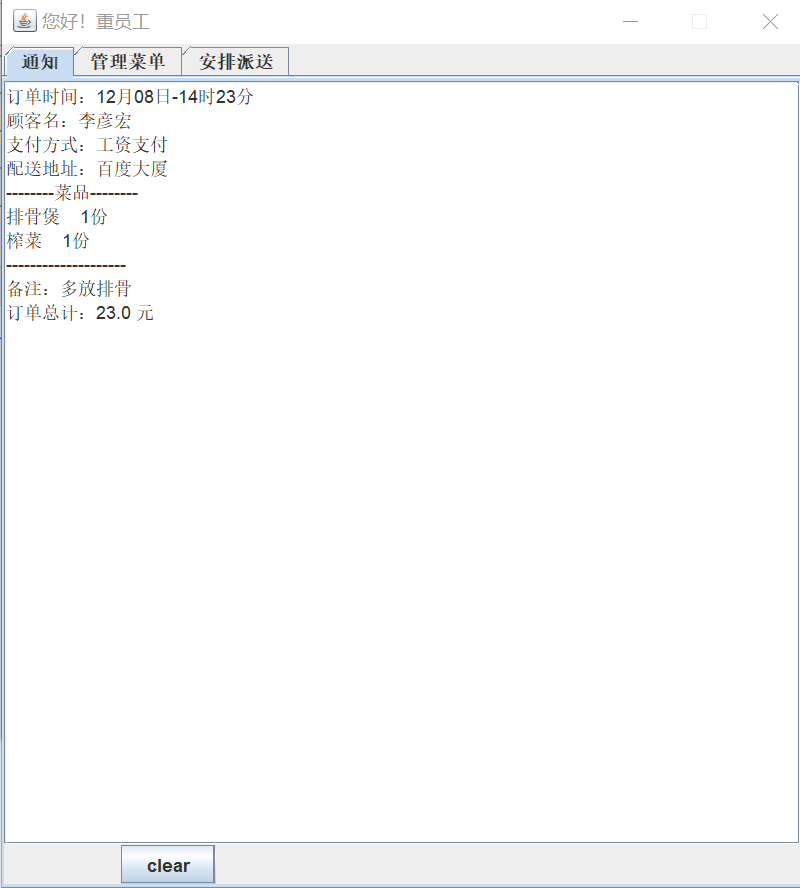


**b. 购物车**



**3．商户界面**

**a. 通知界面**



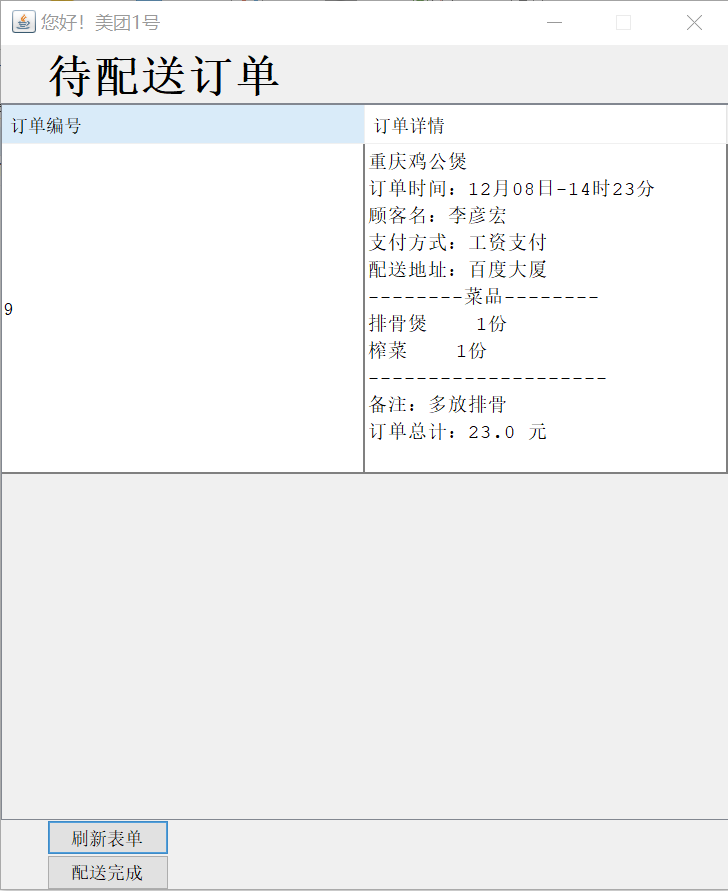
**b. 菜单管理界面**



**c. 请求派送界面**



4. 派送员界面



五. **总结**

客户端与服务器的通信主要由JDBC和网络编程实现。JDBC担任了绝大多数客户端与服务器的通信任务，包括验证账号登录、获取商户信息、获取菜单以及存储订单，数据库存储信息很方便，但是实时性不好，对于新订单生成这件事无法很好地实现。所以为了实时地通知店铺，需要引入网络编程这一工具。

为编写企业级的应用，还需要应用更多的Java开发知识。

# 附录

|  |
| --- |
| */\* \* 系统的登录界面 \*/* **import** java.awt.\*; **import** java.awt.event.ActionEvent; **import** java.awt.event.ActionListener; **import** javax.swing.\*; **import** java.sql.\*; **class** Login **extends** JFrame {  JPanel **panel1**,**panel2**,**panel3**,**panel4**;*// 面板* JRadioButton **radioButton1**,**radioButton2**,**radioButton3**;  ButtonGroup **buttonGroup**;  JLabel **label1**,**label2**,**label3**;*// 标签* JButton **button1**,**button2**;*// 按钮* JTextField **textField**;*// 用户名* JPasswordField **passwordField**;*// 密码  // JDBC* Connection **connection** = **null**;  Statement **statement** = **null**;  ResultSet **resultSet** = **null**;  Login()  {  *// 构造各组件* **panel1** = **new** JPanel();  **panel2** = **new** JPanel();  **panel3** = **new** JPanel();  **panel4** = **new** JPanel();   **label1** = **new** JLabel(**"用户名:"**);  **label2** = **new** JLabel(**"密 码:"**);  **label3** = **new** JLabel(**"身 份:"**);   **radioButton1** = **new** JRadioButton(**"企业员工"**);  **radioButton2** = **new** JRadioButton(**"餐厅员工"**);  **radioButton3** = **new** JRadioButton(**"送餐员"**);  **radioButton1**.setSelected(**true**);*// 默认选择* **buttonGroup** = **new** ButtonGroup();  **buttonGroup**.add(**radioButton1**);  **buttonGroup**.add(**radioButton2**);  **buttonGroup**.add(**radioButton3**);    **button1** = **new** JButton(**"登 录"**);  **button1**.addActionListener(**new** Button1Listener());  **button2** = **new** JButton(**"注 册"**);  **button2**.addActionListener(**new** Button2Listener());   **textField** = **new** JTextField(30);  **passwordField** = **new** JPasswordField(30);   **this**.setLayout(**new** GridLayout(4,1));*// 网格式布局  // 添加组件到面板* **panel1**.add(**label1**);  **panel1**.add(**textField**);  **panel2**.add(**label2**);  **panel2**.add(**passwordField**);  **panel3**.add(**label3**);  **panel3**.add(**radioButton1**);  **panel3**.add(**radioButton2**);  **panel3**.add(**radioButton3**);  **panel4**.add(**button1**);  **panel4**.add(**button2**);   *// 添加面板到Frame* **this**.add(**panel1**);  **this**.add(**panel2**);  **this**.add(**panel3**);  **this**.add(**panel4**);  *// 设置Frame* **this**.setTitle(**"请登录"**);  **this**.setSize(400,200);  **this**.setLocationRelativeTo(**null**);*// 居中显示* **this**.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);  **this**.setVisible(**true**);  **this**.setResizable(**false**);*// 不允许改变大小* **try***// 在构造函数中连接数据库，就不用再在成员函数中连接了* {  Class.*forName*(**"com.mysql.cj.jdbc.Driver"**);  **connection** = DriverManager.*getConnection*(**"jdbc:mysql://192.168.159.131/OrderingSystem"**, **"root"**, **"123456"**);  **statement** = **connection**.createStatement();  }**catch**(Exception ex)  {  ex.printStackTrace();  }  }  **class** Button1Listener **implements** ActionListener  {  **public void** actionPerformed(ActionEvent event)  {  *// 验证用户登录* **button1**.setText(**"登录中..."**);  **if**(**radioButton1**.isSelected() && check(1,**textField**.getText(),String.*valueOf*(**passwordField**.getPassword())))*// 被选中且验证通过* {  JOptionPane.*showMessageDialog*(**null**,**"登录成功！"**);  dispose();*// 销毁登录界面* **new** EStaff(**textField**.getText());*// 打开顾客界面* }  **else if**(**radioButton2**.isSelected() && check(2,**textField**.getText(),String.*valueOf*(**passwordField**.getPassword())))  {  JOptionPane.*showMessageDialog*(**null**,**"登录成功！"**);  dispose();  **new** CStaff(**textField**.getText());  }  **else if**(**radioButton3**.isSelected() && check(3,**textField**.getText(),String.*valueOf*(**passwordField**.getPassword())))  {  JOptionPane.*showMessageDialog*(**null**,**"登录成功！"**);  dispose();  **new** PostMan(**textField**.getText());  }  **else** {  JOptionPane.*showMessageDialog*(**null**,**"登录失败，用户名、密码或身份错误！"**,**"标题"**,JOptionPane.***ERROR\_MESSAGE***);  clear();*// 清楚框中内容* **button1**.setText(**"登 录"**);  }  }  }  **class** Button2Listener **implements** ActionListener  {  **public void** actionPerformed(ActionEvent event)  {  **int** tid = 1;*// 默认只提供顾客用户的注册* **if**(!**radioButton1**.isSelected())  {  JOptionPane.*showMessageDialog*(**null**,**"未开放注册"**,**"标题"**,JOptionPane.***ERROR\_MESSAGE***);  **return**;  }  **try** {  String sql = **"select** *\** **from login where identity = "** + String.*valueOf*(tid) + **" and usr\_name = '"** + **textField**.getText() + **"'"**;  *// System.out.println(sql);// 检验sql语句的正确性* **resultSet** = **statement**.executeQuery(sql);  **if**(**resultSet**.next() == **false**)*// 如果数据库中不存在该用户名，则可以注册* {  sql = **"insert into login values("**+String.*valueOf*(tid)+**",'"**+**textField**.getText()+**"','"**+String.*valueOf*(**passwordField**.getPassword())+**"')"**;  **statement**.executeUpdate(sql);  JOptionPane.*showMessageDialog*(**null**,**"注册成功！"**);  }  **else** {  JOptionPane.*showMessageDialog*(**null**,**"用户名已存在！"**,**"标题"**,JOptionPane.***ERROR\_MESSAGE***);  }  }**catch**(Exception ex)  {  ex.printStackTrace();  }  }  }  *// 形参：用户类型、用户名和密码* **public boolean** check(**int** id,String name,String pwd)  {  String ret = **null**;  **try** {  String sql = **"select pwd from login where identity = "** + String.*valueOf*(id) + **" and usr\_name = '"** + name + **"'"**;  *// System.out.println(sql);* **resultSet** = **statement**.executeQuery(sql);  **if**(**resultSet**.next())  ret = **resultSet**.getString(1);  }**catch**(Exception ex)  {  ex.printStackTrace();  }  **if**(ret == **null** || ret.equals(pwd) == **false**)  **return false**;  **else  return true**;  }  **void** clear()  {  **textField**.setText(**""**);  **passwordField**.setText(**""**);  }  **protected void** finalize()*// 对象消亡时关闭资源* {  **if** (**statement** != **null**)  {  **try** {  **statement**.close();  }**catch**(SQLException e)  {  e.printStackTrace();  }  }  **if** (**connection** != **null**)  {  **try** {  **connection**.close();  }**catch**(SQLException e)  {  e.printStackTrace();  }  }  }  **public static void** main(String[] Args)  {  **try***// 改变UI风格* {  UIManager.*setLookAndFeel*(UIManager.*getSystemLookAndFeelClassName*());  }**catch**(ClassNotFoundException | InstantiationException | IllegalAccessException | UnsupportedLookAndFeelException e)  {  e.printStackTrace();  }  **new** Login();  } }  */\* \* 顾客用户的操作界面 \*/* **import** java.awt.\*; **import** java.awt.event.ActionEvent; **import** java.awt.event.ActionListener; **import** java.awt.event.ItemEvent; **import** java.awt.event.ItemListener; **import** javax.swing.\*; **import** javax.swing.event.ChangeEvent; **import** javax.swing.event.ChangeListener; **import** javax.swing.event.TableModelEvent; **import** javax.swing.event.TableModelListener; **import** javax.swing.table.DefaultTableModel; **import** java.text.SimpleDateFormat; **import** java.sql.\*; **import** java.util.\*; **public class** EStaff **extends** JFrame {  String **userName**;*// 用户名* JTabbedPane **tabbedPane**;*// 选项卡面板* JPanel **panel1**,**panel2**;  JLabel **label1**,**label2**,**label3**,**label4**,**label5**,**label6**,**label7**,**label8**,**label9**;  JRadioButton **radioButton1**,**radioButton2**;  ButtonGroup **buttonGroup**;  JTable **table1**,**table2**,**table3**;  DefaultTableModel **model1**,**model2**,**model3**;  JScrollPane **scrollPane1**,**scrollPane2**,**scrollPane3**;  String[] **strArr1** = {**"名称"**,**"单价"**};  String[] **strArr2** = {**"名称"**,**"特价"**};  String[] **strArr3**= {**"店铺名称"**,**"菜品名称"**,**"单价"**,**"数量"**,**"小计"**};  Font **font** = **new** Font(**"Serif"**,Font.***BOLD***,30);*// 字体* JComboBox<String> **comboBox**;*// 下拉列表框* JTextField **textField1**,**textField2**,**textField3**;  JButton **button1**,**button2**,**button3**;  Connection **connection** = **null**;  Statement **statement** = **null**;  ResultSet **resultSet** = **null**;  **void** init\_comboBox()*// 构造下拉列表框* {  ArrayList<String> dynamicArr = **new** ArrayList<String>();*// 动态数组，用于生成普通数组* **try***// 查询所有的商家名* {  String sql = **"select shop\_name from usr\_shop"**;  **resultSet** = **statement**.executeQuery(sql);  **while**(**resultSet**.next())  dynamicArr.add(**resultSet**.getString(1));  }**catch**(Exception ex)  {  ex.printStackTrace();  }  String[] ShopName = **new** String[dynamicArr.size()];*// 商家名数组* **for**(**int** i = 0; i < dynamicArr.size(); ++i)  ShopName[i] = dynamicArr.get(i);  **comboBox** = **new** JComboBox<String>(ShopName);  **comboBox**.addItemListener(**new** ItemListener()  {  **public void** itemStateChanged(ItemEvent e)  {  *// 选中项被更改* **if**(e.getStateChange() == ItemEvent.***SELECTED***)  {  **model1**.setRowCount(0);*// 清空table1* **model2**.setRowCount(0);*// 清空table2* **try***// 修改表格内容* {  *// table1* String sql = **"select dish\_name,dish\_price from dishes where shop\_name = '"** + **comboBox**.getSelectedItem() + **"'"**;  **resultSet** = **statement**.executeQuery(sql);  **while**(**resultSet**.next())  {  **model1**.addRow(**new** String[]{**resultSet**.getString(1), **resultSet**.getString(2)});  }  *// table2* sql = **"select dish\_na,dish\_pri from discount\_dishes where shop\_na = '"** + **comboBox**.getSelectedItem() + **"'"**;  **resultSet** = **statement**.executeQuery(sql);  **while**(**resultSet**.next())  {  **model2**.addRow(**new** String[]{**resultSet**.getString(1), **resultSet**.getString(2)});  }  }**catch**(Exception ex)  {  ex.printStackTrace();  }  }  }  });  }  **void** init\_panel1()*//绘制1号选项卡* {  **panel1** = **new** JPanel();  **label1** = **new** JLabel(**"选择商家"**);  **label1**.setFont(**font**);  **label2** = **new** JLabel(**"美食"**);  **label2**.setFont(**font**);  **label3** = **new** JLabel(**"今日特价"**);  **label3**.setFont(**font**);  **label4** = **new** JLabel(**"选择数量"**);  **label4**.setFont(**font**);  init\_comboBox();*// 初始化下拉列表框* **model1** = **new** DefaultTableModel(**strArr1**,0);  **table1** = **new** JTable(**model1**)  {  *// 使得单元格可被选中但不能够被修改* **public boolean** isCellEditable(**int** row,**int** column)  {  **return false**;  }  };  **table1**.getTableHeader().setReorderingAllowed(**false**);*// 禁止改变列的相对位置  //table1.setEnabled(false);// 禁止选中单元格* **scrollPane1** = **new** JScrollPane(**table1**);   **model2** = **new** DefaultTableModel(**strArr2**,0);  **table2** = **new** JTable(**model2**)  {  *// 使得单元格可被选中但不能够被修改* **public boolean** isCellEditable(**int** row,**int** column)  {  **return false**;  }  };  **table1**.getTableHeader().setReorderingAllowed(**false**);  **scrollPane2** = **new** JScrollPane(**table2**);   **textField1** = **new** JTextField();  **textField1**.setText(**"1"**);*// 默认数量为1* **button1** = **new** JButton(**"加入订单"**);  **button1**.addActionListener(**new** Button1Listener());   **panel1**.setLayout(**new** BoxLayout(**panel1**,BoxLayout.***Y\_AXIS***));*// 设置布局管理器* **panel1**.add(**label1**);  **panel1**.add(**comboBox**);  **panel1**.add(**label2**);  **panel1**.add(**scrollPane1**);  **panel1**.add(**label3**);  **panel1**.add(**scrollPane2**);  **panel1**.add(**label4**);  **panel1**.add(**textField1**);  **panel1**.add(**button1**);  }  **class** Button1Listener **implements** ActionListener  {  *// 这两张表格未设置选择时互斥，所以一次最好只选择一张表格中的项目* **public void** actionPerformed(ActionEvent event)  {  **int** selectedRow = **table1**.getSelectedRow();  **if**(selectedRow != -1)*// 如果存在被选中的表项* {  **double** sumPrice = Double.*valueOf*(**textField1**.getText())\*Double.*valueOf*(**table1**.getValueAt(selectedRow,1).toString());*// 数量\*单价* **model3**.addRow(**new** String[]{**comboBox**.getSelectedItem().toString() ,**table1**.getValueAt(selectedRow,0).toString(),**table1**.getValueAt(selectedRow,1).toString(),**textField1**.getText(),String.*valueOf*(sumPrice)});*// 加入购物车表格* }  **table1**.clearSelection();*// 取消选中* selectedRow = **table2**.getSelectedRow();  **if**(selectedRow != -1)  {  **double** sumPrice = Double.*valueOf*(**textField1**.getText())\*Double.*valueOf*(**table2**.getValueAt(selectedRow,1).toString());*// 数量\*单价* **model3**.addRow(**new** String[]{**comboBox**.getSelectedItem().toString() ,**table2**.getValueAt(selectedRow,0).toString(),**table2**.getValueAt(selectedRow,1).toString(),**textField1**.getText(),String.*valueOf*(sumPrice)});*// 加入购物车表格* }  **table2**.clearSelection();*// 取消选中* }  }  **void** init\_panel2()*// 绘制2号选项卡* {  **button2** = **new** JButton(**"清空购物车"**);  **button2**.addActionListener(**new** Button2Listener());  **button3** = **new** JButton(**"提交订单"**);  **button3**.addActionListener(**new** Button3Listener());   **label5** = **new** JLabel(**"订单详情"**);  **label5**.setFont(**font**);  **label6** = **new** JLabel(**"总计："**+String.*valueOf*(0));  **label6**.setFont(**font**);  **label7** = **new** JLabel(**"填写配送地址"**);  **label7**.setFont(**font**);  **label8** = **new** JLabel(**"选择付款方式"**);  **label8**.setFont(**font**);  **label9** = **new** JLabel(**"备注"**);  **label9**.setFont(**font**);    **model3** = **new** DefaultTableModel(**strArr3**,0);  **model3**.addTableModelListener(**new** TableModelListener()  {  *// 当表格被修改时，重新计算订单的总价* **public void** tableChanged(TableModelEvent e)  {  **double** sum\_price = 0;  **for**(**int** i = 0; i < **model3**.getRowCount(); ++i)  sum\_price += Double.*valueOf*(**model3**.getValueAt(i,4).toString());  **label6**.setText(**"总计："**+String.*valueOf*(sum\_price)+**" 元"**);  }  });  **table3** = **new** JTable(**model3**);  **table3**.getTableHeader().setReorderingAllowed(**false**);  **table3**.setEnabled(**false**);  **scrollPane3** = **new** JScrollPane(**table3**);   **textField2** = **new** JTextField();  **textField3** = **new** JTextField();   **radioButton1** = **new** JRadioButton(**"工资支付"**);  **radioButton2** = **new** JRadioButton(**"现金支付"**);  **radioButton1**.setSelected(**true**);  **buttonGroup** = **new** ButtonGroup();  **buttonGroup**.add(**radioButton1**);  **buttonGroup**.add(**radioButton2**);   **panel2** = **new** JPanel();  **panel2**.setLayout(**new** BoxLayout(**panel2**,BoxLayout.***Y\_AXIS***));  **panel2**.add(**label5**);  **panel2**.add(**scrollPane3**);  **panel2**.add(**label6**);  **panel2**.add(**button2**);  **panel2**.add(**label7**);  **panel2**.add(**textField2**);  **panel2**.add(**label8**);  **panel2**.add(**radioButton1**);  **panel2**.add(**radioButton2**);  **panel2**.add(**label9**);  **panel2**.add(**textField3**);  **panel2**.add(**button3**);  }  **class** Button2Listener **implements** ActionListener*// 清空购物车按钮的监听者* {  **public void** actionPerformed(ActionEvent event)  {  **model3**.setRowCount(0);*// 清空表格* }  }  **class** Button3Listener **implements** ActionListener*// 提交订单按钮的监听者* {  **public void** actionPerformed(ActionEvent event)  {  *// 目标商家名，此处假设每次只向一家商户下订单，否则会出错* String shopName = **model3**.getValueAt(0,0).toString();  String info = shopName +**"\n"**;   String payMethod = **""**;  **if**(**radioButton1**.isSelected())  payMethod = **"工资支付"**;  **else** payMethod = **"现金支付"**;   String curTime = **new** SimpleDateFormat(**"MM月dd日-HH时mm分"**).format(System.*currentTimeMillis*());   info += **"订单时间："**+curTime+**"\n顾客名："**+**userName**+**"\n支付方式："**+payMethod+**"\n配送地址："**+**textField2**.getText();   info += **"\n--------菜品--------\n"**;  **for**(**int** i = 0; i < **model3**.getRowCount(); ++i)  {  info += **model3**.getValueAt(i,1).toString();  info += **" "**;  info += **model3**.getValueAt(i,3).toString();  info += **"份\n"**;  }  info += **"--------------------\n"**;  info += **"备注："**+**textField3**.getText()+**"\n"**;  info += **"订单"** + **label6**.getText() + **"\n"**;  *// System.out.println(info);// 检验订单信息的格式  // 委托消息发送者将订单信息发给商家* Sender sender = **new** Sender();  sender.send(info);  **try** {  String sql = **"insert into orders(shop\_name,info,who,done) values('"**+shopName+**"','"**+info+**"','nobody',0)"**;  *// System.out.println(sql);* **statement**.executeUpdate(sql);  }**catch**(Exception ex)  {  ex.printStackTrace();  }  }  }  **public** EStaff(String str)  {  **try** {  Class.*forName*(**"com.mysql.cj.jdbc.Driver"**);  **connection** = DriverManager.*getConnection*(**"jdbc:mysql://192.168.159.131/OrderingSystem"**, **"root"**, **"123456"**);  **statement** = **connection**.createStatement();  }**catch**(Exception ex)  {  ex.printStackTrace();  }  **userName** = str;  **tabbedPane** = **new** JTabbedPane(JTabbedPane.***TOP***);  init\_panel1();  init\_panel2();  **tabbedPane**.add(**panel1**,**"Menu"**);  **tabbedPane**.add(**panel2**,**"购物车"**);  **tabbedPane**.addChangeListener(**new** ChangeListener()  {  *// 改变选项卡的大小* **public void** stateChanged(ChangeEvent e)  {  **if**(**tabbedPane**.getSelectedIndex() == 1)  *// 需在外部类中指明是谁的this* EStaff.**this**.setSize(550,500);  **else** EStaff.**this**.setSize(550,600);  }  });  **this**.add(**tabbedPane**);  **this**.setTitle(**"您好！"** + **userName** + **" 先生/女士"**);  **this**.setSize(550,600);  **this**.setLocationRelativeTo(**null**);*// 居中显示* **this**.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);  **this**.setVisible(**true**);  **this**.setResizable(**false**);*// 不允许改变大小* }  **protected void** finalize()*// 对象消亡时关闭资源* {  **if** (**statement** != **null**)  {  **try** {  **statement**.close();  }**catch**(SQLException e)  {  e.printStackTrace();  }  }  **if** (**connection** != **null**)  {  **try** {  **connection**.close();  }**catch**(SQLException e)  {  e.printStackTrace();  }  }  }  **public static void** main(String[] Args)  {  *// 调试用* **new** EStaff(**"乔布斯"**);  } }  */\* \* 商家的操作界面 \*/* **import** java.awt.\*; **import** java.awt.event.ActionEvent; **import** java.awt.event.ActionListener; **import** javax.swing.\*; **import** javax.swing.table.DefaultTableCellRenderer; **import** javax.swing.table.DefaultTableModel; **import** java.io.\*; **import** java.net.\*; **import** java.sql.\*; **import** java.util.\*; **public class** CStaff **extends** JFrame {  String **userName**;  String **shopName**;  BufferedReader **reader**;*// 读取从服务器端发送来的实时消息* Socket **sock**;   JTabbedPane **tabbedPane**;  JPanel **panel1**,**panel2**,**panel3**;  JTextArea **textArea**;  JScrollPane **scrollPane1**,**scrollPane2**,**scrollPane3**,**scrollPane4**;  JButton **button1**,**button2**,**button3**,**button4**,**button5**,**button6**,**button7**;  JLabel **label1**,**label2**,**label3**;  JTable **table1**,**table2**,**table3**;  DefaultTableModel **model1**,**model2**,**model3**;  JComboBox<String> **comboBox**;  String[] **strArr1** = {**"名称"**,**"单价"**};  String[] **strArr2** = {**"名称"**,**"特价"**};  String[] **strArr3** = {**"订单号"**,**"订单"**};  Font **font** = **new** Font(**"Serif"**,Font.***BOLD***,30);  Connection **connection** = **null**;  Statement **statement** = **null**;  ResultSet **resultSet** = **null**;  ArrayList<Integer> **arrList** = **new** ArrayList<Integer>();  **void** getShopName()*// 由商家用户名获得店铺名* {  **try** {  String sql = **"select shop\_name from usr\_shop where usr\_name = '"** + **userName** + **"'"**;  *// System.out.println(sql);* **resultSet** = **statement**.executeQuery(sql);  **if**(**resultSet**.next())  **shopName** = **resultSet**.getString(1);  }**catch**(Exception ex)  {  ex.printStackTrace();  }  }  **void** init\_panel1()  {  **button1** = **new** JButton(**"clear"**);  **button1**.addActionListener(**new** Button1Listener());   **textArea** = **new** JTextArea(**""**);  **textArea**.setEditable(**false**);  **scrollPane1** = **new** JScrollPane(**textArea**);   **panel1** = **new** JPanel();  **panel1**.setLayout(**new** BoxLayout(**panel1**,BoxLayout.***Y\_AXIS***));  **panel1**.add(**scrollPane1**);  **panel1**.add(**button1**);  }  **class** Button1Listener **implements** ActionListener*// clear按钮的监听者* {  **public void** actionPerformed(ActionEvent event)  {  **textArea**.setText(**""**);  }  }  **void** init\_panel2()  {  **label1** = **new** JLabel(**"普通菜单"**);  **label1**.setFont(**font**);  **label2** = **new** JLabel(**"特价菜单"**);  **label2**.setFont(**font**);   **model1** = **new** DefaultTableModel(**strArr1**,0);  **table1** = **new** JTable(**model1**)  {  *// 使得单元格可被选中但不能够被修改* **public boolean** isCellEditable(**int** row,**int** column)  {  **return false**;  }  };  **table1**.getTableHeader().setReorderingAllowed(**false**);  **scrollPane2** = **new** JScrollPane(**table1**);   **model2** = **new** DefaultTableModel(**strArr2**,0);  **table2** = **new** JTable(**model2**)  {  *// 使得单元格可被选中但不能够被修改* **public boolean** isCellEditable(**int** row,**int** column)  {  **return false**;  }  };  **table2**.getTableHeader().setReorderingAllowed(**false**);  **scrollPane3** = **new** JScrollPane(**table2**);   **button2** = **new** JButton(**"更改为特价菜"**);  **button2**.addActionListener(**new** Button2Listener());  **button3** = **new** JButton(**"更改为普通菜"**);  **button3**.addActionListener(**new** Button3Listener());  **button4** = **new** JButton(**"增加"**);  **button4**.addActionListener(**new** Button4Listener());  **button5** = **new** JButton(**"删除"**);  **button5**.addActionListener(**new** Button5Listener());   **panel2** = **new** JPanel();  **panel2**.setLayout(**new** BoxLayout(**panel2**,BoxLayout.***Y\_AXIS***));  **panel2**.add(**label1**);  **panel2**.add(**scrollPane2**);  **panel2**.add(**button2**);  **panel2**.add(**label2**);  **panel2**.add(**scrollPane3**);  **panel2**.add(**button3**);  **panel2**.add(**button4**);  **panel2**.add(**button5**);  loadTable();  }  **void** loadTable()  {  **try** {  *// table1* String sql = **"select dish\_name,dish\_price from dishes where shop\_name = '"** + **shopName** + **"'"**;  **resultSet** = **statement**.executeQuery(sql);  **while**(**resultSet**.next())  {  **model1**.addRow(**new** String[]{**resultSet**.getString(1), **resultSet**.getString(2)});  }  *// table2* sql = **"select dish\_na,dish\_pri from discount\_dishes where shop\_na = '"** + **shopName** + **"'"**;  **resultSet** = **statement**.executeQuery(sql);  **while**(**resultSet**.next())  {  **model2**.addRow(**new** String[]{**resultSet**.getString(1), **resultSet**.getString(2)});  }  }**catch**(Exception ex)  {  ex.printStackTrace();  }  }  **class** Button2Listener **implements** ActionListener  {  *// 更改为普通菜* **public void** actionPerformed(ActionEvent event)  {  **int** selectedRow = **table1**.getSelectedRow();  JFrame frame = **new** JFrame();  JPanel panel = **new** JPanel();  JLabel label = **new** JLabel(**"设置新价格"**);  JTextField textField = **new** JTextField();  JButton button = **new** JButton(**"确认"**);  button.addActionListener(**new** ActionListener()  {  **public void** actionPerformed(ActionEvent event)  {  **try** {  *// 先删除* String sql = **"delete from dishes where dish\_name = '"** + **table1**.getValueAt(selectedRow,0).toString() + **"'"**;  **statement**.executeUpdate(sql);  *// 再添加* sql = **"insert into discount\_dishes values('"**+**shopName**+**"','"**+**table1**.getValueAt(selectedRow,0).toString()+**"',"**+textField.getText()+**")"**;  **statement**.executeUpdate(sql);  }**catch**(Exception ex)  {  ex.printStackTrace();  }  frame.dispose();  *// 刷新两张表格* **model1**.setRowCount(0);  **model2**.setRowCount(0);  loadTable();  }  });  panel.setLayout(**new** BoxLayout(panel,BoxLayout.***Y\_AXIS***));  panel.add(label);  panel.add(textField);  panel.add(button);  frame.getContentPane().add(BorderLayout.***NORTH***,panel);  frame.setSize(300,100);  frame.setLocationRelativeTo(**null**);*// 居中显示* frame.setVisible(**true**);  }  }  **class** Button3Listener **implements** ActionListener  {  **public void** actionPerformed(ActionEvent event)  {  **int** selectedRow = **table2**.getSelectedRow();  JFrame frame = **new** JFrame();  JPanel panel = **new** JPanel();  JLabel label = **new** JLabel(**"设置新价格"**);  JTextField textField = **new** JTextField();  JButton button = **new** JButton(**"确认"**);  button.addActionListener(**new** ActionListener()  {  **public void** actionPerformed(ActionEvent event)  {  **try** {  *// 先删除* String sql = **"delete from discount\_dishes where dish\_na = '"** + **table2**.getValueAt(selectedRow,0).toString() + **"'"**;  **statement**.executeUpdate(sql);  *// 再添加* sql = **"insert into dishes values('"**+**shopName**+**"','"**+**table2**.getValueAt(selectedRow,0).toString()+**"',"**+textField.getText()+**")"**;  **statement**.executeUpdate(sql);  }**catch**(Exception ex)  {  ex.printStackTrace();  }  frame.dispose();  *// 刷新两张表格* **model1**.setRowCount(0);  **model2**.setRowCount(0);  loadTable();  }  });  panel.setLayout(**new** BoxLayout(panel,BoxLayout.***Y\_AXIS***));  panel.add(label);  panel.add(textField);  panel.add(button);  frame.getContentPane().add(BorderLayout.***NORTH***,panel);  frame.setSize(300,100);  frame.setLocationRelativeTo(**null**);*// 居中显示* frame.setVisible(**true**);  }  }  **class** Button4Listener **implements** ActionListener  {  **public void** actionPerformed(ActionEvent event)  {  JFrame frame = **new** JFrame();  JPanel panel = **new** JPanel();  JLabel label1 = **new** JLabel(**"设置菜名"**);  JLabel label2 = **new** JLabel(**"设置价格"**);  JTextField textField1 = **new** JTextField();  JTextField textField2 = **new** JTextField();  JButton button = **new** JButton(**"确认"**);  button.addActionListener(**new** ActionListener()  {  **public void** actionPerformed(ActionEvent event)  {  **try** {  String sql = **"insert into dishes values('"**+**shopName**+**"','"**+textField1.getText()+**"',"**+textField2.getText()+**")"**;  *// System.out.println(sql);* **statement**.executeUpdate(sql);  }**catch**(Exception ex)  {  ex.printStackTrace();  }  frame.dispose();  *// 刷新两张表格* **model1**.setRowCount(0);  **model2**.setRowCount(0);  loadTable();  }  });  panel.setLayout(**new** BoxLayout(panel,BoxLayout.***Y\_AXIS***));  panel.add(label1);  panel.add(textField1);  panel.add(label2);  panel.add(textField2);  panel.add(button);  frame.getContentPane().add(BorderLayout.***NORTH***,panel);  frame.setSize(300,150);  frame.setLocationRelativeTo(**null**);*// 居中显示* frame.setVisible(**true**);  }  }  **class** Button5Listener **implements** ActionListener  {  **public void** actionPerformed(ActionEvent event)  {  JFrame frame = **new** JFrame();  JPanel panel = **new** JPanel();  JLabel label = **new** JLabel(**"输入菜名"**);  JTextField textField = **new** JTextField();  JButton button = **new** JButton(**"确认"**);  button.addActionListener(**new** ActionListener()  {  **public void** actionPerformed(ActionEvent event)  {  **try** {  String sql = **"delete from dishes where dish\_name = '"** + textField.getText() + **"'"**;  *// System.out.println(sql);* **statement**.executeUpdate(sql);  }**catch**(Exception ex)  {  ex.printStackTrace();  }  frame.dispose();  *// 刷新两张表格* **model1**.setRowCount(0);  **model2**.setRowCount(0);  loadTable();  }  });  panel.setLayout(**new** BoxLayout(panel,BoxLayout.***Y\_AXIS***));  panel.add(label);  panel.add(textField);  panel.add(button);  frame.getContentPane().add(BorderLayout.***NORTH***,panel);  frame.setSize(300,100);  frame.setLocationRelativeTo(**null**);*// 居中显示* frame.setVisible(**true**);  }  }  **void** init\_panel3()  {  **label3** = **new** JLabel(**"选择配送员"**);  **model3** = **new** DefaultTableModel(**strArr3**,0);  **table3** = **new** JTable(**model3**)  {  *// 使得单元格可被选中但不能够被修改* **public boolean** isCellEditable(**int** row,**int** column)  {  **return false**;  }  };  **table3**.getTableHeader().setReorderingAllowed(**false**);  **table3**.getColumn(**"订单"**).setCellRenderer(**new** TableCellTextAreaRenderer());*// 自动换行* DefaultTableCellRenderer backGroundColor = **new** DefaultTableCellRenderer()*// 修改表格颜色* {  **public** Component getTableCellRendererComponent(JTable table,Object value,**boolean** isSelected, **boolean** hasFocus, **int** row, **int** column)  {  Component cell = **super**.getTableCellRendererComponent(table,value,isSelected,hasFocus, row, column);  **if**(**arrList**.contains(Integer.*valueOf*(table.getValueAt(row,0).toString())) && column == 0 && cell.isBackgroundSet())  cell.setBackground(Color.***ORANGE***);  **else** cell.setBackground(Color.***WHITE***);  **return** cell;  }  };  **table3**.getColumn(**"订单号"**).setCellRenderer(backGroundColor);  **scrollPane4** = **new** JScrollPane(**table3**);   init\_comboBox();  loadTable3();   **button6** = **new** JButton(**"刷新"**);  **button6**.addActionListener(**new** Button6Listener());  **button7** = **new** JButton(**"配送"**);  **button7**.addActionListener(**new** Button7Listener());   **panel3** = **new** JPanel();  **panel3**.setLayout(**new** BoxLayout(**panel3**,BoxLayout.***Y\_AXIS***));  **panel3**.add(**scrollPane4**);  **panel3**.add(**button6**);  **panel3**.add(**label3**);  **panel3**.add(**comboBox**);  **panel3**.add(**button7**);  }  **class** Button6Listener **implements** ActionListener  {  **public void** actionPerformed(ActionEvent event)  {  loadTable3();  }  }  **void** init\_comboBox()*// 构造下拉列表框* {  ArrayList<String> dynamicArr = **new** ArrayList<String>();  String[] postManName;  **try** {  String sql = **"select usr\_name from login where identity = 3"**;  **resultSet** = **statement**.executeQuery(sql);  **while**(**resultSet**.next())  dynamicArr.add(**resultSet**.getString(1));  }**catch**(Exception ex)  {  ex.printStackTrace();  }  postManName = **new** String[dynamicArr.size()];  **for**(**int** i = 0; i < dynamicArr.size(); ++i)  postManName[i] = dynamicArr.get(i);  **comboBox** = **new** JComboBox<String>(postManName);  }  **class** Button7Listener **implements** ActionListener  {  **public void** actionPerformed(ActionEvent event)  {  *// 配送* String postManName = **comboBox**.getSelectedItem().toString();  String orderNo = **table3**.getValueAt(**table3**.getSelectedRow(),0).toString();  **try** {  String sql = **"update orders set who = '"**+postManName+**"' where no = "**+orderNo;  **statement**.executeUpdate(sql);  }**catch**(Exception ex)  {  ex.printStackTrace();  };  **table3**.clearSelection();  loadTable3();  }  }  **void** loadTable3()  {  **model3**.setRowCount(0);  **arrList**.clear();  **try** {  String sql = **"select no,info from orders where shop\_name = '"**+**shopName**+**"'"**;  **resultSet** = **statement**.executeQuery(sql);  **while**(**resultSet**.next())  {  **model3**.addRow(**new** String[]{**resultSet**.getString(1),**resultSet**.getString(2)});  }  sql = **"select no from orders where who <> 'nobody' and shop\_name = '"**+**shopName**+**"'"**;  **resultSet** = **statement**.executeQuery(sql);  **while**(**resultSet**.next())  {  **arrList**.add(**resultSet**.getInt(1));  }  }**catch**(Exception ex)  {  ex.printStackTrace();  }  }  CStaff(String str)  {  **try** {  Class.*forName*(**"com.mysql.cj.jdbc.Driver"**);  **connection** = DriverManager.*getConnection*(**"jdbc:mysql://192.168.159.131/OrderingSystem"**, **"root"**, **"123456"**);  **statement** = **connection**.createStatement();  }**catch**(Exception ex)  {  ex.printStackTrace();  }   **try** {  *// sock = new Socket("127.0.0.1",5000);// 连接本地主机* **sock** = **new** Socket(**"192.168.159.131"**,5000);  InputStreamReader streamReader = **new** InputStreamReader(**sock**.getInputStream());*// 输入流阅读器，接收来自Socket对象的输入* **reader** = **new** BufferedReader(streamReader);*// 缓冲区阅读器连接到输入流阅读器，流的起点为streamReader* System.***out***.println(**"Networking established"**);  }**catch**(IOException ex)  {  ex.printStackTrace();  }  Thread readerThread = **new** Thread(**new** IncomingReader());  readerThread.start();   **userName** = str;  getShopName();  init\_panel1();  init\_panel2();  init\_panel3();   **tabbedPane** = **new** JTabbedPane(JTabbedPane.***TOP***);  **tabbedPane**.add(**panel1**,**"通知"**);  **tabbedPane**.add(**panel2**,**"管理菜单"**);  **tabbedPane**.add(**panel3**,**"安排派送"**);   **this**.add(**tabbedPane**);  **this**.setTitle(**"您好！"** + **userName**);  **this**.setSize(550,600);  **this**.setLocationRelativeTo(**null**);*// 居中显示* **this**.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);  **this**.setVisible(**true**);  **this**.setResizable(**false**);*// 不允许改变大小* }  **public class** IncomingReader **implements** Runnable  {  **public void** run()  {  String message;  String check;  **try** {  **while**(**true**)*// 实现了正确地接收信息* {  check = **reader**.readLine();*// 读取第一行信息* {  *// 循环读取缓冲区中的信息并打印* **while**((message = **reader**.readLine()) != **null**)  {  **if**(check.equals(**shopName**))*// 只接收发送给自家的消息* **textArea**.append(message+**"\n"**);  **if**(message.equals(**""**))  **break**;  }  }  }  }**catch**(Exception ex)  {  ex.printStackTrace();  }  }  }  **protected void** finalize()*// 对象消亡时关闭资源* {  **if** (**statement** != **null**)  {  **try** {  **statement**.close();  }**catch**(SQLException e)  {  e.printStackTrace();  }  }  **if** (**connection** != **null**)  {  **try** {  **connection**.close();  }**catch**(SQLException e)  {  e.printStackTrace();  }  }  }  **public static void** main(String[] Args)  {  *// 测试* **new** CStaff(**"重员工"**);  } }  */\* \* 配送员的操作界面 \*/* **import** javax.swing.\*; **import** javax.swing.table.DefaultTableModel; **import** java.awt.\*; **import** java.awt.event.ActionEvent; **import** java.awt.event.ActionListener; **import** java.sql.\*; **public class** PostMan {  String **userName**;*// 配送员名* JFrame **frame**;  JPanel **panel**;  JLabel **label**;  JButton **button1**,**button2**;  JTable **table**;  DefaultTableModel **model**;  JScrollPane **scrollPane**;  String[] **strArr** = {**"订单编号"**,**"订单详情"**};  Font **font** = **new** Font(**"Serif"**,Font.***BOLD***,30);  Connection **connection** = **null**;  Statement **statement** = **null**;  ResultSet **resultSet** = **null**;  PostMan(String str)  {  **try** {  Class.*forName*(**"com.mysql.cj.jdbc.Driver"**);  **connection** = DriverManager.*getConnection*(**"jdbc:mysql://192.168.159.131/OrderingSystem"**, **"root"**, **"123456"**);  **statement** = **connection**.createStatement();  }**catch**(Exception ex)  {  ex.printStackTrace();  }   **userName** = str;   **frame** = **new** JFrame();  **panel** = **new** JPanel();  **label** = **new** JLabel(**"待配送订单"**);  **label**.setFont(**font**);   **model** = **new** DefaultTableModel(**strArr**,0);  **table** = **new** JTable(**model**)  {  *// 使得单元格可被选中但不能够被修改* **public boolean** isCellEditable(**int** row,**int** column)  {  **return false**;  }  };  **table**.getTableHeader().setReorderingAllowed(**false**);  **table**.getColumn(**"订单详情"**).setCellRenderer(**new** TableCellTextAreaRenderer());*// 表格内字符串自动换行* **scrollPane** = **new** JScrollPane(**table**);   **button1** = **new** JButton(**"刷新表单"**);  **button1**.addActionListener(**new** Button1Listener());  **button2** = **new** JButton(**"配送完成"**);  **button2**.addActionListener(**new** Button2Listener());  loadTable();   **panel**.setLayout(**new** BoxLayout(**panel**,BoxLayout.***Y\_AXIS***));  **panel**.add(**label**);  **panel**.add(**scrollPane**);  **panel**.add(**button1**);  **panel**.add(**button2**);   **frame**.add(**panel**);  **frame**.setTitle(**"您好！"** + **userName**);  **frame**.setSize(500,600);  **frame**.setLocationRelativeTo(**null**);*// 居中显示* **frame**.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);  **frame**.setVisible(**true**);  **frame**.setResizable(**false**);*// 不允许改变大小* }  **void** loadTable()*// 加载表格内容* {  **model**.setRowCount(0);  **try** {  String sql = **"select no,info from orders where who = '"**+**userName**+**"' and done = 0"** ;  **resultSet** = **statement**.executeQuery(sql);  **while**(**resultSet**.next())  {  **model**.addRow(**new** String[]{**resultSet**.getString(1),**resultSet**.getString(2)});  }  }**catch**(Exception ex)  {  ex.printStackTrace();  }  }  **class** Button1Listener **implements** ActionListener  {  *// 刷新表格按钮* **public void** actionPerformed(ActionEvent event)  {  loadTable();  }  }  **class** Button2Listener **implements** ActionListener  {  *// 配送完成按钮* **public void** actionPerformed(ActionEvent event)  {  String no = **table**.getValueAt(**table**.getSelectedRow(),0).toString();  **try** {  String sql = **"update orders set done = 1 where no = "** + no;  **statement**.executeUpdate(sql);  }**catch**(Exception ex)  {  ex.printStackTrace();  }  loadTable();  }  }  **protected void** finalize()*// 对象消亡时关闭资源* {  **if** (**statement** != **null**)  {  **try** {  **statement**.close();  }**catch**(SQLException e)  {  e.printStackTrace();  }  }  **if** (**connection** != **null**)  {  **try** {  **connection**.close();  }**catch**(SQLException e)  {  e.printStackTrace();  }  }  }  **public static void** main(String[] Args)  {  *// 测试* **new** PostMan(**"美团1号"**);  } }  */\* \* 此对象用于委托服务器端向其他客户端发送消息 \*/* **import** java.io.\*; **import** java.net.\*; **public class** Sender {  PrintWriter **writer**;  Socket **sock**;  Sender()  {  **try** {  *// sock = new Socket("127.0.0.1",5000);// 连接本地主机* **sock** = **new** Socket(**"192.168.159.131"**,5000);*// 由于本人将服务器端程序放到了服务器上运行，故在此这样填写* **writer** = **new** PrintWriter(**sock**.getOutputStream());*// 打印者与Socket对象的输出相连接，流的起点为writer  // System.out.println("Networking established");* }**catch**(IOException ex)  {  ex.printStackTrace();  }  }  **public void** send(String toSend)  {  **writer**.println(toSend);*// 输出流* **writer**.flush();  } }  */\* \* 注意：在网络编程中，对象只分为客户端和服务器端两种 \* 在本程序中，Sender对象也是客户端，是特殊的客户端 \* 而下文所指的客户端是特定语境下的狭义客户端 \* 具体说来就是，Sender对象是只能发送消息的客户端 \* 而下文所指的客户端是只能接收消息的客户端 \* 主线程与所有客户端（广义）建立连接，这种连接是无法区分的 \* 而子线程主要的目的是与Sender对象建立连接并处理来自Sender对象的输入并转发给各狭义上的客户端 \* 由于无法区分，子线程自然分为以下两种情况 \* ①与Sender对象连接的子线程：从缓冲区中成功读取消息，并转发给所有广义上的客户端 \* ②与狭义上的客户端建立连接的子线程：从缓冲区中读不到消息，发送出去的消息为空 \*/* **import** java.io.\*; **import** java.net.\*; **import** java.util.\*; **public class** Server {  ArrayList **clientOutputStreams** = **new** ArrayList();  **public class** SenderHandler **implements** Runnable  {  *// 目的：将来自Sender对象的消息转发给与服务器端建立连接的每一个狭义上的客户端* BufferedReader **reader**;  Socket **sock**;  **public** SenderHandler(Socket senderSocket)*// 内部类的构造函数* {  *// senderSocket的属性是广义上的客户端的套接字  // 但由于只有Sender对象才发送消息  // 所以如此命名  // 虽然也会与狭义上的客户端建立如下连接，但无妨* **try** {  **sock** = senderSocket;  InputStreamReader isReader = **new** InputStreamReader(**sock**.getInputStream());*// 建立连接* **reader** = **new** BufferedReader(isReader);*// 建立连接* }**catch**(Exception ex)  {  ex.printStackTrace();  }  }  **public void** tellEveryone(String message)  {  Iterator it = **clientOutputStreams**.iterator();*// 面向客户端的输出流集合（由服务器端向客户端）* PrintWriter writer;  **while**(it.hasNext())*// 遍历集合* {  **try** {  writer = (PrintWriter) it.next();  writer.print(message + **"\n"**);*// 因为客户端中readLine会等待回车、换行符，所以需要在这里人为添加，避免出现阻塞状态* writer.flush();  }**catch**(Exception ex)  {  ex.printStackTrace();  }  }  }  **public void** run()*// 将消息广播* {  *// 此处的目标是狭义上的客户端  // 虽然消息也会发送到Sender对象，但无妨* String message;  **try** {  *// 从输入缓冲区中循环读取信息  // 对于狭义上的客户端的套接字，此处不会读出任何消息* **while**((message = **reader**.readLine()) != **null**)  {  tellEveryone(message);*// 将收到的信息分发给每个客户端* }  }**catch**(Exception ex)  {  ex.printStackTrace();  }  }  }  **public void** go()*// 主线程* {  **try** {  ServerSocket serverSock = **new** ServerSocket(5000);  **while**(**true**)*// 不断接收来自客户端的连接，对于每一个连接* {  Socket clientSocket = serverSock.accept();  PrintWriter writer = **new** PrintWriter(clientSocket.getOutputStream());*// 建立连接（方向：服务器到客户端）* **clientOutputStreams**.add(writer);*// 保存连接!!!* Thread t = **new** Thread(**new** SenderHandler(clientSocket));*// 建立多个子线程* t.start();  System.***out***.println(**"Got a connection"**);  }  }**catch**(Exception ex)  {  ex.printStackTrace();  }  }  **public static void** main(String[] Args)  {  **new** Server().go();  } }  */\* \* 实现表格中字符串的自动换行功能 \*/* **import** javax.swing.\*; **import** java.awt.\*; **import** javax.swing.table.TableCellRenderer; **public class** TableCellTextAreaRenderer **extends** JTextArea **implements** TableCellRenderer {  **public** TableCellTextAreaRenderer()  {  **this**.setLineWrap(**true**);  **this**.setWrapStyleWord(**true**);  }  **public** Component getTableCellRendererComponent(JTable table, Object value,**boolean** isSelected,**boolean** hasFocus,**int** row,**int** column)  {  *// 计算当前行的最佳高度* **int** maxPreferredHeight = 0;  **for**(**int** i = 0; i < table.getColumnCount(); ++i)  {  **this**.setText(**""**+table.getValueAt(row,i));  **this**.setSize(table.getColumnModel().getColumn(column).getWidth(),0);  maxPreferredHeight = Math.*max*(maxPreferredHeight,getPreferredScrollableViewportSize().**height**);  }  **if**(table.getRowHeight(row) != maxPreferredHeight)  table.setRowHeight(row,maxPreferredHeight);  setText(value == **null** ? **""** : value.toString());  **return this**;  } } |