

# 组成

---

## 1.entity

---

### 1.Customers

```
1 package entity;
2
3 public class Customers {
4     private String customerNo;//会员编号
5     private String customerName;//会员姓名
6     private String phone;//会员电话
7     private int score;//会员积分
8
9     public Customers() {
10         super();
11     }
12
13     public String getCustomerNo() {
14         return customerNo;
15     }
16     public void setCustomerNo(String customerNo) {
17         this.customerNo = customerNo;
18     }
19     public String getCustomerName() {
20         return customerName;
21     }
22     public void setCustomerName(String customerName) {
23         this.customerName = customerName;
24     }
25     public String getPhone() {
26         return phone;
27     }
28     public void setPhone(String phone) {
29         this.phone = phone;
30     }
31     public int getScore() {
32         return score;
33     }
34     public void setScore(int score) {
35         this.score = score;
36     }
37 }
```

```
37  
38 }  
39
```

## 2.ProductInfo

```
1 package entity;  
2  
3 public class ProductInfo extends Products{  
4     private String categoryName;//商品种类名字  
5  
6     public ProductInfo() {  
7         super();  
8     }  
9  
10    public ProductInfo(String categoryName) {  
11        super();  
12        this.categoryName = categoryName;  
13    }  
14  
15    public String getCategoryName() {  
16        return categoryName;  
17    }  
18  
19    public void setCategoryName(String categoryName) {  
20        this.categoryName = categoryName;  
21    }  
22  
23 }  
24
```

## 3.Products

```
1 package entity;  
2  
3 public class Products {  
4     private String productNo;//商品编号  
5     private String categoryId;//种类编号  
6     private String productName;//商品名  
7     private String type;//规格  
8     private String unit;//单位  
9     private int pcount=-1;//库存数量  
10    private double price;//单价  
11    private double discount;//折扣
```

```
12     public Products() {
13         super();
14     }
15
16     public String getProductNo() {
17         return productNo;
18     }
19     public void setProductNo(String productNo) {
20         this.productNo = productNo;
21     }
22     public String getCategoryId() {
23         return categoryId;
24     }
25     public void setCategoryId(String categoryId) {
26         this.categoryId = categoryId;
27     }
28     public String getProductName() {
29         return productName;
30     }
31     public void setProductName(String productName) {
32         this.productName = productName;
33     }
34     public String getType() {
35         return type;
36     }
37     public void setType(String type) {
38         this.type = type;
39     }
40     public String getUnit() {
41         return unit;
42     }
43     public void setUnit(String unit) {
44         this.unit = unit;
45     }
46     public int getCount() {
47         return pcount;
48     }
49     public void setCount(int pcount) {
50         this.pcount = pcount;
51     }
52     public double getPrice() {
53         return price;
54     }
55     public void setPrice(double price) {
56         this.price = price;
```

```

57     }
58     public double getDiscount() {
59         return discount;
60     }
61     public void setDiscount(double discount) {
62         this.discount = discount;
63     }
64
65 }
66

```

## 4.SaleHistory

```

1  package entity;
2
3  import javax.swing.JOptionPane;
4
5  public class SaleHistory {
6      private int id;//记录编号
7      private String productNo;//商品编号
8      private int saleCount;//购买数量
9      private double price;//商品单价
10     private double discount;//商品折扣
11     private int userId;//管理员编号
12     private String saleTime;//售出时间
13     private int customerNo;//消费者编号
14     public SaleHistory() {
15         super();
16     }
17     public SaleHistory(int id, String productNo, int
saleCount, double price,
18                     double discount, int userId, String
saleTime, int customerNo) {
19         super();
20         this.id = id;
21         this.productNo = productNo;
22         this.saleCount = saleCount;
23         this.price = price;
24         this.discount = discount;
25         this.userId = userId;
26         this.saleTime = saleTime;
27         this.customerNo = customerNo;
28     }
29     public int getId() {

```

```
30         return id;
31     }
32     public void setId(int id) {
33         this.id = id;
34     }
35     public String getProductNo() {
36         return productNo;
37     }
38     public void setProductNo(String productNo) {
39         this.productNo = productNo;
40     }
41     public int getSaleCount() {
42         return saleCount;
43     }
44     public void setSaleCount(int saleCount) {
45         this.saleCount = saleCount;
46     }
47     public double getPrice() {
48         return price;
49     }
50     public int setPrice(double price) {
51         int flag=0;
52         if(price<0){
53             flag=0;
54         }
55         else {
56             this.price = price;
57             flag=1;
58         }
59         return flag;
60     }
61     public double getDiscount() {
62         return discount;
63     }
64     public void setDiscount(double discount) {
65         this.discount = discount;
66     }
67     public int getUserId() {
68         return userId;
69     }
70     public void setUserId(int userId) {
71         this.userId = userId;}
72     public String getSaleTime() {
73         return saleTime;
74     }
```

```

75     public void setSaleTime(String saleTime) {
76         this.saleTime = saleTime;
77     }
78     public int getCustomerNo() {
79         return customerNo;
80     }
81     public void setCustomerNo(int customerNo) {
82         this.customerNo = customerNo;
83     }
84
85 }
86

```

## 5.SaleSummary

```

1  package entity;
2
3  public class SaleSummary extends SaleHistory{
4      private double memberDiscount;//会员折扣
5      private double totalAmount;//单行总金额
6      public double getMemberDiscount() {
7          return memberDiscount;
8      }
9      public void setMemberDiscount(double memberDiscount) {
10         this.memberDiscount = memberDiscount;
11     }
12     public double getTotalAmount() {
13         return totalAmount;
14     }
15     public void setTotalAmount(double totalAmount) {
16         this.totalAmount = totalAmount;
17     }
18     public SaleSummary() {
19         super();
20     }
21
22 }
23

```

## 6.ScoreDiscount

```

1  package entity;
2
3  public class ScoreDiscount {

```

```

4     private int minScore;//积分
5     private double discount;//折扣
6     public ScoreDiscount() {
7         super();
8     }
9     public ScoreDiscount(int minScore, double discount) {
10        super();
11        this.minScore = minScore;
12        this.discount = discount;
13    }
14    public int getMinScore() {
15        return minScore;
16    }
17    public void setMinScore(int minScore) {
18        this.minScore = minScore;
19    }
20    public double getDiscount() {
21        return discount;
22    }
23    public void setDiscount(double discount) {
24        this.discount = discount;
25    }
26
27 }
28

```

## 7.Users

```

1     package entity;
2
3     public class Users {
4         private int userId;//管理员编号
5         private String userName;//管理员姓名
6         private String password;//管理员密码
7         private int roleId;//管理员级别
8         public Users() {
9             super();
10        }
11        public Users(int userId, String userName, String password,
12        int roleId) {
13            super();
14            this.userId = userId;
15            this.userName = userName;
16            this.password = password;
17        }
18    }
19

```

```

16         this.roleId = roleId;
17     }
18     public int getUserId() {
19         return userId;
20     }
21     public void setUserId(int userId) {
22         this.userId = userId;
23     }
24     public String getUsername() {
25         return userName;
26     }
27     public void setUsername(String userName) {
28         this.userName = userName;
29     }
30     public String getPassword() {
31         return password;
32     }
33     public void setPassword(String password) {
34         this.password = password;
35     }
36     public int getRoleId() {
37         return roleId;
38     }
39     public void setRoleId(int roleId) {
40         this.roleId = roleId;
41     }
42
43 }
44

```

## 2.dao

### 1.CategorieDao

```

1 package dao;
2
3 import java.sql.Connection;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7
8 public class CategorieDao {
9     public String getCategoryNameDao(String categoryId){ //种
        类编号
    }
}

```



```

10         String categoryName = "";
11         Connection conn=DBManager.getConnection();
12         PreparedStatement stat=null; // PreparedStatement可以有
    效防止sql注入,
13         ResultSet rs=null;
14         try {
15             stat=conn.prepareStatement("select * from products
    where categoryid = ?");
16             stat.setString(1, categoryId);
17             rs=stat.executeQuery();
18             rs.next();
19             categoryName =rs.getString(2);
20         } catch (SQLException e) {
21             e.printStackTrace();
22         }finally{
23             DBManager.closeAll(conn, stat, rs);
24         }
25         return categoryName;
26     }
27 }
28

```

## 2.CustomerDao

```

1  package dao;
2
3  import java.sql.Connection;
4  import java.sql.PreparedStatement;
5  import java.sql.ResultSet;
6  import java.sql.SQLException;
7  import java.util.ArrayList;
8  import java.util.List;
9
10 import javax.swing.JOptionPane;
11
12 import service.ScoreDiscountService;
13
14 import entity.Customers;
15 import entity.ScoreDiscount;
16 import entity.Users;
17
18 public class CustomerDao {
19     public List<Customers> getAllCustomersDao(){
20         List<Customers> list=new ArrayList<Customers>();

```

```

21         Connection conn=DBManager.getConnection();
22         PreparedStatement stat=null;
23         ResultSet rs=null;
24         try {
25             stat=conn.prepareStatement("select * from
customers");
26             rs=stat.executeQuery();
27             while(rs.next()){
28                 Customers customer=new Customers();
29                 customer.setCustomerNo(rs.getString(1));
30                 customer.setCustomerName(rs.getString(2));
31                 customer.setPhone(rs.getString(3));
32                 customer.setScore(rs.getInt(4));
33                 list.add(customer);
34             }
35         } catch (SQLException e) {
36             e.printStackTrace();
37         }finally{
38             DBManager.closeAll(conn, stat, rs);
39         }
40         return list;
41     }
42     public int newCustomersDao(Customers customer){
43         boolean flag = false;
44         String sql = "insert into customers
(customerno,customername,phone,score) values(?,?,?,?)";
45         Object[] objs =
{customer.getCustomerNo(),customer.getCustomerName(),customer
.getPhone(),0};
46         flag = DBManager.executeUpdate(sql, objs);
47
48
49
50
51         int customerNo = 0;
52         Connection conn=DBManager.getConnection();
53         PreparedStatement stat=null;
54         ResultSet rs=null;
55         try {
56             stat=conn.prepareStatement("select customerno
from customers where customername= ? ");
57             stat.setString(1,customer.getCustomerName());
58             rs=stat.executeQuery();
59             if(rs.next()){
60                 customerNo = rs.getInt(1);

```

```

61         }else{
62             JOptionPane.showMessageDialog(null,"新增会员错
        误","错误提示",JOptionPane.ERROR_MESSAGE);
63         }
64     } catch (SQLException e) {
65         e.printStackTrace();
66     }finally{
67         DBManager.closeAll(conn, stat, rs);
68     }
69     return customerNo;
70 }
71 public boolean deleteCustomersDao(String cstNo){
72     boolean flag = false;
73     String sql = "delete from customers where customerno
    =?";
74     Object[] objs = {cstNo};
75     flag = DBManager.executeUpdate(sql, objs);
76     return flag;
77 }
78 public boolean checkCustomerDao(int customerNo){
79     boolean flag = false;
80     Connection conn=DBManager.getConnection();
81     PreparedStatement stat=null;
82     ResultSet rs=null;
83     try {
84         stat=conn.prepareStatement("select * from
        customers where customerno = ?");
85         stat.setInt(1, customerNo);
86         rs=stat.executeQuery();
87         if(rs.next()){
88             flag = true;
89         }else{
90             JOptionPane.showMessageDialog(null,"该会员不存
        在","错误提示",JOptionPane.ERROR_MESSAGE);
91         }
92     } catch (SQLException e) {
93         e.printStackTrace();
94     }finally{
95         DBManager.closeAll(conn, stat, rs);
96     }
97     return flag;
98 }
99 public boolean checkCustomerNoDao(String customerNo){
100     boolean flag = false;
101     Connection conn=DBManager.getConnection();

```

```

102         PreparedStatement stat=null;
103         ResultSet rs=null;
104         try {
105             stat=conn.prepareStatement("select * from
customers where customerno = ?");
106             stat.setString(1, customerNo);
107             rs=stat.executeQuery();
108             if(rs.next()){
109                 JOptionPane.showMessageDialog(null,"该会员编号
已经存在","错误提示",JOptionPane.ERROR_MESSAGE);
110             }else{
111                 flag = true;
112             }
113         } catch (SQLException e) {
114             e.printStackTrace();
115         }finally{
116             DBManager.closeAll(conn, stat, rs);
117         }
118         return flag;
119     }
120     public double getMemberDiscountByCustomerNoDao(int
customerNo){
121         double discount = 1;
122         Connection conn=DBManager.getConnection();
123         PreparedStatement stat=null;
124         ResultSet rs=null;
125         try {
126             stat=conn.prepareStatement("select * from
customers where customerno = ?");
127             stat.setInt(1, customerNo);
128             rs=stat.executeQuery();
129             if(rs.next()){
130                 int score = rs.getInt(4);
131                 ScoreDiscountService sds = new
ScoreDiscountService();
132                 List<ScoreDiscount> list =
sds.getAllScoreDiscountService();
133                 for (ScoreDiscount scoreDiscount : list) {
134                     if(score>=scoreDiscount.getMinScore()){
135                         discount =
scoreDiscount.getDiscount();
136                     }
137                 }
138             }else{

```

```

139         JOptionPane.showMessageDialog(null,"该会员不存在", "错误提示",JOptionPane.ERROR_MESSAGE);
140     }
141     } catch (SQLException e) {
142         e.printStackTrace();
143     }finally{
144         DBManager.closeAll(conn, stat, rs);
145     }
146     return discount;
147 }
148 public boolean updateCustomerScoreDao(int customerNo,int
score){
149     boolean flag = false;
150     String sql = "update customers set score=score+ ?
where customerno = ?";
151     Object[] objs = {score,customerNo};
152     flag = DBManager.executeUpdate(sql, objs);
153     return flag;
154 }
155 }
156

```

### 3.DBManager

```

1 package dao;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.sql.Statement;
9
10 public class DBManager {
11     static String
driverName="com.microsoft.sqlserver.jdbc.SQLServerDriver";
12     static String dbURL =
"jdbc:sqlserver://127.0.0.1:1433;DatabaseName=heng_supermarket
";
13     static String userName="sa";
14     static String userPwd="zhangheng0.0123";
15
16     public static Connection getConnection(){
17         Connection conn=null;

```

```
18         try {
19             Class.forName(driverName);
20             conn=DriverManager.getConnection(dbURL,
21         userName,userPwd);
22             System.out.println("succeed");
23         } catch (Exception e) {
24             System.out.println("failed");
25             e.printStackTrace();
26         }
27         return conn;
28     }
29     public static boolean executeUpdate(String sql,Object[]
30     objs){
31         boolean flag=false;
32         Connection conn=getConnection();
33         PreparedStatement pst=null;
34         try {
35             pst=conn.prepareStatement(sql);
36             for(int i=0;i<objs.length;i++){
37                 pst.setObject(i+1, objs[i]);
38             }
39             int num=pst.executeUpdate();
40             if(num>0){
41                 flag=true;
42             }
43         } catch (SQLException e) {
44             e.printStackTrace();
45         }finally{
46             closeAll(conn, pst, null);
47         }
48         return flag;
49     }
50     public static void closeAll(Connection conn,Statement
51     stat,ResultSet rs){
52         try {
53             if(rs!=null){
54                 rs.close();
55             }
56             if(stat!=null){
57                 stat.close();
58             }
59             if(conn!=null){
60                 conn.close();
61             }
62         }
```

```

60         } catch (SQLException e) {
61             e.printStackTrace();
62         }
63     }
64 }
65

```

## 4.ProductsDao

```

1  package dao;
2
3  import java.awt.Component;
4  import java.sql.Connection;
5  import java.sql.PreparedStatement;
6  import java.sql.ResultSet;
7  import java.sql.SQLException;
8  import java.util.ArrayList;
9  import java.util.List;
10
11 import javax.swing.JOptionPane;
12
13 import service.CategoriesService;
14 import UI.GoodsAdministrator;
15
16 import entity.Customers;
17 import entity.ProductInfo;
18 import entity.Products;
19
20 public class ProductsDao {
21     public List<ProductInfo> getAllProductsDao(){
22         List<ProductInfo> list=new ArrayList<ProductInfo>();
23         Connection conn=DBManager.getConnection();
24         PreparedStatement stat=null;
25         ResultSet rs=null;
26         try {
27             stat=conn.prepareStatement("select * from
products");
28             rs=stat.executeQuery();
29             while(rs.next()){
30                 ProductInfo product = new ProductInfo();
31                 product.setProductNo(rs.getString(1));
32                 product.setCategoryId(rs.getString(2));
33                 product.setProductName(rs.getString(3));
34                 product.setType(rs.getString(4) );

```

```

35         product.setUnit(rs.getString(5));
36         product.setCount(rs.getInt(6));
37         product.setPrice(rs.getDouble(7));
38         product.setDiscount(rs.getDouble(8));
39         CategorieService cs = new CategorieService();
40
41         product.setCategoryName(cs.getCategoryNameService(product.getCategoryId()));
42         list.add(product);
43     } catch (SQLException e) {
44         e.printStackTrace();
45     } finally {
46         DBManager.closeAll(conn, stat, rs);
47     }
48     return list;
49 }
50 public ProductInfo getProductInfoByProductNo(String ProductNo){
51     ProductInfo product = new ProductInfo();
52     Connection conn=DBManager.getConnection();
53     PreparedStatement stat=null;
54     ResultSet rs=null;
55     try {
56         stat=conn.prepareStatement("select * from products where productno = ?");
57         stat.setString(1, ProductNo);
58         rs=stat.executeQuery();
59         if(rs.next()){
60             product.setProductNo(rs.getString(1));
61             product.setCategoryId(rs.getString(2));
62             product.setProductName(rs.getString(3));
63             product.setType(rs.getString(4) );
64             product.setUnit(rs.getString(5));
65             product.setCount(rs.getInt(6));
66             product.setPrice(rs.getDouble(7));
67             product.setDiscount(rs.getDouble(8));
68             CategorieService cs = new CategorieService();
69
70             product.setCategoryName(cs.getCategoryNameService(product.getCategoryId()));
71             JOptionPane.showMessageDialog(null, "查询的商品不存在", "错误提示", JOptionPane.ERROR_MESSAGE);
72         }

```



```

73         } catch (SQLException e) {
74             e.printStackTrace();
75         }finally{
76             DBManager.closeAll(conn, stat, rs);
77         }
78         return product;
79     }
80     public ProductInfo getProductInfoByproductName(String
productName){
81         ProductInfo product = new ProductInfo();
82         Connection conn=DBManager.getConnection();
83         PreparedStatement stat=null;
84         ResultSet rs=null;
85         try {
86             stat=conn.prepareStatement("select * from
products where `name` =?");
87             stat.setString(1, productName);
88             rs=stat.executeQuery();
89             if(rs.next()){
90                 product.setProductNo(rs.getString(1));
91                 product.setCategoryId(rs.getString(2));
92                 product.setProductName(rs.getString(3));
93                 product.setType(rs.getString(4) );
94                 product.setUnit(rs.getString(5));
95                 product.setCount(rs.getInt(6));
96                 product.setPrice(rs.getDouble(7));
97                 product.setDiscount(rs.getDouble(8));
98                 CategorieService cs = new CategorieService();
99
100                 product.setCategoryName(cs.getCategoryNameService(product.ge
tCategoryId()));
101             }else{
102                 JOptionPane.showMessageDialog(null,"查询的商品
不存在","错误提示",JOptionPane.ERROR_MESSAGE);
103             }
104         } catch (SQLException e) {
105             e.printStackTrace();
106         }finally{
107             DBManager.closeAll(conn, stat, rs);
108         }
109         return product;
110     }
111     public boolean newProductsDao(Products product){
        boolean flag = false;

```

```

112         String sql = "insert into products
values(?,?,?,?,?,?,?,?)";
113         Object[] objs =
{product.getProductNo(),product.getCategoryId(),product.getPr
oductName(),product.getType(),product.getUnit()
114
,product.getCount(),product.getPrice(),product.getDiscount()}
;
115         flag = DBManager.executeUpdate(sql, objs);
116         return flag;
117     }
118     public boolean deleteProductsDao(String productNo){
119         boolean flag = false;
120         String sql = "delete from products where productno
=?";
121         Object[] objs = {productNo};
122         flag = DBManager.executeUpdate(sql, objs);
123         return flag;
124     }
125     public boolean updateProductsDao(String productNo,int
pcount,double discount){
126         boolean flag = false;
127         String sql = "update products set pcount =
?,`discount` = ? where productno = ?";
128         Object[] objs = {pcount,discount,productNo};
129         flag = DBManager.executeUpdate(sql, objs);
130         return flag;
131     }
132     public boolean updateProductsCountDao(String
productNo,int soldNumber){
133         boolean flag = false;
134         int pcount = 0;
135         ProductsDao pd = new ProductsDao();
136         pcount =
pd.getProductInfoByProductNo(productNo).getCount()-
soldNumber;
137         String sql = "update products set pcount = ? where
productno = ?";
138         Object[] objs = {pcount,productNo};
139         flag = DBManager.executeUpdate(sql, objs);
140         return flag;
141     }
142     public boolean checkProductsNoDao(String ProductsNo){
143         boolean flag = false;
144         Connection conn=DBManager.getConnection();

```

```

145         PreparedStatement stat=null;
146         ResultSet rs=null;
147         try {
148             stat=conn.prepareStatement("select * from
products where productno = ?");
149             stat.setString(1, ProductsNo);
150             rs=stat.executeQuery();
151             if(rs.next()){
152                 JOptionPane.showMessageDialog(null,"该商品编号
已经存在","错误提示",JOptionPane.ERROR_MESSAGE);
153             }else{
154                 flag = true;
155             }
156         } catch (SQLException e) {
157             e.printStackTrace();
158         }finally{
159             DBManager.closeAll(conn, stat, rs);
160         }
161         return flag;
162     }
163 }
164

```

## 5.SaleHistoryDao

```

1 package dao;
2
3 import entity.SaleHistory;
4 /*
5  *
6  *
7  * */
8 public class SaleHistoryDao {
9     public boolean newSaleHistoryDao(SaleHistory saleHistory){
10         boolean flag = false;
11         if (saleHistory.getCustomerNo()==0) {
12             System.out.println("这个人的信息没必要插入");
13             return flag;
14         }
15         else{
16             String sql = "insert into
salehistory(productno,salecount,price,discount,userID,saletime
,customerNo) values(?,?,?,?,?,?,?)";

```

```

17         Object[] objs =
    {saleHistory.getProductNo(),saleHistory.getSaleCount(),saleHis
    tory.getPrice(),saleHistory.getDiscount()
18
    ,saleHistory.getUserId(),saleHistory.getSaleTime(),saleHistory
    .getCustomerNo()};
19         flag = DBManager.executeUpdate(sql, objs);
20         System.out.println("插入在当前收银台的这个会员的消费信
    息");
21         return flag;
22     }
23 }
24 }
25

```

## 6.ScoreDiscountDao

```

1  package dao;
2
3  import java.sql.Connection;
4  import java.sql.PreparedStatement;
5  import java.sql.ResultSet;
6  import java.sql.SQLException;
7  import java.util.ArrayList;
8  import java.util.List;
9
10 import entity.ScoreDiscount;
11
12 public class ScoreDiscountDao {
13     public List<ScoreDiscount> getAllScoreDiscountDao(){
14         List<ScoreDiscount> list=new ArrayList<ScoreDiscount>
    ();
15         Connection conn=DBManager.getConnection();
16         PreparedStatement stat=null;
17         ResultSet rs=null;
18         try {
19             stat=conn.prepareStatement("select distinct
    c.score,p.discount from customers c, products p");
20             rs=stat.executeQuery();
21             while(rs.next()){
22                 ScoreDiscount sc = new ScoreDiscount();
23                 sc.setMinScore(rs.getInt(1));
24                 sc.setDiscount(rs.getFloat(2));
25                 list.add(sc);

```

```

26         }
27     } catch (SQLException e) {
28         e.printStackTrace();
29     }finally{
30         DBManager.closeAll(conn, stat, rs);
31     }
32     return list;
33 }
34 }

```

## 7.UsersDao

```

1  package dao;
2
3  import java.sql.Connection;
4  import java.sql.PreparedStatement;
5  import java.sql.ResultSet;
6  import java.sql.SQLException;
7  import java.util.ArrayList;
8  import java.util.List;
9
10 import javax.swing.JOptionPane;
11
12 import service.CategorieService;
13
14 import entity.ProductInfo;
15 import entity.Users;
16 /*
17  * UsersDao类
18  * 连接数据库获取返回的对象
19  * 将各个数据赋值给变量
20  * */
21 public class UsersDao {
22     public List<Users> getAllUsersDao(){
23         List<Users> list=new ArrayList<Users>();//n个用户的数据
24         Connection conn=DBManager.getConnection();//连接数据库
25         PreparedStatement stat=null;//PreparedStatement 接口继承
26         承了Statement
27         ResultSet rs=null;//返回的对象
28         try {
29             stat=conn.prepareStatement("select * from
30             users");
31             rs=stat.executeQuery();//用statement类的
32             executeQuery()方法来下达select指令以查询数据库,

```

```

30         while(rs.next()){
31             Users user=new Users();
32             user.setUserId(rs.getInt(1));//返回的第一个数据
33             user.setUserName(rs.getString(2));//返回的第二
   个数据
34             user.setPassword(rs.getString(3));//返回的第三
   个数据
35             user.setRoleId(rs.getInt(4));
36             list.add(user);
37         }
38     } catch (SQLException e) {
39         e.printStackTrace();
40     }finally{
41         DBManager.closeAll(conn, stat, rs);
42     }
43     return list;
44 }
45 /*
46  * checkUsersDao判断用户信息是否正确
47  * 判断name password 角色是否对应
48  * */
49     public boolean checkUsersDao(String UserName,String
   Pwd,int roleId){
50         boolean flag = false;
51         if(UserName.isEmpty()){
52             return flag;
53         }else if(Pwd.isEmpty()){
54             return flag;
55         }else{
56             UsersDao user = new UsersDao();
57             List<Users> l =user.getAllUsersDao();
58             boolean tag = false;
59             for (Users a : l) {
60                 if(UserName.equals(a.getUserName()) &&
   Pwd.equals(a.getPassword()) && roleId == a.getRoleId()){
61                     tag = true;
62                     flag = true;
63                 }
64             }
65             if(!tag){
66                 return flag;
67             }
68         }
69         return flag;
70     }

```

```

71      /*
72      * 新建用户
73      *
74      * */
75
76
77      public boolean newUsersDao(Users user){
78          boolean flag = false;
79          String sql = "insert into users values(?,?,?,?)";
80          Object[] objs =
{user.getUserId(),user.getUserName(),user.getPassword(),user.
getRoleId()};
81          flag = DBManager.executeUpdate(sql, objs);
82          return flag;
83      }
84      public boolean deleteUsersDao(int userId){
85          boolean flag = false;
86          String sql = "delete from users where userId = ?";
87          Object[] objs = {userId};
88          flag = DBManager.executeUpdate(sql, objs);
89          return flag;
90      }
91      public int getUserIdByUserName(String userName){
92          int userId = 1000;
93          Connection conn=DBManager.getConnection();
94          PreparedStatement stat=null;
95          ResultSet rs=null;
96          try {
97              stat=conn.prepareStatement("select userId from
users where username = ?");
98              stat.setString(1, userName);
99              rs=stat.executeQuery();
100             if(rs.next()){
101                 userId = rs.getInt(1);
102             }else{
103                 JOptionPane.showMessageDialog(null,"查询的管理
员信息不存在","错误提示",JOptionPane.ERROR_MESSAGE);
104             }
105         } catch (SQLException e) {
106             e.printStackTrace();
107         }finally{
108             DBManager.closeAll(conn, stat, rs);
109         }
110         return userId;
111     }

```

```

112     public Users getUsersByUserId(int userId){
113         Users user = new Users();
114         Connection conn=DBManager.getConnection();
115         PreparedStatement stat=null;
116         ResultSet rs=null;
117         try {
118             stat=conn.prepareStatement("select * from users
119 where userId =?");
120             stat.setInt(1, userId);
121             rs=stat.executeQuery();
122             if(rs.next()){
123                 user.setUserId(rs.getInt(1));
124                 user.setUserName(rs.getString(2));
125                 user.setPassword(rs.getString(3));
126                 user.setRoleId(rs.getInt(4));
127             }else{
128                 JOptionPane.showMessageDialog(null,"查询的管理
129 员信息不存在","错误提示",JOptionPane.ERROR_MESSAGE);
130             }
131         } catch (SQLException e) {
132             e.printStackTrace();
133         }finally{
134             DBManager.closeAll(conn, stat, rs);
135         }
136         return user;
137     }
138     public boolean updateUserPasswordDao(int userId,String
139 password){
140         boolean flag = false;
141         String sql = "update users set password = ? where
142 userId = ?";
143         Object[] objs = {password,userId};
144         flag = DBManager.executeUpdate(sql, objs);
145         return flag;
146     }
147 }

```

## 3.service



# 1. CategorieService

```
1 package service;
2
3 import dao.CategorieDao;
4
5 public class CategorieService {
6     public String getCategoryNameService(String categoryId){
7         CategorieDao cd = new CategorieDao();
8         return cd.getCategoryNameDao(categoryId);
9     }
10 }
11
```

# 2. CustomerService

```
1 package service;
2
3 import java.util.List;
4
5 import dao.CustomerDao;
6
7 import entity.Customers;
8
9 public class CustomersService {
10     public List<Customers> getAllCustomersService(){
11         CustomerDao cd = new CustomerDao();
12         return cd.getAllCustomersDao();
13     }
14     public int newCustomersService(Customers customer){
15         CustomerDao cd = new CustomerDao();
16         return cd.newCustomersDao(customer);
17     }
18     public boolean deleteCustomersService(String cstNo){
19         CustomerDao cd = new CustomerDao();
20         return cd.deleteCustomersDao(cstNo);
21     }
22     public boolean checkCustomerDao(int customerNo){
23         CustomerDao cd = new CustomerDao();
24         return cd.checkCustomerDao(customerNo);
25     }
26     public double getMemberDiscountByCustomerNoService(int
customerNo){
27         CustomerDao cd = new CustomerDao();
```

```

28         return
cd.getMemberDiscountByCustomerNoDao(customerNo);
29     }
30     public boolean updateCustomerScoreDao(int customerNo,int
score){
31         CustomerDao cd = new CustomerDao();
32         return cd.updateCustomerScoreDao(customerNo, score);
33     }
34     public boolean checkCustomerNoDao(String customerNo){
35         CustomerDao cd = new CustomerDao();
36         return cd.checkCustomerNoDao(customerNo);
37     }
38 }
39

```

### 3.ProductsService

```

1  package service;
2
3  import java.util.List;
4
5  import dao.ProductsDao;
6
7  import entity.ProductInfo;
8  import entity.Products;
9
10 public class ProductsService {
11     public List<ProductInfo> getAllProductsService(){
12         ProductsDao pd = new ProductsDao();
13         return pd.getAllProductsDao();
14     }
15     public ProductInfo getProductInfoByProductNoService(String
ProductNo){
16         ProductsDao pd = new ProductsDao();
17         return pd.getProductInfoByProductNo(ProductNo);
18     }
19
20     public ProductInfo
getProductInfoByproductNameService(String productName){
21         ProductsDao pd = new ProductsDao();
22         return pd.getProductInfoByproductName(productName);
23     }
24     public boolean newProductsService(Products product){
25         ProductsDao pd = new ProductsDao();

```

```

26         return pd.newProductsDao(product);
27     }
28     public boolean deleteProductsService(String productNo){
29         ProductsDao pd = new ProductsDao();
30         return pd.deleteProductsDao(productNo);
31     }
32     public boolean updateProductsService(String productNo,int
count,double discount){
33         ProductsDao pd = new ProductsDao();
34         return pd.updateProductsDao(productNo, count,
discount);
35     }
36     public boolean updateProductsCountService(String
productNo,int soldNumber){
37         ProductsDao pd = new ProductsDao();
38         return pd.updateProductsCountDao(productNo,
soldNumber);
39     }
40     public boolean checkProductsNoDao(String ProductsNo){
41         ProductsDao pd = new ProductsDao();
42         return pd.checkProductsNoDao(ProductsNo);
43     }
44 }
45

```

## 4.SaleHistoryService

```

1  package service;
2
3  import dao.SaleHistoryDao;
4  import entity.SaleHistory;
5
6  public class SaleHistoryService {
7      public boolean newSaleHistoryService(SaleHistory
saleHistory){
8          saleHistoryDao shd = new SaleHistoryDao();
9          return shd.newSaleHistoryDao(saleHistory);
10     }
11 }
12

```

## 5.ScoreDiscountService

```
1 package service;
2
3 import java.util.List;
4
5 import dao.ScoreDiscountDao;
6
7 import entity.ScoreDiscount;
8
9 public class ScoreDiscountService {
10     public List<ScoreDiscount> getAllScoreDiscountService(){
11         ScoreDiscountDao sdd = new ScoreDiscountDao();
12         return sdd.getAllScoreDiscountDao();
13     }
14 }
15
```

## 6.UsersService

```
1 package service;
2
3 import java.util.List;
4
5 import dao.UsersDao;
6
7 import entity.Users;
8
9 public class UsersService {
10     public List<Users> getAllUsersService(){
11         UsersDao ud = new UsersDao();
12         return ud.getAllUsersDao();
13     }
14     public boolean checkUsersService(String UserName,String
15 Pwd,int roleId){
16         UsersDao ud = new UsersDao();
17         return ud.checkUsersDao(UserName, Pwd, roleId);
18     }
19     public boolean deleteUsersService(int userId){
20         UsersDao ud = new UsersDao();
21         return ud.deleteUsersDao(userId);
22     }
23 }
```

```

22     public boolean newUsersService(Users user){
23         UsersDao ud = new UsersDao();
24         return ud.newUsersDao(user);
25     }
26     public int getUserIdByUserNameService(String userName){
27         UsersDao ud = new UsersDao();
28         return ud.getUserIdByUserName(userName);
29     }
30     public Users getUsersByUserIdService(int userId){
31         UsersDao ud = new UsersDao();
32         return ud.getUsersByUserId(userId);
33     }
34     public boolean updateUserPasswordService(int userId,String
password){
35         UsersDao ud = new UsersDao();
36         return ud.updateUserPasswordDao(userId, password);
37     }
38 }
39

```

## 4.UI

### 1.AdministratorManagement

```

1  package UI;
2
3  import java.util.List;
4
5  import javax.swing.JFrame;
6  import javax.swing.JPanel;
7  import javax.swing.border.EmptyBorder;
8  import javax.swing.table.DefaultTableModel;
9  import javax.swing.JOptionPane;
10 import javax.swing.JScrollPane;
11 import javax.swing.JButton;
12 import javax.swing.JTable;
13
14 import entity.Users;
15
16 import service.UsersService;
17 import java.awt.event.ActionListener;
18 import java.awt.event.ActionEvent;
19
20 public class AdministratorManagement extends JFrame {

```

```

21
22     private JPanel contentPane;
23     private JTable table;
24
25     /**
26      * Launch the application.
27      */
28     public static void main(String[] args) {
29         AdministratorManagement frame = new
AdministratorManagement();
30         frame.setVisible(true);
31     }
32
33     /**
34      * Create the frame.
35      */
36     public AdministratorManagement() {
37         this.setTitle("管理员信息管理");
38         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
39         setBounds(500, 200, 680, 390);
40         contentPane = new JPanel();
41         contentPane.setBorder(new EmptyBorder(40, 40, 40,
42 40));
43         setContentPane(contentPane);
44         contentPane.setLayout(null);
45
46         JScrollPane scrollPane = new JScrollPane();
47         scrollPane.setBounds(50, 85, 520, 192);
48         contentPane.add(scrollPane);
49
50         String[] cols={"员工编号","用户名","密码","管理员级别"}; //
建立一个数组
51         final DefaultTableModel model = new
DefaultTableModel(cols,0); //建立一个有cols数组长度的表
52         table = new JTable(model);
53         UsersService us = new UsersService();
54         List<Users> rows = us.getAllUsersService();
55         for (Users users : rows) {
56             Object[] row =
{users.getUserId(),users.getUserName(),users.getPassword(),us
ers.getRoleId()};
57             model.addRow(row);
58         }
59         scrollPane.setViewportView(table);

```

```

60         JButton btnNewButton = new JButton("新增");
61         btnNewButton.addActionListener(new ActionListener() {
62             public void actionPerformed(ActionEvent e) {
63                 AdministratorManagement.this.dispose();
64                 NewAdministrator na = new NewAdministrator();
65                 na.setVisible(true);
66             }
67         });
68         btnNewButton.setBounds(307, 23, 103, 33);
69         contentPane.add(btnNewButton);
70
71         JButton button_1 = new JButton("删除");
72         button_1.addActionListener(new ActionListener() {
73             public void actionPerformed(ActionEvent e) {
74                 int rowIndex = table.getSelectedRow();
75                 if(rowIndex<0){
76
77                     JOptionPane.showMessageDialog(AdministratorManagement.this,
78 "请选择要删除的行", "错误提示", JOptionPane.ERROR_MESSAGE);
79                 }else{
80                     int userId = (Integer)
81 model.getValueAt(rowIndex, 0);
82
83                     if(JOptionPane.showConfirmDialog(AdministratorManagement.this,
84 "确定删除第"+(rowIndex+1)+"行?")==0){
85                         UsersService us = new UsersService();
86                         if(us.deleteUsersService(userId)){
87
88                             JOptionPane.showMessageDialog(AdministratorManagement.this,
89 "删除成功!");
90
91                             AdministratorManagement.this.dispose();
92                             AdministratorManagement frame =
93 new AdministratorManagement();
94                             frame.setVisible(true);
95                         }
96                     }
97                 }
98             }
99         });
100         button_1.setBounds(451, 23, 103, 33);
101         contentPane.add(button_1);
102
103         JButton btnNewButton_1 = new JButton("返回主管理员界
面");

```

```

95         btnNewButton_1.addActionListener(new ActionListener()
96         {
97             public void actionPerformed(ActionEvent e) {
98                 AdministratorManagement.this.dispose();
99                 RootAdministrator ra = new
100 RootAdministrator();
101                 ra.setVisible(true);
102             }
103         });
104         btnNewButton_1.setBounds(469, 297, 155, 43);
105         contentPane.add(btnNewButton_1);
106     }
107 }

```

## 2.Cashier

```

1  package UI;
2
3  import java.awt.event.ActionEvent;
4  import java.awt.event.ActionListener;
5
6  import javax.swing.JFrame;
7  import javax.swing.JPanel;
8  import javax.swing.border.EmptyBorder;
9  import javax.swing.JMenuBar;
10 import javax.swing.JMenu;
11 import javax.swing.JMenuItem;
12 import javax.swing.JOptionPane;
13 import javax.swing.SwingConstants;
14 import javax.swing.JLabel;
15
16 import service.CustomerService;
17 import service.UsersService;
18
19 import entity.Users;
20 import javax.swing.JTextField;
21 import javax.swing.JButton;
22 import java.awt.Font;
23
24 public class Cashier extends JFrame {
25
26     private JPanel contentPane;

```



```

27     private JTextField userName;
28     private JTextField textField;
29     private JTextField textField_1;
30     private JTextField textField_2;
31     private JTextField textField_3;
32
33     /**
34      * Launch the application.
35      */
36     public static void main(String[] args) {
37
38         Cashier frame = new Cashier(1003);
39         frame.setVisible(true);
40     }
41
42     /**
43      * Create the frame.
44      */
45     public Cashier(final int userId) {
46         this.setTitle("收银员界面");
47         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
48         setBounds(400, 200, 523, 370);
49
50         JMenuBar menuBar = new JMenuBar(); // 菜单栏
51         setJMenuBar(menuBar);
52
53         JMenu mnNewMenu = new JMenu("收银员信息管理"); // 构造一个新 JMenu，用提供的字符串作为其文本。
54         // mnNewMenu.setMnemonic('M'); // 键盘助记符
55
56         mnNewMenu.setHorizontalAlignment(SwingConstants.LEFT); // 设置对齐方式
57
58         menuBar.add(mnNewMenu);
59
60         contentPane = new JPanel();
61         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
62         setContentPane(contentPane);
63         contentPane.setLayout(null);
64
65         JLabel cashierId = new JLabel("New label");
66         cashierId.setText("当前操作收银员编号:" + userId);
67         cashierId.setBounds(55, 267, 155, 15);
68         contentPane.add(cashierId);

```

```

69         final JButton button = new JButton("会员收银");
70         button.setFont(new Font("微软雅黑 Light", Font.PLAIN,
21));
71         button.setBounds(72, 91, 150, 88);
72         contentPane.add(button);
73
74         final JButton button_1 = new JButton("非会员收银");
75         button_1.setFont(new Font("微软雅黑 Light",
Font.PLAIN, 21));
76         button_1.setBounds(264, 91, 155, 88);
77         contentPane.add(button_1);
78
79
80         JMenuItem jmi1 = new JMenuItem("修改密码");
81         // jmi1.setMnemonic('C');
82         jmi1.setHorizontalAlignment(SwingConstants.CENTER);
83         jmi1.addActionListener(new ActionListener() {
84             @Override
85             public void actionPerformed(ActionEvent e) {
86                 button.setVisible(false);
87                 button_1.setVisible(false);
88                 UsersService us = new UsersService();
89                 final Users changeUser =
us.getUsersByUserIdService(userId);
90                 //显示修改密码界面
91                 {
92                     JLabel labelun = new JLabel("用户名:");
93                     labelun.setBounds(88, 40, 54, 15);
94                     contentPane.add(labelun);
95
96                     userName = new JTextField();
97
98                     userName.setText(changeUser.getUserName());
99                     userName.setEditable(false);
100                    userName.setBounds(206, 37, 134, 21);
101                    contentPane.add(userName);
102                    userName.setColumns(10);
103
104                    JLabel label = new JLabel("请输入原密码:");
105                    label.setBounds(88, 81, 94, 15);
106                    contentPane.add(label);
107
108                    textField = new JTextField();
109                    textField.setColumns(10);
110                    textField.setBounds(206, 78, 134, 21);

```

```
110         contentPane.add(textField);
111
112         JLabel label_1 = new JLabel("请输入新密
码:");
113         label_1.setBounds(88, 122, 94, 15);
114         contentPane.add(label_1);
115
116         textField_1 = new JTextField();
117         textField_1.setColumns(10);
118         textField_1.setBounds(206, 119, 134, 21);
119         contentPane.add(textField_1);
120
121         JLabel label_2 = new JLabel("请确认新密
码:");
122         label_2.setBounds(88, 169, 94, 15);
123         contentPane.add(label_2);
124
125         textField_2 = new JTextField();
126         textField_2.setColumns(10);
127         textField_2.setBounds(206, 166, 134, 21);
128         contentPane.add(textField_2);
129
130         JButton btnNewButton = new JButton("确认修
改");
131         btnNewButton.addActionListener(new
ActionListener() {
132             @Override
133             public void
actionPerformed(ActionEvent e) {
134                 String psd = textField.getText();
135                 String newPsd =
textField_1.getText();
136                 String newPsdC =
textField_2.getText();
137
138                 if(changeUser.getPassword().equals(psd)){
139                     if(newPsd.equals(newPsdC)){
140                         UsersService us1 = new
UsersService();
141
142                         if(us1.updateUserPasswordService(userId, newPsdC)){
143                             JOptionPane.showMessageDialog(Cashier.this, "修改成功!");
144                             Cashier.this.dispose();
145                         }
146                     }
147                 }
148             }
149         });
150     }
151 }
```

```

143                                     Login frame = new
Login();
144
    frame.setVisible(true);
145                                     }else{
146
    JOptionPane.showMessageDialog(Cashier.this, "修改失败!");
147                                     }
148                                     }else{
149
    JOptionPane.showMessageDialog(Cashier.this, "两次输入不一
致!", "错误提示", JOptionPane.ERROR_MESSAGE);
150                                     }
151                                     }else{
152
    JOptionPane.showMessageDialog(Cashier.this, "原始密码错
误!", "错误提示", JOptionPane.ERROR_MESSAGE);
153                                     }
154                                     }
155                                     });
156                                     btnNewButton.setBounds(117, 216, 93, 23);
157                                     contentPane.add(btnNewButton);
158
159                                     JButton btnNewButton_1 = new JButton("取
消");
160                                     btnNewButton_1.addActionListener(new
ActionListener() {
161                                     @Override
162                                     public void
actionPerformed(ActionEvent e) {
163                                     Cashier.this.dispose();
164                                     Cashier frame = new
Cashier(userId);
165                                     frame.setVisible(true);
166
167                                     }
168                                     });
169                                     btnNewButton_1.setBounds(235, 216, 93,
23);
170                                     contentPane.add(btnNewButton_1);
171                                     //刷新页面
172                                     Cashier.this.repaint();
173                                     }
174
175

```

```

176
177         }
178     });
179     mnNewMenu.add(jmi1);
180     JMenuItem Cancellation= new JMenuItem("注销");
181     // Cancellation.setMnemonic('w');
182
183     Cancellation.setHorizontalAlignment(SwingConstants.CENTER);
184     Cancellation.addActionListener(new ActionListener() {
185         @Override
186         public void actionPerformed(ActionEvent e) {
187             Cashier.this.dispose();
188             Login l = new Login();
189             l.setVisible(true);
190         }
191     });
192     mnNewMenu.add(Cancellation);
193
194     JMenuItem exit = new JMenuItem("退出");
195     // exit.setMnemonic('R');
196     exit.setHorizontalAlignment(SwingConstants.CENTER);
197     exit.addActionListener(new ActionListener() {
198         @Override
199         public void actionPerformed(ActionEvent e) {
200             Cashier.this.dispose();
201         }
202     });
203     mnNewMenu.add(exit);
204
205     //会员收银
206     button.addActionListener(new ActionListener() {
207         public void actionPerformed(ActionEvent e) {
208             button.setVisible(false);
209             button_1.setVisible(false);
210             JLabel lblNewLabel = new JLabel("请输入会员卡卡
211 号");
212             lblNewLabel.setBounds(101, 86, 109, 15);
213             contentPane.add(lblNewLabel);
214
215             textField_3 = new JTextField();
216             textField_3.setBounds(220, 83, 155, 21);
217             contentPane.add(textField_3);
218             textField_3.setColumns(10);

```

```

219
220         JButton btnNewButton_2 = new JButton("收银");
221         btnNewButton_2.addActionListener(new
ActionListener() {
222             public void actionPerformed(ActionEvent
e) {
223                 String customerNo1 =
textField_3.getText();
224                 int customerNo = 0;
225                 try {
226                     customerNo =
Integer.parseInt(customerNo1);
227                 } catch (NumberFormatException e1) {
228                     e1.printStackTrace();
229                 }
230                 CustomerService cs = new
CustomerService();
231                 if(cs.checkCustomerDao(customerNo)){
232                     Cashier.this.dispose();
233                     CollectMoney cm = new
CollectMoney(userId, customerNo);
234                     //System.out.println("cm"+userId);
235                     cm.setVisible(true);
236                 }
237             }
238         });
239         btnNewButton_2.setBounds(127, 190, 93, 23);
240         contentPane.add(btnNewButton_2);
241
242         JButton button = new JButton("返回主窗口");
243         button.addActionListener(new ActionListener()
{
244             public void actionPerformed(ActionEvent
e) {
245                 Cashier.this.dispose();
246                 Cashier c = new Cashier(userId);
247                 c.setVisible(true);
248             }
249         });
250         button.setBounds(263, 190, 120, 23);
251         contentPane.add(button);
252         Cashier.this.repaint();
253
254     }

```

```

255     });
256     //非会员收银
257     button_1.addActionListener(new ActionListener() {
258         public void actionPerformed(ActionEvent e) {
259             Cashier.this.dispose();
260             CollectMoney cm = new CollectMoney(userId,0);
261             cm.setVisible(true);
262         }
263     });
264     JButton btnNewButton_2 = new JButton("返回");
265     btnNewButton_2.addActionListener(new ActionListener()
266     {
267         public void actionPerformed(ActionEvent e) {
268             Cashier.this.dispose();
269             Login am = new Login();
270             am.setVisible(true);
271         }
272     });
273     btnNewButton_2.setBounds(361, 254,113, 43);
274     contentPane.add(btnNewButton_2);
275 }
276 }
277

```

### 3.CollectMoney

```

1  package UI;
2
3  import java.awt.EventQueue;
4
5  import javax.swing.JFrame;
6  import javax.swing.JLabel;
7  import javax.swing.JPanel;
8  import javax.swing.border.EmptyBorder;
9  import javax.swing.table.DefaultTableModel;
10 import javax.swing.JOptionPane;
11 import javax.swing.JTable;
12 import javax.swing.JScrollPane;
13 import javax.swing.JTextField;
14 import javax.swing.JButton;
15
16 import service.CustomerService;
17 import service.ProductsService;

```

```
18
19 import entity.ProductInfo;
20 import entity.SaleHistory;
21 import entity.SaleSummary;
22 import service.SaleHistoryService;
23
24 import java.awt.event.ActionListener;
25 import java.awt.event.ActionEvent;
26 import java.text.DateFormat;
27 import java.text.DecimalFormat;
28 import java.text.SimpleDateFormat;
29 import java.util.Calendar;
30
31 /*
32  * 结算界面
33  * */
34 public class CollectMoney extends JFrame {
35
36     private JPanel contentPane;
37     private double totalAmount = 0;
38     private JTable table;
39     private JTextField proNo;
40     private JTextField number;
41     private SaleSummary[] ss = new SaleSummary[20];
42
43     /**
44      * Launch the application.
45      */
46     public static void main(String[] args) {
47         CollectMoney frame = new CollectMoney(1003, 0);
48         frame.setVisible(true);
49     }
50
51     /**
52      * Create the frame.
53      */
54     public CollectMoney(final int userId, final int
customerNo) {
55         this.setTitle("收银界面");
56         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
57         setBounds(400, 200, 599, 383);
58         contentPane = new JPanel();
59         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
60         setContentPane(contentPane);
61         contentPane.setLayout(null);
```



```
62
63         JLabel lblNewLabel = new JLabel("当前操作收银员编号:" +
        userId);
64         lblNewLabel.setBounds(27, 313, 159, 21);
65         contentPane.add(lblNewLabel);
66
67         final JLabel lblNewLabel_1 = new JLabel("购物总价:" +
        totalAmount);
68         lblNewLabel_1.setBounds(406, 316, 106, 15);
69         contentPane.add(lblNewLabel_1);
70
71         JScrollPane scrollPane = new JScrollPane();
72         scrollPane.setBounds(10, 66, 563, 237);
73         contentPane.add(scrollPane);
74
75         String[] cols = { "产品编号", "数量", "单价", "商品折扣",
        "收银员编号", "销售时间", "会员编号",
76             "会员折扣", "总价格" };
77         final DefaultTableModel model = new
        DefaultTableModel(cols, 0);
78         table = new JTable(model);
79         scrollPane.setViewportView(table);
80
81         JLabel lblNewLabel_2 = new JLabel("商品编号:");
82         lblNewLabel_2.setBounds(10, 10, 70, 15);
83         contentPane.add(lblNewLabel_2);
84
85         proNo = new JTextField();
86         proNo.setBounds(90, 7, 89, 21);
87         contentPane.add(proNo);
88         proNo.setColumns(10);
89
90         JLabel lblNewLabel_3 = new JLabel("会员编号:" +
        customerNo);
91         lblNewLabel_3.setBounds(242, 316, 106, 15);
92         contentPane.add(lblNewLabel_3);
93
94         JLabel lblNewLabel_4 = new JLabel("数量:");
95         lblNewLabel_4.setBounds(10, 41, 54, 15);
96         contentPane.add(lblNewLabel_4);
97
98         number = new JTextField();
99         number.setColumns(10);
100        number.setBounds(90, 38, 89, 21);
101        contentPane.add(number);
```



```

136         CustomerService cs = new
CustomerService();
137         //保留两位小数
138         DecimalFormat df = new
DecimalFormat("#.00");
139         /*
140         * 以下两行是判断会员折扣的
141         * */
142
143         s.setMemberDiscount(Double.valueOf(df.format(cs.getMemberDis
countByCustomerNoService(customerNo))));
144         double amount = pi.getPrice() *
num * pi.getDiscount()*s.getMemberDiscount();
145         //
146         System.out.println("amount: "+
amount);
147
148         amount =
Double.valueOf(df.format(amount));
149         s.setTotalAmount(amount);
150         // 添加到数组中
151         int i = 0;
152         for (; ss[i] != null; i++) {
153         }
154         ss[i] = s;
155         // 增加总金额
156         totalAmount += amount;
157         totalAmount =
Double.valueOf(df.format(totalAmount));
158         Object[] row = {
s.getProductNo(), s.getSaleCount(),
159         s.getPrice(),
s.getDiscount(), s.getUserId(), time,
160         customerNo,
s.getMemberDiscount(),
161         s.getTotalAmount() };
162         model.addRow(row);
163         proNo.setText("");
164         number.setText("");
165         lblNewLabel_1.setText("购物总价:"
+ totalAmount);
166         }} else if(pi.getCount() != -1){
167         JOptionPane.showMessageDialog(null,
"库存不足!剩余库存为" +
pi.getCount(), "错误提示",

```

```

168         JOptionPane.ERROR_MESSAGE);
169     }
170 }
171 }
172 });
173 btnNewButton.setBounds(223, 6, 93, 23);
174 contentPane.add(btnNewButton);
175
176 JButton button = new JButton("删除");
177 button.addActionListener(new ActionListener() {
178     public void actionPerformed(ActionEvent e) {
179         int rowIndex = table.getSelectedRow();
180         if(rowIndex<0){
181
182             JOptionPane.showMessageDialog(CollectMoney.this, "请选择要删除
183             的行", "错误提示", JOptionPane.ERROR_MESSAGE);
184         }else{
185
186             if(JOptionPane.showConfirmDialog(CollectMoney.this, "确定删除
187             第"+(rowIndex+1)+"行?")==0){
188                 int i = rowIndex;
189                 model.removeRow(i);
190                 double ramount =
191                 ss[i].getPrice()*ss[i].getSaleCount()*ss[i].getDiscount()*ss[
192                 i].getMemberDiscount();
193                 totalAmount -= ramount;
194                 lblNewLabel_1.setText("购物总价:" +
195                 totalAmount);
196                 for(;ss[i]!=null;i++){
197                     ss[i] = ss[i+1];
198                 }
199                 ss[i] = null;
200             }
201         }
202     }
203 });
204 button.setBounds(223, 33, 93, 23);
205 contentPane.add(button);
206
207 JButton btnNewButton_1 = new JButton("清空");
208 btnNewButton_1.addActionListener(new ActionListener()
209 {
210     public void actionPerformed(ActionEvent e) {
211         model.setRowCount(0);
212         ss = new SalesSummary[20];

```

```

205         totalAmount = 0;
206         lblNewLabel_1.setText("购物总价:" +
totalAmount);
207     }
208 });
209 btnNewButton_1.setBounds(338, 6, 93, 23);
210 contentPane.add(btnNewButton_1);
211
212 JButton btnNewButton_2 = new JButton("结算");
213 btnNewButton_2.addActionListener(new ActionListener()
{
214     public void actionPerformed(ActionEvent e) {
215         for(int i = 0;ss[i]!=null;i++){
216             SaleHistory sh = new SaleHistory();
217             sh.setProductNo(ss[i].getProductNo());
218             sh.setSaleCount(ss[i].getSaleCount());
219             sh.setPrice(ss[i].getPrice());
220             sh.setDiscount(ss[i].getDiscount());
221             sh.setUserId(ss[i].getUserId());
222             sh.setSaleTime(ss[i].getSaleTime());
223             sh.setCustomerNo(ss[i].getCustomerNo());
224             SaleHistoryService shs = new
SaleHistoryService();
225             shs.newSaleHistoryService(sh);
226             ProductsService ps = new
ProductsService();
227
228             ps.updateProductsCountService(ss[i].getProductNo(),
ss[i].getSaleCount());
229             CollectMoney.this.dispose();
230
231             if(customerNo!=0){
232                 System.out.println("新增积分");
233                 CustomerService cs = new
CustomerService();
234                 cs.updateCustomerScoreDao(customerNo,
(int)(totalAmount));
235             }
236
237             System.out.println("算账");
238             GiveChange gc = new
GiveChange(totalAmount,userId,customerNo);
239             gc.setVisible(true);
240

```

```

241         }
242     });
243     btnNewButton_2.setBounds(454, 7, 106, 49);
244     contentPane.add(btnNewButton_2);
245
246     JButton btnNewButton_3 = new JButton("返回");
247     btnNewButton_3.addActionListener(new ActionListener()
248     {
249         public void actionPerformed(ActionEvent e) {
250             CollectMoney.this.dispose();
251             Cashier c = new Cashier(userId);
252             c.setVisible(true);
253         }
254     });
255     btnNewButton_3.setBounds(338, 33, 93, 23);
256     contentPane.add(btnNewButton_3);
257 }
258 }
259

```

## 4.CustomerManagement

```

1  package UI;
2
3  import java.util.List;
4
5  import javax.swing.JFrame;
6  import javax.swing.JPanel;
7  import javax.swing.border.EmptyBorder;
8  import javax.swing.table.DefaultTableModel;
9  import javax.swing.JOptionPane;
10 import javax.swing.JScrollPane;
11 import javax.swing.JTable;
12 import javax.swing.JButton;
13
14 import entity.Customers;
15
16 import service.CustomerService;
17 import java.awt.event.ActionListener;
18 import java.awt.event.ActionEvent;
19
20 public class CustomerManagement extends JFrame {

```

```
21
22     private JPanel contentPane;
23     private JTable table;
24
25     /**
26      * Launch the application.
27      */
28     public static void main(String[] args) {
29         CustomerManagement frame = new CustomerManagement();
30         frame.setVisible(true);
31     }
32
33     /**
34      * Create the frame.
35      */
36     public CustomerManagement() {
37         this.setTitle("会员信息管理");
38         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
39         setBounds(400, 200, 450, 340);
40         contentPane = new JPanel();
41         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
42         setContentPane(contentPane);
43         contentPane.setLayout(null);
44
45         JScrollPane scrollPane = new JScrollPane();
46         scrollPane.setBounds(10, 58, 414, 165);
47         contentPane.add(scrollPane);
48
49         String[] cols={"会员编号","会员用户名","会员联系方式","会员
积分"};
50         final DefaultTableModel model = new
DefaultTableModel(cols,0);
51         table = new JTable(model);
52         CustomerService cs = new CustomerService();
53         List<Customers> rows = cs.getAllCustomersService();
54         for (Customers customers : rows) {
55             Object[] row =
{customers.getCustomerNo(),customers.getCustomerName(),custom
ers.getPhone(),customers.getScore()};
56             model.addRow(row);
57         }
58         scrollPane.setViewportView(table);
59
60         JButton button = new JButton("新增");
61         button.addActionListener(new ActionListener() {
```

```

62         public void actionPerformed(ActionEvent e) {
63             CustomerManagement.this.dispose();
64             NewCustomer nc = new NewCustomer();
65             nc.setVisible(true);
66         }
67     });
68     button.setBounds(214, 25, 93, 23);
69     contentPane.add(button);
70
71     JButton button_1 = new JButton("删除");
72     button_1.addActionListener(new ActionListener() {
73         public void actionPerformed(ActionEvent e) {
74             int rowIndex = table.getSelectedRow();
75             if(rowIndex<0){
76
77                 JOptionPane.showMessageDialog(CustomerManagement.this, "请选
78                 择要删除的行", "错误提示", JOptionPane.ERROR_MESSAGE);
79             }else{
80                 String cstNo =
81                 String.valueOf(model.getValueAt(rowIndex, 0));
82
83                 if(JOptionPane.showConfirmDialog(CustomerManagement.this,
84                 "确定删除第"+(rowIndex+1)+"行?")==0){
85                     CustomerService cs = new
86                     CustomerService();
87                     if(cs.deleteCustomersService(cstNo)){
88
89                         JOptionPane.showMessageDialog(CustomerManagement.this, "删除
90                         成功!");
91
92                         CustomerManagement.this.dispose();
93
94                         CustomerManagement frame = new
95                         CustomerManagement();
96
97                         frame.setVisible(true);
98                     }
99                 }
100             }
101         }
102     });
103     button_1.setBounds(331, 25, 93, 23);
104     contentPane.add(button_1);
105
106     JButton button_2 = new JButton("返回主界面");
107     button_2.addActionListener(new ActionListener() {
108         public void actionPerformed(ActionEvent e) {

```



```

97         CustomerManagement.this.dispose();
98         RootAdministrator ra = new
RootAdministrator();
99         ra.setVisible(true);
100     }
101 });
102 button_2.setBounds(279, 233, 145, 26);
103 contentPane.add(button_2);
104
105 }
106 }
107

```

## 5. GiveChange

```

1  package UI;
2
3  import java.awt.EventQueue;
4
5  import javax.swing.JFrame;
6  import javax.swing.JPanel;
7  import javax.swing.border.EmptyBorder;
8  import javax.swing.JLabel;
9  import javax.swing.JOptionPane;
10
11 import java.awt.Font;
12 import javax.swing.JTextField;
13 import javax.swing.JButton;
14 import java.awt.event.ActionListener;
15 import java.awt.event.ActionEvent;
16 import java.text.DecimalFormat;
17
18 public class GiveChange extends JFrame {
19
20     private JPanel contentPane;
21     private JTextField textField;
22     private double change = 0;
23
24     /**
25      * Launch the application.
26      */
27     public static void main(String[] args) {
28         // EventQueue.invokeLater(new Runnable() {

```

```

29 //          public void run() {
30 //              try {
31 //                  GiveChange frame = new
GiveChange(0,1003,0);
32 //                      frame.setVisible(true);
33 //              } catch (Exception e) {
34 //                  e.printStackTrace();
35 //              }
36 //          }
37 //      });
38         GiveChange frame = new GiveChange(0,1003,0);
39         frame.setVisible(true);
40     }
41
42     /**
43      * Create the frame.
44      * @param customerNo
45      */
46     public GiveChange(final double totalAmount, final int
userId, final int customerNo) {
47         this.setTitle("找零界面");
48         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
49         setBounds(400, 200, 450, 300);
50         contentPane = new JPanel();
51         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
52         setContentPane(contentPane);
53         contentPane.setLayout(null);
54
55         JLabel lblNewLabel = new JLabel("商品总金
额:"+totalAmount);
56         lblNewLabel.setFont(new Font("微软雅黑", Font.PLAIN,
23));
57         lblNewLabel.setBounds(56, 33, 294, 57);
58         contentPane.add(lblNewLabel);
59
60         JLabel lblNewLabel_1 = new JLabel("支付金额:");
61         lblNewLabel_1.setFont(new Font("微软雅黑", Font.PLAIN,
16));
62         lblNewLabel_1.setBounds(56, 100, 87, 32);
63         contentPane.add(lblNewLabel_1);
64
65         textField = new JTextField();
66         textField.setBounds(153, 107, 130, 21);
67         contentPane.add(textField);
68         textField.setColumns(10);

```

```

69
70     final JLabel label = new JLabel("找零金额:"+change);
71     label.setFont(new Font("微软雅黑", Font.PLAIN, 16));
72     label.setBounds(56, 183, 186, 32);
73     contentPane.add(label);
74
75     JButton btnNewButton = new JButton("确认支付");
76     btnNewButton.addActionListener(new ActionListener() {
77         public void actionPerformed(ActionEvent e) {
78             change = Double.valueOf(textField.getText())-
totalAmount;
79             DecimalFormat df = new DecimalFormat("#.00");
80             change = Double.valueOf(df.format(change));
81             if(change<0){
82                 JOptionPane.showMessageDialog(null,"付款不
足, 请支付"+totalAmount+"元","错误提
示",JOptionPane.ERROR_MESSAGE);}
83             else{
84                 label.setText("找零金额:"+change);
85             }
86
87
88         }
89     });
90     btnNewButton.setBounds(50, 150, 93, 23);
91     contentPane.add(btnNewButton);
92
93     JButton button = new JButton("返回");
94     button.addActionListener(new ActionListener() {
95         public void actionPerformed(ActionEvent e) {
96             GiveChange.this.dispose();
97             Cashier frame = new Cashier(userId);
98             frame.setVisible(true);
99         }
100     });
101     button.setBounds(173, 150, 93, 23);
102     contentPane.add(button);
103 }
104 }
105

```

## 6.GoodsAdministrator

```
1  package UI;
2
3  import java.awt.BorderLayout;
4  import java.awt.EventQueue;
5  import java.util.List;
6
7  import javax.swing.JFrame;
8  import javax.swing.JPanel;
9  import javax.swing.border.EmptyBorder;
10 import javax.swing.table.DefaultTableModel;
11 import javax.swing.JOptionPane;
12 import javax.swing.JScrollPane;
13 import javax.swing.JLabel;
14 import javax.swing.JTextField;
15 import javax.swing.JButton;
16 import javax.swing.JTable;
17
18 import entity.ProductInfo;
19
20 import service.ProductsService;
21 import java.awt.event.ActionListener;
22 import java.awt.event.ActionEvent;
23
24 public class GoodsAdministrator extends JFrame {
25
26     private JPanel contentPane;
27     private JTextField ProductNo;
28     private JTextField ProductName;
29     private JTable table;
30
31     /**
32      * Launch the application.
33      */
34     public static void main(String[] args) {
35
36         GoodsAdministrator frame = new GoodsAdministrator();
37         frame.setVisible(true);
38     }
39
40     /**
41      * Create the frame.
42      */
```

```
43     public GoodsAdministrator() {
44         this.setTitle("商品管理员界面");
45         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
46         setBounds(400, 200, 603, 377);
47         contentPane = new JPanel();
48         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
49         setContentPane(contentPane);
50         contentPane.setLayout(null);
51
52         JScrollPane scrollPane = new JScrollPane();
53         scrollPane.setBounds(10, 53, 570, 224);
54         contentPane.add(scrollPane);
55
56         String[] cols={"商品编号","商品类型","商品名称","规格","单
57         位","数量","单价","折扣"};
58         final DefaultTableModel model = new
59         DefaultTableModel(cols,0);
60         table = new JTable(model);
61         scrollPane.setViewportView(table);
62
63         JLabel lblNewLabel = new JLabel("商品编号:");
64         lblNewLabel.setBounds(36, 290, 70, 15);
65         contentPane.add(lblNewLabel);
66
67         ProductNo = new JTextField();
68         ProductNo.setBounds(116, 287, 102, 21);
69         contentPane.add(ProductNo);
70         ProductNo.setColumns(10);
71
72         JLabel label = new JLabel("商品名称:");
73         label.setBounds(241, 290, 68, 15);
74         contentPane.add(label);
75
76         ProductName = new JTextField();
77         ProductName.setColumns(10);
78         ProductName.setBounds(308, 287, 102, 21);
79         contentPane.add(ProductName);
80
81         JButton button = new JButton("查询指定商品");
82         button.addActionListener(new ActionListener() {
83             public void actionPerformed(ActionEvent e) {
84                 String productNo = "";
85                 String productName = "";
86                 productNo = ProductNo.getText();
```

```

86         productName = ProductName.getText();
87         ProductsService ps = new ProductsService();
88         ProductInfo productInfo = new ProductInfo();
89         if(!productNo.isEmpty()){
90             model.setRowCount(0);
91             productInfo =
ps.getProductInfoByProductNoService(productNo);
92             Object[] row =
{productInfo.getProductNo(),productInfo.getCategoryName(),pro
ductInfo.getProductNo(),productInfo.getType()
93
,productInfo.getUnit(),productInfo.getCount(),productInfo.get
Price(),productInfo.getDiscount()};
94             model.addRow(row);
95             }else if(!productName.isEmpty()){
96                 model.setRowCount(0);
97                 productInfo =
ps.getProductInfoByproductNameService(productName);
98                 Object[] row =
{productInfo.getProductNo(),productInfo.getCategoryName(),pro
ductInfo.getProductNo(),productInfo.getType()
99
,productInfo.getUnit(),productInfo.getCount(),productInfo.get
Price(),productInfo.getDiscount()};
100                 model.addRow(row);
101                 }else{
102
JOptionPane.showMessageDialog(GoodsAdministrator.this, "请输
入要查询的商品信息","错误提示",JOptionPane.ERROR_MESSAGE);
103                 }
104             }
105         });
106         button.setBounds(451, 287, 117, 23);
107         contentPane.add(button);
108
JButton btnNewButton = new JButton("增加新商品");
109         btnNewButton.addActionListener(new ActionListener() {
110             public void actionPerformed(ActionEvent e) {
111                 GoodsAdministrator.this.dispose();
112                 NewProducts np = new NewProducts();
113                 np.setVisible(true);
114             }
115         });
116         btnNewButton.setBounds(308, 10, 117, 23);
117         contentPane.add(btnNewButton);
118

```

```

119
120         JButton btnNewButton_1 = new JButton("删除商品信息");
121         btnNewButton_1.addActionListener(new ActionListener()
122         {
123             public void actionPerformed(ActionEvent e) {
124                 int rowIndex = table.getSelectedRow();
125                 if(rowIndex<0){
126
127                     JOptionPane.showMessageDialog(GoodsAdministrator.this, "请选
128                     择要删除的行", "错误提示", JOptionPane.ERROR_MESSAGE);
129                 }else{
130                     String productNo = (String)
131                     model.getValueAt(rowIndex, 0);
132
133                     if(JOptionPane.showConfirmDialog(GoodsAdministrator.this,
134                     "确定删除第"+(rowIndex+1)+"行?")==0){
135                         ProductsService ps = new
136                         ProductsService();
137
138                         if(ps.deleteProductsService(productNo)){
139
140                             JOptionPane.showMessageDialog(GoodsAdministrator.this, "删除
141                             成功!");
142
143                             GoodsAdministrator.this.dispose();
144                             GoodsAdministrator frame = new
145                             GoodsAdministrator();
146                             frame.setVisible(true);
147                         }
148                     }
149                 }
150             });
151         btnNewButton_1.setBounds(446, 10, 122, 23);
152         contentPane.add(btnNewButton_1);
153
154         JButton showAllProducts = new JButton("查询所有商品");
155         showAllProducts.addActionListener(new
156         ActionListener() {
157             public void actionPerformed(ActionEvent e) {
158                 model.setRowCount(0);
159                 ProductsService ps = new ProductsService();
160                 List<ProductInfo> rows =
161                 ps.getAllProductsService();
162                 for (ProductInfo productInfo : rows) {

```

```

150         Object[] row =
        {productInfo.getProductNo(),productInfo.getCategoryName(),pro
        ductInfo.getProductNo(),productInfo.getType()
151
        ,productInfo.getUnit(),productInfo.getCount(),productInfo.get
        Price(),productInfo.getDiscount()};
152         model.addRow(row);
153     }
154 }
155 });
156 showAllProducts.setBounds(27, 10, 117, 23);
157 contentPane.add(showAllProducts);
158
159 JButton button_1 = new JButton("更新库存信息");
160 button_1.addActionListener(new ActionListener() {
161     public void actionPerformed(ActionEvent e) {
162         int rowIndex = table.getSelectedRow();
163         if(rowIndex<0){
164
165             JOptionPane.showMessageDialog(GoodsAdministrator.this, "请选
166             择要更新的行","错误提示",JOptionPane.ERROR_MESSAGE);
167         }else{
168             Object[] row =
169             {model.getValueAt(rowIndex, 0),model.getValueAt(rowIndex,
170             2),model.getValueAt(rowIndex, 5),model.getValueAt(rowIndex,
171             7)};
172
173             GoodsAdministrator.this.dispose();
174             UpdateProducts up = new
175             UpdateProducts(row);
176             up.setVisible(true);
177         }
178     }
179 });
180 button_1.setBounds(172, 10, 126, 23);
181 contentPane.add(button_1);
182 JButton btnNewButton_6 = new JButton("返回");
183 btnNewButton_6.addActionListener(new ActionListener()
184 {
185     public void actionPerformed(ActionEvent e) {
186         GoodsAdministrator.this.dispose();
187         RootAdministrator am = new
188         RootAdministrator();
189         am.setVisible(true);
190     }
191 });

```



```

183         btnNewButton_6.setBounds(451, 317, 117, 23);
184         contentPane.add(btnNewButton_6);
185     }
186 }
187

```

## 7.Login

```

1  package UI;
2
3  import java.awt.*;
4  import javax.swing.JFrame;
5  import javax.swing.JPanel;
6  import javax.swing.JPasswordField;
7  import javax.swing.border.EmptyBorder;
8  import javax.swing.JLabel;
9  import javax.swing.JOptionPane;
10 import javax.swing.JTextField;
11 import javax.swing.JComboBox;
12 import javax.swing.DefaultComboBoxModel;
13 import javax.swing.ImageIcon;
14 import javax.swing.JButton;
15
16 import service.UserService;
17
18 import java.awt.event.ActionListener;
19 import java.awt.event.ActionEvent;
20 import java.awt.event.ItemEvent;
21 import java.awt.event.ItemListener;
22
23 public class Login extends JFrame {
24
25     private JPanel contentPane;//面板
26     private JTextField userName;
27     final JPasswordField pwd =new JPasswordField();//密码不可更
    改
28     private int roleId = 0;//角色
29
30     /**
31      * Launch the application.
32      */
33     public static void main(String[] args) {
34         Login frame = new Login();
35         frame.setVisible(true);

```

```

36
37     }
38
39     /**
40      * Create the frame.
41      */
42     public Login() {
43
44         this.setTitle("超市管理系统");//标题
45         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
46         setBounds(400, 200, 470, 400);
47         contentPane = new JPanel();
48         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));//
控制面板边界
49         setContentPane(contentPane);
50         contentPane.setLayout(null);
51         JLabel beijing = new JLabel();
52         /*beijing.setBounds(400, 200, 450, 300);*/
53         ImageIcon beijing1 = new ImageIcon("E:\\平时作业\\大二
上\\JAVA\\算法个人作业\\Final project\\src\\UI\\背景.jpg");
54         beijing.setBounds(0,0,520,436);
55         beijing.setIcon(new ImageIcon(beijing1.getImage()));
56
57         JLabel Title = new JLabel("超市管理系统");
58         Title.setBounds(125,15,400,30);
59         Title.setFont(new Font("黑体",Font.PLAIN,26));
60         contentPane.add(Title);
61
62         JLabel Jusername = new JLabel("员工名:");//标签
63         Jusername.setBounds(82, 98, 54, 15);
64         Jusername.setFont(new Font("宋体",Font.BOLD,14));
65         contentPane.add(Jusername);
66
67         userName = new JTextField();
68         userName.setBounds(186,95,141,21);
69         contentPane.add(userName);    //输入文本框
70
71         JLabel Jpwd = new JLabel("密码:");
72         Jpwd.setBounds(82, 144, 79, 15);
73         Jpwd.setFont(new Font("宋体",Font.BOLD,14));
74         contentPane.add(Jpwd);
75
76         pwd.setBounds(186,137,141,21);
77         pwd.setColumns(10);    //限制长度
78         contentPane.add(pwd);

```

```

79
80     JLabel label_1 = new JLabel("员工类别:");
81     label_1.setBounds(80, 186, 100, 15);
82     label_1.setFont(new Font("宋体", Font.BOLD, 14));
83     contentPane.add(label_1);
84
85
86
87     final JComboBox comboBox = new JComboBox(); //将按钮
或可编辑字段与下拉列表组合的组件
88     comboBox.setModel(new DefaultComboBoxModel(new
String[] {"主管员", "商品管理员", "收银员"}));
89     //利用DefaultComboBoxModel类可以很方便地做到动态更改
JComboBox的项目值
90     comboBox.addItemListener(new ItemListener() { //下
拉列表框的选项监听，被选中的显示，未被选中的隐藏。
91         @Override
92         public void itemStateChanged(ItemEvent e) {
93             if(ItemEvent.SELECTED==e.getStateChange()){
94                 roleId = comboBox.getSelectedIndex();
95             }
96         }
97     });
98     comboBox.setBounds(186, 186, 79, 21);
99     contentPane.add(comboBox);
100
101     JButton button = new JButton("登录");
102     button.addActionListener(new ActionListener() {
//设置一个监听
103         public void actionPerformed(ActionEvent e) {
104             String UserName = userName.getText(); //获
取输入的名字
105             String Pwd = pwd.getText(); //获取输
入的密码
106             UsersService user = new UsersService();
//用户相关服务
107             if(user.checkUsersService(UserName, Pwd,
roleId)){ //将输入的三个参数与数据库中的进行比对
108                 Login.this.dispose(); //使用dispose()
方法关闭的窗体可以使用pack 或 show 方法恢复，并且可以恢复到dispose前的
状态
109                 if(roleId == 0){//主管员
110                     RootAdministrator ra = new
RootAdministrator();
111                     ra.setVisible(true);

```

```

112         }else if(roleId == 1){//货物管理
113             GoodsAdministrator ga = new
GoodsAdministrator();
114             ga.setVisible(true);
115         }else if(roleId == 2){//收银员
116             UsersService us = new UsersService();
117             int userId =
us.getUserIdByUserNameService(Username);
118             Cashier c = new Cashier(userId);
119             c.setVisible(true);
120         }
121     }else{
122         JOptionPane.showMessageDialog(Login.this,
"登陆失败!");
123     }
124 }
125 });
126 button.setBounds(100, 243, 93, 43);
127 contentPane.add(button);
128
129 JButton button_1 = new JButton("退出");
130 button_1.addActionListener(new ActionListener() {
131     public void actionPerformed(ActionEvent e) {
132         Login.this.dispose();
133     }
134 });
135 button_1.setBounds(264, 243, 93, 43);
136 contentPane.add(button_1);
137 this.add(beijing);
138 }
139
140 }
141

```

## 8.NewAdministrator

```

1  package UI;
2
3  import java.awt.BorderLayout;
4  import java.awt.EventQueue;
5
6  import javax.swing.JFrame;
7  import javax.swing.JPanel;
8  import javax.swing.JPasswordField;

```

```
9  import javax.swing.border.EmptyBorder;
10 import javax.swing.ButtonGroup;
11 import javax.swing.JLabel;
12 import javax.swing.JOptionPane;
13 import javax.swing.JTextField;
14 import javax.swing.JRadioButton;
15 import javax.swing.JButton;
16
17 import service.UsersService;
18
19 import entity.Users;
20
21 import java.awt.event.ActionListener;
22 import java.awt.event.ActionEvent;
23
24 public class NewAdministrator extends JFrame {
25
26     private JPanel contentPane;
27     private JTextField userName;
28     private JTextField userId;
29     final JPasswordField pwd =new JPasswordField();
30     /**
31      * Launch the application.
32      */
33     public static void main(String[] args) {
34         NewAdministrator frame = new NewAdministrator();
35         frame.setVisible(true);
36     }
37
38     /**
39      * Create the frame.
40      */
41     public NewAdministrator() {
42         this.setTitle("新增管理员");
43         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
44         setBounds(400, 200, 505, 319);
45         contentPane = new JPanel();
46         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
47         setContentPane(contentPane);
48         contentPane.setLayout(null);
49
50         JLabel labe0 = new JLabel("员工编号:");
51         labe0.setBounds(85, 7, 104, 15);
52         contentPane.add(labe0);
53         userId = new JTextField();
```

```
54     userId.setColumns(10);
55     userId.setBounds(186, 7, 168, 21);
56     contentPane.add(userId);
57
58     JLabel label = new JLabel("用户名:");
59     label.setBounds(85, 50, 54, 15);
60     contentPane.add(label);
61
62     userName = new JTextField();
63     userName.setColumns(10);
64     userName.setBounds(186, 50, 168, 21);
65     contentPane.add(userName);
66
67     JLabel label_1 = new JLabel("密码:");
68     label_1.setBounds(85, 95, 54, 15);
69     contentPane.add(label_1);
70
71     pwd.setColumns(10);
72     pwd.setBounds(186, 92, 168, 21);
73     contentPane.add(pwd);
74
75
76
77     JLabel label_2 = new JLabel("管理员级别:");
78     label_2.setBounds(85, 144, 66, 15);
79     contentPane.add(label_2);
80
81     final JRadioButton rdbtnNewRadioButton = new
JRadioButton("主管管理员");
82     rdbtnNewRadioButton.setBounds(173, 140, 93, 23);
83     contentPane.add(rdbtnNewRadioButton);
84
85     final JRadioButton radioButton = new JRadioButton("货
物管理员");
86     radioButton.setBounds(268, 140, 94, 23);
87     contentPane.add(radioButton);
88
89     final JRadioButton radioButton_1 = new
JRadioButton("收银员");
90     radioButton_1.setBounds(364, 140, 85, 23);
91     contentPane.add(radioButton_1);
92     ButtonGroup bg = new ButtonGroup();
93     bg.add(rdbtnNewRadioButton);
94     bg.add(radioButton);
95     bg.add(radioButton_1);
```

```

96
97     JButton btnNewButton = new JButton("提交");
98     btnNewButton.addActionListener(new ActionListener() {
99         public void actionPerformed(ActionEvent e) {
100             Users user = new Users();           //连接User
101             String s = userId.getText();
102             int i = Integer.parseInt(s);        //Integer的默认值为null
103             user.setUserId(i);
104             user.setUserName(userName.getText());
105             user.setPassword(pwd.getText());
106             if(rdbtnNewRadioButton.isSelected()){
107                 user.setRoleId(0);           //对应的相应地级别
108             }else if(radioButton.isSelected()){
109                 user.setRoleId(1);
110             }else if(radioButton_1.isSelected()){
111                 user.setRoleId(2);
112             }else{
113
114                 JOptionPane.showMessageDialog(NewAdministrator.this, "请选择
115                 管理员类别!", "错误提示", JOptionPane.ERROR_MESSAGE);
116             }
117             UsersService us = new UsersService();
118             if(us.newUsersService(user)){
119
120                 JOptionPane.showMessageDialog(NewAdministrator.this, "添加管
121                 理员成功!");
122             }else{
123
124                 JOptionPane.showMessageDialog(NewAdministrator.this, "添加管
125                 理员失败!");
126             }
127         }
128     });
129     btnNewButton.setBounds(106, 194, 93, 23);
130     contentPane.add(btnNewButton);
131
132     JButton btnNewButton_1 = new JButton("返回");
133     btnNewButton_1.addActionListener(new ActionListener()
134     {
135         public void actionPerformed(ActionEvent e) {
136             NewAdministrator.this.dispose();
137             AdministratorManagement am = new
138             AdministratorManagement();

```

```

131         am.setVisible(true);
132     }
133 });
134     btnNewButton_1.setBounds(241, 194, 93, 23);
135     contentPane.add(btnNewButton_1);
136 }
137 }
138

```

## 9.NewCustomer

```

1  package UI;
2
3  import java.awt.BorderLayout;
4  import java.awt.EventQueue;
5
6  import javax.swing.JFrame;
7  import javax.swing.JPanel;
8  import javax.swing.border.EmptyBorder;
9  import javax.swing.JLabel;
10 import javax.swing.JOptionPane;
11 import javax.swing.JTextField;
12 import javax.swing.JButton;
13
14 import service.CustomerService;
15
16 import entity.Customers;
17
18 import java.awt.event.ActionListener;
19 import java.awt.event.ActionEvent;
20
21 public class NewCustomer extends JFrame {
22
23
24     private JPanel contentPane;
25     private JTextField cstName;
26     private JTextField telephone;
27     private JTextField cstNo;
28
29     /**
30      * Launch the application.
31      */
32     public static void main(String[] args) {

```



```
33
34     NewCustomer frame = new NewCustomer();
35     frame.setVisible(true);
36 }
37
38 /**
39  * Create the frame.
40  */
41 public NewCustomer() {
42     this.setTitle("新增会员");
43     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
44     setBounds(400, 200, 450, 300);
45     contentPane = new JPanel();
46     contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
47     setContentPane(contentPane);
48     contentPane.setLayout(null);
49
50     JLabel lblNewLabel = new JLabel("会员编号:");
51     lblNewLabel.setBounds(80, 35, 76, 15);
52     contentPane.add(lblNewLabel);
53
54     cstNo = new JTextField();
55     cstNo.setBounds(166, 32, 148, 21);
56     contentPane.add(cstNo);
57     cstNo.setColumns(10);
58
59     JLabel label = new JLabel("会员名:");
60     label.setBounds(89, 80, 54, 15);
61     contentPane.add(label);
62
63     cstName = new JTextField();
64     cstName.setColumns(10);
65     cstName.setBounds(166, 77, 148, 21);
66     contentPane.add(cstName);
67
68     JLabel label_1 = new JLabel("会员联系方式:");
69     label_1.setBounds(58, 131, 85, 15);
70     contentPane.add(label_1);
71
72     telephone = new JTextField();
73     telephone.setColumns(10);
74     telephone.setBounds(166, 128, 148, 21);
75     contentPane.add (telephone);
76
77     JButton btnNewButton = new JButton("提交");
```

```

78         btnNewButton.addActionListener(new ActionListener() {
79             public void actionPerformed(ActionEvent e) {
80                 Customers customer = new Customers();
81                 customer.setCustomerNo(cstNo.getText());
82                 customer.setCustomerName(cstName.getText());
83                 customer.setPhone(telephone.getText());
84                 CustomerService cs = new CustomerService();
85                 //if(cs.checkCustomerNoDao(cstNo.getText())){
86                 int flag=cs.newCustomersService(customer);
87                 if(flag != 0){
88
89                     JOptionPane.showMessageDialog(NewCustomer.this, "添加会员成功!
90 会员号码是"+customer.getCustomerNo());
91                 }else{
92
93                     JOptionPane.showMessageDialog(NewCustomer.this, "添加会员失
94 败!");
95                 }
96                 //}
97             }
98         });
99         btnNewButton.setBounds(99, 189, 93, 23);
100         contentPane.add(btnNewButton);
101
102         JButton button = new JButton("返回");
103         button.addActionListener(new ActionListener() {
104             public void actionPerformed(ActionEvent e) {
105                 NewCustomer.this.dispose();
106                 CustomerManagement cm = new
107 CustomerManagement();
108                 cm.setVisible(true);
109             }
110         });
111         button.setBounds(243, 189, 93, 23);
112         contentPane.add(button);
113     }
114 }

```

## 10.NewProducts

```
1  package UI;
2
3  import java.awt.BorderLayout;
4  import java.awt.EventQueue;
5
6  import javax.swing.JFrame;
7  import javax.swing.JPanel;
8  import javax.swing.border.EmptyBorder;
9  import javax.swing.JLabel;
10 import javax.swing.JOptionPane;
11 import javax.swing.JTextField;
12 import javax.swing.JComboBox;
13 import javax.swing.JButton;
14 import javax.swing.DefaultComboBoxModel;
15
16 import service.ProductsService;
17
18 import entity.Products;
19
20 import java.awt.event.ActionListener;
21 import java.awt.event.ActionEvent;
22 import java.awt.event.ItemEvent;
23 import java.awt.event.ItemListener;
24
25 public class NewProducts extends JFrame {
26
27     private JPanel contentPane;
28     private JTextField ProductNo;
29     private JTextField ProductName;
30     private JTextField Type;
31     private JTextField Unit;
32     private JTextField pCount;
33     private JTextField Price;
34     private JTextField Discount;
35     private String categoryId = "000";
36
37     /**
38      * Launch the application.
39      */
40     public static void main(String[] args) {
41
42         NewProducts frame = new NewProducts();
```

```
43         frame.setVisible(true);
44     }
45
46     /**
47      * Create the frame.
48      */
49     public NewProducts() {
50         this.setTitle("新增商品信息");
51         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
52         setBounds(400, 200, 450, 407);
53         contentPane = new JPanel();
54         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
55         setContentPane(contentPane);
56         contentPane.setLayout(null);
57
58         JLabel lblNewLabel = new JLabel("商品编号:");
59         lblNewLabel.setBounds(67, 22, 72, 15);
60         contentPane.add(lblNewLabel);
61
62         ProductNo = new JTextField();
63         ProductNo.setBounds(164, 19, 144, 21);
64         contentPane.add(ProductNo);
65         ProductNo.setColumns(10);
66
67         JLabel label = new JLabel("商品名称:");
68         label.setBounds(67, 50, 72, 15);
69         contentPane.add(label);
70
71         ProductName = new JTextField();
72         ProductName.setColumns(10);
73         ProductName.setBounds(164, 47, 144, 21);
74         contentPane.add(ProductName);
75
76         JLabel label_1 = new JLabel("商品类别:");
77         label_1.setBounds(67, 75, 72, 15);
78         contentPane.add(label_1);
79
80         JLabel label_2 = new JLabel("规格:");
81         label_2.setBounds(67, 119, 54, 15);
82         contentPane.add(label_2);
83
84         Type = new JTextField();
85         Type.setColumns(10);
86         Type.setBounds(164, 116, 144, 21);
87         contentPane.add(Type);
```

```
88
89     JLabel label_3 = new JLabel("单位:");
90     label_3.setBounds(67, 147, 54, 15);
91     contentPane.add(label_3);
92
93     Unit = new JTextField();
94     Unit.setColumns(10);
95     Unit.setBounds(164, 144, 144, 21);
96     contentPane.add(Unit);
97
98     JLabel label_4 = new JLabel("数量:");
99     label_4.setBounds(67, 175, 54, 15);
100    contentPane.add(label_4);
101
102    pCount = new JTextField();
103    pCount.setColumns(10);
104    pCount.setBounds(164, 172, 144, 21);
105    contentPane.add(pCount);
106
107    JLabel label_5 = new JLabel("售价:");
108    label_5.setBounds(67, 203, 54, 15);
109    contentPane.add(label_5);
110
111    Price = new JTextField();
112    Price.setColumns(10);
113    Price.setBounds(164, 200, 144, 21);
114    contentPane.add(Price);
115
116    JLabel label_6 = new JLabel("折扣:");
117    label_6.setBounds(67, 231, 54, 15);
118    contentPane.add(label_6);
119
120    Discount = new JTextField();
121    Discount.setColumns(10);
122    Discount.setBounds(164, 228, 144, 21);
123    contentPane.add(Discount);
124
125    final JComboBox comboBox = new JComboBox();
126    comboBox.setModel(new DefaultComboBoxModel(new
String[] {" ", "水果", "主食", "蔬菜", "文体办公", "家电五金", "洗浴
用品"}));
127    comboBox.addItemListener(new ItemListener() {
128        @Override
129        public void itemStateChanged(ItemEvent e) {
130            if(ItemEvent.SELECTED==e.getStateChange()){
```

```

131         categoryId =
comboBox.getSelectedItemAt().toString();
132         //categoryId = "00"+
(comboBox.getSelectedIndex()+1);    //转换为数字
133     }
134 }
135 });
136 comboBox.setBounds(164, 78, 82, 21);
137 contentPane.add(comboBox);
138
139 JButton btnNewButton = new JButton("保存");
140 btnNewButton.addActionListener(new ActionListener() {
141     public void actionPerformed(ActionEvent e) {
142         Products product = new Products();
143         if(ProductNo.getText().isEmpty()){
144             JOptionPane.showMessageDialog(null,"商品编
号不能为空","错误提示",JOptionPane.ERROR_MESSAGE);
145         }else{
146
147             product.setProductNo(ProductNo.getText());
148             product.setCategoryId(categoryId);
149             if(ProductName.getText().isEmpty()){
150                 JOptionPane.showMessageDialog(null,"商品名称不能为空","错误提
示",JOptionPane.ERROR_MESSAGE);
151             }else{
152                 product.setProductName(ProductName.getText());
153                 product.setType(Type.getText());
154                 if(Unit.getText().isEmpty()){
155                     JOptionPane.showMessageDialog(null,"商品单位不能为空","错误提
示",JOptionPane.ERROR_MESSAGE);
156                 }else{
157                     product.setUnit(Unit.getText());
158                     if(pCount.getText().isEmpty()){
159                         JOptionPane.showMessageDialog(null,"库存数量不能为空","错误提
示",JOptionPane.ERROR_MESSAGE);
160                     }else{
161                         product.setCount(Integer.valueOf(pCount.getText()));
162                         String str1=Price.getText();
163                         int a = 0;
164                         try {

```

```

164             a =
Integer.parseInt(str1);
165         } catch
(NumberFormatException e1) {
166             e1.printStackTrace();
167         }
168
169         if(Price.getText().isEmpty() || a<=0){
170             JOptionPane.showMessageDialog(null,"单价不能为空或负数","错误提
示",JOptionPane.ERROR_MESSAGE);
171         }else{
172             product.setPrice(Double.valueOf(Price.getText()));
173
174             if(Discount.getText().isEmpty()){
175                 JOptionPane.showMessageDialog(null,"折扣不能为空","错误提
示",JOptionPane.ERROR_MESSAGE);
176             }else{
177                 product.setDiscount(Double.valueOf(Discount.getText()));
178                 ProductsService ps =
new ProductsService();
179
180                 if(ps.checkProductsNoDao(ProductNo.getText())){
181                     if(ps.newProductsService(product)){
182                         JOptionPane.showMessageDialog(NewProducts.this, "添加商品成
功!");
183                         NewProducts.this.dispose();
184                         NewProducts
np = new NewProducts();
185                         np.setVisible(true);
186                     }else{
187                         JOptionPane.showMessageDialog(NewProducts.this, "添加商品失
败!");
188                     }
189                 }
190             }
191         }
192     }
193 }

```

```

189         }
190     }
191 }
192 }
193 }
194 });
195 btnNewButton.setBounds(115, 275, 175, 23);
196 contentPane.add(btnNewButton);
197
198 JButton button = new JButton("返回商品管理界面");
199 button.addActionListener(new ActionListener() {
200     public void actionPerformed(ActionEvent e) {
201         NewProducts.this.dispose();
202         GoodsAdministrator ga = new
GoodsAdministrator();
203         ga.setVisible(true);
204     }
205 });
206 button.setBounds(115, 316, 175, 23);
207 contentPane.add(button);
208 }
209 }
210

```

## 11.RootAdministrator

```

1  package UI;
2
3  import java.awt.*;
4
5  import javax.swing.JFrame;
6  import javax.swing.JPanel;
7  import javax.swing.border.EmptyBorder;
8  import javax.swing.JLabel;
9  import javax.swing.JButton;
10 import java.awt.event.ActionListener;
11 import java.awt.event.ActionEvent;
12
13 public class RootAdministrator extends JFrame {
14
15     private JPanel contentPane;
16
17     /**
18      * Launch the application.

```



```

19     */
20     public static void main(String[] args) {
21         RootAdministrator frame = new RootAdministrator();
22         frame.setVisible(true);
23     }
24
25     /**
26      * Create the frame.
27      */
28     public RootAdministrator() {
29         this.setTitle("主管理员界面");
30         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
31         setBounds(400, 200, 500, 500);
32         contentPane = new JPanel();
33         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
34         setContentPane(contentPane);
35         contentPane.setLayout(null);    //清空布局管理器
36
37         //修改字体大小
38         JLabel lblNewLabel = new JLabel("管理员,请选择要执行的功能
模块");
39         lblNewLabel.setFont(new Font("微软雅黑", Font.PLAIN,
22));
40         lblNewLabel.setSize(new Dimension(20,20));
41         lblNewLabel.setBounds(78, 33, 318, 49);
42         contentPane.add(lblNewLabel);
43
44         JButton btnNewButton = new JButton("管理员工信息");
45
46         btnNewButton.setFont(new Font("微软雅黑", Font.PLAIN,
17));
47
48         btnNewButton.addActionListener(new ActionListener() {
49             public void actionPerformed(ActionEvent e) {
50                 RootAdministrator.this.dispose();
51                 AdministratorManagement am = new
AdministratorManagement(); //打开员工信息管理界面
52                 am.setVisible(true);
53             }
54         });
55         btnNewButton.setBounds(170, 120, 159, 41);
56         contentPane.add(btnNewButton);
57
58         JButton button = new JButton("管理顾客信息");
59         button.addActionListener(new ActionListener() {

```

```

60         public void actionPerformed(ActionEvent e) {
61             RootAdministrator.this.dispose();
62             CustomerManagement cm = new
CustomerManagement();    // 打开会员信息管理界面
63             cm.setVisible(true);
64         }
65     });
66     button.setFont(new Font("微软雅黑", Font.PLAIN, 17));
67     button.setBounds(170, 200, 159, 41);
68     contentPane.add(button);
69
70     JButton btnNewButton1 = new JButton("管理商品信息");
71     btnNewButton1.setFont(new Font("微软雅黑", Font.PLAIN,
17));
72     btnNewButton1.addActionListener(new ActionListener() {
73         public void actionPerformed(ActionEvent e) {
74             RootAdministrator.this.dispose();
75             GoodsAdministrator qh = new
GoodsAdministrator();    // 打开商品信息管理界面
76             qh.setVisible(true);
77         }
78     });
79     btnNewButton1.setBounds(170, 280, 159, 41);
80     contentPane.add(btnNewButton1);
81
82     JButton btnNewButton_1 = new JButton("返回登录界面");
83     btnNewButton_1.addActionListener(new ActionListener()
{
84         public void actionPerformed(ActionEvent e) {
85             RootAdministrator.this.dispose();
86             Login lg = new Login();    // 返回登陆界
面
87             lg.setVisible(true);
88         }
89     });
90     btnNewButton_1.setBounds(350, 360, 117, 35);
91     contentPane.add(btnNewButton_1);
92 }
93
94 }
95

```

## 12.UpdateProducts

```
1  package UI;
2
3  import java.awt.BorderLayout;
4  import java.awt.EventQueue;
5
6  import javax.swing.JFrame;
7  import javax.swing.JPanel;
8  import javax.swing.border.EmptyBorder;
9  import javax.swing.JLabel;
10 import javax.swing.JOptionPane;
11 import javax.swing.JTextField;
12 import javax.swing.JButton;
13
14 import service.ProductsService;
15
16 import java.awt.event.ActionListener;
17 import java.awt.event.ActionEvent;
18
19 public class UpdateProducts extends JFrame {
20
21     private JPanel contentPane;
22     private JTextField productName;
23     private JTextField addCount;
24     private JTextField discount;
25
26     /**
27      * Launch the application.
28      */
29     public static void main(String[] args) {
30
31         Object[] row = {"dd","ww","ww","1"};
32         UpdateProducts frame = new UpdateProducts(row);
33         frame.setVisible(true);
34     }
35
36     /**
37      * Create the frame.
38      */
39     public UpdateProducts(final Object[] row) {
40         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
41         setBounds(400, 200, 450, 300);
42         contentPane = new JPanel();
```

```
43     contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
44     setContentPane(contentPane);
45     contentPane.setLayout(null);
46
47     JLabel lblNewLabel = new JLabel("商品名称:");
48     lblNewLabel.setBounds(98, 46, 68, 15);
49     contentPane.add(lblNewLabel);
50
51     productName = new JTextField();
52     productName.setEditable(false);
53     productName.setText((String) row[1]);
54     productName.setBounds(192, 43, 151, 21);
55     contentPane.add(productName);
56     productName.setColumns(10);
57
58     JLabel label = new JLabel("新增库存:");
59     label.setBounds(98, 94, 68, 15);
60     contentPane.add(label);
61
62     addCount = new JTextField();
63     addCount.setColumns(10);
64     addCount.setBounds(192, 91, 151, 21);
65     contentPane.add(addCount);
66
67     JLabel label_1 = new JLabel("折扣:");
68     label_1.setBounds(98, 139, 54, 15);
69     contentPane.add(label_1);
70
71     discount = new JTextField();
72     discount.setText(String.valueOf(row[3]));
73     discount.setColumns(10);
74     discount.setBounds(192, 136, 151, 21);
75     contentPane.add(discount);
76
77     JButton btnNewButton = new JButton("更新");
78     btnNewButton.addActionListener(new ActionListener() {
79         public void actionPerformed(ActionEvent e) {
80             int pcount = (Integer)(row[2]) +
81 Integer.valueOf(addCount.getText());
82             double Discount =
83 Double.valueOf(discount.getText());
84             String productNo = (String) row[0];
85             ProductsService ps = new ProductsService();
86             if(ps.updateProductsService(productNo,
87 pcount, Discount)){
```

```

85         JOptionPane.showMessageDialog(UpdateProducts.this, "更新库存信
            息成功!当前库存:"+pcount+"当前折扣:"+Discount);
86         }else{
87
88             JOptionPane.showMessageDialog(UpdateProducts.this, "更新库存信
            息失败!");
89
90         }
91     });
92     btnNewButton.setBounds(112, 190, 93, 23);
93     contentPane.add(btnNewButton);
94
95     JButton button = new JButton("返回");
96     button.addActionListener(new ActionListener() {
97         public void actionPerformed(ActionEvent e) {
98             UpdateProducts.this.dispose();
99             GoodsAdministrator ga = new
GoodsAdministrator();
100             ga.setVisible(true);
101         }
102     });
103     button.setBounds(231, 190, 93, 23);
104     contentPane.add(button);
105 }
106 }
107
108

```

## 5.main

### 1.main

```

1 package main;
2 import UI.Login;
3 public class main {
4     public static void main(String[] args) {
5         Login logoooin = new Login();
6         logoooin.setVisible(true);
7
8     }
9 }

```

## 6.iml

```
1  .iml文件
2
3
4  <?xml version="1.0" encoding="UTF-8"?>
5  <module type="JAVA_MODULE" version="4">
6    <component name="NewModuleRootManager"
LANGUAGE_LEVEL="JDK_1_8">
7      <output
url="file://$MODULE_DIR$/out/production/Supermarket" />
8      <output-test
url="file://$MODULE_DIR$/out/test/Supermarket" />
9      <exclude-output />
10     <content url="file://$MODULE_DIR$">
11       <sourceFolder url="file://$MODULE_DIR$/src"
isTestSource="false" />
12     </content>
13     <orderEntry type="inheritedJdk" />
14     <orderEntry type="sourceFolder" forTests="false" />
15     <orderEntry type="library" name="mssql-jdbc-6.4.0.jre8"
level="project" />
16   </component>
17 </module>
```