1.项目概述

1.1.项目演示

1. 运行 “饿了么商家后台管理系统”， 演示应用程序效果。

1.2.项目目标

1. 本项目为课程级贯穿项目中的第一个项目（**JDBC项目**、前端项目、javaWeb项目）。
2. 本项目完成后，学员将能够使用javaSE+JDBC+MySql技术开发基于控制台的C/S结构应用程序。

1.3.项目中所涉及到相关知识点

- 封装JDBC
- 封装DAO
- 领域模型中的实体类
- 增删改查操作
- 多条件模糊查询
- JDBC事务管理
- 表的主外键关系

1.4.数据库设计

1.4.1.DB一览表

No	表名称	中文名	说明
1	business	商家表	存储所有商家信息
2	food	食品表	存储每个商家所拥有的所有食品信息
3	admin	管理员表	存储管理员信息

1.5.2.表结构

约束类型标识： PK：primary key 主键 FK：foreign key 外键 NN：not null 非空 UQ：unique 唯一索引 AI：auto increment 自增

1.5.2.1.business（商家表）

No	字段名	数据类型	size	默认值	约束	说明
1	businessId	int			PK、AI、NN	商家编号
2	password	varchar	20		NN	密码
3	businessName	varchar	40		NN	商家名称
4	businessAddress	varchar	50			商家地址
5	businessExplain	varchar	40			商家介绍
6	starPrice	decimal	(5,2)	0.00		起送费
7	deliveryPrice	decimal	(5,2)	0.00		配送费

1.5.2.2.food（食品表）

No	字段名	数据类型	size	默认值	约束	说明
1	foodId	int			PK、AI、NN	食品编号
2	foodName	varchar	30		NN	食品名称
3	foodExplain	varchar	30			食品介绍
4	foodPrice	decimal	(5,2)		NN	食品价格
5	businessId	int			FK、NN	所属商家编号

1.5.2.3.admin（管理员表）

No	字段名	数据类型	size	默认值	约束	说明
1	adminId	int			PK、AI、NN	管理员编号
2	adminName	varchar	20		NN、UQ	管理员名称
3	password	varchar	20		NN	密码

2.项目搭建

2.1.开发环境检查

1. 开发工具：SpringToolSuite（STS）
2. 检查开发工具的jdk配置：jdk8
3. 检查开发工具的文件编码配置：utf-8

2.2.搭建javaWeb工程总体架构

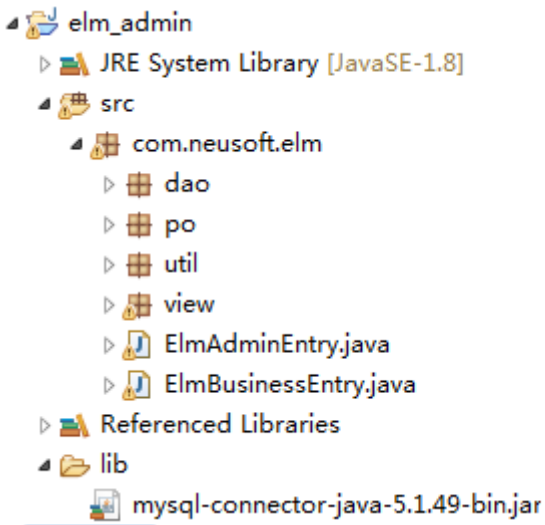
2.2.1.工程类型

创建工程类型：java Project

2.2.2.导入jar包

1. mysql-connector-java-bin.jar

2.2.3.工程目录结构



4.项目代码

4.1.领域模型（PO）

4.1.1.Business

```
package com.neusoft.elm.po;

public class Business {

    private Integer businessId;
    private String password;
    private String businessName;
    private String businessAddress;
    private String businessExplain;
    private Double starPrice;
    private Double deliveryPrice;

    @Override
    public String toString() {
        return "\n商家编号："+this.businessId+
               "\n商家名称："+this.businessName+
               "\n商家地址："+this.businessAddress+
               "\n商家介绍："+this.businessExplain+
               "\n起送费：¥"+this.starPrice+
               "\n配送费：¥"+this.deliveryPrice+"\n";
    }

    //get、set ... ...
}
```

4.1.2.Food

```
package com.neusoft.elm.po;

public class Food {

    private Integer foodId;
    private String foodName;
    private String foodExplain;
    private Double foodPrice;
    private Integer businessId;

    @Override
    public String toString() {
        return "\n食品编号："+this.foodId+
               "\n食品名称："+this.foodName+
               "\n食品介绍："+this.foodExplain+
               "\n食品价格："+this.foodPrice+
               "\n所属商家："+this.businessId;
    }


    //get、set ... ...
```

```
}
```

4.1.3.Admin

```
package com.neusoft.elm.po;

public class Admin {

    private Integer adminId;
    private String adminName;
    private String password;

    //get、set ... ...
}
```

4.2.Dao层代码

4.2.1.Business

```
package com.neusoft.elm.dao;

import java.util.List;

import com.neusoft.elm.po.Business;

public interface BusinessDao {

    public List<Business> listBusiness(String businessName,String businessAddress);
    public int saveBusiness(String businessName);
    public int removeBusiness(int businessId);

    public Business getBusinessByIdByPass(Integer businessId,String password);
    public Business getBusinessById(Integer businessId);
    public int updateBusiness(Business business);
    public int updateBusinessByPassword(Integer businessId,String password);
}
```

```
package com.neusoft.elm.dao.impl;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import java.util.List;
```

```
import com.neusoft.elm.dao.BusinessDao;
import com.neusoft.elm.po.Admin;
import com.neusoft.elm.po.Business;
import com.neusoft.elm.util.DBUtil;

public class BusinessDaoImpl implements BusinessDao{

    private Connection con = null;
    private PreparedStatement pst = null;
    private ResultSet rs = null;

    @Override
    public List<Business> listBusiness(String businessName,String businessAddress) {
        List<Business> list = new ArrayList<>();
        StringBuffer sql = new StringBuffer("select * from business where 1=1 ");
        if(businessName!=null&&!businessName.equals("")) {
            sql.append(" and businessName like '%"+businessName+"%' ");
        }
        if(businessAddress!=null&&!businessAddress.equals("")) {
            sql.append(" and businessAddress like '%"+businessAddress+"%' ");
        }
        try {
            con = DBUtil.getConnection();
            pst = con.prepareStatement(sql.toString());
            rs = pst.executeQuery();
            while(rs.next()) {
                Business business = new Business();
                business.setBusinessId(rs.getInt("businessId"));
                business.setPassword(rs.getString("password"));
                business.setBusinessName(rs.getString("businessName"));
                business.setBusinessAddress(rs.getString("businessAddress"));
                business.setBusinessExplain(rs.getString("businessExplain"));
                business.setStarPrice(rs.getDouble("starPrice"));
                business.setDeliveryPrice(rs.getDouble("deliveryPrice"));
                list.add(business);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            DBUtil.close(rs, pst, con);
        }
        return list;
    }

    @Override
    public int saveBusiness(String businessName) {
        int businessId = 0;
        String sql = "insert into business(businessName,password) values(?,'123')";
        try {
            con = DBUtil.getConnection();
            //设置返回自增长列值

            pst = con.prepareStatement(sql,PreparedStatement.RETURN_GENERATED_KEYS);
```

```
            pst.setString(1, businessName);
            pst.executeUpdate();
            //获取自增长列值（一行一列）
            rs = pst.getGeneratedKeys();
            if(rs.next()) {
                businessId = rs.getInt(1);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            DBUtil.close(rs, pst, con);
        }
        return businessId;
    }

    @Override
    public int removeBusiness(int businessId) {
        int result = 0;
        String delFootSql = "delete from food where businessId=?";
        String delBusinessSql = "delete from business where businessId=?";
        try {
            con = DBUtil.getConnection();
            //开启一个事务
            con.setAutoCommit(false);

            pst = con.prepareStatement(delFootSql);
            pst.setInt(1, businessId);
            pst.executeUpdate();

            pst = con.prepareStatement(delBusinessSql);
            pst.setInt(1, businessId);
            result = pst.executeUpdate();

            con.commit();
        } catch (SQLException e) {
            result = 0;
            try {
                con.rollback();
            } catch (SQLException e1) {
                e1.printStackTrace();
            }
            e.printStackTrace();
        } finally {
            DBUtil.close(null, pst, con);
        }
        return result;
    }

    @Override
    public Business getBusinessByIdByPass(Integer businessId,String password) {
        Business business = null;
        String sql = "select * from business where businessId=? and password=?";

        try {
```

```
            con = DBUtil.getConnection();
            pst = con.prepareStatement(sql.toString());
            pst.setInt(1, businessId);
            pst.setString(2, password);
            rs = pst.executeQuery();
            while(rs.next()) {
                business = new Business();
                business.setBusinessId(rs.getInt("businessId"));
                business.setPassword(rs.getString("password"));
                business.setBusinessName(rs.getString("businessName"));
                business.setBusinessAddress(rs.getString("businessAddress"));
                business.setBusinessExplain(rs.getString("businessExplain"));
                business.setStarPrice(rs.getDouble("starPrice"));
                business.setDeliveryPrice(rs.getDouble("deliveryPrice"));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            DBUtil.close(rs, pst, con);
        }
        return business;
    }

    @Override
    public Business getBusinessById(Integer businessId) {
        Business business = null;
        String sql = "select * from business where businessId=?";
        try {
            con = DBUtil.getConnection();
            pst = con.prepareStatement(sql.toString());
            pst.setInt(1, businessId);
            rs = pst.executeQuery();
            while(rs.next()) {
                business = new Business();
                business.setBusinessId(rs.getInt("businessId"));
                business.setPassword(rs.getString("password"));
                business.setBusinessName(rs.getString("businessName"));
                business.setBusinessAddress(rs.getString("businessAddress"));
                business.setBusinessExplain(rs.getString("businessExplain"));
                business.setStarPrice(rs.getDouble("starPrice"));
                business.setDeliveryPrice(rs.getDouble("deliveryPrice"));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            DBUtil.close(rs, pst, con);
        }
        return business;
    }

    @Override
    public int updateBusiness(Business business) {

        int result = 0;
```

```
        String sql = "update business set
businessName=?,businessAddress=?,businessExplain=?,starPrice=?,deliveryPrice=? where
businessId=?";
        try {
            con = DBUtil.getConnection();
            pst = con.prepareStatement(sql);
            pst.setString(1, business.getBusinessName());
            pst.setString(2, business.getBusinessAddress());
            pst.setString(3, business.getBusinessExplain());
            pst.setDouble(4, business.getStarPrice());
            pst.setDouble(5, business.getDeliveryPrice());
            pst.setInt(6, business.getBusinessId());
            result = pst.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            DBUtil.close(null, pst, con);
        }
        return result;
    }

    @Override
    public int updateBusinessByPassword(Integer businessId,String password) {
        int result = 0;
        String sql = "update business set password=? where businessId=?";
        try {
            con = DBUtil.getConnection();
            pst = con.prepareStatement(sql);
            pst.setString(1, password);
            pst.setInt(2, businessId);
            result = pst.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            DBUtil.close(null, pst, con);
        }
        return result;
    }
}
```

4.2.2.Food

```
package com.neusoft.elm.dao;

import java.util.List;

import com.neusoft.elm.po.Food;

public interface FoodDao {
```

```
    public List<Food> listFoodByBusinessId(Integer businessId);
    public int saveFood(Food food);
    public Food getFoodById(Integer foodId);
    public int updateFood(Food food);
    public int removeFood(Integer foodId);
}
```

```
package com.neusoft.elm.dao.impl;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import com.neusoft.elm.dao.FoodDao;
import com.neusoft.elm.po.Food;
import com.neusoft.elm.util.DBUtil;

public class FoodDaoImpl implements FoodDao{

    private Connection con = null;
    private PreparedStatement pst = null;
    private ResultSet rs = null;

    @Override
    public List<Food> listFoodByBusinessId(Integer businessId) {
        List<Food> list = new ArrayList<>();
        String sql = "select * from food where businessId=?";
        try {
            con = DBUtil.getConnection();
            pst = con.prepareStatement(sql);
            pst.setInt(1, businessId);
            rs = pst.executeQuery();
            while(rs.next()) {
                Food food = new Food();
                food.setFoodId(rs.getInt("foodId"));
                food.setFoodName(rs.getString("foodName"));
                food.setFoodExplain(rs.getString("foodExplain"));
                food.setFoodPrice(rs.getDouble("foodPrice"));
                food.setBusinessId(rs.getInt("businessId"));
                list.add(food);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            DBUtil.close(rs, pst, con);
        }
        return list;
    }
```

```
    @Override
    public int saveFood(Food food) {
        int result = 0;
        String sql = "insert into food values(null,?,?,?,?)";
        try {
            con = DBUtil.getConnection();
            pst = con.prepareStatement(sql);
            pst.setString(1, food.getFoodName());
            pst.setString(2, food.getFoodExplain());
            pst.setDouble(3, food.getFoodPrice());
            pst.setInt(4, food.getBusinessId());
            result = pst.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            DBUtil.close(null, pst, con);
        }
        return result;
    }

    @Override
    public Food getFoodById(Integer foodId) {
        Food food = null;
        String sql = "select * from food where foodId=?";
        try {
            con = DBUtil.getConnection();
            pst = con.prepareStatement(sql);
            pst.setInt(1, foodId);
            rs = pst.executeQuery();
            while(rs.next()) {
                food = new Food();
                food.setFoodId(rs.getInt("foodId"));
                food.setFoodName(rs.getString("foodName"));
                food.setFoodExplain(rs.getString("foodExplain"));
                food.setFoodPrice(rs.getDouble("foodPrice"));
                food.setBusinessId(rs.getInt("businessId"));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            DBUtil.close(rs, pst, con);
        }
        return food;
    }

    @Override
    public int updateFood(Food food) {
        int result = 0;
        String sql = "update food set foodName=?,foodExplain=?,foodPrice=? where foodId=?";
        try {
            con = DBUtil.getConnection();
            pst = con.prepareStatement(sql);

            pst.setString(1, food.getFoodName());
```

```
            pst.setString(2, food.getFoodExplain());
            pst.setDouble(3, food.getFoodPrice());
            pst.setInt(4, food.getFoodId());
            result = pst.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            DBUtil.close(null, pst, con);
        }
        return result;
    }

    @Override
    public int removeFood(Integer foodId) {
        int result = 0;
        String sql = "delete from food where foodId=?";
        try {
            con = DBUtil.getConnection();
            pst = con.prepareStatement(sql);
            pst.setInt(1, foodId);
            result = pst.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            DBUtil.close(null, pst, con);
        }
        return result;
    }
}
```

4.2.3.Admin

```
package com.neusoft.elm.dao;

import com.neusoft.elm.po.Admin;

public interface AdminDao {

    public Admin getAdminByNameByPass(String adminName,String password);
}
```

```
package com.neusoft.elm.dao.impl;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;


import com.neusoft.elm.dao.AdminDao;
```

```
import com.neusoft.elm.po.Admin;
import com.neusoft.elm.util.DBUtil;

public class AdminDaoImpl implements AdminDao{

    private Connection con = null;
    private PreparedStatement pst = null;
    private ResultSet rs = null;

    @Override
    public Admin getAdminByNameByPass(String adminName, String password) {
        Admin admin = null;
        String sql = "select * from admin where adminName=? and password=?";
        try {
            con = DBUtil.getConnection();
            pst = con.prepareStatement(sql);
            pst.setString(1, adminName);
            pst.setString(2, password);
            rs = pst.executeQuery();
            while(rs.next()) {
                admin = new Admin();
                admin.setAdminId(rs.getInt("adminId"));
                admin.setAdminName(rs.getString("adminName"));
                admin.setPassword(rs.getString("password"));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            DBUtil.close(rs, pst, con);
        }
        return admin;
    }
}
```

4.3.View层代码

4.3.1.Business

```
package com.neusoft.elm.view;

import com.neusoft.elm.po.Business;

public interface BusinessView {

    public void listBusinessAll();
    public void listBusiness();
    public void saveBusiness();
    public void removeBusiness();
```

```
    public Business login();
    public void showBusiness(Integer businessId);
    public void editBusiness(Integer businessId);
    public void updateBusinessByPassword(Integer businessId);
}
```

```
package com.neusoft.elm.view.impl;

import java.util.List;
import java.util.Scanner;

import com.neusoft.elm.dao.BusinessDao;
import com.neusoft.elm.dao.impl.BusinessDaoImpl;
import com.neusoft.elm.po.Business;
import com.neusoft.elm.view.BusinessView;

public class BusinessViewImpl implements BusinessView{

    private Scanner input = new Scanner(System.in);

    @Override
    public void listBusinessAll() {
        BusinessDao dao = new BusinessDaoImpl();
        List<Business> list = dao.listBusiness(null,null);
        System.out.println("商家编号\t商家名称\t商家地址\t商家介绍\t起送费\t配送费");
        for(Business b : list) {

System.out.println(b.getBusinessId()+"\t"+b.getBusinessName()+"\t"+b.getBusinessAddress()+"\t"+b
.getBusinessExplain()+"\t"+b.getStarPrice()+"\t"+b.getDeliveryPrice());
        }
    }

    @Override
    public void listBusiness() {
        String businessName = "";
        String businessAddress = "";

        String inputStr = "";
        System.out.println("是否需要输入商家名称关键词(y/n): ");
        inputStr = input.next();
        if(inputStr.equals("y")) {
            System.out.println("请输入商家名称关键词: ");
            businessName = input.next();
        }

        System.out.println("是否需要输入商家地址关键词(y/n): ");
        inputStr = input.next();
        if(inputStr.equals("y")) {
            System.out.println("请输入商家地址关键词: ");
            businessAddress = input.next();
        }
```

```
        BusinessDao dao = new BusinessDaoImpl();
        List<Business> list = dao.listBusiness(businessName, businessAddress);
        System.out.println("商家编号\t商家名称\t商家地址\t商家介绍\t起送费\t配送费");
        for(Business b : list) {

System.out.println(b.getBusinessId()+"\t"+b.getBusinessName()+"\t"+b.getBusinessAddress()+"\t"+b
.getBusinessExplain()+"\t"+b.getStarPrice()+"\t"+b.getDeliveryPrice());
        }
    }

    @Override
    public void saveBusiness() {
        System.out.println("请输入商家名称: ");
        String businessName = input.next();
        BusinessDao dao = new BusinessDaoImpl();
        int businessId = dao.saveBusiness(businessName);
        if(businessId>0) {
            System.out.println("新建商家成功! 商家编号为: "+businessId);
        }else {
            System.out.println("新建商家失败! ");
        }
    }

    @Override
    public void removeBusiness() {
        System.out.println("请输入商家编号: ");
        int businessId = input.nextInt();

        BusinessDao dao = new BusinessDaoImpl();
        System.out.println("确认要删除吗(y/n): ");
        if(input.next().equals("y")) {
            int result = dao.removeBusiness(businessId);
            if(result==1) {
                System.out.println("删除商家成功! ");
            }else {
                System.out.println("删除商家失败! ");
            }
        }
    }

    @Override
    public Business login() {
        System.out.println("请输入商家编号: ");
        int businessId = input.nextInt();
        System.out.println("请输入密码: ");
        String password = input.next();

        BusinessDao dao = new BusinessDaoImpl();
        return dao.getBusinessByIdByPass(businessId, password);
    }

    @Override

    public void showBusiness(Integer businessId) {
```

```
        BusinessDao dao = new BusinessDaoImpl();
        Business business = dao.getBusinessById(businessId);
        System.out.println(business);
    }

    @Override
    public void editBusiness(Integer businessId) {
        //先将商家信息查询出来显示，然后用户才能更新
        BusinessDao dao = new BusinessDaoImpl();
        Business business = dao.getBusinessById(businessId);
        System.out.println(business);

        String inputStr = "";
        System.out.println("是否修改商家名称(y/n)：");
        inputStr = input.next();
        if(inputStr.equals("y")) {
            System.out.println("请输入新的商家名称：");
            business.setBusinessName(input.next());
        }

        System.out.println("是否修改商家地址(y/n)：");
        inputStr = input.next();
        if(inputStr.equals("y")) {
            System.out.println("请输入新的商家地址：");
            business.setBusinessAddress(input.next());
        }

        System.out.println("是否修改商家介绍(y/n)：");
        inputStr = input.next();
        if(inputStr.equals("y")) {
            System.out.println("请输入新的商家介绍：");
            business.setBusinessExplain(input.next());
        }

        System.out.println("是否修改起送费(y/n)：");
        inputStr = input.next();
        if(inputStr.equals("y")) {
            System.out.println("请输入新的起送费：");
            business.setStarPrice(input.nextDouble());
        }

        System.out.println("是否修改配送费(y/n)：");
        inputStr = input.next();
        if(inputStr.equals("y")) {
            System.out.println("请输入新的配送费：");
            business.setDeliveryPrice(input.nextDouble());
        }

        int result = dao.updateBusiness(business);
        if(result>0) {
            System.out.println("\n修改商家信息成功！\n");
        }else {

            System.out.println("\n修改商家信息失败！\n");
```

```
        }
    }

    @Override
    public void updateBusinessByPassword(Integer businessId) {
        BusinessDao dao = new BusinessDaoImpl();
        Business business = dao.getBusinessById(businessId);

        System.out.println("\n请输入旧密码：");
        String oldPass = input.next();
        System.out.println("\n请输入新密码：");
        String password = input.next();
        System.out.println("\n请再次输入新密码：");
        String beginPassword = input.next();

        if(!business.getPassword().equals(oldPass)) {
            System.out.println("\n旧密码输入错误！");
        }else if(!password.equals(beginPassword)) {
            System.out.println("\n两次输入密码不一致！");
        }else {
            int result = dao.updateBusinessByPassword(businessId, password);
            if(result>0) {
                System.out.println("\n修改密码成功！");
            }else {
                System.out.println("\n修改密码失败！");
            }
        }
    }
}
```

4.3.2.Food

```
package com.neusoft.elm.view;

import java.util.List;

import com.neusoft.elm.po.Food;

public interface FoodView {

    public List<Food> showFoodList(Integer businessId);
    public void saveFood(Integer businessId);
    public void updateFood(Integer businessId);
    public void removeFood(Integer businessId);
}
```

```
package com.neusoft.elm.view.impl;


import java.util.List;
```

```
import java.util.Scanner;

import com.neusoft.elm.dao.FoodDao;
import com.neusoft.elm.dao.impl.FoodDaoImpl;
import com.neusoft.elm.po.Food;
import com.neusoft.elm.view.FoodView;

public class FoodViewImpl implements FoodView{

    private Scanner input = new Scanner(System.in);

    @Override
    public List<Food> showFoodList(Integer businessId) {
        FoodDao dao = new FoodDaoImpl();
        List<Food> list = dao.listFoodByBusinessId(businessId);
        System.out.println("食品编号\t食品名称\t食品介绍\t食品价格");
        for(Food food : list) {

System.out.println(food.getFoodId()+"\t"+food.getFoodName()+"\t"+food.getFoodExplain()+"\t"+food
.getFoodPrice());
        }
        return list;
    }

    @Override
    public void saveFood(Integer businessId) {
        Food food = new Food();
        System.out.println("请输入食品名称：");
        food.setFoodName(input.next());
        System.out.println("请输入食品介绍：");
        food.setFoodExplain(input.next());
        System.out.println("请输入食品价格：");
        food.setFoodPrice(input.nextDouble());
        food.setBusinessId(businessId);

        FoodDao dao = new FoodDaoImpl();
        int result = dao.saveFood(food);
        if(result>0) {
            System.out.println("\n新增食品成功！\n");
        }else {
            System.out.println("\n新增食品失败！\n");
        }
    }

    @Override
    public void updateFood(Integer businessId) {
        FoodDao dao = new FoodDaoImpl();
        List<Food> list = showFoodList(businessId);

        if(list.size()==0) {
            System.out.println("没有任何食品！");
        }else {

            System.out.println("请选择要更新的食品编号：");
```

```
            int foodId = input.nextInt();
            Food food = dao.getFoodById(foodId);
            System.out.println(food);

            String inputStr = "";
            System.out.println("是否更新食品名称(y/n): ");
            inputStr = input.next();
            if(inputStr.equals("y")) {
                System.out.println("请输入新的食品名称: ");
                food.setFoodName(input.next());
            }

            System.out.println("是否更新食品介绍(y/n): ");
            inputStr = input.next();
            if(inputStr.equals("y")) {
                System.out.println("请输入新的食品介绍: ");
                food.setFoodExplain(input.next());
            }

            System.out.println("是否更新食品价格(y/n): ");
            inputStr = input.next();
            if(inputStr.equals("y")) {
                System.out.println("请输入新的食品价格: ");
                food.setFoodPrice(input.nextDouble());
            }

            int result = dao.updateFood(food);
            if(result>0) {
                System.out.println("\n修改食品成功! \n");
            }else {
                System.out.println("\n修改食品失败! \n");
            }
        }
    }

    @Override
    public void removeFood(Integer businessId) {
        FoodDao dao = new FoodDaoImpl();
        List<Food> list = showFoodList(businessId);

        if(list.size()==0) {
            System.out.println("没有任何食品! ");
        }else {
            System.out.println("请选择要删除的食品编号: ");
            int foodId = input.nextInt();

            System.out.println("确认要删除吗(y/n): ");
            if(input.next().equals("y")) {
                int result = dao.removeFood(foodId);
                if(result>0) {
                    System.out.println("\n删除食品成功! \n");
                }else {

                    System.out.println("\n删除食品失败! \n");
```

```
                }
            }
        }
    }
}
```

4.3.3.Admin

```
package com.neusoft.elm.view;

import com.neusoft.elm.po.Admin;

public interface AdminView {

    public Admin login();
}
```

```
package com.neusoft.elm.view.impl;

import java.util.Scanner;

import com.neusoft.elm.dao.AdminDao;
import com.neusoft.elm.dao.impl.AdminDaoImpl;
import com.neusoft.elm.po.Admin;
import com.neusoft.elm.view.AdminView;

public class AdminViewImpl implements AdminView{

    private Scanner input = new Scanner(System.in);

    @Override
    public Admin login() {
        System.out.println("请输入管理员名称：");
        String adminName = input.next();
        System.out.println("请输入密码：");
        String password = input.next();

        AdminDao dao = new AdminDaoImpl();
        return dao.getAdminByNameByPass(adminName, password);
    }
}
```

4.4.程序入口

4.4.1.管理员入口

```
package com.neusoft.elm;

import java.util.Scanner;

import com.neusoft.elm.po.Admin;
import com.neusoft.elm.view.AdminView;
import com.neusoft.elm.view.BusinessView;
import com.neusoft.elm.view.impl.AdminViewImpl;
import com.neusoft.elm.view.impl.BusinessViewImpl;

public class ElmAdminEntry {

    public void work() {
        Scanner input = new Scanner(System.in);

        System.out.println("------------------------------------------------------------");
        System.out.println("|\t\t\t 饿了么后台管理系统  \t\t\t|");
        System.out.println("------------------------------------------------------------");

        AdminView adminView = new AdminViewImpl();
        BusinessView businessView = new BusinessViewImpl();
        Admin admin = adminView.login();

        //登录
        if(admin!=null) {
            int menu = 0;
            while(menu!=5) {
                //输出主菜单
                System.out.println("\n========= 1.所有商家列表=2.搜索商家=3.新建商家=4.删除商家=5.
退出系统 =========");
                System.out.println("请输入你的选择: ");
                menu = input.nextInt();
                switch(menu) {
                    case 1:
                        businessView.listBusinessAll();
                        break;
                    case 2:
                        businessView.listBusiness();
                        break;
                    case 3:
                        businessView.saveBusiness();
                        break;
                    case 4:
                        businessView.removeBusiness();
                        break;
                    case 5:
                        System.out.println("----------------------欢迎下次光临饿了么后台管理系统-
----------------------");
                        break;

                    default:
```

```
                        System.out.println("没有这个选项！\n");
                        break;
                }
            }

        }else {
            System.out.println("\n管理员名称或密码输入错误!\n");
        }
    }

    public static void main(String[] args) {
        new ElmAdminEntry().work();
    }

}
```

4.4.2.商家入口

```
package com.neusoft.elm;

import java.util.Scanner;

import com.neusoft.elm.po.Business;
import com.neusoft.elm.view.BusinessView;
import com.neusoft.elm.view.FoodView;
import com.neusoft.elm.view.impl.BusinessViewImpl;
import com.neusoft.elm.view.impl.FoodViewImpl;

public class ElmBusinessEntry {

    public void work() {
        Scanner input = new Scanner(System.in);

        System.out.println("------------------------------------------------------------");
        System.out.println("|\t\t\t 饿了么后台管理系统  \t\t\t|");
        System.out.println("------------------------------------------------------------");

        BusinessView businessView = new BusinessViewImpl();

        //商家登录
        Business business = businessView.login();

        if(business!=null) {
            int menu = 0;
            while(menu!=5) {
                //输出一级菜单
                System.out.println("\n======= 一级菜单（商家管理）1.查看商家信息=2.修改商家信息=3.更
新密码=4.所属商品管理=5.退出系统=======");
                System.out.println("请输入你的选择：");
                menu = input.nextInt();


                switch(menu) {
```

```
                    case 1:
                        businessView.showBusiness(business.getBusinessId());
                        break;
                    case 2:
                        businessView.editBusiness(business.getBusinessId());
                        break;
                    case 3:
                        businessView.updateBusinessByPassword(business.getBusinessId());
                        break;
                    case 4:
                        foodManager(business.getBusinessId());
                        break;
                    case 5:
                        System.out.println("----------------------欢迎下次光临饿了么后台管理系统-
----------------------");
                        break;
                    default:
                        System.out.println("没有这个选项！\n");
                        break;
                }
            }
        }else {
            System.out.println("商家编号或密码输入错误！");
        }

    }

    private void foodManager(int businessId) {
        Scanner input = new Scanner(System.in);

        FoodView foodView = new FoodViewImpl();

        int menu = 0;
        while(menu!=5) {
            //输出二级菜单
            System.out.println("\n======= 二级菜单（食品管理）1.查看食品列表=2.新增食品=3.修改食品
=4.删除食品=5.返回一级菜单 =======");
            System.out.println("请输入你的选择：");
            menu = input.nextInt();

            switch(menu) {
                case 1:
                    foodView.showFoodList(businessId);
                    break;
                case 2:
                    foodView.saveFood(businessId);
                    break;
                case 3:
                    foodView.updateFood(businessId);
                    break;
                case 4:
                    foodView.removeFood(businessId);

                    break;
```

```
                case 5:
                    break;
                default:
                    System.out.println("没有这个选项！\n");
                    break;
            }
        }
    }

    public static void main(String[] args) {
        new ElmBusinessEntry().work();
    }
}
```