长 沙 学 院

软件工程基础实训Ⅱ 实训

|  |  |
| --- | --- |
| **题目** | **基于C/S架构的餐饮管理系统** |
| **学院** | **计算机工程与应用数学** |
| **专业(班级)** | **软件工程（21软件01）** |
| **姓名** | **徐智强** |
| **学号** | **B20210304106** |
| **指导教师** | **杨刚, 潘怡** |
| **起止日期** | **2023.06.13～2023.06.24** |

实训任务书

**课程名称：软件工程基础实训II**

**实训题目：基于C/S架构的餐饮管理系统**

# 已知技术参数和设计要求：

1. 问题描述（功能要求）：

本次案例将完成一个基于C/S架构的餐饮管理系统。

（1）.进行餐饮管理系统的功能设计。

（2）.设计餐饮管理系统的数据库。

（3）.利用JAVA的GUI类设计开发友好的人机接口。

（4）.利用JAVA语言设计监听用户操作和JDBC进行数据库的处理。

餐饮管理系统主要有6大模块：

（1）.员工管理：对员工实现增删改查。

（2）.客户管理：对客户实现增删改查。

（3）.餐台管理：对餐台实现增删改查。

（4）.菜品管理：对菜品分类、菜品实现增删改查。

（5）.点菜管理：服务员对某客户、某一空闲餐台实行开台，同时实现点菜，将餐台号与所点的菜品对应起来，分别显示出来，并记录开台时间。

（6）.结账管理：收银员对某一餐台通过统计显示消费的菜品清单统计出消费金额，通过手动输入实收金额进行找零的计算，并显示，完成结账的操作，并记录成统计数据。

基础实训II有2周时间40学时，学生基本必做任务：独立完成基于C/S架构的餐饮管理系统。

选做扩展：菜品推荐、销售统计。

**2. 运行环境要求：**

（1）客户端：

windows操作系统，JDK1.6，Eclipse。

（2）服务器：

MySQL5.6或 SQL Server 2008 数据库服务器。

3. 技术要求：

1） 掌握软件工程的需求分析和系统设计方法。

2) 掌握数据库的分析与设计，完成餐饮管理系统数据库的设计。

3) 掌握JAVA的GUI设计方法，完成登录、主界面、菜品分类管理界面、菜品管理界面等人机接口。

4） 掌握JAVA面向对象程序设计方法，设计系统所需的各种类。

5) 掌握JDBC方法，对数据库进行操作，完成主程序和菜品分类、菜品管理等功能。

### 实训工作量：

40课时

### 工作计划：

1.班级

20软件1-4

2.课时及教室安排

见实训课程工作计划表。

实训报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学生姓名 | 徐智强 | 学号 | B20210304106 | 班级 | 软件工程（21软件01） |
| 专业 | 软件工程 | | 指导教师姓名 | 杨刚，潘怡 | |
| 项目名称 | 基于C/S架构的餐饮管理系统 | | | | |
| **实训过程**：（说明实训各阶段的工作过程，包括需求分析、系统设计、编码实现、系统测试、交付实施等阶段的主要工作。）  需求分析：随着我国市场经济的飞速发展，各行业都呈现出生机勃勃的发展景象，其中餐饮行业的发展尤为突出。但其中的服务及其繁琐，人工压力大而且管理不变。根据餐饮系统的流程，完成从用户登录到点餐，到结账收银的信息化管理，使用计算机对餐饮行业信息进行管理。  系统设计  员工管理：对员工实现增删改查  客户管理：对客户实现增删改查  餐台管理：对餐台实现增删改查，并且可以查看餐台的使用状态  菜品管理：对菜品分类、菜品实现增删改查  点餐管理：可实时对客户自动显示推荐菜品，可以通过选择不同的菜系进行精确的菜品选择。  结账管理：收银员对某一餐台通过统计显示消费菜品统计出消费金额，通过手动输入金额找零的计算，并显示，完成结账操作。  编码实现：运行环境是idea，数据库采用mySQL8.0,JDK采用17。主要采用JDBC框架实现对数据和java的GUI界面结合，实现对用户的进行数据的动态显示。没有采用传统的文本管理模式。  无系统测试和交付实施。 | | | | | |
| **实训成果**：  参见基于C/S架构的餐饮管理系统设计说明书。 | | | | | |
| **实训总结**：（包括心得体会、存在的问题和改进方向。）  通过此次实训，我更加对javaGUI的界面以及数据库的知识有了更深层次的了解，并且对于idea中的Datebase插件有了更加熟练的掌握。我通过使用数据库来对数据进行的使用和操作，并学习通过在java中通过jdbc来连接数据库。  在此次实训也发现了了许多的不足，在一开始的构思代码中存在不少的逻辑错误，也显现了自身数据库知识3的不扎实。对Java的抽象工厂设计模式也并不熟悉，无法自己独立写出来。 | | | | | |

基于C/S架构的餐饮管理系统

设计说明书

作者：徐智强（B20210304106）

计算机工程与应用数学学院

2022年 06 月 24日

**摘要**

随着我国市场经济的飞速发展，各行业都呈现出生机勃勃的发展景象，其中餐饮行业的发展尤为突出。但其中的服务及其繁琐，人工压力大而且管理不变。根据餐饮系统的流程，完成从用户登录到点餐，到结账收银的信息化管理，使用计算机对餐饮行业信息进行管理。

本餐饮管理系统采用C/S架构，其中运行环境是Eclipse，数据库采用mySQL8.0,JDK采用11。主要采用JDBC框架实现对数据和java的GUI界面结合，实现对用户的进行数据的动态显示。没有采用传统的文本管理模式。

**关键词：**C/S架构，餐饮管理系统，java

目录

[1 引言 1](#_Toc11882504)

[1.1 编写目的 1](#_Toc11882505)

[1.2 参考资料 1](#_Toc11882506)

[2 需求规约 2](#_Toc11882507)

[2.1 功能需求 2](#_Toc11882508)

[2.2 界面需求 2](#_Toc11882509)

[2.3 数据需求 2](#_Toc11882510)

[3 系统设计 5](#_Toc11882511)

[3.1 运行环境 5](#_Toc11882512)

[3.2 系统静态结构设计 5](#_Toc11882513)

[3.3 人机接口设计 6](#_Toc11882514)

[3.4 系统管理模块设计说明 8](#_Toc11882515)

[3.6 菜品管理模块设计说明 1](#_Toc11882516)1

[3.7 业务模块设计说明 1](#_Toc11882516)3

[4 数据库设计 1](#_Toc11882517)5

[4.1 数据库环境说明 1](#_Toc11882518)5

[4.2 数据库的命名规则 1](#_Toc11882519)5

[4.3 逻辑结构设计 1](#_Toc11882520)5

[4.4 物理结构设计 1](#_Toc11882521)6

[4.5 安全性设计 2](#_Toc11882522)0

[5 测试用例设计 2](#_Toc11882523)2

[附录 24](#_Toc11882524)

[附录1 程序运行结果 24](#_Toc11882525)

[附录2 程序源代码 2](#_Toc11882526)8

# 1 引言

## 1.1 编写目的

本文档是“餐饮管理系统”的软件设计说明书。编写目的是：明确软件需求、概要设计、详细设计和功能测试用例，为软件后续开发和维护提供指导。本文档的读者对象是需求分析人员、系统设计人员、系统开发人员、系统测试人员。

## 1.2 参考资料

列出有关资料的名称、作者、文件编号或版本等。

[1] 王珊，萨师煊. 数据库系统开发[M]. 北京: 高等教育出版社，2014．

[2] 戴远泉，程宁，胡文杰.Java高级程序设计.北京：人民邮电出版社，2022.

# 

# 2 需求规约

## 2.1 功能需求

本餐饮管理系统是基于小型餐饮店面和其相对应的餐饮业务为基础，突出管理，从专业角度出发，提供科学有效的管理模式。

本餐饮管理系统有如下的作用：

1. 能够针对中餐为其菜品的多样性和特色化的服务提供标准化的管理。
2. 能够提供符合餐饮企业自身要求的较科学的标准话、流程化管理，解决餐饮行业专业人才欠缺的问题。
3. 能够依据订餐、点餐、结账等环节的繁重化及复杂化问题，实现强化管理、降低成本、堵漏节流等效益。
4. 能够针对企业的经营现状做出科学的分析，使得企业对市场的应变能力得到提高。

2.2 界面需求

(1)登陆界面

主要实现用户的登录，然后进行页面的跳转到系统的主界面。

1. 系统的主界面

餐饮管理的核心，是跳转各个界面的中间键，也是各个界面的显示的载体。

1. 点菜管理界面

可以实现客户的点餐的操作，使点餐的数据进入数据库，以便其他界面的数据的显示。

1. 菜品管理界面

显示餐馆含有的菜品，可以实现对菜品的增加和删除，以及菜品的状态。

1. 菜品分类管理界面

显示餐馆含有的菜品，可以实现对菜品的增加和删除。

1. 结账界面

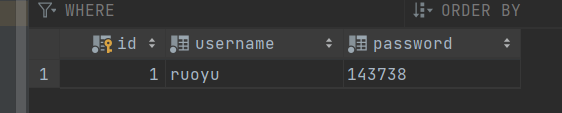
把所有的订单全部显示，如何可以点击对其进行结账处理，并且显示，客户的账单以及他所点单的菜品。也有折扣的处理。

1. 帮助界面

显示开发者的信息。

## 2.3 数据需求

表2.1 用户实体的数据



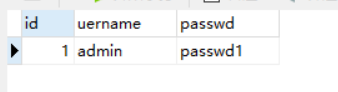


表2.2员工实体

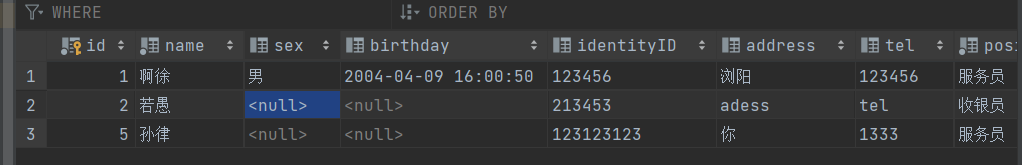


表2.3vip客户实体

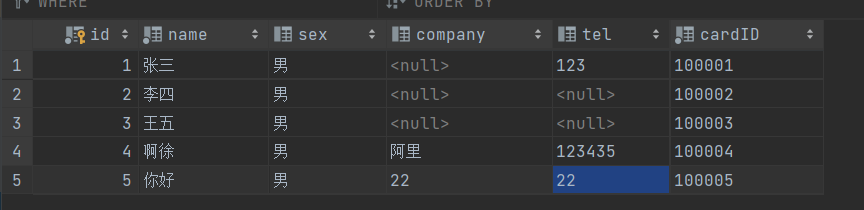


表2.4菜系实体



表2.5菜品实体

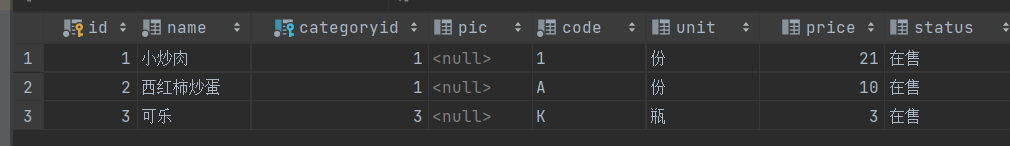


表2.6餐台实体

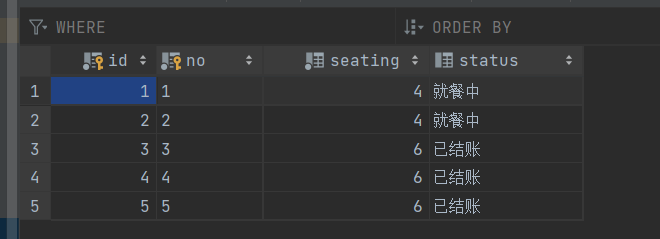


表2.7订单信息实体

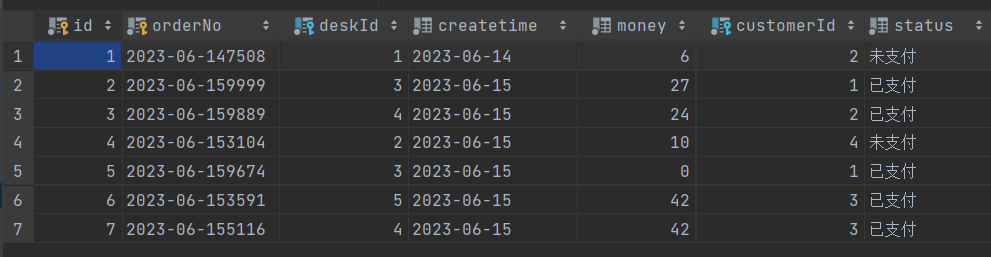
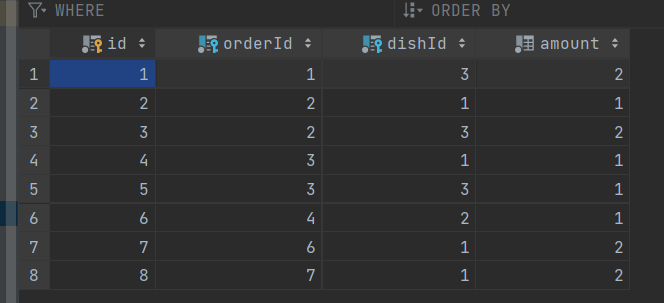


表2.8订单明细实体



# 3 系统设计

利用mysql作为后端数据库。

以jdbc为连接介质。

## 3.1 运行环境

本餐饮管理系统采用Java语言进行开发，JDK采用17，开发工具使用IDEA，数据库使用 8.0.25。

## 3.2 系统静态结构设计

根据该企业的具体情况，系统主要功能设计有六大部分，分别为员工管理、客户管理、餐台管理、菜品管理、点菜管理、结账管理。如图所示：

餐厅管理系统

餐台管理

员工管理

客户管理

点餐管理

结账管理

图3.1 餐饮管理包含图

员工管理：对员工实现增删改查

客户管理：对客户实现增删改查

餐台管理：对餐台实现增删改查，并且可以查看餐台的使用状态

菜品管理：对菜品分类、菜品实现增删改查

点餐管理：可实时对客户自动显示推荐菜品，可以通过选择不同的菜系进行精确的菜品选择。

结账管理：收银员对某一餐台通过统计显示消费菜品统计出消费金额，通过手动输入金额找零的计算，并显示，完成结账操作，并且记录统计数据。

系统静态结构说明

本系统是由多个静态的界面构成，主要是登录界面和餐饮管理界面。其中的核心是餐饮管理界面，它是各个界面的跳转的枢纽。餐饮管理界面含有系统管理、餐台管理、菜品管理、业务管理、结账管理。

## 3.3 人机接口设计

3.3.1 设计应用类

本系统的流程界面图是

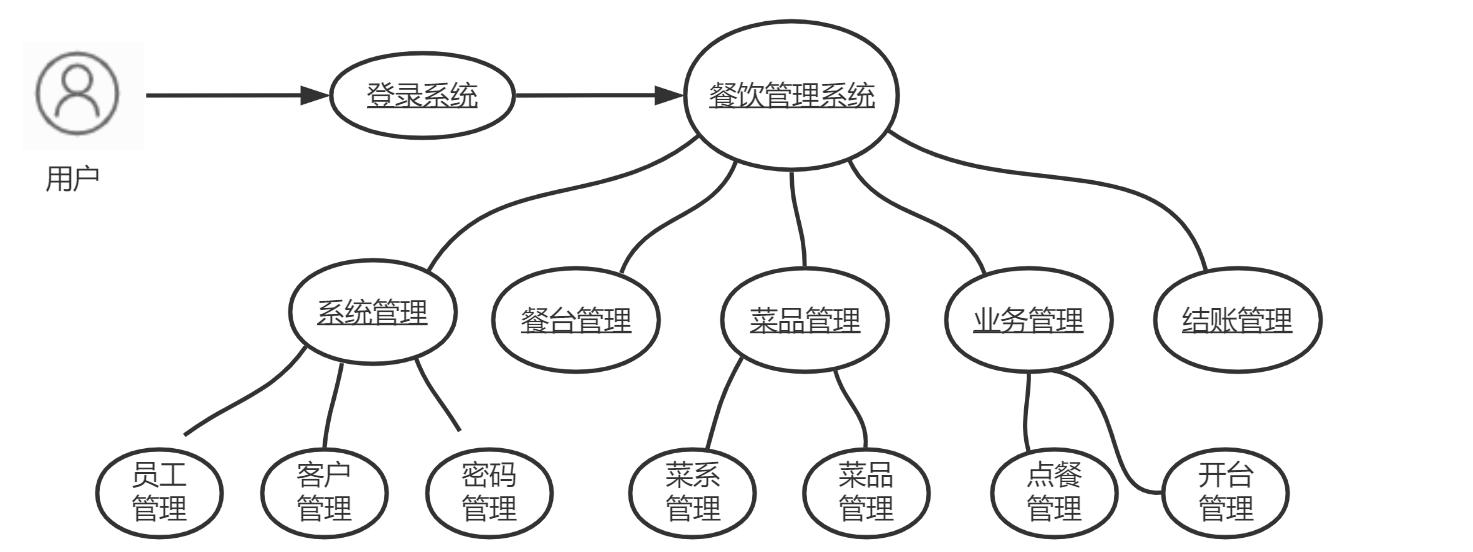


图3.2 餐饮管理的设计类图

3.3.2 用户接口设计



图3.3 用户登录图

用户登录界面图主要是输入正确的用户密码和用户账号，如何成功以后跳转到餐饮管理的主界面。



图3.4餐饮系统的主界面

餐饮系统的主界面其中包含员工管理、餐台管理、菜品分类管理、菜品管理、业务管理、结账管理。在点击其中各个管理的界面

## 3.4 员工管理模块设计说明



图3.5 员工管理图

员工管理界面可以实现对员工的添加和删除。可以方便用户对店里面的员工进行管理。缓解用户的压力。添加按钮是实现对员工的添加，删除对员工的离职，保留可以对员工的信息修改。

## 3.5菜品管理模块设计说明

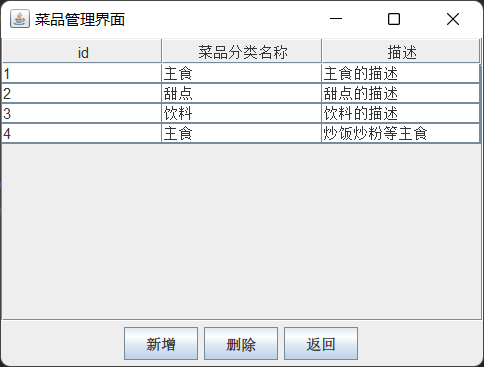


图3.9 菜系管理图

菜系管理界面可以让用户简便的管理本饭店的菜系。对餐管的菜系添加、删除和修改更加方便。添加按钮是实现对菜系的添加，删除是对菜系的剔除，保留可以对菜系的信息修改。

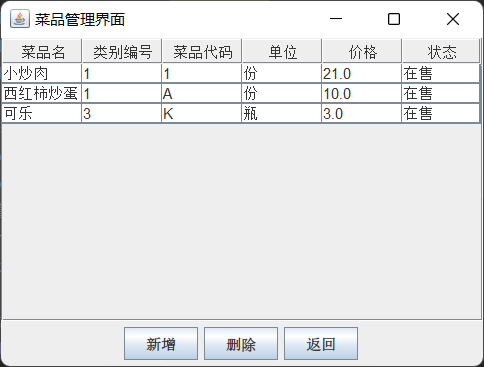


图3.10 菜品管理图

菜品管理界面可以让用户简便的管理本饭店的菜系。对餐管的菜品添加、删除和修改菜品的状态，但没有的时候可以自行设计，防止客户体验感下降。添加按钮是实现对菜品的添加，删除是对菜品的剔除，保留可以对菜品的信息修改。

3.6业务管理模块设计说明

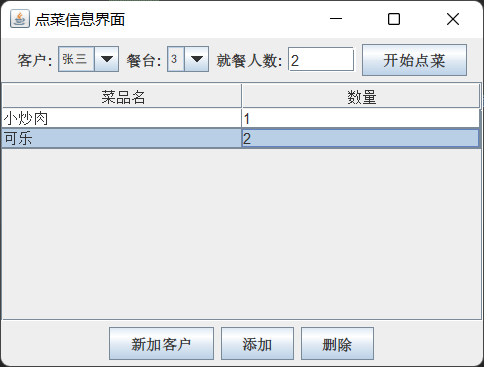


图3.11 点餐管理图

点餐管理界面可以让客户简便点自己喜欢菜品。界面可以自动显示提示框，让员工可以为客户点餐。提高客户和员工的体验感。添加按钮可以输入需要添加的餐品，删除可以去除客户不要的菜品，开始点菜可以传送要厨房，以及方便结账的管理

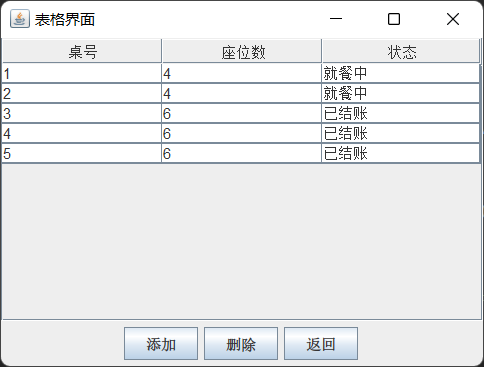


图3.12 开台管理图

餐台管理界面可以让客户进行餐台的预约，让用户可以及时处理订餐和预约。是餐台的管理井井有条。

3.7结账管理模块设计说明



图3.13 结账管理图

结账管理界面是可以选择要结账的桌号，点击结账查询按钮，会根据桌号显示客户的订单的详细。点击确定按钮，会对金额进行计算。

# 4 数据库设计

## 4.1 数据库环境说明

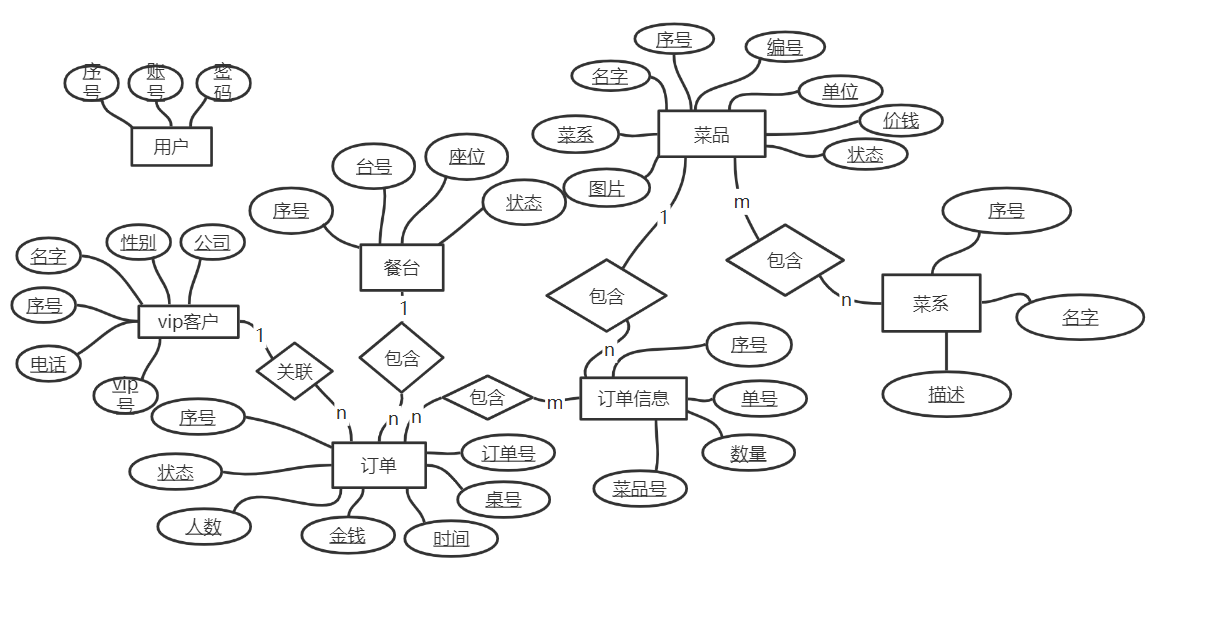
数据库采用用mySQL8.0,运行的环境是Navicat

## 4.2 数据库的命名规则

命名规则是不与MySQL中的关键子一样。

## 4.3 逻辑结构设计

E-R图



关系模型

管理员（序号，账号，密码）序号是主键

员工（编号，名字，性别，身份证，地址，电话，职位，状态）编号是主键

客户（编号，名字，性别，公司，电话，VIP号）编号是主键

餐台（序号，编号，状态，座位数）序号、编号是主键

菜系（编号，名称，描述）编号是主键

菜品（序号，名字，菜系，图片，编号，单位，价格，状态）编号是主键，菜品类是外键

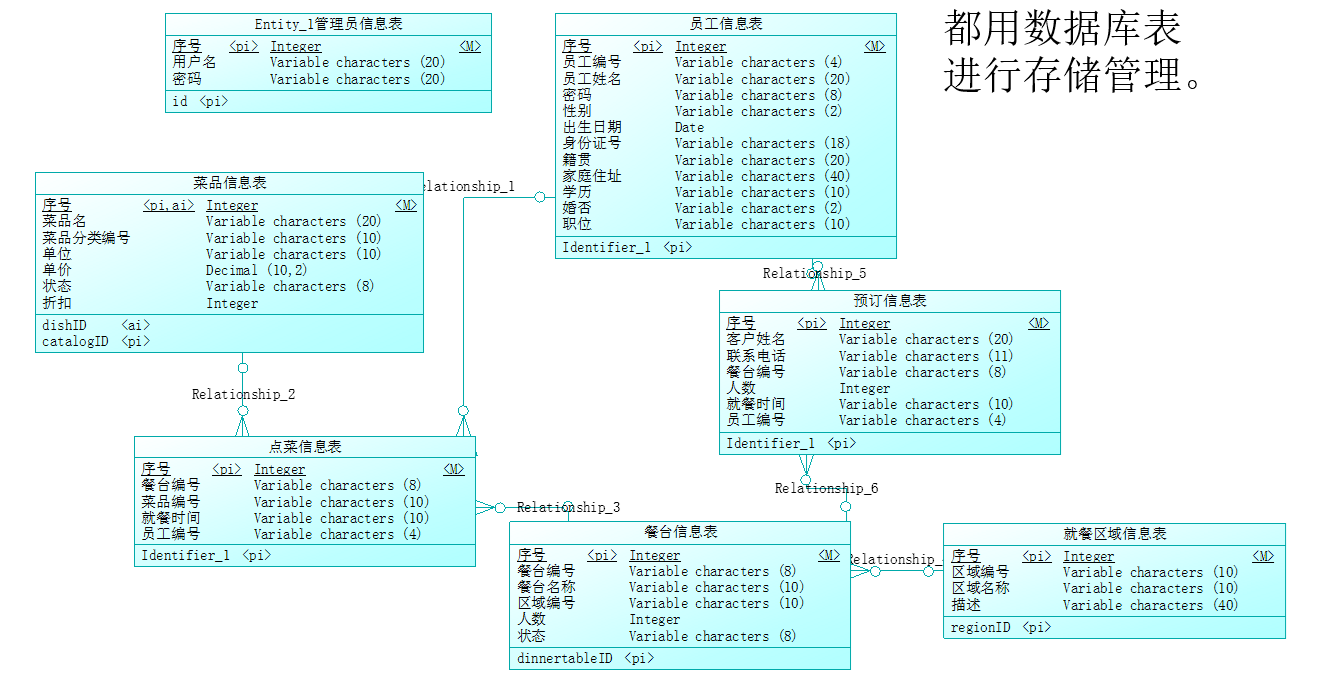
订单（序号，订单编号，餐台，时间，vip号，状态，价格，人数）序号是主键，客户编号是外键，餐台编号是外键

订单信息（序号，订单编号，菜品编号，数量）序号是主键，菜品序号是外键，餐台编号是外键

范式分析

该数据关系存在传递依赖，所以是2NF

逻辑模型图



## 4.4 物理结构设计

4.4.1 数据库表

表4.1 表汇总

|  |  |
| --- | --- |
| 表 名 | 功 能 说 明 |
| category | 记录菜系信息 |
| Dish | 记录菜品信息 |
| Desk | 记录餐桌信息 |
| User | 记录用户信息 |
| Staff | 记录员工信息 |
| Order | 记录订单信息 |
| orderitem | 记录订单详情 |
| customer | 记录客户信息 |

表4.2 菜系管理表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空Y/N | 唯一Y/N | 区别度 | 默认值 | 约束条件/说明 |
| 1 | Id | Int(10) | N | Y |  |  | 主键 |
| 2 | name | Varchar(20) | N | N |  |  |  |
| 3 | desrib | Varchar(20) | Y | N |  |  |  |

表4.3 菜品管理表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空Y/N | 唯一Y/N | 区别度 | 默认值 | 约束条件/说明 |
| 1 | Id | Int(10) | N | Y |  |  | 主键 |
| 2 | name | Varchar(20) | N | N |  |  |  |
| 3 | categoryid | Int(10) | N | N |  |  | 外键 |
| 4 | pic | Int(10) | N | N |  |  |  |
| 5 | code | Varchar(10) | N | N |  |  |  |
| 6 | unit | Varchar(10) | Y | N |  |  |  |
| 7 | price | Double(4) | Y | N |  |  |  |
| 8 | status | Varchar(4) | Y | N |  |  |  |

表4.4 餐桌管理表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空Y/N | 唯一Y/N | 区别度 | 默认值 | 约束条件/说明 |
| 1 | Id | Int(10) | N | Y |  |  | 主键 |
| 2 | no | Varchar(8) | N | Y |  |  | 主键 |
| 3 | seating | int(10) | Y | N |  |  |  |
| 4 | status | varchar | N | N |  |  |  |

表4.5 用户管理表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空Y/N | 唯一Y/N | 区别度 | 默认值 | 约束条件/说明 |
| 1 | Id | Int(10) | N | Y |  |  | 主键 |
| 2 | uername | Varchar(20) | N | N |  |  |  |
| 3 | passwd | Carchar(20) | N | N |  |  |  |

表4.6 员工管理表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空Y/N | 唯一Y/N | 区别度 | 默认值 | 约束条件/说明 |
| 1 | Id | Int(10) | N | Y |  |  | 主键 |
| 2 | name | Varchar(20) | N | N |  |  |  |
| 3 | sex | varchar(4) | Y | N |  |  |  |
| 4 | indentytyID | vharchar(18) | N | N |  |  |  |
| 5 | address | Varchar(10) | Y | N |  |  |  |
| 6 | tel | Varchar(11) | Y | N |  |  |  |
| 7 | position | Double(4) | N | N |  |  |  |
| 8 | freeze | Varchar(4) | YN | N |  |  |  |

表4.7 订单管理表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空Y/N | 唯一Y/N | 区别度 | 默认值 | 约束条件/说明 |
| 1 | Id | Int(10) | N | Y |  |  | 主键 |
| 2 | orderno | Varchar(20) | Y | N |  |  | 外键 |
| 3 | deskid | varchar(4) | Y | N |  |  | 外键 |
| 4 | creatime | date | Y | N |  |  |  |
| 5 | customerid | Varchar(10) | Y | N |  |  | 外键 |
| 6 | status | Varchar(11) | Y | N |  |  |  |
| 7 | number | int | Y | N |  |  |  |
| 8 | money | double | Y | N |  |  |  |

表4.8 订单详情表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空Y/N | 唯一Y/N | 区别度 | 默认值 | 约束条件/说明 |
| 1 | Id | Int(10) | N | Y |  |  | 主键 |
| 2 | orderID | Varchar(8) | N | N |  |  | 外键 |
| 3 | Dislisid | int(10) | N | N |  |  | 外键 |
| 4 | amout | Int(10) | Y | N |  |  |  |

## 4.5 安全性设计

4.5.1 防止用户直接操作数据库的方法

1.用户只能用帐号登录到应用软件，通过应用软件访问数据库，没有其他途径修改数据库数据。

2.用户只能通过应用软件提供接口修改可修改的数据。

4.5.2 用户帐号密码的加密方法

用户账号加密的方法：

1. 用户只能用帐号登录到应用软件，通过应用软件访问数据库，而没有其他途径修改账户密码。
2. 用户帐号密码的加密方法。对用户帐号的密码进行加密处理，确保在任何地方都不会出现密码的明文。

# 5 测试用例设计

表5.1 登录-注册测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能 | 用例编号 | 操作步骤  （输入条件） | 测试数据 | 预期结果 | 实际结果/判定 |
| 登录 | 1 | 输入‘用户名’和‘密码’ | admin,passwd1 | 登录成功 | 通过 |
| 添加员工 | 2 | 输入员工信息 | 按照要求输入数据 | 添加成功 | 通过 |
| 修改员工 | 3 | 选择员工信息并修改 |  | 修改成功 | 通过 |
| 删除员工 | 4 | 选择员工删除 |  | 删除成功 | 通过 |
| 添加客户 | 5 | 输入客户信息 |  | 添加成功 | 通过 |
| 修改客户 | 6 | 选择客户信息并修改 | 按照要求输入数据 | 修改成功 | 通过 |
| 删除客户 | 7 | 选择客户并删除 |  | 删除成功 | 通过 |
| 修改密码 | 8 | 输入旧密码以及新密码 |  | 修改成功 | 通过 |
| 添加餐台 | 9 | 输入餐台信息 | 按照要求输入数据 | 添加成功 | 通过 |
| 修改餐台 | 10 | 选择餐台信息并修改 |  | 修改成功 | 通过 |
| 删除餐台 | 11 | 选择餐台并删除 |  | 删除成功 | 通过 |
| 添加菜品类 | 12 | 输入菜品类信息 | 小吃，好吃 | 添加成功 | 通过 |
| 修改菜品类 | 13 | 选择菜品类信息并修改 |  | 修改成功 | 通过 |
| 删除菜品类 | 14 | 选择菜品类并删除 |  | 删除成功 | 通过 |
| 添加菜品 | 15 | 输入菜品信息 |  | 添加成功 | 通过 |
| 修改菜品 | 16 | 选择菜品信息并修改 |  | 修改成功 | 通过 |
| 删除菜品 | 17 | 选择菜品并删除 |  | 删除成功 | 通过 |
| 点菜管理 | 18 | 选择客户、就餐人数、餐台、以及所需菜品进行点单 |  | 点菜成功 | 通过 |
| 结账 | 19 | 选择未支付的订单进行结账 | 编号为5的订单 | 结账成功 | 通过 |

# 附录

## 附录1 程序运行结果



图1 登录界面



图2 系统主界面



图3 员工管理界面



图4 客户新加界面

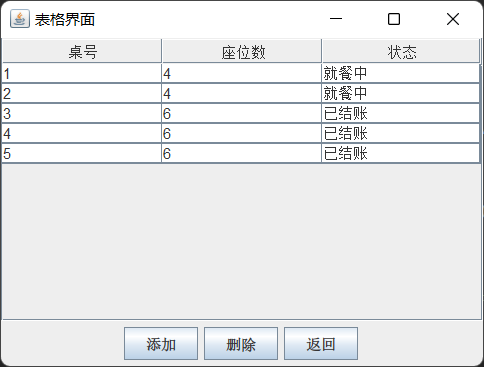


图5 餐台管理界面

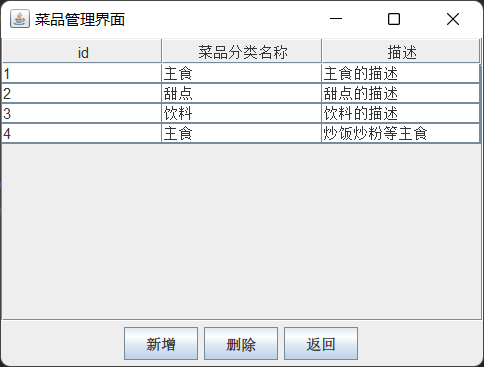


图6菜系管理界面

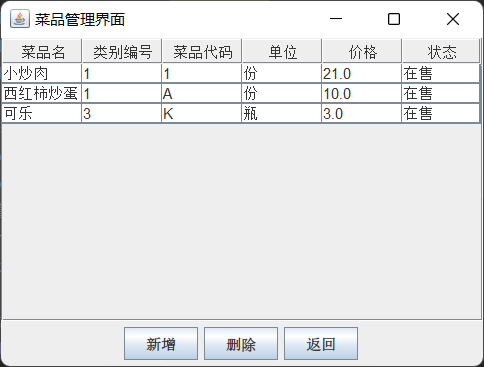


图7菜系管理界面

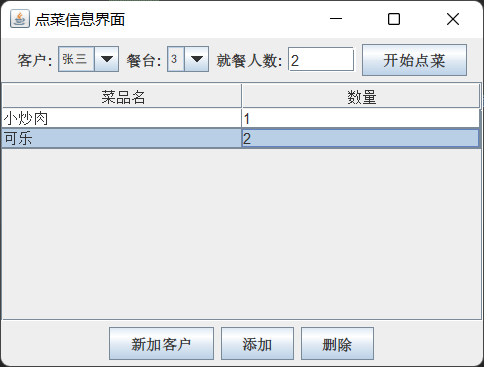


图8点餐管理界面



图9结账管理界面

## 附录2 程序源代码

package com.canying.bianjie.cantai;  
  
import com.canying.kongzhi.CategoryDAOImpl;  
import com.canying.kongzhi.DeskDAOImpl;  
import com.canying.kongzhi.DishDAOImpl;  
import com.canying.kongzhi.JDBCConection;  
import com.canying.shiti.Category;  
import com.canying.shiti.Desk;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.Connection;  
  
public class AddCantai extends JFrame {  
  
 public AddCantai() {  
 //创建一个主要框架,将其命名为”登录"  
 JFrame jFrame = new JFrame("菜品分类新加");  
  
 //设置窗口大小  
 jFrame.setSize(900, 507);  
  
 //先将布局管理器置为null  
 jFrame.setLayout(null);  
  
  
 //添加标签【学生管理系统】  
 JLabel textStudentManage = new JLabel("请输入新增的餐台信息");//创建一个标签,并命名为“欢迎使用餐饮管理系统“  
 textStudentManage.setForeground(new Color(0x0010FF));//设置字体颜色  
 textStudentManage.setFont(new Font("黑体", Font.*PLAIN*, 50));//设置字体大小  
 textStudentManage.setBounds(200, 50, 800, 100);//设置标签的绝对位置  
 jFrame.add(textStudentManage);//向框架中添加组件【标签(学生管理系统)】  
  
  
 //添加标签【用户名】  
 JLabel textUser = new JLabel("餐台编号:");  
 textUser.setForeground(new Color(0xFF0000));  
 textUser.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 textUser.setBounds(160, 140, 200, 100);  
 jFrame.add(textUser);  
  
 //添加输入框【用户名输入框】  
 JTextField user = new JTextField(20);  
 user.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 user.setSelectedTextColor(new Color(0xFF0000));  
 user.setBounds(330, 170, 280, 40);  
 jFrame.add(user);  
  
 //添加标签【密码】  
 JLabel textPassword = new JLabel("座位数:");  
 textPassword.setForeground(new Color(0xFF0000));  
 textPassword.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 textPassword.setBounds(200, 200, 200, 100);  
 jFrame.add(textPassword);  
  
 //添加密码输入框【密码】  
 JTextField password = new JTextField(20);  
 password.setBounds(330, 230, 280, 40);  
 password.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 password.setSelectedTextColor(new Color(0xFF0000));  
 jFrame.add(password);  
  
 //添加按钮【登录】  
 JButton jButton = new JButton("确定");  
 jButton.setForeground(new Color(0x023BF6));  
 jButton.setBackground(new Color(0x38FF00));  
 jButton.setFont(new Font("黑体", Font.*PLAIN*, 20));  
 jButton.setBorderPainted(false);  
 jButton.setBounds(300, 330, 100, 50);  
 jFrame.add(jButton);  
  
  
 JButton register = new JButton("退出");  
 register.setForeground(new Color(0x0029FF));  
 register.setBackground(new Color(0xECA871));  
 register.setFont(new Font("黑体", Font.*PLAIN*, 20));  
 register.setBorderPainted(false);  
 register.setBounds(500, 330, 100, 50);  
 jFrame.add(register);  
  
 register.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 jFrame.setVisible(false);  
 new Cantai();  
 }  
 });  
  
  
 //对按钮事件进行处理  
 jButton.addActionListener((e -> {  
 JDBCConection db = new JDBCConection();  
 Connection conn = db.getConn();  
 String no = new String(user.getText());  
 String seating = new String(password.getText());  
 if (no.equals("") || seating.equals("")) {  
 JOptionPane.*showMessageDialog*(jFrame, "请输入完整的信息", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 } else {  
 int id = Integer.*valueOf*(no);  
 int seatings = Integer.*valueOf*(seating);  
 Desk cantai = new Desk(id, no, seatings);  
 DeskDAOImpl deskDAO = new DeskDAOImpl();  
  
 deskDAO.add(conn, cantai);  
  
  
 JOptionPane.*showMessageDialog*(jFrame, "添加成功", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 user.setText("");  
 password.setText("");  
 }  
 }));  
  
 //设置相对位置：屏幕中间  
 jFrame.setLocationRelativeTo(null);  
  
 //确保使用窗口关闭按钮，能够正常退出，结束进程！  
 jFrame.setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  
  
 //禁止对窗口大小进行缩放处理  
 jFrame.setResizable(false);  
  
 //设置可见  
 jFrame.setVisible(true);  
 }  
}

package com.canying.bianjie.cantai;  
  
import com.canying.bianjie.RestaurantSystem;  
import com.canying.kongzhi.CategoryDAOImpl;  
import com.canying.kongzhi.DeskDAOImpl;  
import com.canying.kongzhi.JDBCConection;  
  
import javax.swing.\*;  
import javax.swing.table.DefaultTableModel;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
  
public class Cantai extends JFrame implements ActionListener {  
 private JTable table;  
 private DefaultTableModel tableModel;  
 private JButton addButton, deleteButton, reButton;  
 JDBCConection db = new JDBCConection();  
 Connection conn = db.getConn();  
  
 public Cantai() {  
 // 设置窗口标题和大小  
 setTitle("表格界面");  
 setSize(400, 300);  
  
 // 创建表格和表格模型  
 tableModel = new DefaultTableModel();  
 tableModel.addColumn("桌号");  
 tableModel.addColumn("座位数");  
 tableModel.addColumn("状态");  
  
 table = new JTable(tableModel);  
  
 // 添加组件到窗口中  
 Container container = getContentPane();  
 container.setLayout(new BorderLayout());  
 container.add(new JScrollPane(table), BorderLayout.*CENTER*);  
  
 addButton = new JButton("添加");  
 addButton.addActionListener(this);  
 deleteButton = new JButton("删除");  
 deleteButton.addActionListener(this);  
 reButton = new JButton("返回");  
 reButton.addActionListener(this);  
  
 JPanel buttonPanel = new JPanel();  
 buttonPanel.setLayout(new FlowLayout());  
 buttonPanel.add(addButton);  
 buttonPanel.add(deleteButton);  
 buttonPanel.add(reButton);  
 container.add(buttonPanel, BorderLayout.*SOUTH*);  
  
  
 addButton.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 setVisible(false);  
 new AddCantai();  
 }  
 });  
 deleteButton.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 int selectedRow = table.getSelectedRow();  
 if (selectedRow >= 0) {  
 int id = 0;  
 try {  
 String str = ((String) tableModel.getValueAt(selectedRow, 0));  
 id = Integer.*parseInt*(str);  
 tableModel.removeRow(selectedRow);  
 DeskDAOImpl deskDAO = new DeskDAOImpl();  
 deskDAO.delete(conn, id);  
 } catch (NumberFormatException e1) {  
 System.*out*.println("字符串无法转换为整数类型");  
 }  
  
 }  
 }  
 });  
 reButton.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 setVisible(false);  
 new RestaurantSystem();  
 }  
 });  
  
  
 try {  
 // 查询餐台数据  
 String sql = "SELECT id,no,seating,status,status FROM desk";  
 PreparedStatement stmt = conn.prepareStatement(sql);  
 ResultSet rs = stmt.executeQuery();  
  
 // 将查询结果添加到表格模型中EFWCDSFHJGF  
 while (rs.next()) {  
 // int id = rs.getInt("id");  
 String no = rs.getString("no");  
 int seating = rs.getInt("seating");  
 String status = rs.getString("status");  
 tableModel.addRow(new Object[]{no, seating, status});  
 }  
 // 设置窗口居中显示  
 setLocationRelativeTo(null);  
  
 // 设置窗口可见  
 setVisible(true);  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 }  
  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
  
 }  
  
 public static void main(String[] args) {  
 new Cantai();  
 }  
}

package com.canying.bianjie.category;  
  
import com.canying.kongzhi.CategoryDAOImpl;  
import com.canying.kongzhi.JDBCConection;  
import com.canying.shiti.Category;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.Connection;  
  
public class AddDishFrame extends JFrame {  
  
 public AddDishFrame() {  
 //创建一个主要框架,将其命名为”登录"  
 JFrame jFrame = new JFrame("菜品分类新加");  
  
 //设置窗口大小  
 jFrame.setSize(900, 507);  
  
 //先将布局管理器置为null  
 jFrame.setLayout(null);  
  
  
 //添加标签【学生管理系统】  
 JLabel textStudentManage = new JLabel("请输入新增的菜品分类");//创建一个标签,并命名为“欢迎使用餐饮管理系统“  
 textStudentManage.setForeground(new Color(0x0010FF));//设置字体颜色  
 textStudentManage.setFont(new Font("黑体", Font.*PLAIN*, 50));//设置字体大小  
 textStudentManage.setBounds(200, 50, 800, 100);//设置标签的绝对位置  
 jFrame.add(textStudentManage);//向框架中添加组件【标签(学生管理系统)】  
  
  
 //添加标签【用户名】  
 JLabel textUser = new JLabel("菜品分类名:");  
 textUser.setForeground(new Color(0xFF0000));  
 textUser.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 textUser.setBounds(160, 140, 200, 100);  
 jFrame.add(textUser);  
  
 //添加输入框【用户名输入框】  
 JTextField user = new JTextField(20);  
 user.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 user.setSelectedTextColor(new Color(0xFF0000));  
 user.setBounds(330, 170, 280, 40);  
 jFrame.add(user);  
  
 //添加标签【密码】  
 JLabel textPassword = new JLabel("描述 :");  
 textPassword.setForeground(new Color(0xFF0000));  
 textPassword.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 textPassword.setBounds(200, 200, 200, 100);  
 jFrame.add(textPassword);  
  
 //添加密码输入框【密码】  
 JTextField password = new JTextField(20);  
 password.setBounds(330, 230, 280, 40);  
 password.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 password.setSelectedTextColor(new Color(0xFF0000));  
 jFrame.add(password);  
  
 //添加按钮【登录】  
 JButton jButton = new JButton("确定");  
 jButton.setForeground(new Color(0x023BF6));  
 jButton.setBackground(new Color(0x38FF00));  
 jButton.setFont(new Font("黑体", Font.*PLAIN*, 20));  
 jButton.setBorderPainted(false);  
 jButton.setBounds(300, 330, 100, 50);  
 jFrame.add(jButton);  
  
  
 JButton register = new JButton("退出");  
 register.setForeground(new Color(0x0029FF));  
 register.setBackground(new Color(0xECA871));  
 register.setFont(new Font("黑体", Font.*PLAIN*, 20));  
 register.setBorderPainted(false);  
 register.setBounds(500, 330, 100, 50);  
 jFrame.add(register);  
  
 register.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 jFrame.setVisible(false);  
 new caipinfenleibiao();  
 }  
 });  
  
  
 //对按钮事件进行处理  
 jButton.addActionListener((e -> {  
 JDBCConection db = new JDBCConection();  
 Connection conn = db.getConn();  
 String name = new String(user.getText());  
 String describ = new String(password.getText());  
 if (name.equals("") || describ.equals("")) {  
 JOptionPane.*showMessageDialog*(jFrame, "请输入完整的信息", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 } else {  
 Category category = new Category(name, describ);  
 CategoryDAOImpl categoryDAO = new CategoryDAOImpl();  
  
 int id = categoryDAO.getmaxid(conn);  
 Category.*setId*(id + 1);  
 categoryDAO.add(conn, category);  
  
  
 JOptionPane.*showMessageDialog*(jFrame, "添加成功", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 user.setText("");  
 password.setText("");  
 }  
 }));  
  
 //设置相对位置：屏幕中间  
 jFrame.setLocationRelativeTo(null);  
  
 //确保使用窗口关闭按钮，能够正常退出，结束进程！  
 jFrame.setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  
  
 //禁止对窗口大小进行缩放处理  
 jFrame.setResizable(false);  
  
 //设置可见  
 jFrame.setVisible(true);  
 }  
  
  
}

package com.canying.bianjie.category;  
  
import com.canying.bianjie.RestaurantSystem;  
import com.canying.kongzhi.CategoryDAOImpl;  
import com.canying.kongzhi.JDBCConection;  
import com.canying.shiti.Category;  
  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
  
import javax.swing.\*;  
import javax.swing.table.DefaultTableModel;  
  
public class caipinfenleibiao extends JFrame {  
 private JTable table;  
 private DefaultTableModel tableModel;  
 private JButton addButton;  
 private JButton deleteButton;  
 private JButton saveButton;  
 private JButton modifyButton;  
  
 JDBCConection db = new JDBCConection();  
 Connection conn = db.getConn();  
  
 public caipinfenleibiao() {  
 // 设置窗口标题和大小  
 setTitle("菜品管理界面");  
 setSize(400, 300);  
  
 // 创建表格和表格模型  
 tableModel = new DefaultTableModel();  
 tableModel.addColumn("id");  
 tableModel.addColumn("菜品分类名称");  
 tableModel.addColumn("描述");  
  
 table = new JTable(tableModel);  
  
 // 创建新增和删除按钮  
 addButton = new JButton("新增");  
 deleteButton = new JButton("删除");  
 saveButton = new JButton("返回");  
 modifyButton = new JButton("修改");  
  
 // 添加按钮事件监听器  
 addButton.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 setVisible(false);  
 new AddDishFrame();  
 }  
 });  
  
 deleteButton.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 // 删除选中的行  
 int selectedRow = table.getSelectedRow();  
 if (selectedRow >= 0) {  
 int id = 0;  
 try {  
 String str = ((String) tableModel.getValueAt(selectedRow, 0));  
 id = Integer.*parseInt*(str);  
 tableModel.removeRow(selectedRow);  
 CategoryDAOImpl categoryDAO = new CategoryDAOImpl();  
 categoryDAO.delete(conn, id);  
 } catch (NumberFormatException e1) {  
 System.*out*.println("字符串无法转换为整数类型");  
 }  
  
 }  
 }  
 });  
 saveButton.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 // 从表格中检索数据并更新数据库  
 setVisible(false);  
 new RestaurantSystem();  
 }  
 });  
  
 modifyButton.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 // 获取选中的行  
 int selectedRow = table.getSelectedRow();  
 if (selectedRow >= 0) {  
 // 获取选中行的数据  
// String name = (String) tableModel.getValueAt(selectedRow, 1);  
// String description = (String) tableModel.getValueAt(selectedRow, 2);  
//  
// // 更新数据库中的数据  
// try {  
// // 建立与数据库的连接  
// JDBCConection db = new JDBCConection();  
// Connection conn = db.getConn();  
//  
// // 更新数据  
// String sql = "UPDATE category SET describ=? WHERE name=?";  
// PreparedStatement stmt = conn.prepareStatement(sql);  
// stmt.setString(1, description);  
// stmt.setString(2, name);  
// stmt.executeUpdate();  
//  
// // 关闭数据库连接和PreparedStatement对象  
// stmt.close();  
// conn.close();  
//  
// // 向用户显示一个消息，表示数据已保存  
// JOptionPane.showMessageDialog(null, "数据已保存。");  
// } catch (SQLException ex) {  
// ex.printStackTrace();  
// }  
 }  
 }  
 });  
  
 // 添加组件到窗口中  
 Container container = getContentPane();  
 container.setLayout(new BorderLayout());  
  
 JPanel buttonPanel = new JPanel();  
 buttonPanel.add(addButton);  
 buttonPanel.add(deleteButton);  
 buttonPanel.add(saveButton);  
  
 container.add(buttonPanel, BorderLayout.*SOUTH*);  
 container.add(new JScrollPane(table), BorderLayout.*CENTER*);  
  
  
 // 从数据库中获取菜品数据并显示  
 try {  
 // 连接到MySQL数据库  
 JDBCConection db = new JDBCConection();  
 Connection conn = db.getConn();  
  
 // 查询菜品数据  
 String sql = "SELECT id,name, describ FROM category";  
 PreparedStatement stmt = conn.prepareStatement(sql);  
 ResultSet rs = stmt.executeQuery();  
  
 // 将查询结果添加到表格模型中EFWCDSFHJGF  
 while (rs.next()) {  
 int id = rs.getInt("id");  
 String name = rs.getString("name");  
 String description = rs.getString("describ");  
 tableModel.addRow(new Object[]{id, name, description});  
 }  
  
 // 关闭数据库连接和查询结果  
 rs.close();  
 stmt.close();  
 conn.close();  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
  
  
 // 设置窗口居中显示  
 setLocationRelativeTo(null);  
  
 // 设置窗口可见  
 setVisible(true);  
 }  
 // 在GUI中增加一个新的按钮，用于保存数据  
  
 public static void main(String[] args) {  
 new caipinfenleibiao();  
 }  
}

package com.canying.bianjie.diancai;  
  
  
import com.canying.kongzhi.\*;  
import com.canying.shiti.Customer;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.Connection;  
  
public class Addcus extends JFrame {  
  
 public Addcus() {  
 JDBCConection db = new JDBCConection();  
 Connection conn = db.getConn();  
 //创建一个主要框架,将其命名为”登录"  
 JFrame jFrame = new JFrame("客户新加");  
  
 //设置窗口大小  
 jFrame.setSize(700, 690);  
  
 //先将布局管理器置为null  
 jFrame.setLayout(null);  
  
  
 //添加标签【学生管理系统】  
 JLabel textStudentManage = new JLabel("请输入新增的客户信息");//创建一个标签,并命名为“欢迎使用餐饮管理系统“  
 textStudentManage.setForeground(new Color(0x0010FF));//设置字体颜色  
 textStudentManage.setFont(new Font("黑体", Font.*PLAIN*, 50));//设置字体大小  
 textStudentManage.setBounds(80, 30, 800, 100);//设置标签的绝对位置  
 jFrame.add(textStudentManage);//向框架中添加组件【标签(学生管理系统)】  
  
  
 //添加标签【用户名】  
 JLabel textUser = new JLabel("姓名:");  
 textUser.setForeground(new Color(0xFF0000));  
 textUser.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 textUser.setBounds(200, 120, 200, 100);  
 jFrame.add(textUser);  
  
 //添加输入框【用户名输入框】  
 JTextField user = new JTextField(20);  
 user.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 user.setSelectedTextColor(new Color(0xFF0000));  
 user.setBounds(310, 150, 280, 40);  
 jFrame.add(user);  
  
 //添加标签【密码】  
 JLabel textPassword = new JLabel("性别:");  
 textPassword.setForeground(new Color(0xFF0000));  
 textPassword.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 textPassword.setBounds(200, 180, 200, 100);  
 jFrame.add(textPassword);  
  
 //添加密码输入框【密码】  
// JTextField password = new JTextField(20);  
// password.setBounds(310, 210, 280, 40);  
// password.setFont(new Font("黑体", Font.PLAIN, 18));  
// password.setSelectedTextColor(new Color(0xFF0000));  
// jFrame.add(password);  
  
 JComboBox<String> sexSelector = new JComboBox<>();  
 sexSelector.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 sexSelector.setBounds(310, 210, 280, 40);  
 jFrame.add(sexSelector);  
  
 sexSelector.addItem("男");  
 sexSelector.addItem("女");  
  
 //添加标签【密码】  
 JLabel unit = new JLabel("单位:");  
 unit.setForeground(new Color(0xFF0000));  
 unit.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 unit.setBounds(200, 240, 200, 100);  
 jFrame.add(unit);  
  
 //添加密码输入框【密码】  
 JTextField unit\_ = new JTextField(20);  
 unit\_.setBounds(310, 270, 280, 40);  
 unit\_.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 unit\_.setSelectedTextColor(new Color(0xFF0000));  
 jFrame.add(unit\_);  
  
 //添加标签【密码】  
 JLabel price = new JLabel("电话:");  
 price.setForeground(new Color(0xFF0000));  
 price.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 price.setBounds(200, 300, 200, 100);  
 jFrame.add(price);  
  
 //添加密码输入框【密码】  
 JTextField price\_ = new JTextField(20);  
 price\_.setBounds(310, 330, 280, 40);  
 price\_.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 price\_.setSelectedTextColor(new Color(0xFF0000));  
 jFrame.add(price\_);  
  
  
 //添加按钮【登录】  
 JButton jButton = new JButton("确定");  
 jButton.setForeground(new Color(0x023BF6));  
 jButton.setBackground(new Color(0x38FF00));  
 jButton.setFont(new Font("黑体", Font.*PLAIN*, 20));  
 jButton.setBorderPainted(false);  
 jButton.setBounds(180, 590, 100, 50);  
 jFrame.add(jButton);  
  
  
 JButton register = new JButton("退出");  
 register.setForeground(new Color(0x0029FF));  
 register.setBackground(new Color(0xECA871));  
 register.setFont(new Font("黑体", Font.*PLAIN*, 20));  
 register.setBorderPainted(false);  
 register.setBounds(400, 590, 100, 50);  
 jFrame.add(register);  
  
 register.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 jFrame.setVisible(false);  
 new Diancai();  
 }  
 });  
  
 //对按钮事件进行处理  
 jButton.addActionListener((e -> {  
  
 String name = user.getText();  
 String sex = (String) sexSelector.getSelectedItem();  
 // String brithday = fenlei\_id.getText();  
 String companey = new String(unit\_.getText());  
 String tel = new String(price\_.getText());  
  
  
 if (name.equals("") || tel.equals("") || companey.equals("")) {  
 JOptionPane.*showMessageDialog*(jFrame, "请输入完整的信息", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 } else {  
 CustomerDAOImpl customerDAO = new CustomerDAOImpl();  
 int id = customerDAO.getMaxid(conn) + 1;  
 int cardid = id + 100000;  
 String cardid1 = Integer.*toString*(cardid);  
 Customer customer = new Customer(id, name, sex, companey, tel, cardid1);  
  
 customerDAO.add(conn, customer);  
  
 JOptionPane.*showMessageDialog*(jFrame, "添加成功!会员号:" + cardid1, "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 user.setText("");  
 unit\_.setText("");  
 price\_.setText("");  
 }  
 }));  
  
 //设置相对位置：屏幕中间  
 jFrame.setLocationRelativeTo(null);  
  
 //确保使用窗口关闭按钮，能够正常退出，结束进程！  
 jFrame.setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  
  
 //禁止对窗口大小进行缩放处理  
 jFrame.setResizable(false);  
  
 //设置可见  
 jFrame.setVisible(true);  
 }  
  
}

package com.canying.bianjie.diancai;  
  
  
import com.canying.kongzhi.\*;  
  
import com.canying.shiti.Order;  
import com.canying.shiti.Orderitem;  
  
import javax.swing.\*;  
import javax.swing.table.DefaultTableModel;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.Connection;  
  
import java.text.SimpleDateFormat;  
import java.util.\*;  
import java.sql.Date;  
import java.util.List;  
  
public class Diancai extends JFrame implements ActionListener {  
 private JTable table;  
 private DefaultTableModel tableModel;  
 private JButton addButton, deleteButton, startButton, addCusButton;  
 JDBCConection db = new JDBCConection();  
 Connection conn = db.getConn();  
  
 public Diancai() {  
 // 设置窗口标题和大小  
 setTitle("点菜信息界面");  
 setSize(400, 300);  
 JPanel labelPanel = new JPanel();  
 labelPanel.add(new JLabel("客户:"));  
 JComboBox<String> customerSelector = new JComboBox<>();  
 customerSelector.setFont(new Font("黑体", Font.*PLAIN*, 10));  
 //categorySelector.setBounds(330, 230, 280, 40);  
 labelPanel.add(customerSelector);  
  
//从数据库中读取类别列表，并添加到下拉框中  
 CustomerDAOImpl customerDAO = new CustomerDAOImpl();  
 java.util.List<String> categoryList = customerDAO.findAllName(conn);  
 for (String s : categoryList) {  
  
 customerSelector.addItem(s);  
 }  
  
 labelPanel.add(new JLabel("餐台:"));  
 JComboBox<String> CantaiSelector = new JComboBox<>();  
 CantaiSelector.setFont(new Font("黑体", Font.*PLAIN*, 10));  
 //categorySelector.setBounds(330, 230, 280, 40);  
 labelPanel.add(CantaiSelector);  
  
//从数据库中读取类别列表，并添加到下拉框中  
 DeskDAOImpl deskDAO = new DeskDAOImpl();  
 java.util.List<String> caitaiList = (List) deskDAO.findAllName(conn);  
 for (String s : caitaiList) {  
  
 CantaiSelector.addItem(s);  
 }  
 labelPanel.add(new JLabel("就餐人数:"));  
 JTextField num = new JTextField(5);  
 labelPanel.add(num);  
 startButton = new JButton("开始点菜");  
 labelPanel.add(startButton);  
 // 创建表格和表格模型  
 tableModel = new DefaultTableModel();  
 tableModel.addColumn("菜品名");  
 tableModel.addColumn("数量");  
  
  
 table = new JTable(tableModel);  
  
 // 添加组件到窗口中  
 Container container = getContentPane();  
 container.setLayout(new BorderLayout());  
 container.add(labelPanel, BorderLayout.*NORTH*);  
 container.add(new JScrollPane(table), BorderLayout.*CENTER*);  
  
 addCusButton = new JButton("新加客户");  
 addCusButton.addActionListener(this);  
 addButton = new JButton("添加");  
 addButton.addActionListener(this);  
 deleteButton = new JButton("删除");  
 deleteButton.addActionListener(this);  
  
  
 JPanel buttonPanel = new JPanel();  
 buttonPanel.setLayout(new FlowLayout());  
 buttonPanel.add(addCusButton);  
 buttonPanel.add(addButton);  
 buttonPanel.add(deleteButton);  
 container.add(buttonPanel, BorderLayout.*SOUTH*);  
  
  
 addCusButton.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 setVisible(false);  
 new Addcus();  
 }  
 });  
 addButton.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 tableModel.addRow(new Object[]{"", ""});  
 }  
 });  
 deleteButton.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
  
 int selectedRow = table.getSelectedRow();  
 tableModel.removeRow(selectedRow);  
 }  
 });  
 startButton.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
// if(num.getText().equals("")){  
// JOptionPane.showMessageDialog(Diancai.this, "请输入完整的信息", "提示", JOptionPane.INFORMATION\_MESSAGE);  
// return;  
// }  
  
 // 创建SimpleDateFormat对象，指定日期格式  
 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");  
// 获取当前时间  
 java.util.Date utilDate = new java.util.Date();  
 Date sqlDate = new Date(utilDate.getTime());  
 System.*out*.println(sqlDate);  
  
// 格式化时间为字符串  
 String currentTime = sdf.format(utilDate);  
 System.*out*.println(currentTime);  
  
  
// 生成随机数  
 Random random = new Random();  
 int num = random.nextInt(9000) + 1000;  
  
// 将随机数添加到日期字符串后面  
 String orderNo = currentTime + num;  
 System.*out*.println(orderNo);  
  
 String customername = (String) customerSelector.getSelectedItem();  
 String deskId = (String) CantaiSelector.getSelectedItem();  
 int id = Integer.*valueOf*(deskId);  
 System.*out*.println(id);  
 CustomerDAOImpl customerDAO = new CustomerDAOImpl();  
 int customerId = customerDAO.findIdByName(conn, customername);  
 System.*out*.println("客户id:" + customerId);  
 OrderDaAOImpl orderDAO = new OrderDaAOImpl();  
 int dinddanid = orderDAO.getmaxid(conn) + 1;  
 System.*out*.println("订单id:" + dinddanid);  
  
  
 Map<Integer, Integer> dishes = new HashMap<>();  
 double sum = 0;  
 for (int i = 0; i < tableModel.getRowCount(); i++) {  
 String dishName = (String) tableModel.getValueAt(i, 0);  
 String dishCount = (String) tableModel.getValueAt(i, 1);  
 int num2 = Integer.*valueOf*(dishCount);  
  
 if (dishName != null && !dishName.isEmpty()) {  
 DishDAOImpl dishDAO = new DishDAOImpl();  
 sum += dishDAO.findMoneyByName(conn, dishName) \* num2;  
 System.*out*.println(sum);  
  
 int dishid = dishDAO.findIdByName(conn, dishName);  
 dishes.put(dishid, num2);  
  
  
 } else {  
 JOptionPane.*showMessageDialog*(Diancai.this, "请输入菜品信息", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 }  
 }  
  
 Order order = new Order(dinddanid, orderNo, id, sqlDate, customerId, sum, "未支付", num);  
  
 orderDAO.add(conn, order);  
 OrderItemDAOImpl orderItemDAO = new OrderItemDAOImpl();  
 for (Map.Entry<Integer, Integer> entry : dishes.entrySet()) {  
 int orderid = orderItemDAO.getmaxid(conn) + 1;  
 int ortid = entry.getKey();  
 System.*out*.println(ortid);  
 int bynum = entry.getValue();  
 System.*out*.println(bynum);  
 Orderitem orderitem = new Orderitem(orderid, dinddanid, ortid, bynum);  
 orderItemDAO.add(conn, orderitem);  
 }  
  
 JOptionPane.*showMessageDialog*(Diancai.this, "下单成功", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 deskDAO.upStatus(conn, id, "就餐中");  
 setVisible(false);  
 new Diancai();  
 System.*out*.println(dishes);  
 }  
  
 });  
  
  
 // 设置窗口居中显示  
 setLocationRelativeTo(null);  
 // 设置窗口可见  
 setVisible(true);  
  
 }  
  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
  
 }  
  
 public static void main(String[] args) {  
 new Diancai();  
 }  
}

package com.canying.bianjie.dish;  
  
import com.canying.kongzhi.CategoryDAOImpl;  
import com.canying.kongzhi.DishDAOImpl;  
import com.canying.kongzhi.JDBCConection;  
import com.canying.shiti.Category;  
import com.canying.shiti.Dish;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.Connection;  
import java.util.List;  
  
public class AddDish extends JFrame {  
  
 public AddDish() {  
 JDBCConection db = new JDBCConection();  
 Connection conn = db.getConn();  
 //创建一个主要框架,将其命名为”登录"  
 JFrame jFrame = new JFrame("菜品新加");  
  
 //设置窗口大小  
 jFrame.setSize(700, 650);  
  
 //先将布局管理器置为null  
 jFrame.setLayout(null);  
  
  
 //添加标签【学生管理系统】  
 JLabel textStudentManage = new JLabel("请输入新增的菜品信息");//创建一个标签,并命名为“欢迎使用餐饮管理系统“  
 textStudentManage.setForeground(new Color(0x0010FF));//设置字体颜色  
 textStudentManage.setFont(new Font("黑体", Font.*PLAIN*, 50));//设置字体大小  
 textStudentManage.setBounds(100, 50, 800, 100);//设置标签的绝对位置  
 jFrame.add(textStudentManage);//向框架中添加组件【标签(学生管理系统)】  
  
  
 //添加标签【用户名】  
 JLabel textUser = new JLabel("菜品名称:");  
 textUser.setForeground(new Color(0xFF0000));  
 textUser.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 textUser.setBounds(180, 140, 200, 100);  
  
  
 jFrame.add(textUser);  
  
 //添加输入框【用户名输入框】  
 JTextField user = new JTextField(20);  
 user.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 user.setSelectedTextColor(new Color(0xFF0000));  
 user.setBounds(330, 170, 280, 40);  
  
 //user.setHorizontalAlignment(JTextField.CENTER);  
  
 jFrame.add(user);  
  
 //添加标签【密码】  
 JLabel textPassword = new JLabel("菜品分类:");  
 textPassword.setForeground(new Color(0xFF0000));  
 textPassword.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 textPassword.setBounds(180, 200, 200, 100);  
 jFrame.add(textPassword);  
  
// //添加密码输入框【密码】  
// JTextField password = new JTextField(20);  
// password.setBounds(330, 230, 280, 40);  
// password.setFont(new Font("黑体", Font.PLAIN, 18));  
// password.setSelectedTextColor(new Color(0xFF0000));  
// jFrame.add(password);  
  
 //添加标签【密码】  
 JLabel fenleiid = new JLabel("菜品代号:");  
 fenleiid.setForeground(new Color(0xFF0000));  
 fenleiid.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 fenleiid.setBounds(180, 260, 200, 100);  
 jFrame.add(fenleiid);  
  
 //添加密码输入框【密码】  
 JTextField fenlei\_id = new JTextField(20);  
 fenlei\_id.setBounds(330, 290, 280, 40);  
 fenlei\_id.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 fenlei\_id.setSelectedTextColor(new Color(0xFF0000));  
 jFrame.add(fenlei\_id);  
  
  
 //添加下拉框【类别选择器】  
 JComboBox<String> categorySelector = new JComboBox<>();  
 categorySelector.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 categorySelector.setBounds(330, 230, 280, 40);  
 jFrame.add(categorySelector);  
  
//从数据库中读取类别列表，并添加到下拉框中  
 DishDAOImpl dishDAO1 = new DishDAOImpl();  
 List<String> categoryList = (List) dishDAO1.findAllName(conn);  
 for (String s : categoryList) {  
 System.*out*.println(s);  
 categorySelector.addItem(s);  
 }  
 //添加标签【密码】  
 JLabel unit = new JLabel("单位:");  
 unit.setForeground(new Color(0xFF0000));  
 unit.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 unit.setBounds(220, 320, 200, 100);  
 jFrame.add(unit);  
  
 //添加密码输入框【密码】  
 JTextField unit\_ = new JTextField(20);  
 unit\_.setBounds(330, 350, 280, 40);  
 unit\_.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 unit\_.setSelectedTextColor(new Color(0xFF0000));  
 jFrame.add(unit\_);  
  
 //添加标签【密码】  
 JLabel price = new JLabel("价格:");  
 price.setForeground(new Color(0xFF0000));  
 price.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 price.setBounds(220, 380, 200, 100);  
 jFrame.add(price);  
  
 //添加密码输入框【密码】  
 JTextField price\_ = new JTextField(20);  
 price\_.setBounds(330, 410, 280, 40);  
 price\_.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 price\_.setSelectedTextColor(new Color(0xFF0000));  
 jFrame.add(price\_);  
  
 //添加标签【密码】  
 JLabel status = new JLabel("状态:");  
 status.setForeground(new Color(0xFF0000));  
 status.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 status.setBounds(220, 440, 200, 100);  
 jFrame.add(status);  
  
 //添加密码输入框【密码】  
 JTextField status\_ = new JTextField(20);  
 status\_.setBounds(330, 470, 280, 40);  
 status\_.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 status\_.setSelectedTextColor(new Color(0xFF0000));  
 jFrame.add(status\_);  
  
  
 //添加按钮【登录】  
 JButton jButton = new JButton("确定");  
 jButton.setForeground(new Color(0x023BF6));  
 jButton.setBackground(new Color(0x38FF00));  
 jButton.setFont(new Font("黑体", Font.*PLAIN*, 20));  
 jButton.setBorderPainted(false);  
 jButton.setBounds(180, 530, 100, 50);  
 jFrame.add(jButton);  
  
  
 JButton register = new JButton("退出");  
 register.setForeground(new Color(0x0029FF));  
 register.setBackground(new Color(0xECA871));  
 register.setFont(new Font("黑体", Font.*PLAIN*, 20));  
 register.setBorderPainted(false);  
 register.setBounds(400, 530, 100, 50);  
 jFrame.add(register);  
  
 register.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 jFrame.setVisible(false);  
 new Dishbiao();  
 }  
 });  
  
 //对按钮事件进行处理  
 jButton.addActionListener((e -> {  
  
 String name = new String(user.getText());  
 String categoryid = (String) categorySelector.getSelectedItem();  
 System.*out*.println(categoryid);  
 String code = new String(fenlei\_id.getText());  
 String unit1 = new String(unit\_.getText());  
 String price1 = new String(price\_.getText());  
 String status1 = new String(status\_.getText());  
 if (name.equals("") || code.equals("") || unit1.equals("") || price1.equals("") || status1.equals("")) {  
 JOptionPane.*showMessageDialog*(jFrame, "请输入完整的信息", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 } else {  
 DishDAOImpl dishDAO = new DishDAOImpl();  
 int id2 = dishDAO.findIdByName(conn, categoryid);  
 System.*out*.println(id2);  
 double price2 = Integer.*valueOf*(price1);  
 int id = DishDAOImpl.*getmaxid*(conn);  
 Dish.*setId*(id);  
 Dish dish = new Dish(name, id2, code, unit1, price2, status1);  
  
 System.*out*.println(123);  
  
 dishDAO.add(conn, dish);  
 System.*out*.println(123);  
  
 JOptionPane.*showMessageDialog*(jFrame, "添加成功", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 user.setText("");  
 fenlei\_id.setText("");  
 unit\_.setText("");  
 price\_.setText("");  
 status\_.setText("");  
 }  
 }));  
  
 //设置相对位置：屏幕中间  
 jFrame.setLocationRelativeTo(null);  
  
 //确保使用窗口关闭按钮，能够正常退出，结束进程！  
 jFrame.setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  
  
 //禁止对窗口大小进行缩放处理  
 jFrame.setResizable(false);  
  
 //设置可见  
 jFrame.setVisible(true);  
 }  
  
}

package com.canying.bianjie.dish;  
  
  
//import com.canying.bianjie.RestaurantSystem;  
import com.canying.bianjie.RestaurantSystem;  
import com.canying.kongzhi.CategoryDAOImpl;  
import com.canying.kongzhi.DishDAOImpl;  
import com.canying.kongzhi.JDBCConection;  
  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
  
import javax.swing.\*;  
import javax.swing.table.DefaultTableModel;  
  
public class Dishbiao extends JFrame {  
 private JTable table;  
 private DefaultTableModel tableModel;  
 private JButton addButton;  
 private JButton deleteButton;  
 private JButton saveButton;  
  
 public Dishbiao() {  
 JDBCConection db = new JDBCConection();  
 Connection conn = db.getConn();  
 // 设置窗口标题和大小  
 setTitle("菜品管理界面");  
 setSize(400, 300);  
  
 // 创建表格和表格模型  
 tableModel = new DefaultTableModel();  
 tableModel.addColumn("菜品名");  
 tableModel.addColumn("类别编号");  
 tableModel.addColumn("菜品代码");  
 tableModel.addColumn("单位");  
 tableModel.addColumn("价格");  
 tableModel.addColumn("状态");  
  
  
 table = new JTable(tableModel);  
  
 // 创建新增和删除按钮  
 addButton = new JButton("新增");  
 deleteButton = new JButton("删除");  
 saveButton = new JButton("返回");  
  
 // 添加按钮事件监听器  
 addButton.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 // 在表格中添加一行空数据  
 setVisible(false);  
 new AddDish();  
 }  
 });  
  
 deleteButton.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 // 删除选中的行  
 int selectedRow = table.getSelectedRow();  
  
 if (selectedRow >= 0) {  
  
 DishDAOImpl dishDAO = new DishDAOImpl();  
 try {  
 String s = ((String) tableModel.getValueAt(selectedRow, 0));  
 tableModel.removeRow(selectedRow);  
  
 dishDAO.delete(conn, s);  
 } catch (NumberFormatException e1) {  
 System.*out*.println("字符串无法转换为整数类型");  
 }  
 }  
 }  
 });  
 saveButton.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 setVisible(false);  
 new RestaurantSystem();  
 }  
 });  
  
 // 添加组件到窗口中  
 Container container = getContentPane();  
 container.setLayout(new BorderLayout());  
  
 JPanel buttonPanel = new JPanel();  
 buttonPanel.add(addButton);  
 buttonPanel.add(deleteButton);  
 buttonPanel.add(saveButton);  
  
 container.add(buttonPanel, BorderLayout.*SOUTH*);  
 container.add(new JScrollPane(table), BorderLayout.*CENTER*);  
  
 // 从数据库中获取菜品数据并显示  
 try {  
 // 查询菜品数据  
 String sql = "SELECT name,categoryid,code,unit,price,status FROM dish";  
 PreparedStatement stmt = conn.prepareStatement(sql);  
 ResultSet rs = stmt.executeQuery();  
  
 // 将查询结果添加到表格模型中EFWCDSFHJGF  
 while (rs.next()) {  
 String name = rs.getString("name");  
 int categoryid = rs.getInt("categoryid");  
 String code = rs.getString("code");  
 String unit = rs.getString("unit");  
 String price = rs.getString("price");  
 String status = rs.getString("status");  
 tableModel.addRow(new Object[]{name, categoryid, code, unit, price, status});  
 }  
  
  
 // 设置窗口居中显示  
 setLocationRelativeTo(null);  
  
 // 设置窗口可见  
 setVisible(true);  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 }  
  
 public static void main(String[] args) {  
 new Dishbiao();  
 }  
}

package com.canying.bianjie.staff;  
  
  
import com.canying.bianjie.dish.Dishbiao;  
import com.canying.kongzhi.\*;  
import com.canying.shiti.Category;  
import com.canying.shiti.Dish;  
import com.canying.shiti.Staff;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.Connection;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.List;  
  
public class AddStaff extends JFrame {  
  
 public AddStaff() {  
 JDBCConection db = new JDBCConection();  
 Connection conn = db.getConn();  
 //创建一个主要框架,将其命名为”登录"  
 JFrame jFrame = new JFrame("员工新加");  
  
 //设置窗口大小  
 jFrame.setSize(700, 690);  
  
 //先将布局管理器置为null  
 jFrame.setLayout(null);  
  
  
 //添加标签【学生管理系统】  
 JLabel textStudentManage = new JLabel("请输入新增的员工信息");//创建一个标签,并命名为“欢迎使用餐饮管理系统“  
 textStudentManage.setForeground(new Color(0x0010FF));//设置字体颜色  
 textStudentManage.setFont(new Font("黑体", Font.*PLAIN*, 50));//设置字体大小  
 textStudentManage.setBounds(80, 30, 800, 100);//设置标签的绝对位置  
 jFrame.add(textStudentManage);//向框架中添加组件【标签(学生管理系统)】  
  
  
 //添加标签【用户名】  
 JLabel textUser = new JLabel("姓名:");  
 textUser.setForeground(new Color(0xFF0000));  
 textUser.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 textUser.setBounds(200, 120, 200, 100);  
 jFrame.add(textUser);  
  
 //添加输入框【用户名输入框】  
 JTextField user = new JTextField(20);  
 user.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 user.setSelectedTextColor(new Color(0xFF0000));  
 user.setBounds(310, 150, 280, 40);  
 jFrame.add(user);  
  
 //添加标签【密码】  
 JLabel textPassword = new JLabel("性别:");  
 textPassword.setForeground(new Color(0xFF0000));  
 textPassword.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 textPassword.setBounds(200, 180, 200, 100);  
 jFrame.add(textPassword);  
  
 //添加密码输入框【密码】  
// JTextField password = new JTextField(20);  
// password.setBounds(310, 210, 280, 40);  
// password.setFont(new Font("黑体", Font.PLAIN, 18));  
// password.setSelectedTextColor(new Color(0xFF0000));  
// jFrame.add(password);  
  
 JComboBox<String> sexSelector = new JComboBox<>();  
 sexSelector.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 sexSelector.setBounds(310, 210, 280, 40);  
 jFrame.add(sexSelector);  
  
 sexSelector.addItem("男");  
 sexSelector.addItem("女");  
  
 //添加标签【密码】  
 JLabel unit = new JLabel("身份证号:");  
 unit.setForeground(new Color(0xFF0000));  
 unit.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 unit.setBounds(160, 240, 200, 100);  
 jFrame.add(unit);  
  
 //添加密码输入框【密码】  
 JTextField unit\_ = new JTextField(20);  
 unit\_.setBounds(310, 270, 280, 40);  
 unit\_.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 unit\_.setSelectedTextColor(new Color(0xFF0000));  
 jFrame.add(unit\_);  
  
 //添加标签【密码】  
 JLabel price = new JLabel("地址:");  
 price.setForeground(new Color(0xFF0000));  
 price.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 price.setBounds(200, 300, 200, 100);  
 jFrame.add(price);  
  
 //添加密码输入框【密码】  
 JTextField price\_ = new JTextField(20);  
 price\_.setBounds(310, 330, 280, 40);  
 price\_.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 price\_.setSelectedTextColor(new Color(0xFF0000));  
 jFrame.add(price\_);  
  
 //添加标签【密码】  
 JLabel status = new JLabel("电话:");  
 status.setForeground(new Color(0xFF0000));  
 status.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 status.setBounds(200, 360, 200, 100);  
 jFrame.add(status);  
  
 //添加密码输入框【密码】  
 JTextField status\_ = new JTextField(20);  
 status\_.setBounds(310, 390, 280, 40);  
 status\_.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 status\_.setSelectedTextColor(new Color(0xFF0000));  
 jFrame.add(status\_);  
  
 //添加标签【密码】  
 JLabel position = new JLabel("职务:");  
 position.setForeground(new Color(0xFF0000));  
 position.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 position.setBounds(200, 420, 200, 100);  
 jFrame.add(position);  
  
 JComboBox<String> categorySelector = new JComboBox<>();  
 categorySelector.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 categorySelector.setBounds(310, 450, 280, 40);  
 jFrame.add(categorySelector);  
  
 categorySelector.addItem("服务员");  
 categorySelector.addItem("收银员");  
  
 //添加标签【密码】  
 JLabel freeze = new JLabel("是否在职:");  
 freeze.setForeground(new Color(0xFF0000));  
 freeze.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 freeze.setBounds(160, 480, 200, 100);  
 jFrame.add(freeze);  
  
 //添加下拉框【类别选择器】  
 JComboBox<String> ifSelector = new JComboBox<>();  
 ifSelector.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 ifSelector.setBounds(310, 510, 280, 40);  
 jFrame.add(ifSelector);  
  
 ifSelector.addItem("是");  
 ifSelector.addItem("否");  
  
  
 //添加按钮【登录】  
 JButton jButton = new JButton("确定");  
 jButton.setForeground(new Color(0x023BF6));  
 jButton.setBackground(new Color(0x38FF00));  
 jButton.setFont(new Font("黑体", Font.*PLAIN*, 20));  
 jButton.setBorderPainted(false);  
 jButton.setBounds(180, 590, 100, 50);  
 jFrame.add(jButton);  
  
  
 JButton register = new JButton("退出");  
 register.setForeground(new Color(0x0029FF));  
 register.setBackground(new Color(0xECA871));  
 register.setFont(new Font("黑体", Font.*PLAIN*, 20));  
 register.setBorderPainted(false);  
 register.setBounds(400, 590, 100, 50);  
 jFrame.add(register);  
  
 register.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 jFrame.setVisible(false);  
 new Staffbiao();  
 }  
 });  
  
 //对按钮事件进行处理  
 jButton.addActionListener((e -> {  
  
 String name = user.getText();  
 String sex = (String) sexSelector.getSelectedItem();  
 String identityID = new String(unit\_.getText());  
 String adress = new String(price\_.getText());  
 String tel = new String(status\_.getText());  
 String position1 = (String) categorySelector.getSelectedItem();  
 String freeze1 = (String) ifSelector.getSelectedItem();  
 if (freeze1.equals("") || name.equals("") || identityID.equals("") || adress.equals("") || tel.equals("")) {  
 JOptionPane.*showMessageDialog*(jFrame, "请输入完整的信息", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 } else {  
 EmployeeDaoImpl employeeDao = new EmployeeDaoImpl();  
  
 int id = EmployeeDaoImpl.*getmaxid*(conn);  
 Staff.*setNum*(id + 1);  
 Staff staff = new Staff(name, sex, identityID, adress, tel, position1, freeze1);  
  
  
 EmployeeDaoImpl employeeDao1 = new EmployeeDaoImpl();  
 employeeDao1.add(conn, staff);  
  
  
 JOptionPane.*showMessageDialog*(jFrame, "添加成功", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 user.setText("");  
 unit\_.setText("");  
 price\_.setText("");  
 status\_.setText("");  
 }  
 }));  
  
 //设置相对位置：屏幕中间  
 jFrame.setLocationRelativeTo(null);  
  
 //确保使用窗口关闭按钮，能够正常退出，结束进程！  
 jFrame.setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  
  
 //禁止对窗口大小进行缩放处理  
 jFrame.setResizable(false);  
  
 //设置可见  
 jFrame.setVisible(true);  
 }  
}

package com.canying.bianjie.staff;  
  
  
import com.canying.bianjie.RestaurantSystem;  
import com.canying.kongzhi.CategoryDAOImpl;  
import com.canying.kongzhi.DishDAOImpl;  
import com.canying.kongzhi.EmployeeDaoImpl;  
import com.canying.kongzhi.JDBCConection;  
  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.\*;  
  
import javax.swing.\*;  
import javax.swing.table.DefaultTableModel;  
  
public class Staffbiao extends JFrame {  
 private JTable table;  
 private DefaultTableModel tableModel;  
 private JButton addButton;  
 private JButton deleteButton;  
 private JButton saveButton;  
  
 public Staffbiao() {  
 JDBCConection db = new JDBCConection();  
 Connection conn = db.getConn();  
 // 设置窗口标题和大小  
 setTitle("菜品管理界面");  
 setSize(600, 300);  
  
 // 创建表格和表格模型  
 tableModel = new DefaultTableModel();  
 tableModel.addColumn("序号");  
 tableModel.addColumn("用户名");  
 tableModel.addColumn("性别");  
 tableModel.addColumn("身份证号");  
 tableModel.addColumn("家庭住址");  
 tableModel.addColumn("电话");  
 tableModel.addColumn("职位");  
 tableModel.addColumn("是否在职");  
  
  
 table = new JTable(tableModel);  
  
 // 创建新增和删除按钮  
 addButton = new JButton("新增");  
 deleteButton = new JButton("删除");  
 saveButton = new JButton("返回");  
  
 // 添加按钮事件监听器  
 addButton.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 // 在表格中添加一行空数据  
 setVisible(false);  
 new AddStaff();  
 }  
 });  
  
 deleteButton.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 // 删除选中的行  
 int selectedRow = table.getSelectedRow();  
 if (selectedRow >= 0) {  
  
 EmployeeDaoImpl employeeDao = new EmployeeDaoImpl();  
 try {  
 String s = ((String) tableModel.getValueAt(selectedRow, 0));  
 int id = Integer.*valueOf*(s);  
 tableModel.removeRow(selectedRow);  
  
 employeeDao.delete(conn, id);  
 } catch (NumberFormatException e1) {  
 System.*out*.println("字符串无法转换为整数类型");  
 }  
 }  
 }  
 });  
 saveButton.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 setVisible(false);  
 new RestaurantSystem();  
 }  
 });  
  
 // 添加组件到窗口中  
 Container container = getContentPane();  
 container.setLayout(new BorderLayout());  
  
 JPanel buttonPanel = new JPanel();  
 buttonPanel.add(addButton);  
 buttonPanel.add(deleteButton);  
 buttonPanel.add(saveButton);  
  
 container.add(buttonPanel, BorderLayout.*SOUTH*);  
 container.add(new JScrollPane(table), BorderLayout.*CENTER*);  
  
 // 从数据库中获取菜品数据并显示  
 try {  
 // 查询菜品数据  
 String sql = "SELECT \* FROM staff";  
 PreparedStatement stmt = conn.prepareStatement(sql);  
 ResultSet rs = stmt.executeQuery();  
  
 // 将查询结果添加到表格模型中EFWCDSFHJGF  
 while (rs.next()) {  
 int id = rs.getInt("id");  
 String name = rs.getString("name");  
 String sex = rs.getString("sex");  
 String identityID = rs.getString("identityID");  
 String adress = rs.getString("address");  
 String tel = rs.getString("tel");  
 String position = rs.getString("position");  
 String freeze = rs.getString("freeze");  
  
 tableModel.addRow(new Object[]{id, name, sex, identityID, adress, tel, position, freeze});  
 }  
  
  
 // 设置窗口居中显示  
 setLocationRelativeTo(null);  
  
 // 设置窗口可见  
 setVisible(true);  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 }  
}

package com.canying.bianjie;  
  
import com.canying.bianjie.diancai.Diancai;  
  
import com.canying.bianjie.zhaoling.ChangeCalculator;  
import com.canying.kongzhi.EmployeeDaoImpl;  
import com.canying.kongzhi.JDBCConection;  
import com.canying.kongzhi.UserDaoImpl;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.io.IOException;  
import java.sql.Connection;  
  
public class LoginFranme {  
 public static void main(String[] args) {  
  
 JDBCConection db = new JDBCConection();  
 Connection conn = db.getConn();  
 //创建一个主要框架,将其命名为”登录"  
 JFrame jFrame = new JFrame("欢迎进入餐饮管理系统");  
  
 //设置窗口大小  
 jFrame.setSize(900, 507);  
  
 //先将布局管理器置为null  
 jFrame.setLayout(null);  
  
  
 //添加标签【学生管理系统】  
 JLabel textStudentManage = new JLabel("欢迎进入餐饮管理系统");//创建一个标签,并命名为“欢迎使用餐饮管理系统“  
 textStudentManage.setForeground(new Color(0x0010FF));//设置字体颜色  
 textStudentManage.setFont(new Font("黑体", Font.*PLAIN*, 50));//设置字体大小  
 textStudentManage.setBounds(200, 50, 800, 100);//设置标签的绝对位置  
 jFrame.add(textStudentManage);//向框架中添加组件【标签(学生管理系统)】  
  
 // 图片需要调用ImageIcon类 构造方法（文件路径+文件名.格式）  
 //图片是内容 不是组件 所以图片对象必须要绑定到组件上才能显示到界面上  
 ImageIcon icon = new ImageIcon("C:\\Users\\XZQ\\Desktop\\桥牌照片\\照片\\shu.jpg");  
 JLabel labIcon = new JLabel(icon); // 用标签来接收图片，实例化JLabel标签对象，该对象显示icon图标  
 Dimension dim = new Dimension(400, 300); //调用Dimension类的构造方法，实例化dim对象 尺寸 dimension  
 labIcon.setPreferredSize(dim); //优先考虑尺寸  
 jFrame.setBackground(null);  
 jFrame.add(labIcon); //将标签对象 labicon加载到窗体frame上  
  
 //添加标签【用户名】  
 JLabel textUser = new JLabel("用户名:");  
 textUser.setForeground(new Color(0xFF0000));  
 textUser.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 textUser.setBounds(200, 140, 200, 100);  
 jFrame.add(textUser);  
  
 //添加输入框【用户名输入框】  
 JTextField user = new JTextField(20);  
 user.setFont(new Font("黑体", Font.*PLAIN*, 18));  
 user.setSelectedTextColor(new Color(0xFF0000));  
 user.setBounds(330, 170, 280, 40);  
 jFrame.add(user);  
  
 //添加标签【密码】  
 JLabel textPassword = new JLabel("密码 :");  
 textPassword.setForeground(new Color(0xFF0000));  
 textPassword.setFont(new Font("黑体", Font.*PLAIN*, 30));  
 textPassword.setBounds(200, 200, 200, 100);  
 jFrame.add(textPassword);  
  
 //添加密码输入框【密码】  
 JPasswordField password = new JPasswordField(20);  
 password.setBounds(330, 230, 280, 40);  
 jFrame.add(password);  
  
 //添加按钮【登录】  
 JButton jButton = new JButton("登录");  
 jButton.setForeground(new Color(0x023BF6));  
 jButton.setBackground(new Color(0x38FF00));  
 jButton.setFont(new Font("黑体", Font.*PLAIN*, 20));  
 jButton.setBorderPainted(false);  
 jButton.setBounds(300, 330, 100, 50);  
 jFrame.add(jButton);  
  
 //添加按钮【注册】  
 JButton register = new JButton("退出");  
 register.setForeground(new Color(0x0029FF));  
 register.setBackground(new Color(0xECA871));  
 register.setFont(new Font("黑体", Font.*PLAIN*, 20));  
 register.setBorderPainted(false);  
 register.setBounds(500, 330, 100, 50);  
 jFrame.add(register);  
  
 register.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 jFrame.setVisible(false);  
 }  
 });  
  
  
 //对按钮事件进行处理  
 jButton.addActionListener((e -> {  
  
 String yonghu = new String(user.getText());  
 String pwd = new String(password.getPassword());  
  
  
 if (user.getText().equals("") && pwd.equals("")) {  
 JOptionPane.*showMessageDialog*(jFrame, "请输入完整的信息", "提示", JOptionPane.*INFORMATION\_MESSAGE*);  
 } else {  
 //不为空，查账户  
 UserDaoImpl userDao = new UserDaoImpl();  
  
 if (userDao.finduser(conn, yonghu)) {//管理员中找到名字  
 if (pwd.equals(userDao.findpwd(conn, yonghu))) {  
 new RestaurantSystem();  
 }  
 jFrame.setVisible(false);//将登录界面设定为不可见  
 // new yonghujiemian().StudentMainInterface(yonghu);  
 } else {  
 //管理员中没找到，职工表中找  
 EmployeeDaoImpl employeeDao = new EmployeeDaoImpl();  
 if (employeeDao.findStaff(conn, yonghu)) {  
 if (pwd.equals(employeeDao.findPwd(conn, yonghu))) {  
 if (employeeDao.findposition(conn, yonghu, pwd).equals("服务员")) {  
 new Diancai();  
 jFrame.setVisible(false);  
 } else {  
 System.*out*.println(123);  
 System.*out*.println(employeeDao.findposition(conn, yonghu, pwd));  
 new ChangeCalculator();  
  
 }  
  
 }  
 }  
 user.setText("");  
 password.setText("");  
 }  
 }  
 }));  
  
 //设置相对位置：屏幕中间  
 jFrame.setLocationRelativeTo(null);  
  
 //确保使用窗口关闭按钮，能够正常退出，结束进程！  
 jFrame.setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  
  
 //禁止对窗口大小进行缩放处理  
 jFrame.setResizable(false);  
  
 //设置可见  
 jFrame.setVisible(true);  
 }  
  
}

package com.canying.bianjie;  
  
import com.canying.bianjie.cantai.Cantai;  
import com.canying.bianjie.category.caipinfenleibiao;  
import com.canying.bianjie.diancai.Diancai;  
import com.canying.bianjie.dish.Dishbiao;  
import com.canying.bianjie.staff.Staffbiao;  
import com.canying.bianjie.zhaoling.ChangeCalculator;  
  
import javax.swing.\*;  
import javax.swing.border.Border;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class RestaurantSystem extends JFrame implements ActionListener {  
 private JButton orderButton, menuButton, employeeButton, exitButton, Button1, Button2, Button3;  
 private JPanel contentPanel;  
  
 public RestaurantSystem() {  
 // 设置窗口标题和大小  
 setTitle("餐饮系统");  
 setSize(600, 300);  
  
 // 创建组件  
 orderButton = new JButton("员工管理");  
 orderButton.addActionListener(this);  
 menuButton = new JButton("餐台管理");  
 menuButton.addActionListener(this);  
 employeeButton = new JButton("菜品分类管理");  
 employeeButton.addActionListener(this);  
 Button1 = new JButton("菜品管理");  
 Button1.addActionListener(this);  
 Button3 = new JButton("业务管理");  
 Button3.addActionListener(this);  
 Button2 = new JButton("结账管理");  
 Button2.addActionListener(this);  
 exitButton = new JButton("退出");  
 exitButton.addActionListener(this);  
  
 // 设置按钮的边框和间距  
 Border emptyBorder = BorderFactory.*createEmptyBorder*(5, 10, 5, 10);  
 orderButton.setBorder(emptyBorder);  
 menuButton.setBorder(emptyBorder);  
 employeeButton.setBorder(emptyBorder);  
 Button1.setBorder(emptyBorder);  
 Button2.setBorder(emptyBorder);  
 Button3.setBorder(emptyBorder);  
 exitButton.setBorder(emptyBorder);  
  
 // 将按钮添加到面板中  
 JPanel buttonPanel = new JPanel();  
 buttonPanel.setLayout(new BoxLayout(buttonPanel, BoxLayout.*X\_AXIS*));  
 buttonPanel.add(Box.*createHorizontalGlue*());  
 buttonPanel.add(orderButton);  
 buttonPanel.add(Box.*createHorizontalStrut*(10));  
 buttonPanel.add(menuButton);  
 buttonPanel.add(Box.*createHorizontalStrut*(10));  
 buttonPanel.add(employeeButton);  
 buttonPanel.add(Box.*createHorizontalStrut*(10));  
 buttonPanel.add(Button1);  
 buttonPanel.add(Box.*createHorizontalStrut*(10));  
 buttonPanel.add(Button3);  
 buttonPanel.add(Box.*createHorizontalStrut*(10));  
 buttonPanel.add(Button2);  
 buttonPanel.add(Box.*createHorizontalStrut*(10));  
 buttonPanel.add(exitButton);  
 buttonPanel.add(Box.*createHorizontalGlue*());  
  
 // 将面板添加到窗口中  
 Container container = getContentPane();  
 container.setLayout(new BorderLayout());  
 container.add(buttonPanel, BorderLayout.*NORTH*);  
  
 // 设置窗口居中显示  
 setLocationRelativeTo(null);  
  
 // 设置窗口可见  
 setVisible(true);  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == orderButton) {  
 setVisible(false);  
 new Staffbiao();  
 } else if (e.getSource() == menuButton) {  
 setVisible(false);  
 new Cantai();  
 } else if (e.getSource() == employeeButton) {  
 setVisible(false);  
 new caipinfenleibiao();  
 } else if (e.getSource() == Button1) {  
 setVisible(false);  
 new Dishbiao();  
 } else if (e.getSource() == Button2) {  
 new ChangeCalculator();  
 } else if (e.getSource() == Button3) {  
 new Diancai();  
 } else if (e.getSource() == exitButton) {  
 // 退出程序  
 System.*exit*(0);  
 }  
 }  
  
 public static void main(String[] args) {  
 new RestaurantSystem();  
 }  
}

package com.canying.kongzhi;  
  
import com.canying.shiti.Category;  
  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
  
public class CategoryDAOImpl implements IBaseDao<Category> {  
 private PreparedStatement preStmt;  
  
 @Override  
 public void add(Connection conn, Category category) {  
 try {  
 String sql = "insert into category (id,name,describ) " + "values (?,?,?)";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setInt(1, category.getId());  
 preStmt.setString(2, category.getName());  
 preStmt.setString(3, category.getDescrib());  
 preStmt.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void delete(Connection conn, int id) {  
 try {  
 String sql = "delete from category where id= ?";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setInt(1, id);  
 preStmt.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void update(Connection conn, int id, String val) {  
 try {  
 String sql = "update category set name =? where id = ?";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setString(1, val);  
 preStmt.setInt(2, id);  
 preStmt.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void cha(Connection conn) {  
 try {  
 String sql = "select \* from staff";  
 preStmt = conn.prepareStatement(sql);  
 ResultSet rs = preStmt.executeQuery();  
 way(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 public int getmaxid(Connection conn) {  
 int id = 0;  
 try {  
 String sql = "SELECT MAX(id) FROM category";  
 preStmt = conn.prepareStatement(sql);  
 ResultSet rs = preStmt.executeQuery();  
 if (rs.next()) {  
 id = rs.getInt(1);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 return id;  
  
 }  
  
 public void way(ResultSet rs) {  
 try {  
 while (rs.next()) {  
 System.*out*.print(rs.getInt("id") + "\t");  
 System.*out*.print(rs.getString("username") + "\t");//可以用大写，数据库大小写不敏感  
 System.*out*.print(rs.getString("password") + "\t");  
 System.*out*.print("\n");  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
}

package com.canying.kongzhi;  
  
import com.canying.shiti.Customer;  
  
import javax.management.MBeanAttributeInfo;  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.Statement;  
import java.util.ArrayList;  
import java.util.List;  
  
public class CustomerDAOImpl implements IBaseDao<Customer> {  
 private static PreparedStatement *preStmt*;  
  
 private Statement stmt;  
 public List<String> findAllName(Connection conn){  
 List<String> List=new ArrayList<>();  
 try {  
 String sql = "SELECT name FROM customer";  
 *preStmt* = conn.prepareStatement(sql);  
 ResultSet rs = *preStmt*.executeQuery();  
 while(rs.next()) {  
 List.add(rs.getString("name"));  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return List;  
 }  
 @Override  
 public void add(Connection conn, Customer customer) {  
 try {  
 String sql = "insert into customer (id,name,sex,company,tel,cardID) " + "values (?,?,?,?,?,?)";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setInt(1, customer.getId());  
 *preStmt*.setString(2, customer.getName());  
 *preStmt*.setString(3, customer.getSex());  
 *preStmt*.setString(4, customer.getCompany());  
 *preStmt*.setString(5, customer.getTel());  
 *preStmt*.setString(6, customer.getCardID());  
 *preStmt*.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void delete(Connection conn, int id) {  
 try {  
 String sql = "delete from customer where id= ?";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setInt(1, id);  
 *preStmt*.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
  
 @Override  
 public void update(Connection conn, int id, String val) {  
 try {  
 String sql = "update customer set name =? where id = ?";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setString(1, val);  
 *preStmt*.setInt(2, id);  
 *preStmt*.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void cha(Connection conn) {  
 try {  
 String sql = "select \* from staff";  
 *preStmt* = conn.prepareStatement(sql);  
 ResultSet rs = *preStmt*.executeQuery();  
 way(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 public int findIdByName(Connection conn,String name) {  
 int id=1;  
 try {  
 String sql = "select id from customer where name = ?";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setString(1, name);  
 ResultSet rs = preStmt.executeQuery();  
 if(rs.next()) {  
 id = rs.getInt("id");  
 }  
 return id;  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return id;  
 }  
 public static String findNameById(Connection conn,int id) {  
 String name="";  
 try {  
 String sql = "select name from customer where id = ?";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setInt(1, id);  
 ResultSet rs = preStmt.executeQuery();  
 if(rs.next()) {  
 name = rs.getString("name");  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return name;  
 }  
 public void way(ResultSet rs) {  
 try {  
 while (rs.next()) {  
 System.out.print(rs.getInt("id") + "\t");  
 System.out.print(rs.getString("username") + "\t");//可以用大写，数据库大小写不敏感  
 System.out.print(rs.getString("password") + "\t");  
 System.out.print("\n");  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
  
  
 public int getMaxid(Connection conn) {  
 int id = 0;  
 try {  
 String sql = "SELECT MAX(id) FROM customer";  
 *preStmt* = conn.prepareStatement(sql);  
 ResultSet rs = *preStmt*.executeQuery();  
 if (rs.next()) {  
 id = rs.getInt(1);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 return id;  
  
 }  
}

package com.canying.kongzhi;  
  
import com.canying.shiti.Desk;  
  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.util.ArrayList;  
import java.util.List;  
  
public class DeskDAOImpl implements IBaseDao<Desk>{  
 private PreparedStatement preStmt;  
 public void upStatus(Connection conn,int id,String status){  
 try {  
 String sql = "update desk set status =? where id = ?";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setString(1, status);  
 preStmt.setInt(2, id);  
 preStmt.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 @Override  
 public void add(Connection conn, Desk desk) {  
 try {  
 String sql = "insert into desk (id,no,seating,status) " + "values (?,?,?,?)";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setInt(1, desk.getId());  
 preStmt.setString(2, desk.getNo());  
 preStmt.setInt(3, desk.getSeating());  
 preStmt.setString(4, desk.getStatus());  
 preStmt.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void delete(Connection conn, int id) {  
 try {  
 String sql = "delete from desk where id= ?";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setInt(1, id);  
 preStmt.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void update(Connection conn, int id, String val) {  
 try {  
 String sql = "update desk set seating =? where id = ?";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setString(1, val);  
 preStmt.setInt(2, id);  
 preStmt.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void cha(Connection conn) {  
 try {  
 String sql = "select \* from staff";  
 preStmt = conn.prepareStatement(sql);  
 ResultSet rs = preStmt.executeQuery();  
 way(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 public void way(ResultSet rs) {  
 try {  
 while (rs.next()) {  
 System.*out*.print(rs.getInt("id") + "\t");  
 System.*out*.print(rs.getString("username") + "\t");//可以用大写，数据库大小写不敏感  
 System.*out*.print(rs.getString("password") + "\t");  
 System.*out*.print("\n");  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 public List<String> findAllName(Connection conn) {  
 List<String> list=new ArrayList<String>();  
  
 try {  
 String sql = "SELECT \* FROM desk WHERE status NOT IN ('已预定', '就餐中')";  
 preStmt = conn.prepareStatement(sql);  
 ResultSet rs = preStmt.executeQuery();  
 while (rs.next()) {  
 list.add(rs.getString("no"));  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return list;  
 }  
  
 public Object findAllName2(Connection conn) {  
  
 List<String> list=new ArrayList<String>();  
  
 try {  
 String sql = "SELECT \* FROM desk WHERE status NOT IN ('已预定', '就餐中')";  
 preStmt = conn.prepareStatement(sql);  
 ResultSet rs = preStmt.executeQuery();  
 while (rs.next()) {  
 list.add(rs.getString("no"));  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return list;  
 }  
  
 public int getMaxId(Connection conn) {  
 int id = 0;  
 try {  
 String sql = "SELECT MAX(id) FROM desk";  
 preStmt = conn.prepareStatement(sql);  
 ResultSet rs = preStmt.executeQuery();  
 if (rs.next()) {  
 id = rs.getInt(1);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 return id;  
  
 }  
}

package com.canying.kongzhi;  
  
import com.canying.shiti.Dish;  
  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.util.ArrayList;  
import java.util.List;  
  
public class DishDAOImpl implements IBaseDao<Dish>{  
 private static PreparedStatement *preStmt*;  
  
 public int findMoneyByName(Connection conn,String name){  
 int money=0;  
 try {  
 String sql = "SELECT price FROM dish where name = ?";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setString(1, name);  
 ResultSet rs = *preStmt*.executeQuery();  
 if (rs.next()) {  
 money =(int)rs.getDouble("price");  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return money;  
 }  
 public static int getmaxid(Connection conn) {  
 int id = 0;  
 try {  
 String sql = "SELECT MAX(id) FROM dish";  
 *preStmt* = conn.prepareStatement(sql);  
 ResultSet rs = *preStmt*.executeQuery();  
 if (rs.next()) {  
 id = rs.getInt(1);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 return id;  
 }  
  
 @Override  
 public void add(Connection conn, Dish dish) {  
 try {  
 String sql = "insert into dish (id,name,categoryid,pic,code,unit,price,status) " + "values (?,?,?,?,?,?,?,?)";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setInt(1, dish.getId());  
 *preStmt*.setString(2, dish.getName());  
 *preStmt*.setInt(3, dish.getCategoryid());  
 *preStmt*.setBlob(4, dish.getPic());  
 *preStmt*.setString(5, dish.getCode());  
 *preStmt*.setString(6, dish.getUnit());  
 *preStmt*.setDouble(7, dish.getPrice());  
 *preStmt*.setString(8, dish.getStatus());  
 *preStmt*.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void delete(Connection conn, int id) {  
 try {  
 String sql = "delete from dish where id= ?";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setInt(1, id);  
 *preStmt*.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 public void delete(Connection conn, String name) {  
 try {  
 String sql = "delete from dish where name= ?";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setString(1, name);  
 *preStmt*.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void update(Connection conn, int id, String val) {  
 try {  
 String sql = "update dish set name =? where id = ?";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setString(1, val);  
 *preStmt*.setInt(2, id);  
 *preStmt*.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void cha(Connection conn) {  
 try {  
 String sql = "select \* from staff";  
 *preStmt* = conn.prepareStatement(sql);  
 ResultSet rs = *preStmt*.executeQuery();  
 way(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 public void way(ResultSet rs) {  
 try {  
 while (rs.next()) {  
 System.*out*.print(rs.getInt("id") + "\t");  
 System.*out*.print(rs.getString("username") + "\t");//可以用大写，数据库大小写不敏感  
 System.*out*.print(rs.getString("password") + "\t");  
 System.*out*.print("\n");  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 public List<String> findAllName(Connection conn){  
 List<String> List=new ArrayList<String>();  
 try {  
 String sql = "select name from category";  
 *preStmt* = conn.prepareStatement(sql);  
 ResultSet rs = *preStmt*.executeQuery();  
 while (rs.next()) {  
 System.*out*.print(rs.getString("name") + "\t");  
 List.add(rs.getString("name"));  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return List;  
 }  
 public int findIdByName(Connection conn,String name){  
 int id=0;  
 try {  
 String sql = "select id from dish where name = ?";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setString(1, name);  
  
 *preStmt*.execute();  
 ResultSet rs = *preStmt*.executeQuery();  
 while (rs.next()) {  
 id=Integer.*valueOf*(rs.getInt("id"));  
 }  
 return id;  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return id;  
 }  
  
  
}

package com.canying.kongzhi;  
  
import com.canying.shiti.Staff;  
import com.sun.source.tree.ConditionalExpressionTree;  
  
import java.beans.Statement;  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.List;  
  
public class EmployeeDaoImpl implements IBaseDao<Staff> {  
 private static PreparedStatement *preStmt*;  
  
 private Statement stmt;  
 public boolean findStaff(Connection conn,String user){  
 boolean res=false;  
 try {  
 String sql = "SELECT name FROM staff where name = ?";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setString(1,user);  
 ResultSet rs = *preStmt*.executeQuery();  
 if (rs.next()) {  
 res=true;  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return res;  
 }  
  
 public String findPwd(Connection conn,String user){  
 String res="";  
 try {  
 String sql = "SELECT identityID FROM staff where name = ?";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setString(1,user);  
 ResultSet rs = *preStmt*.executeQuery();  
 if (rs.next()) {  
 res=rs.getString("identityID");  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return res;  
 }  
 public String findposition(Connection conn,String user,String pwd){  
 String res="";  
 try {  
 String sql = "SELECT position FROM staff where name = ? && identityID = ? ";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setString(1,user);  
 *preStmt*.setString(2,pwd);  
 ResultSet rs = *preStmt*.executeQuery();  
 if (rs.next()) {  
 res=rs.getString("position");  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return res;  
 }  
 public static int getmaxid(Connection conn) {  
 int id=1;  
 try {  
 String sql = "SELECT MAX(id) FROM staff";  
 *preStmt* = conn.prepareStatement(sql);  
 ResultSet rs = *preStmt*.executeQuery();  
 if (rs.next()) {  
 id = rs.getInt(1);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return id;  
  
 }  
  
 @Override  
 public void add(Connection conn, Staff staff) {  
 try {  
 String sql = "insert into staff (id,name,sex,identityID,address,tel,position,freeze) values (?,?,?,?,?,?,?,?)";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setInt(1, staff.getId());  
 *preStmt*.setString(2, staff.getName());  
 *preStmt*.setString(3, staff.getSex());  
 // preStmt.setDate(4, (java.sql.Date) staff.getBirthday());  
 *preStmt*.setString(4, staff.getIdentityID());  
 *preStmt*.setString(5, staff.getAddress());  
 *preStmt*.setString(6, staff.getTel());  
 *preStmt*.setString(7, staff.getPosition());  
 *preStmt*.setString(8, staff.getFreeze());  
 *preStmt*.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void delete(Connection conn, int id) {  
 try {  
 String sql = "delete from staff where id= ?";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setInt(1, id);  
 *preStmt*.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void update(Connection conn, int id, String val) {  
 try {  
 String sql = "update staff set name =? where id = ?";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setString(1, val);  
 *preStmt*.setInt(2, id);  
 *preStmt*.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void cha(Connection conn) {  
 try {  
 String sql = "select \* from staff";  
 *preStmt* = conn.prepareStatement(sql);  
 ResultSet rs = *preStmt*.executeQuery();  
 way(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void way(ResultSet rs) {  
 try {  
 while (rs.next()) {  
 System.*out*.print(rs.getInt("id") + "\t");  
 System.*out*.print(rs.getString("username") + "\t");//可以用大写，数据库大小写不敏感  
 System.*out*.print(rs.getString("password") + "\t");  
 System.*out*.print("\n");  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
}

package com.canying.kongzhi;  
  
import java.sql.Connection;  
  
public interface IBaseDao<T> {  
 public void add(Connection conn,T t);  
 public void delete(Connection conn,int id);  
 public void update(Connection conn,int id,String val);  
 public void cha(Connection conn);  
}

package com.canying.kongzhi;  
  
import com.canying.shiti.User;  
  
import java.sql.\*;  
  
  
public class JDBCConection {  
 String driver = "com.mysql.cj.jdbc.Driver";  
 //127.0.0.1 localhost  
 String url = "jdbc:mysql://localhost:3306/restaurant?useUnicode=true&characterEncoding=utf8";  
 String user = "root";  
 String password = "123456";  
 static private Connection *conn*;  
 private static Statement *stmt*;  
 private static PreparedStatement *preStmt*;  
  
 public JDBCConection() {  
 try {  
 Class.*forName*(driver);  
 *conn* = DriverManager.*getConnection*(url, user, password);  
 System.*out*.println("数据库连接成功!");  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 }  
  
 public Connection getConn() {  
 return *conn*;  
 }  
  
}

package com.canying.kongzhi;  
  
import com.canying.shiti.Order;  
import com.mysql.cj.x.protobuf.MysqlxCrud;  
  
import java.sql.Connection;  
import java.sql.Date;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.util.ArrayList;  
import java.util.List;  
  
public class OrderDaAOImpl implements IBaseDao<Order>{  
 private PreparedStatement preStmt;  
 public List<String> zhaolingdesk(){  
 JDBCConection db=new JDBCConection();  
 Connection conn=db.getConn();  
 List<String> List=new ArrayList<>();  
 //找桌号数量  
 DeskDAOImpl deskDAO=new DeskDAOImpl();  
 int deskid=deskDAO.getMaxId(conn);  
 System.*out*.println(deskid);  
for (int i=1;i<=deskid;i++){  
 int orid=getstatus(conn,i);  
 String sta=getstatus2(conn,orid);  
 if (sta.equals("未支付")){  
 List.add(Integer.*toString*(i));  
 }  
}  
  
 return List;  
 }  
 public int getstatus(Connection conn,int deskid){  
 int id=0;  
 try {  
  
 String sql = "SELECT MAX(id) FROM `order` WHERE deskid = ?";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setInt(1, deskid);  
 ResultSet rs = preStmt.executeQuery();  
 if (rs.next()) {  
 id=rs.getInt(1);  
 }  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return id;  
 }  
  
  
  
 public String getstatus2(Connection conn,int id){  
 String status="已支付";  
 try {  
 String sql = "SELECT status FROM `order` WHERE id = ?";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setInt(1, id);  
 ResultSet rs = preStmt.executeQuery();  
 if (rs.next()) {  
 status=rs.getString(1);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return status;  
 }  
 public int getmaxid(Connection conn) {  
 int id = 1;  
 try {  
 String sql = "SELECT MAX(id) FROM `order`";  
 preStmt = conn.prepareStatement(sql);  
 ResultSet rs = preStmt.executeQuery();  
 if (rs.next()) {  
 id = rs.getInt(1);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 return id;  
 }  
 public double findMoneyById(Connection conn,int id) {  
 double money = 0.00;  
 try {  
 String sql = "SELECT money FROM `order` where id = ?";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setInt(1, id);  
 ResultSet rs = preStmt.executeQuery();  
 if (rs.next()) {  
 money = rs.getDouble(1);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 return money;  
 }  
 @Override  
 public void add(Connection conn, Order order) {  
 try {  
 String sql = "insert into `order` (id,orderNo,deskid,createtime,money,customerid,status,number) values (?,?,?,?,?,?,?,?)";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setInt(1, order.getId());  
 preStmt.setString(2,order.getOrderNo());  
 preStmt.setInt(3,order.getDeskid());  
 preStmt.setDate(4, (Date) order.getCreatetime());  
 preStmt.setDouble(5,order.getMoney());  
 preStmt.setInt(6,order.getCustomerid());  
 preStmt.setString(7,order.getStatus());  
 preStmt.setInt(8,order.getNumber());  
  
 preStmt.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void delete(Connection conn, int id) {  
 try {  
 String sql = "delete from order where id= ?";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setInt(1, id);  
 preStmt.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void update(Connection conn, int id, String val) {  
 try {  
 String sql = "update orderi set deskid =? where id = ?";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setString(1, val);  
 preStmt.setInt(2, id);  
 preStmt.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 public void updateStatus(Connection conn, int id, String val) {  
 try {  
 String sql = "update `order` set status = ? where id = ?";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setString(1, val);  
 preStmt.setInt(2, id);  
 preStmt.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void cha(Connection conn) {  
 try {  
 String sql = "select \* from user";  
 preStmt = conn.prepareStatement(sql);  
 ResultSet rs = preStmt.executeQuery();  
 way(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 public int name\_id(Connection conn,int id){  
  
 int id1=0;  
 try {  
 String sql = "SELECT customerId FROM `order` where id =?";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setInt(1, id);  
 ResultSet rs = preStmt.executeQuery();  
 if (rs.next()) {  
 id1 = rs.getInt(1);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return id1;  
 }  
 public void way(ResultSet rs) {  
 try {  
 while (rs.next()) {  
 System.*out*.print(rs.getInt("id") + "\t");  
 System.*out*.print(rs.getString("username") + "\t");//可以用大写，数据库大小写不敏感  
 System.*out*.print(rs.getString("password") + "\t");  
 System.*out*.print("\n");  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
}

package com.canying.kongzhi;  
  
import com.canying.shiti.Orderitem;  
  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
  
public class OrderItemDAOImpl implements IBaseDao<Orderitem>{  
 private PreparedStatement preStmt;  
 @Override  
 public void add(Connection conn, Orderitem orderitem) {  
 try {  
 String sql = "insert into orderitem (id,orderId,dishId,amount) " + "values (?,?,?,?)";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setInt(1, orderitem.getId());  
 preStmt.setInt(2,orderitem.getOrdeid());  
 preStmt.setInt(3,orderitem.getDishid());  
 preStmt.setDouble(4,orderitem.getAmount());  
 preStmt.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void delete(Connection conn, int id) {  
 try {  
 String sql = "delete from orderitem where id= ?";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setInt(1, id);  
 preStmt.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void update(Connection conn, int id, String val) {  
 try {  
 String sql = "update orderitem set amount =? where id = ?";  
 preStmt = conn.prepareStatement(sql);  
 preStmt.setString(1, val);  
 preStmt.setInt(2, id);  
 preStmt.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void cha(Connection conn) {  
 try {  
 String sql = "select \* from user";  
 preStmt = conn.prepareStatement(sql);  
 ResultSet rs = preStmt.executeQuery();  
 way(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 public void way(ResultSet rs) {  
 try {  
 while (rs.next()) {  
 System.*out*.print(rs.getInt("id") + "\t");  
 System.*out*.print(rs.getString("username") + "\t");//可以用大写，数据库大小写不敏感  
 System.*out*.print(rs.getString("password") + "\t");  
 System.*out*.print("\n");  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 public int getmaxid(Connection conn) {  
 int id = 0;  
 try {  
 String sql = "SELECT MAX(id) FROM orderitem";  
 preStmt = conn.prepareStatement(sql);  
 ResultSet rs = preStmt.executeQuery();  
 if (rs.next()) {  
 id = rs.getInt(1);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 return id;  
  
 }  
 public void findmingxi(Connection conn,int deskid){  
  
  
 }  
}

package com.canying.kongzhi;  
  
import com.canying.shiti.User;  
  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.Statement;  
import java.util.ArrayList;  
import java.util.List;  
  
  
public class UserDaoImpl implements IBaseDao<User>{  
 private static PreparedStatement *preStmt*;  
  
 private Statement stmt;  
  
 public static int getmaxid(Connection conn) {  
 int id=0;  
 try {  
 String sql = "SELECT MAX(id) FROM user";  
 *preStmt* = conn.prepareStatement(sql);  
 ResultSet rs = *preStmt*.executeQuery();  
 if (rs.next()) {  
 id = rs.getInt(1);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return id;  
 }  
  
public boolean finduser(Connection conn,String user){  
 boolean res=false;  
 try {  
 String sql = "SELECT username FROM user where username = ?";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setString(1,user);  
 ResultSet rs = *preStmt*.executeQuery();  
 if (rs.next()) {  
 res=true;  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return res;  
}  
  
 public String findpwd(Connection conn,String user){  
 String res="";  
 try {  
 String sql = "SELECT password FROM user where username = ?";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setString(1,user);  
 ResultSet rs = *preStmt*.executeQuery();  
 if (rs.next()) {  
 res=rs.getString("password");  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return res;  
 }  
 @Override  
 public void add (Connection conn,User user) {  
 try {  
 String sql = "insert into user (id,username,password) " + "values (?,?,?)";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setInt(1, user.getId());  
 *preStmt*.setString(2,user.getUsername());  
 *preStmt*.setString(3,user.getPassword());  
 *preStmt*.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void delete(Connection conn,int id) {  
  
 try {  
 String sql = "delete from user where id= ?";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setInt(1, id);  
 *preStmt*.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void update(Connection conn,int id,String val) {  
 try {  
 String sql = "update user set password =? where id = ?";  
 *preStmt* = conn.prepareStatement(sql);  
 *preStmt*.setString(1, val);  
 *preStmt*.setInt(2, id);  
 *preStmt*.execute();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 }  
  
 @Override  
 public void cha(Connection conn) {  
 try {  
 String sql = "select \* from user";  
 *preStmt* = conn.prepareStatement(sql);  
 ResultSet rs = *preStmt*.executeQuery();  
 way(rs);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void way(ResultSet rs) {  
 try {  
 while (rs.next()) {  
 System.*out*.print(rs.getInt("id") + "\t");  
 System.*out*.print(rs.getString("username") + "\t");//可以用大写，数据库大小写不敏感  
 System.*out*.print(rs.getString("password") + "\t");  
 System.*out*.print("\n");  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
}

package com.canying.shiti;  
  
public class Category {  
 private static int *id*=1;  
 private String name;  
 private String describ;  
public Category(String name,String describ){  
 this.*id*=*id*;*id*++;  
 this.name=name;  
 this.describ=describ;  
  
}  
 public int getId() {  
 return *id*;  
 }  
  
 public static void setId(int id) {  
 Category.*id* = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getDescrib() {  
 return describ;  
 }  
  
 public void setDescrib(String describ) {  
 this.describ = describ;  
 }  
}

package com.canying.shiti;  
  
public class Customer {  
 private int id;  
 private String name;  
 private String sex;  
 private String company;  
 private String tel;  
 private String cardID;  
  
 public Customer(int id, String name, String sex, String companey, String tel, String cardid) {  
 this.id=id;  
 this.name=name;  
 this.sex=sex;  
 this.company=companey;  
 this.tel=tel;  
 this.cardID=cardid;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getSex() {  
 return sex;  
 }  
  
 public void setSex(String sex) {  
 this.sex = sex;  
 }  
  
 public String getCompany() {  
 return company;  
 }  
  
 public void setCompany(String company) {  
 this.company = company;  
 }  
  
 public String getTel() {  
 return tel;  
 }  
  
 public void setTel(String tel) {  
 this.tel = tel;  
 }  
  
 public String getCardID() {  
 return cardID;  
 }  
  
 public void setCardID(String cardID) {  
 this.cardID = cardID;  
 }  
}

package com.canying.shiti;  
  
public class Desk {  
 private int id;  
 private String no;  
 private int seating;  
 private String status;  
  
 public Desk(int id, String no, int seatings) {  
 this.id=id;  
 this.no=no;  
 this.seating=seatings;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getNo() {  
 return no;  
 }  
  
 public void setNo(String no) {  
 this.no = no;  
 }  
  
 public int getSeating() {  
 return seating;  
 }  
  
 public void setSeating(int seating) {  
 this.seating = seating;  
 }  
  
 public String getStatus() {  
 return status;  
 }  
  
 public void setStatus(String status) {  
 this.status = status;  
 }  
}

package com.canying.shiti;  
  
import java.sql.Blob;  
import java.sql.Connection;  
import java.util.Spliterators;  
  
public class Dish {  
 private static int *id*=1;  
 private String name;  
 private int categoryid;  
 private Blob pic;  
 private String code;  
 private String unit;  
 private double price;  
 private String status;  
  
 public Dish(String name, int categoryid, Blob pic,String code,String unit,Double price,String status){  
 this.*id*=*id*;*id*++;  
 this.name=name;  
 this.categoryid=categoryid;  
 this.pic=pic;  
 this.code=code;  
 this.unit=unit;  
 this.price=price;  
 this.status=status;  
 }  
  
 public Dish(String name, int categoryid, String code, String unit, double price, String status) {  
 this.*id*=*id*;*id*++;  
 this.name=name;  
 this.categoryid=categoryid;  
 this.code=code;  
 this.unit=unit;  
 this.price=price;  
 this.status=status;  
 }  
  
  
 public int getId() {  
 return *id*;  
 }  
  
 public static void setId(int id) {  
 Dish.*id* = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public int getCategoryid() {  
 return categoryid;  
 }  
  
 public void setCategoryid(int categoryid) {  
 this.categoryid = categoryid;  
 }  
  
 public Blob getPic() {  
 return pic;  
 }  
  
 public void setPic(Blob pic) {  
 this.pic = pic;  
 }  
  
 public String getCode() {  
 return code;  
 }  
  
 public void setCode(String code) {  
 this.code = code;  
 }  
  
 public String getUnit() {  
 return unit;  
 }  
  
 public void setUnit(String unit) {  
 this.unit = unit;  
 }  
  
 public double getPrice() {  
 return price;  
 }  
  
 public void setPrice(double price) {  
 this.price = price;  
 }  
  
 public String getStatus() {  
 return status;  
 }  
  
 public void setStatus(String status) {  
 this.status = status;  
 }  
  
}

package com.canying.shiti;  
  
import java.util.Date;  
  
public class Order {  
 private int id=1;  
 private String orderNo;  
 private int deskid;  
 private Date createtime;  
 private double money;  
 private int customerid;  
 private String status;  
 private int number;  
  
 public Order(int id,String orderNo, int deskid,Date createtime,int customerid,double money,String status,int number){  
 this.id=id;  
 this.orderNo=orderNo;  
 this.deskid=deskid;  
 this.createtime=createtime;  
 this.money=money;  
 this.customerid=customerid;  
 this.status=status;  
 this.number=number;  
 }  
  
  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getOrderNo() {  
 return orderNo;  
 }  
  
 public void setOrderNo(String orderNo) {  
 this.orderNo = orderNo;  
 }  
  
 public int getDeskid() {  
 return deskid;  
 }  
  
 public void setDeskid(int deskid) {  
 this.deskid = deskid;  
 }  
  
 public Date getCreatetime() {  
 return createtime;  
 }  
  
 public void setCreatetime(Date createtime) {  
 this.createtime = createtime;  
 }  
  
 public double getMoney() {  
 return money;  
 }  
  
 public void setMoney(double money) {  
 this.money = money;  
 }  
  
 public int getCustomerid() {  
 return customerid;  
 }  
  
 public void setCustomerid(int customerid) {  
 this.customerid = customerid;  
 }  
  
 public String getStatus() {  
 return status;  
 }  
  
 public void setStatus(String status) {  
 this.status = status;  
 }  
  
 public int getNumber() {  
 return number;  
 }  
  
 public void setNumber(int number) {  
 this.number = number;  
 }  
}

package com.canying.shiti;  
  
public class Orderitem {  
 private int id;  
 private int ordeid;  
 private int dishid;  
 private double amount;  
  
 public Orderitem(int orderid, int dinddanid, int dishid, int num2) {  
 this.id=orderid;  
 this.ordeid=dinddanid;  
 this.dishid=dishid;  
 this.amount=num2;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public int getOrdeid() {  
 return ordeid;  
 }  
  
 public void setOrdeid(int ordeid) {  
 this.ordeid = ordeid;  
 }  
  
 public int getDishid() {  
 return dishid;  
 }  
  
 public void setDishid(int dishid) {  
 this.dishid = dishid;  
 }  
  
 public double getAmount() {  
 return amount;  
 }  
  
 public void setAmount(double amount) {  
 this.amount = amount;  
 }  
}

package com.canying.shiti;  
  
  
import java.util.Date;  
  
public class Staff {  
 private static int *num*=1;  
 private int id=1;  
 private String name;  
 private String sex;  
 private Date birthday;  
 private String identityID;  
 private String address;  
 private String tel;  
 private String position;  
 private String freeze;  
  
 public Staff(String name, String sex, Date date, String identityID, String adress, String tel, String position1, String freeze1) {  
 this.id=*num*;*num*++;  
 this.name=name;  
 this.tel=sex;  
 this.identityID=identityID;  
 this.address=adress;  
 this.tel=tel;  
 this.position=position1;  
 this.freeze=freeze1;  
  
 this.birthday=date;  
 }  
  
 public Staff(String name, String sex, String identityID, String adress, String tel, String position1, String freeze1) {  
 this.id=*num*;*num*++;  
 this.name=name;  
 this.tel=sex;  
 this.identityID=identityID;  
 this.address=adress;  
 this.tel=tel;  
 this.position=position1;  
 this.freeze=freeze1;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public static void setNum(int id) {  
 Staff.*num* = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getSex() {  
 return sex;  
 }  
  
 public void setSex(String sex) {  
 this.sex = sex;  
 }  
  
 public Date getBirthday() {  
 return birthday;  
 }  
  
 public void setBirthday(Date birthday) {  
 this.birthday = birthday;  
 }  
  
 public String getIdentityID() {  
 return identityID;  
 }  
  
 public void setIdentityID(String identityID) {  
 this.identityID = identityID;  
 }  
  
 public String getAddress() {  
 return address;  
 }  
  
 public void setAddress(String address) {  
 this.address = address;  
 }  
  
 public String getTel() {  
 return tel;  
 }  
  
 public void setTel(String tel) {  
 this.tel = tel;  
 }  
  
 public String getPosition() {  
 return position;  
 }  
  
 public void setPosition(String position) {  
 this.position = position;  
 }  
  
 public String getFreeze() {  
 return freeze;  
 }  
  
 public void setFreeze(String freeze) {  
 this.freeze = freeze;  
 }  
}

package com.canying.shiti;  
  
public class User {  
 private static int *num*=1;  
 private int id;  
 private String username;  
 private String password;  
 public String getUsername() {  
 return username;  
 }  
  
 public User(String username,String password){  
 this.id=*num*;*num*++;  
 this.username=username;  
 this.password=password;  
 }  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public void setUsername(String username) {  
 this.username = username;  
 }  
  
 public String getPassword() {  
 return password;  
 }  
  
 public void setPassword(String password) {  
 this.password = password;  
 }  
}