

目录

1. [数据库的使用](#)
 - (1) [hosue_owner](#)
 - (2) [house](#)
 - (3) [department](#)
 - (4) [staff](#)
2. [db 库](#)
 - (1) [DBINFO](#)
 - (2) [SQLString](#)
 - (3) [insert](#)
 - (4) [delete](#)
 - (5) [select](#)
 - (6) [update](#)
3. [staff 库](#)
 - (1) [Staff](#)
 - (2) [Manager](#)
 - (3) [chargeStaff](#)
4. [cs 库](#)
5. [chargeDisplay 库](#)
6. [managerDisplay 库](#)

一 . 数据库的使用

在项目文件夹下的 database 文件夹下存放着导出数据库文件 cs.sql。根据和数据库文件一起存放的 readme.txt 导入数据库。

对此数据库存在 2 名用户： manager 和 staff（需要在导入数据库后自行创建这两个用户）
manager 具有对 staff 表增删改查的权限

staff 具有对 house 表查的权限

该数据库共包含 4 个表 house_owner、house、department 和 staff

- (1) house_owner——业主信息表

| 名称 | 数据类型 | 属性 | 描述 |
|--------------|---------|----|--------|
| owner_id | varchar | 主键 | 业主号 |
| owner_name | varchar | - | 业主姓名 |
| work_place | varchar | - | 业主工作地址 |
| phone_number | varchar | - | 业主手机号码 |

- (2) house——房屋信息表

| 名称 | 数据类型 | 属性 | 描述 |
|----------|-------------|----|---------|
| house_id | varchar | 主键 | 房屋号 |
| area | double | - | 房屋面积 |
| room | double | - | 房屋的房间数量 |
| owner_id | varchar(20) | 外键 | 房主 ID |

(3) department——部门信息表

| 名称 | 数据类型 | 属性 | 描述 |
|-----------------|---------|----|-------|
| department_id | varchar | 主键 | 部门号 |
| department_name | varchar | - | 部门名称 |
| manager | varchar | - | 部门负责人 |
| phone_number | varchar | - | 部门电话 |

(4) staff——员工信息表

| 名称 | 数据类型 | 属性 | 描述 |
|---------------|---------------|----|------------------|
| staff_id | varchar | 主键 | 员工号 |
| staff_name | varchar | - | 员工姓名 |
| birthday | date | - | 生日 |
| gender | enum('M','F') | - | 性别, M 为男, F 为女 |
| address | varchar | - | 员工住址 |
| phone | varchar | - | 员工电话 |
| department_id | varchar | 外键 | 员工所属部门号 |
| job | enum('M','S') | - | 职位, M 为经理, S 为员工 |
| pwd | varchar | - | 员工密码 |

(5) bill——账单表

| 名称 | 数据类型 | 属性 | 描述 |
|-------------|----------------------------|----|-----------------|
| property | varchar(20) | - | 物业费 |
| clean | varchar(20) | - | 卫生费 |
| water | varchar(20) | - | 水费 |
| electricity | varchar(20) | - | 电费 |
| time_stamp | date | - | 添加此条记录的时间 |
| house_id | varchar | - | 房屋号 |
| state | ENUM('unordered','unpaid') | - | 状态, 未出单、未支付 |
| staff_id | varchar(20) | - | 插入此条收费记录的员工的 ID |

二. db 库

db 库主要包含于数据库相关的操作类

DNINFO:

此类无方法。该类为 delete、insert、select、update 类提供支持数据支持。

SQLString:

该类负责为 delete、insert、select、update 类构造 SQL 语句

| 方法 | 返回类型 | 参数 | 描述 |
|--------------|--------|--------------------|------------------------------|
| construct | String | (String[] content) | 接受一个数组，将其拼接后返回 |
| insertColumn | String | (String[] content) | 接受一个数组，将其拼接为 insert 语句要注入的字段 |
| insertValues | String | (String[] content) | 接受一个数组，将其拼接为 insert 语句要注入的内容 |

insert:

该类负责执行 SQL 语句中的 insert 语句

| 方法 | 返回类型 | 参数 | 描述 |
|-----------|------|---|--|
| insertSet | void | (String user, String pass, String table, String[] column, String[] content) | user、pass 为执行此操作的对象的用户名和密码，table 是要插入的表的名字，column 是要插入的字段的集合，content 是要插入的内容 |

delete:

该类负责执行 SQL 语句中的 delete 语句

| 方法 | 返回类型 | 参数 | 描述 |
|-----------|------|--|--|
| deleteSet | void | (String user, String pass, String table, String column, String sign) | user、pass 为执行此操作的对象的用户名和密码，table 是执行操作的表的名字，column 是字段名，sign 是要删除的行对应的 column 的值 |

select:

该类负责执行 SQL 语句中的 select 语句

| 方法 | 返回类型 | 参数 | 描述 |
|-----------|------------------------------|---|--|
| selectSet | Object | (String user, String pass, String table, String content, String[] selectInfo) | user、pass 为执行此操作的对象的用户名和密码，table 是要选择的表的名字，content 是要选择的字段，selectInfo 是附加信息，用于构造 where 部分语句，其内容为一个包含两个字符串的数组，第一个字符串表示字段名，第二个字符串表示该字段的值。不需要限定查找字段时传入 null |
| selectSet | List<HashMap<String,Object>> | (String user, String pass, String table, String[] content, String[] selectInfo) | user、pass 为执行此操作的对象的用户名和密码，table 是要选择的表的名字，content 是要选择的字段构成的数组，selectInfo 是附加信息，用于构造 where 部分语句，其内容为一个包含两个字符串的数组，第一个字符串表示字段名，第二个字符串表示该字段的值。 |

| | | | |
|--|--|--|--|
| | | | 不需要 selectInfo 时传入 null。 由 select 得到的每一行信息构成一个 HashMap 储存在 List 中。 HashMap 中的 key 是字段名， value 是字段对应的值 |
|--|--|--|--|

用例：

```
List<HashMap<String,Object>> recieve=(List<HashMap<String, Object>>) .selectSet("staff","staff","house","*",null);
for(HashMap<String,Object> m:recieve) {
    System.out.println(m);
}
```

输出：

```
{area=100.0, house_id=A0101, electricitycost=100.0, watercost=6.0, room=4.0}
{area=100.0, house_id=A0102, electricitycost=60.0, watercost=4.0, room=4.0}
{area=78.0, house_id=A0201, electricitycost=100.0, watercost=9.0, room=4.0}
{area=94.0, house_id=A0301, electricitycost=50.0, watercost=8.0, room=4.0}
{area=120.0, house_id=A0503, electricitycost=70.0, watercost=4.0, room=4.0}
{area=98.6, house_id=A1608, electricitycost=102.0, watercost=6.0, room=1.0}
{area=102.0, house_id=A2200, electricitycost=33.0, watercost=5.0, room=4.0}
{area=102.0, house_id=A2201, electricitycost=10.0, watercost=1.0, room=4.0}
```

对比数据库：

```
mysql> select * from house;
+-----+-----+-----+-----+-----+
| house_id | area | room | watercost | electricitycost |
+-----+-----+-----+-----+-----+
| A0101    | 100  | 4    | 6         | 100             |
| A0102    | 100  | 4    | 4         | 60              |
| A0201    | 78   | 4    | 9         | 100             |
| A0301    | 94   | 4    | 8         | 50              |
| A0503    | 120  | 4    | 4         | 70              |
| A1608    | 98.6 | 1    | 6         | 102             |
| A2200    | 102  | 4    | 5         | 33              |
| A2201    | 102  | 4    | 1         | 10              |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

update:

该类负责执行 SQL 语句中的 delete 语句

| 方法 | 返回类型 | 参数 | 描述 |
|----|------|----|----|
|----|------|----|----|

| | | | |
|-----------|------|--|--|
| updateSet | void | (String user, String pass, String table, String[] column, String content[], String[] sign) | user、pass 为执行此操作的对象的用户名和密码，table 是要跟新的表的名字，content 是要选择的字段构成的数组，sign 是附加信息，用于构造 where 部分语句，其内容为一个包含两个字符串的数组，第一个字符串表示字段名，第二个字符串表示该字段的值。 |
|-----------|------|--|--|

三 . staff 库

staff 库主要包含和员工相关的库

Staff:

员工类，基类，本身没什么用

| 方法 | 返回类型 | 参数 | 描述 |
|------------|--------|------------------|---------------------------------|
| Staff | - | (String staffID) | staff 类的构造函数，接受一个 String 字符串员工号 |
| getStaffID | String | - | 返回此对象的员工号 |

Manager:

经理类，Staff 类的子类

| 方法 | 返回类型 | 参数 | 描述 |
|------------|--------|------------------|----------------------------------|
| Manager | - | (String staffID) | Manager 的构造函数，接受一个 String 字符串员工号 |
| getStaffID | String | - | 返回员工号 |

| | | | |
|---------|--------|---|-------------------|
| getUser | String | - | 返回 Manager 的数据库账号 |
| getPass | String | - | 返回 Manager 的数据库密码 |

chargeStaff:

收费员工类，Staff 类的子类

| 方法 | 返回类型 | 参数 | 描述 |
|-----------|----------|------------------|---|
| charge | double[] | (String houseID) | houseID 表示要收费的房屋房屋号，返回一个收取费用的数组。数组包含物业费和卫生费 |
| houseInfo | double[] | (String houseID) | 返回房屋的面积和房间数量信息 |
| getID | String | - | 返回员工号 |
| getUser | String | - | 返回 chargeStaff 的数据库账号 |
| getPass | String | - | 返回 chargeStaff 的数据库密码 |

四 . cs 库

本库是整个程序的主显示界面库。cs 是 ChargeSystem 的缩写。

DisPlay:

整个程序的主显示界面

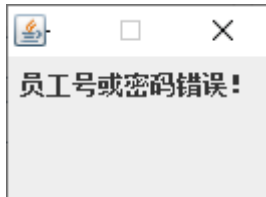


logIn:

进行登录判断的模块

failedToLogin:

登录失败后显示的界面



judgingPosition:

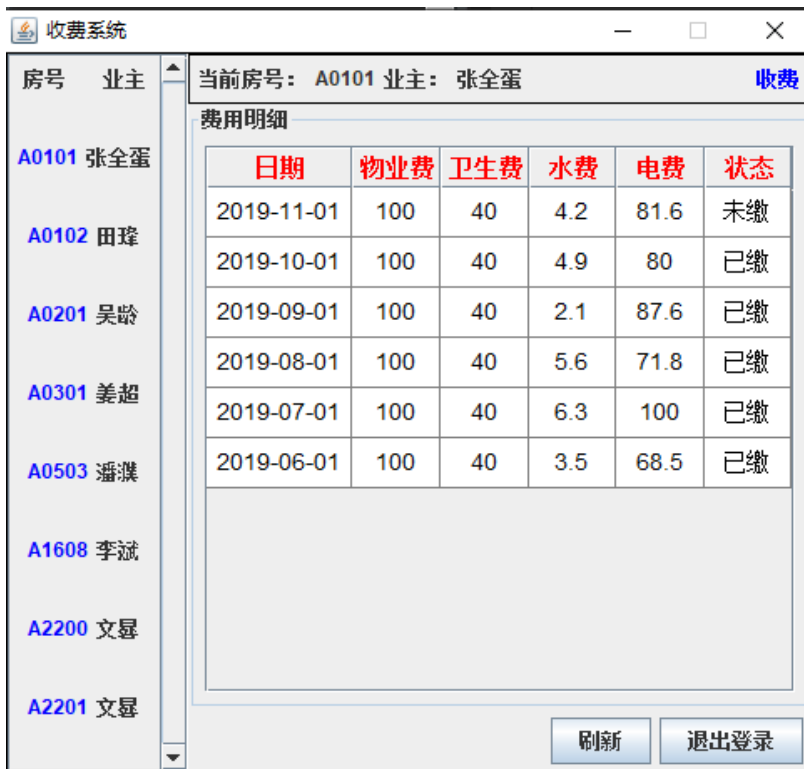
判断登录员工的职位的模块

五 . chargeDisplay 库

本库是收费员工的显示界面

Main:

收费员工的主显示界面



charge:

收费界面

测试窗口

房号: A0101 业主: 张全蛋

| 序号 | 收费类型 | 数量 | 金额 |
|----|------|-------|-------|
| 1 | 物业费 | 100.0 | 100.0 |
| 2 | 卫生费 | 4.0 | 40.0 |
| 3 | 水费 | | |
| 4 | 电费 | | |
| 合计 | | | |

日期: 2019-12-08 员工: 011

确定 打印

ChargeSuccess:

收费成功后的提示界面

收费成功

确定

myButton:

用于方便的管理自定义的按钮样式, 内部包含两个方法, 返回不同风格的按钮

myLabel:

用于方便的管理自定义的标签样式, 内部包含两个方法, 返回不同风格的标签

myTable:

用于方便的管理自定义的表格样式, 内部包含两个方法, 返回不同风格的表格

reservedDigits:

用于返回保留一位小数的小数

toChinese:

将数字字符串转换为中文大写的字符串

六 . managerDisplay 库

尚待施工