

夏季短学期

实践成绩评定表

**姓 名： 田畅**

**学 号： 1705990244**

**专 业： 软件1701**

**指导教师： 罗养霞**

**日 期：** 2019.07.19

**教务处制表**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实践教学开展时间 | | 2019年 6 月 24 日——2019年 7月 19 日 | | | | | | |
| 实践形式（在选项后打√） | | 集中 | | √ | | 分散 | |  |
| 实践单位或地点 | | 实验楼418 | | | | | | |
| 指导教师 | | 罗养霞 | 职称 | 副教授 | 学院 | | 信息学院计算计系 | |
| 实践内容及收获（后附实践报告） | 一、实践内容  1.启动仪式+项目背景讲解——计算机的发展及基础知  2.程序语言背景学习+上机实践（环境搭建+程序语言上机实践）  3.《停车场管理系统项目》学习与实践  4.项目总结答辩  二、实践目的  通过理论联系实际，巩固所学知识并且学习新的知识，提高处理实际问题  的能力提高所学知识与现实系统的结合能力，努力解决并实现现实生活中  的问题，并为自己能顺利与社会环境接轨做准备。  三、成果展示  1.软件——停车场管理系统  四、主要收获  1.通过实践周的学习，使我进一步了解所学习的基础知识，了解计算机技术和信息管理技术，较为系统的掌握计算机应用技能和信息管理技能从 而提高从事实际工作的能力。  2.在实践中雷老师带领我们血洗并且实践，带我们了解JAVA、HTML、静  态网页 （JSP）、运用软件制作静态网页模型，还有就是通过分组协作和明确的分工提高了每一个人的分析解决问题能力。  3,《停车场管理系统》的技术运用不但有新的技术还有一些学习过的基础知识这就要求我们不但要对旧的知识要熟练的掌握，还要有学习新技术的  能力。同时这也就再一次说明这次实践对我们的知识体系有了很大的扩充  提升。 | | | | | | | |
| 成绩  评定 | （综合评语）    成绩： 指导教师签字：  年 月 日 | | | | | | | |
| 系意见 | 签字（盖章）：  年 月 日 | | | | | | | |
| 学院意见 | 签字（盖章）：  年 月 日 | | | | | | | |

注：**1.此表在下一学期开学一周内交给指导教师。**

**2.实践成果附后。**

**3.实践内容收获填写不下，可附页。**

**4.成绩分为优秀、良好、中等、合格、不合格，共五级。**

**5.本表由学院存档。**



**信息学院**

**夏季短学期实践报告**

**成果形式： 实践报告**

**成果名称： 《停车场管理系统》**

学生姓名**： 田畅**

学 号**： 1705990244**

专 业**： 软件工程**

班 级**： 软件工程1701**

指导教师**： 罗养霞**

完成时间： 2019年 7 月 19日

**《停车场管理系统》**

**一、实践目的**

作为大二新生的我感觉时间过得飞快，犹如白驹过隙，一晃大一一年已经完了，经过了紧张的考试。我进入了我们在大学的第一个实践周，刚开始我的心情是十分的激动地，抱着对实践周的神秘性我渐渐地迎来实践周。虽然时间不长但是我也在其中学习到了不少的新知识。

我了解到了一个我未曾接触过的计算机设计语言——JAVA和超文本标记语言——HTML，在时间周中我主要用JAVA语言去写一些JSP网页页面。JAVA和C++一样都是面向对象的一种编程语言其含有C++的有点但去掉了C++的缺点是一门功能强大的语言，在雷老师的讲解下我对JAVA及HTML这两门语言有了深刻的认识。我的见解如下：

**1.JAVA** 是一门面向对象编程语言，因此Java语言具有功能强大和简单易用两个特征。Java语言作为静态面向对象编程语言的代表，极好地实现了面向对象理论，允许程序员以优雅的思维方式进行复杂的编程 。

**2.HTML** 标准通用标记语言下的一个应用。HTML不是**编程语言**，而是一种标记语言，是网页制作所必备的。**“**超文本**”**就是指页面内可以包含图片链接，甚至音乐、程序等非文字元素来充作标记。

HTML具有：简易性、可扩展性、平台无关性、通用性等特点。

**3.**关于JAVA语言和HTML超文本标记语言，前者是一种编程语言而后者却不是编程语言这就直接决定了两个语言的性质用途。就我个人而言在学习的过程中我比较喜欢HTML语言虽然在该项目中HTML的使用不是很多，但是我对HTML设计的网页我也很喜欢我可以根据我的想法去设计我想要的网页。去添加一些音乐、视频、图片。

**二、实践内容**

**1.**启动仪式+项目背景讲解——计算机的发展及基础知

**2.**程序语言背景学习+上机实践（环境搭建+程序语言上机实践）

**3.**《停车场管理系统项目》学习与实践

**4.**项目总结答辩

**三、实践目的**

. 通过实践周的学习，使我进一步知道所学习的基础知识并且提高自身的知识储备将强知识使用实践能力，了解计算机技术和信息管理技术和实用技术和学习计算机技术，较为系统的掌握和应用计算机应用技能和信息管理技能以便以提高从事实际工作的能力。

透过理论联系实际，巩固所学知识，提高处理实际问题的能力提高所学知识与现实系统结合的能力，只有不断的学习才能不落伍不退步一直向前，努力解决并实现现实生活中的问题。

**四、实践过程**

**1.《**停车场管理系统**》**

**（1）**实验目的

通过实验来达到熟练运用基础知识和学习新的知识的能力，从而达到温故知新的效果，通过JAVA语言及HTML语言的配合去完成项目。

**（2）**目的和背景

伴随着科技的快速发展，人们的生活水平越来越高交通工具的越来越普及。汽车作为主要的交通工具之一有着举足轻重的左右。人们生活水平的提高，汽车的数量也不断地增多，伴随而来的办事停车难，停车乱收费等问题。以前落后的人力停车管理即将被高科技化的自动停车管理系统所取代这是社会发展的必然趋势也是大势所趋，智能化、自动化的停车场管理系统节省了大量时间和人力物资消耗，提高了停车收费，车位安排的效率。停车场管理系统基于普通住宅小区的小型停车场和大型公共专用停车场，用来对车辆的计时收费、停车区域选取的管理等功能来解决停车等问题。

停车场管理系统是集成自动化技术、计算机技术为一体的系统，能够避免和调整停车场入口处及出口处的交通阻塞现象及收费不标准问题，同时便利小区和城市街道的停车计费问题合理的安排车位。还能满足使用者和管理者对停车场、安全、性能以及管理上的需要。所以研究1停车场管理系统对于我的的生活有着巨大的帮助。

电脑为Windows10-x64，采用Axure RP Pro 6进行网页模型设计，MyEclipse进行代码开发，sql sever 2008数据库软件。

**（4）**项目框架

该项目主要采用JAVA语言进行逻辑结构的编写，其框架构成主要有三个主要模块配合数据库来实现全部功能：

1.管理员登录功能模块：

该系统是为管理员快速便捷管理停车场的系统软件，通过在输入框输入管理员账号和密码进行登录后，系统会将输入内容与数据库的管理员表进行对比，做出相应。

2.用户信息管理功能模块：

**显示用户信息**：管理员登陆系统以后，在界面的主体位置显示出所有的已进入该停车场的用户的详细信息；

**管理用户信息：**登录该系统的管理员可以对用户信息进行增、删、改、查等操作来达到相应的目的；

**管理停车卡：**登录该系统的管理员为用户提供办卡，退卡等操作；

3.收费管理功能模块：

记录停车场停车的用户信息，并且记录用户进入停车场的时间和离开时间计算应收费用。

并且提供车辆离开停车场是信息的删除，及离开时间的修改。提供了根据卡号查寻车位的查询功能

**（5）**模块功能实现

1. 管理员登录功能模块

模块功能图：

图1 管理员登陆模块流程逻辑图

|  |
| --- |
|  |
|  |  |

算法：

管理员通过输入管理员账号和密码登陆系统

\*算法思路

首先，检查管理员账号是否为空；

然后，把（管理员账号，密码）组成sql语句。执行sql语句，将管理员账号和密码与数据库保存的所有管理员信息做对比；

最后，若数据库中匹配到输入信息,则提示登陆成功，跳转到用户信息界面；否则提示“账号或密码错误”来告诉登录人员继续输入或输出；

流程逻辑：

图2管理员登陆模块流程逻辑图

|  |
| --- |
|  |
|  |  |

1. 用户信息管理功能模块

模块功能图：

图3 用户信息功能模块功能模块图

|  |
| --- |
|  |
|  |  |

算法：

管理员通过输入管理员账号和密码登陆系统

\*查询算法思路

首先，检索到某一用户；

然后，把（用户的卡号，姓名）组成sql语句。执行sql语句，将用户信息从数据库中提取出来；

最后，若检索到该用户的信息输出到查询界面；

\*办卡算法思路

首先，输入办理停车卡所需要的信息（卡号，姓名，电话和车牌号）；

然后，把（用户的卡号，姓名）组成sql语句。执行sql语句，将办卡信息保存到用户信息数据库中以便程序后面调用；

最后，将主界面的所有用户信息更新一次；

流程逻辑：

|  |
| --- |
|  |
|  |  |

图4用户信息功能模块功能模块图

1. 收费功能模块

模块功能图：

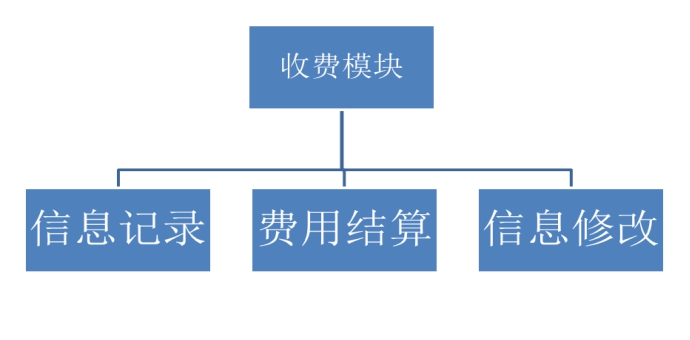


图5收费功能模块功能模块图

算法：

\*算法思路

首先，通过SQL语句把用户表上面的信息查询并且投影到界面上面显示；

然后，添加新的用户信息，把（卡号，车位号，开始时间）编程SQL语句。执行对应SQL语句，将用户信息直接存入数据库，当车辆离开时通过修改（将信息通过SQL插入表中）离开时间计算出用户停车所用的总时间计算出总费用。 最后，根据数据库设计，限制文本输入框的字符数等信息。如果用户输错，利用Javascript进行页面验证登录，即进行相应提示。

流程逻辑：

图6管理员收费功能模块状态图

|  |
| --- |
|  |
|  |  |

1. 系统界面设计
2. 登录界面

实现功能：

实现简单的登录功能，输入管理员的卡号和密码后，点击“登陆”按钮即可进入停车场收费管理系统进行相应的操作。

|  |
| --- |
|  |
|  |  |

1. 用户信息界面

实现功能：

在管理员进入收费系统后，界面主体为停车场车辆的详细信息。

管理员可以在界面上边的输入框输入用户姓名后点击“查询”按钮进行查询，或者直接点击“全部”按钮查看所有的车辆信息。也可以通过点击“添加”按钮，“删除”按钮，“修改”按钮对所有的车辆和车位信息进行相应的添加，删除，和修改操作。

界面的左半部分是所有车辆的详细信息，界面的右半部分是停车场的基本信息，包括停车场剩余车位数和停车场已用车位数。

通过点击界面下边的“收费表”按钮，即可进入收费表界面。

|  |
| --- |
|  |
|  |
|  |  |

1. 收费界面

实现功能：

主要功能是用来查询车辆收费情况，其中可以显示出车辆的信息以及停车开始计费时间以及结束时间。

查询时可以点击“全部”按钮查看停车场所有车辆的详细收费情况，也可以在输入框输入用户卡号后点击“查询”按钮，来查询该用户具体的收费情况。

在该界面也可通过点击“添加”，“删除”和“修改”按钮进行用户车辆收费信息的添加，删除和修改。

点击“返回用户信息”按钮，即可返回用户信息界面。

|  |
| --- |
|  |
|  |  |

1. 添加用户信息界面

实现功能：

进入添加用户信息界面后，再输入框输入卡号，密码，姓名，余额，电话和车牌信息后点击“添加”按钮，即可将用户信息添加至用户信息表中。

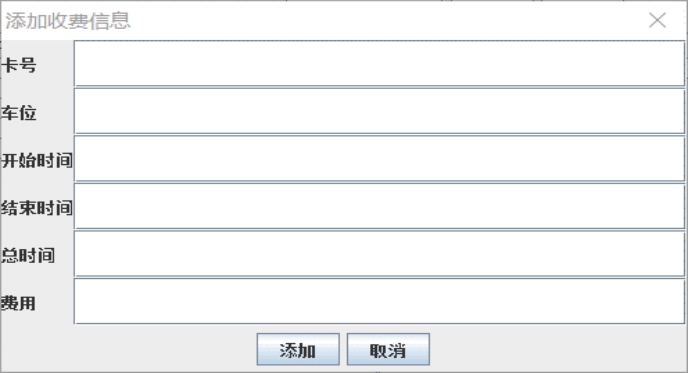
点击“取消”按钮后，可以撤销插入用户信息操作。

|  |
| --- |
|  |
|  |  |

1. 添加收费界面

实现功能：

进入添加收费信息界面后，再输入框输入卡号，车位，开始时间，结束时间，总时间和费用后点击“添加”按钮，即可将收费信息添加至收费信息表中。点击“取消”按钮后，可以撤销插入收费信息操作。



1. **数据库设计**

**（1）**设计目的

伴随着科技的飞速发展，交通工具的越来越普及。汽车作为人类社会中最主要的交通工具之一，起着重大作用。随着人们生活水平的提高，汽车的数量也不断地增多，伴随而来的办事停车难，停车乱收费等问题。以前落后的人力停车管理即将被高科技化的自动停车管理系统所取代这是社会发展的必然趋势也是大师所起，自动化的停车场管理系统节省了大量时间和人力物资消耗，提高了停车收费，车位安排的效率。停车场管理系统基于普通住宅小区的小型停车场和大型公共专用停车场，用来对车辆的计时收费、停车区域选取的管理等功能来解决停车等问题。而数据库将配合软件程序进行运行来达到相应的目的。

**（2）**数据库结构定义

|  |
| --- |
|  |
|  |  |

账号：管理员账号

密码：管理员密码

卡号：用户办理的卡号

车位：目前车辆所在车位

开始时间：车辆进入停车场的时间

结束时间：车辆驶出停车场的时间

总时间：计算得出停车时间

费用：根据时间和单价计算得出的所需收取的费用

剩余车位：停车场内剩余的车位数

已用车位：停车场内已用的车位数



|  |
| --- |
|  |
|  |  |

|  |
| --- |
|  |
|  |  |

卡号:用户办理的卡号

姓名：用户姓名

余额：用户卡内所剩余额

电话：用户电话

车牌号：用户车牌号

**（3）**概念结构设计

|  |
| --- |
|  |
|  |  |

**（4）**逻辑结构设计

|  |
| --- |
|  |
|  |  |

**（5）**关系模型

停车场信息：

停车场信息（剩余车位，已用车位）

收费表：

收费表（卡号，车位，开始时间，结束时间，总时间，费用）

用户信息：

用户信息（卡号，姓名，余额，电话，车牌号）

管理员：

管理员（账号，密码）

**（6）**属性表

1. **代码分析**

**（1）**界面设计代码

package Model;

import java.sql.ResultSet;

import java.util.Vector;

import javax.swing.table.AbstractTableModel;

import Helper.\*;

public class adminModel extends AbstractTableModel{

private Vector columnNames;

private Vector rowDates;

//

public adminModel()

{

String sql = "select \* from 用户信息";

String []parass = {};

}

//========增删改学生

public boolean cudStu(String sql, String []parass)

{

return new SqlHelper().cudExecutes(sql, parass);

}

//========查询学生

public void queryStu(String sql, String []parass)

{

SqlHelper sqlHelper = null;

//========初始化JTable信息

columnNames = new Vector();

rowDates = new Vector();

columnNames.add("卡号");

columnNames.add("姓名"); columnNames.add("余额");

columnNames.add("电话");

columnNames.add("车牌号");

try {

sqlHelper = new SqlHelper();

ResultSet rs = sqlHelper.queryExecutes(sql, parass);

while(rs.next()) {

Vector row = new Vector();

row.add(rs.getString(1));

row.add(rs.getString(2));

row.add(rs.getString(3));

row.add(rs.getString(4));

row.add(rs.getString(5));

rowDates.add(row);

}

} catch (Exception e) {

// TODO: handle exception

} finally {

sqlHelper.close();

}

}

public void squeryStu(String sqls, String []parass)

{

SqlHelper sqlHelper = null;

//========初始化JTable信息

columnNames = new Vector();

rowDates = new Vector();

columnNames.add("剩余车位"); columnNames.add("已用车位");

try {

sqlHelper = new SqlHelper();

ResultSet rs = sqlHelper.queryExecutes(sqls, parass);

while(rs.next()) {

Vector row = new Vector();

row.add(rs.getString(1));

row.add(rs.getString(2));

rowDates.add(row);

}

} catch (Exception e) {

// TODO: handle exception

} finally {

sqlHelper.close();

}

}

@Override

public int getColumnCount() {

// TODO Auto-generated method stub

return this.columnNames.size();

}

@Override

public int getRowCount() {

// TODO Auto-generated method stub

return this.rowDates.size();

}

@Override

public Object getValueAt(int row, int col) {

// TODO Auto-generated method stub

if(!rowDates.isEmpty())

return ((Vector)this.rowDates.get(row)).get(col);

else

return null;

}

@Override

public String getColumnName(int column) {

// TODO Auto-generated method stub

return (String)this.columnNames.get(column);

}

}

package Manage;

import ADD.adminAddDialog;

import Model.adminModel;

import Manage.adminManage;

import Update.adminUpdateDialog;

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

public class adminManage extends JFrame implements ActionListener

{

/\*\*

\* @param args

\*/

public static void main(String[] args)

{

// TODO Auto-generated method stub

new adminManage();

}

//========面板控件

private JLabel queryLab = null;

private JTextField queryTxt = null;

private JButton queryBtn = null;

private JButton allBtn = null;

private JTable resultTb = null;

private JTable resultT = null;

private JScrollPane jsp = null;

private JScrollPane js = null;

private JButton addBtn = null;

private JButton deleteBtn = null;

private JButton updateBtn = null;

private JButton chargeBtn = null;

private JPanel top = null;

private JPanel bottom = null;

//========

private adminModel sm = null;

private adminModel st= null;

//构造函数

public adminManage()

{

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*初始化面板控件\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//========查询栏

super("停车场管理系统");

queryLab = new JLabel("请输入卡号:");

queryTxt = new JTextField(10);

queryBtn = new JButton("查询");

allBtn = new JButton("全部");

//......添加查询栏监听

queryBtn.addActionListener(this);

queryBtn.setActionCommand("query");

allBtn.addActionListener(this);

allBtn.setActionCommand("all");

//========增删改栏

addBtn = new JButton("办卡");

deleteBtn = new JButton("退卡");

updateBtn = new JButton("充值");

chargeBtn = new JButton("收费表");

//......添加增删改栏监听

addBtn.addActionListener(this);

addBtn.setActionCommand("add");

deleteBtn.addActionListener(this);

deleteBtn.setActionCommand("delete");

updateBtn.addActionListener(this);

updateBtn.setActionCommand("update");

chargeBtn.addActionListener(this);

chargeBtn.setActionCommand("charge");

//========创建窗口整体布局

//......顶层查询栏

top = new JPanel();

top.add(queryLab);

top.add(queryTxt);

top.add(queryBtn);

top.add(allBtn);

//......底层增删改栏

bottom = new JPanel();

bottom.add(addBtn);

bottom.add(deleteBtn);

bottom.add(updateBtn);

bottom.add(chargeBtn);

//......中间层显示栏

sm = new adminModel();

String sql = "select \* from 用户信息";

sm.queryStu(sql, null);

resultTb = new JTable(sm);

jsp = new JScrollPane(resultTb);

st = new adminModel();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

String sqls = "select \* from 停车场信息";

st.squeryStu(sqls, null);

resultT = new JTable(st);

js = new JScrollPane(resultT);

//......构建整体布局

this.add(top,BorderLayout.NORTH);

this.add(jsp,BorderLayout.CENTER);

this.add(bottom,BorderLayout.SOUTH);

this.add(js,BorderLayout.EAST);

//========设置窗口属性

this.setSize(1000, 800);

this.setLocationRelativeTo(null);

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

js.setPreferredSize(new Dimension(200, 50));

this.setVisible(true);

this.setResizable(false);

}

//监听

@Override

public void actionPerformed(ActionEvent e)

{

// TODO Auto-generated method stub

if(e.getActionCommand().equals("query")) {

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*查询\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//========获取输入用户的姓名

String name = queryTxt.getText().trim();

if(name.length() != 0) {

//========姓名输入有效时，执行查询

//......定义参数

String sql = "select \* from 用户信息 where 卡号=?";

String []parass = {name};

//......更新模型

jtableUpdate(sql, parass);

} else {

//========姓名为空时，设置提醒

JOptionPane.showMessageDialog(this, "卡号输入不能为空");

}

} else if(e.getActionCommand().equals("add")) {

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*添加\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

new adminAddDialog(this, "办卡信息", true);

String sql = "select \* from 用户信息";

jtableUpdate(sql, null);

} else if(e.getActionCommand().equals("all")) {

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*全部显示\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

String sql = "select \* from 用户信息";

jtableUpdate(sql, null);

} else if(e.getActionCommand().equals("delete")) {

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*删除\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//========获取选择行号

int rowNum = this.resultTb.getSelectedRow();

if(rowNum == -1) {

JOptionPane.showMessageDialog(this, "请选择一行");

return ;

}

//========获取员工ID号

String stuId = (String)sm.getValueAt(rowNum, 0);

//========删除用户

String sql = "delete from 用户信息 where 卡号=?";

String []parass = {stuId};

adminModel tmp = new adminModel();

tmp.cudStu(sql, parass);

//========更新模型

sql = "select \* from 用户信息";

jtableUpdate(sql, null);

} else if(e.getActionCommand().equals("update")) {

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*修改\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//========获取选择行号

int rowNum = this.resultTb.getSelectedRow();

if(rowNum == -1) {

JOptionPane.showMessageDialog(this, "请选择一行");

return ;

}

new adminUpdateDialog(this, "充值信息", true, sm, rowNum);

String sql = "select \* from 用户信息";

jtableUpdate(sql, null);

}

else if(e.getActionCommand().equals("charge"))

{ this.dispose();

new StudentManage();

String sql = "select \* from 收费表";

jtableUpdate(sql, null);

}

}

//========更新JTable内数据

public void jtableUpdate(String sql, String[] parass)

{

//......创建模型

sm = new adminModel();

sm.queryStu(sql, parass);

//......更新显示

resultTb.setModel(sm);

}

}

package Update;

import Model.StuModel;

import java.awt.BorderLayout;

import java.awt.Frame;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import Model.adminModel;

import Update.\*;

import javax.swing.JButton;

import javax.swing.JDialog;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class StuUpdateDialog extends JDialog implements ActionListener{

//=========面板控件

//......左侧标题栏

private JLabel IDLab,posLab,beginTimeLab,endTimeLab,TolTimeLab,feeLab;

//......右侧信息选择填写栏

private JTextField IDTxt,posTxt,beginTimeTxt,endTimeTxt,TolTimeTxt,feeTxt;

//......添加和取消按钮

private JButton addBtn,cancelBtn;

//......布局控件

private JPanel left,center,bottom;

//构造函数

public StuUpdateDialog(Frame owner, String title, boolean modal, StuModel sm, int rowNum)

{

//========重写父类方法

super(owner, title, modal);

//========左侧标签栏

IDLab = new JLabel("卡号:");

posLab = new JLabel("车位:");

beginTimeLab = new JLabel("开始时间:");

endTimeLab = new JLabel("结束时间:");

TolTimeLab = new JLabel("总时间: ");

feeLab= new JLabel("费用：");

//========右侧信息填写栏

IDTxt = new JTextField();

IDTxt.setText((String)sm.getValueAt(rowNum, 0));

IDTxt.setEditable(false);

posTxt = new JTextField();

posTxt.setText((String)sm.getValueAt(rowNum, 1));

beginTimeTxt = new JTextField();

beginTimeTxt.setText((String)sm.getValueAt(rowNum, 2));

endTimeTxt = new JTextField();

endTimeTxt.setText((String)sm.getValueAt(rowNum, 3));

TolTimeTxt=new JTextField();

TolTimeTxt.setText((String)sm.getValueAt(rowNum, 4));

feeTxt =new JTextField();

feeTxt.setText((String)sm.getValueAt(rowNum, 5));

//========添加和取消按钮

addBtn = new JButton("修改");

cancelBtn = new JButton("取消");

//......添加监听

addBtn.addActionListener(this);

addBtn.setActionCommand("update");

cancelBtn.addActionListener(this);

cancelBtn.setActionCommand("cancel");

//========创建布局

//......创建左边栏

left = new JPanel();

left.setLayout(new GridLayout(6, 1));

left.add(IDLab); left.add(posLab);

left.add(beginTimeLab);left.add(endTimeLab);

left.add(TolTimeLab);left.add(feeLab);

//......创建右边栏

center = new JPanel();

center.setLayout(new GridLayout(6, 1));

center.add(IDTxt); center.add(posTxt);

center.add(beginTimeTxt); center.add(endTimeTxt);

center.add(TolTimeTxt);

center.add(feeTxt);

//========底层添加和取消按钮

bottom = new JPanel();

bottom.add(addBtn);

bottom.add(cancelBtn);

//========整体布局

this.add(left,BorderLayout.WEST);

this.add(center,BorderLayout.CENTER);

this.add(bottom,BorderLayout.SOUTH);

//========设置窗口属性

this.setSize(300, 250);

this.setLocationRelativeTo(null);

this.setResizable(false);

this.setVisible(true);

}

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

if(e.getActionCommand().equals("update")) {

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*修改学生信息\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

StuModel tmp = new StuModel();

String sql = "update 收费表 set 车位=?,开始时间=?,结束时间=? ,总时间=?,费用=? where 卡号=?";

String []parass = {posTxt.getText(), beginTimeTxt.getText(),endTimeTxt.getText()

,TolTimeTxt.getText(),feeTxt.getText(),IDTxt.getText()};

if(!tmp.cudStu(sql, parass))

JOptionPane.showMessageDialog(this, "修改用户信息失败");

//========关闭窗口

this.dispose();

} else if(e.getActionCommand().equals("cancel")) {

//========关闭窗口

this.dispose();

}

}

}

**（2）**功能模块代码

package Update;

import Model.adminModel;

import java.awt.BorderLayout;

import java.awt.Frame;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JDialog;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class adminUpdateDialog extends JDialog implements ActionListener{

//=========面板控件

//......左侧标题栏

private JLabel idLab,pwdLab,remainLab,nameLab,telLab,signLab;

//......右侧信息选择填写栏

private JTextField idTxt,pwdTxt,remainTxt,nameTxt,telTxt,signTxt;

//......添加和取消按钮

private JButton addBtn,cancelBtn;

//......布局控件

private JPanel left,center,bottom;

public JTextField getRemainTxt() {

return remainTxt;

}

//构造函数

public adminUpdateDialog(Frame owner, String title, boolean modal, adminModel sm, int rowNum)

{

//========重写父类方法

super(owner, title, modal);

//========左侧标签栏

idLab = new JLabel("卡号:");

remainLab = new JLabel("充值金额:");

signLab= new JLabel("车牌：");

//========右侧信息填写栏

idTxt = new JTextField();

idTxt.setText((String)sm.getValueAt(rowNum, 0));

idTxt.setEditable(false);

remainTxt = new JTextField();

remainTxt.setText("0");

signTxt =new JTextField();

signTxt.setText((String)sm.getValueAt(rowNum, 4));

//========添加和取消按钮

addBtn = new JButton("修改");

cancelBtn = new JButton("取消");

//......添加监听

addBtn.addActionListener(this);

addBtn.setActionCommand("update");

cancelBtn.addActionListener(this);

cancelBtn.setActionCommand("cancel");

//========创建布局

//......创建左边栏

left = new JPanel();

left.setLayout(new GridLayout(3, 1));

left.add(idLab);

left.add(remainLab);left.add(signLab);

//......创建右边栏

center = new JPanel();

center.setLayout(new GridLayout(3, 1));

center.add(idTxt);

center.add(remainTxt);

center.add(signTxt);

//========底层添加和取消按钮

bottom = new JPanel();

bottom.add(addBtn);

bottom.add(cancelBtn);

//========整体布局

this.add(left,BorderLayout.WEST);

this.add(center,BorderLayout.CENTER);

this.add(bottom,BorderLayout.SOUTH);

//========设置窗口属性

this.setSize(300, 250);

this.setLocationRelativeTo(null);

this.setResizable(false);

this.setVisible(true);

}

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

if(e.getActionCommand().equals("update")) {

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*修改用户信息\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

String s=remainTxt.getText();

int i= Integer.parseInt(s);

adminModel tmp = new adminModel();

if(i>=0)

{

String sql = "update 用户信息 set 余额+=? where 卡号=?";

String []parass = {remainTxt.getText(), idTxt.getText()};

if(!tmp.cudStu(sql, parass))

JOptionPane.showMessageDialog(this, "修改用户信息失败");}

else

JOptionPane.showMessageDialog(this, "充值金额不能小于0");

//========关闭窗口

this.dispose();

} else if(e.getActionCommand().equals("cancel")) {

//========关闭窗口

this.dispose();

}

}

}

package Model;

import java.sql.ResultSet;

import java.util.Vector;

import javax.swing.table.AbstractTableModel;

import Helper.\*;

public class StuModel extends AbstractTableModel{

private Vector columnNames;

private Vector rowDates;

//

public StuModel()

{

String sql = "select \* from 收费表";

String []parass = {};

}

//========增删改用户

public boolean cudStu(String sql, String []parass)

{

return new SqlHelper().cudExecutes(sql, parass);

}

//========查询用户

public void queryStu(String sql, String []parass)

{

SqlHelper sqlHelper = null;

//========初始化JTable信息

columnNames = new Vector();

rowDates = new Vector();

columnNames.add("卡号"); columnNames.add("车位");

columnNames.add("开始时间"); columnNames.add("结束时间");

columnNames.add("总时间");

columnNames.add("费用");

try {

sqlHelper = new SqlHelper();

ResultSet rs = sqlHelper.queryExecutes(sql, parass);

while(rs.next()) {

Vector row = new Vector();

row.add(rs.getString(1));

row.add(rs.getString(2));

row.add(rs.getString(3));

row.add(rs.getString(4));

row.add(rs.getString(5));

row.add(rs.getString(6));

rowDates.add(row);

}

} catch (Exception e) {

// TODO: handle exception

} finally {

sqlHelper.close();

}

}

public void squeryStu(String sqls, String []parass)

{

SqlHelper sqlHelper = null;

//========初始化JTable信息

columnNames = new Vector();

rowDates = new Vector();

columnNames.add("剩余车位"); columnNames.add("已用车位");

try {

sqlHelper = new SqlHelper();

ResultSet rs = sqlHelper.queryExecutes(sqls, parass);

while(rs.next()) {

Vector row = new Vector();

row.add(rs.getString(1));

row.add(rs.getString(2));

rowDates.add(row);

}

} catch (Exception e) {

// TODO: handle exception

} finally {

sqlHelper.close();

}

}

@Override

public int getColumnCount() {

// TODO Auto-generated method stub

return this.columnNames.size();

}

@Override

public int getRowCount() {

// TODO Auto-generated method stub

return this.rowDates.size();

}

@Override

public Object getValueAt(int row, int col) {

// TODO Auto-generated method stub

if(!rowDates.isEmpty())

return ((Vector)this.rowDates.get(row)).get(col);

else

return null;

}

@Override

public String getColumnName(int column) {

// TODO Auto-generated method stub

return (String)this.columnNames.get(column);

}

}

package ADD;

import java.awt.BorderLayout;

import java.awt.Dialog;

import java.awt.Frame;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import Model.\*;

import javax.swing.\*;

public class adminAddDialog extends JDialog implements ActionListener{

//=========面板控件

//......左侧标题栏

private JLabel idLab,pwdLab,remainLab,nameLab,telLab,signLab;

//......右侧信息选择填写栏

private JTextField idTxt,pwdTxt,remainTxt,nameTxt,telTxt,signTxt;

//......添加和取消按钮

private JButton addBtn,cancelBtn;

//......布局控件

private JPanel left,center,bottom;

//构造函数

public adminAddDialog(Frame owner, String title, boolean modal)

{

//========重写父类方法

super(owner, title, modal);

//========左侧标签栏

idLab = new JLabel("卡号：");

nameLab = new JLabel("姓名： ");

remainLab = new JLabel("余额: ");

telLab = new JLabel("电话: ");

signLab= new JLabel("车牌号：");

//========右侧信息填写栏

idTxt = new JTextField();

pwdTxt = new JTextField();

nameTxt = new JTextField();

remainTxt = new JTextField();

telTxt=new JTextField();

signTxt=new JTextField();

//========添加和取消按钮

addBtn = new JButton("添加");

cancelBtn = new JButton("取消");

//......添加监听

addBtn.addActionListener(this);

addBtn.setActionCommand("add");

cancelBtn.addActionListener(this);

cancelBtn.setActionCommand("cancel");

//========创建布局

//......创建左边栏

left = new JPanel();

left.setLayout(new GridLayout(5, 1));

left.add(idLab);

left.add(nameLab);left.add(remainLab);

left.add(telLab);left.add(signLab);

//......创建右边栏

center = new JPanel();

center.setLayout(new GridLayout(5, 1));

center.add(idTxt);

center.add(nameTxt); center.add(remainTxt);

center.add(telTxt);;center.add(signTxt);

//========底层添加和取消按钮

bottom = new JPanel();

bottom.add(addBtn);

bottom.add(cancelBtn);

//========整体布局

this.add(left,BorderLayout.WEST);

this.add(center,BorderLayout.CENTER);

this.add(bottom,BorderLayout.SOUTH);

//========设置窗口属性

this.setSize(500, 300);

this.setLocationRelativeTo(null);

this.setResizable(false);

this.setVisible(true);

JFrame jf= new JFrame("停车场管理系统");

jf.setVisible(true);

}

@Override

public void actionPerformed(ActionEvent e)

{this.setLocationRelativeTo(null);

// TODO Auto-generated method stub

if(e.getActionCommand().equals("add")) {

StuModel tmp = new StuModel();

String sql = "insert into 用户信息 values(?,?,?,?,?)";

String []parass = {idTxt.getText(),nameTxt.getText(),remainTxt.getText()

,telTxt.getText(),signTxt.getText()};

if(!tmp.cudStu(sql, parass))

JOptionPane.showMessageDialog(this, "添加用户信息失败");

//========关闭窗口

this.dispose();

} else if(e.getActionCommand().equals("cancel")) {

//========关闭窗口

this.dispose();

}

}

}

**（3）**数据库代码

**package Helper;**

**import java.sql.\*;**

**public class SqlHelper {**

**//========数据库**

**private Connection ct = null;**

**private PreparedStatement ps = null;**

**private ResultSet rs = null;**

**private String driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";**

**private String url = "jdbc:sqlserver://127.0.0.1:1433;database=停车场管理系统";**

**private String user = "sa";**

**private String passwd = "1";**

**//========查询**

**public ResultSet queryExecutes(String sql, String []parass)**

**{**

**try {**

**//========1、加载驱动**

**Class.forName(driver);**

**//========2、连接**

**ct = DriverManager.getConnection(url, user, passwd);**

**//========3、创建PreparedStatement**

**ps = ct.prepareStatement(sql);**

**//========4、给问号赋值**

**if(parass != null) {**

**for(int i = 0; i < parass.length; i++) {**

**ps.setString(i + 1, parass[i]);**

**}**

**}**

**//========5、执行**

**rs = ps.executeQuery();**

**} catch (Exception e) {**

**// TODO: handle exception**

**e.printStackTrace();**

**} finally {**

**//this.close();**

**}**

**//========返回值**

**return rs;**

**}**

**//========增删改**

**public boolean cudExecutes(String sql, String []parass)**

**{**

**boolean b = true;**

**try {**

**//========1、加载驱动**

**Class.forName(driver);**

**//========2、连接**

**ct = DriverManager.getConnection(url, user, passwd);**

**//========3、创建PreparedStatement**

**ps = ct.prepareStatement(sql);**

**//========4、给问号赋值**

**for(int i = 0; i < parass.length; i++) {**

**ps.setString(i + 1, parass[i]);**

**}**

**//========5、执行**

**if(ps.executeUpdate() != 1) b = false;**

**} catch (Exception e) {**

**// TODO: handle exception**

**b = false;**

**e.printStackTrace();**

**} finally {**

**this.close();**

**}**

**//========返回值**

**return b;**

**}**

**//========关闭资源**

**public void close()**

**{**

**try {**

**if(rs!=null) rs.close();**

**if(ps!=null) ps.close();**

**if(ct!=null) ct.close();**

**} catch (Exception e2) {**

**// TODO: handle exception**

**e2.printStackTrace();**

**}**

**}**

**}**

**五、实践周收货**

**1.**通过实践周的学习，使我进一步了解所学的基础知识和新的知识，了解计算机技术和信息管理技术和制作项目基本操作方法，比较全面的掌握计算机应用技能和信息管理技能从而提高从事实际工作的技能水平。

**2.**在实践中雷老师带领我们学习并实践，利用JAVA、HTM、SQL等语言和工具完成了一个完整的项目《停车场管理系统》，让我更加喜欢计算机，在各个小的细节分析并解决相应的问题提出对应的方法提高了实际工作能力和解决问题的能力。

**3.**最后回看这几天在校的实践周生活，我十分的激动并却也十分的感激在老师的讲解之下我从新对人工智能有了新的认识，也认识到了JAVA这种面向对象的语言和超文本标记语言HTML。对于一个大二新生来说见识的还不够这不但为我们增加了见识也提醒了我们不能做温水内面的青蛙要敢于去学习新的知识、敢于去挑战新的技术、敢于求不断挑战自己，只有这样才能在这竞争激烈的时代生存。这个实践周我同老师也进行许多的交流这也使我对自己未来人生道路有了更加明确的目标，我学习到了新的技术也学到了许多感悟和方法这是

**4.**在工作总结中，老师分析了计算机行业目前所需要的人才短缺，人工智能、大数据、云计算、游戏开发，研发管理，区块链等方面都需要我们去努力学习，刻苦钻研，所以要努力学习，经跟时代发展。