 ****

**软 件 学 院**

**实践成果一报告**

**课程名称： 实践成果一**

**课题名称： 图书管理系统开发**

**专 业： 软件工程**

**班 级： RB软工卓161**

**学 号： 201619140220**

**学生姓名： 崔烁豪**

**指导教师： 韩玉民**

2018年1月

**中原工学院软件学院**

**实践成果任务书**

|  |  |  |  |
| --- | --- | --- | --- |
| **姓名** | 崔烁豪 | **班级** | 卓越161班 |
| **题目** | **图书管理系统开发** | | |
| **设**  **计**  **任**  **务** | 图书管理系统是图书馆必不可少的软件系统。方便图书管理，实现图书智能管理，为图书馆提供快速的图书信息检索功能，快捷的图书借阅和归还流程。系统主要功能包括：  1.读者管理  对读者基本信息进行管理，包括读者基本信息的增删改查，读者基本信息主要包括读者编号、姓名、性别、职业、出生日期、身份证号、电话、登记日期等。  2.图书管理  对图书信息进行基本的记录、分类，包括图书的编号、书名、作者、价格、录入时间、页数、操作员。  3.借阅管理  对读者的借阅的图书进行管理，包括读者借阅的书名、书号、借阅时间、归还时间、操作员。  在后续实践课题中可开发功能更全的Web版本。  数据存储要求：采用数据库存储数据。  开发工具：MySQL 5.6/SQL Server，Eclipse JavaEE/My Eclipse | | |
| **时**  **间**  **进**  **度** | 第1周（2017-09-11~2017-09-15）：对需求进行深入分析，并设计出架构与数据库，仔细理解需求，规划功能。  第2周（2017-09-16~2017-10-20）：学习并完成每一步的开发，在过程中发现问题并改进问题，学习使用第三方的API来完成系统，并进行下一步的功能该进，并优化界面。  第3周（2017-09-21~2017-11-05）：调试测试，通过实际用户的操作，提出建议并修改，撰写报告。 | | |
| **原 主**  **始 要**  **资 参**  **料 考**  **与 文**  **献** | [1] 刘瑞新.张义兵.SQL Sever数据库技术及应用教程[M]：电子工业出版社  [2] 欧二强.JAVA编程手记—从实践中学习JAVA[M]：清华大学出版社 | | |

**摘 要**

通过这次的课题完善了窗口的组件优化，增加用户的体验，简便用户操作过程，来使用户或管理员更好的去管理图书系统的信息。

我做的是图书信息管理和读者信息管理，在这两个模块中，对图书信息、读者信息的数据库进行界面化展示和操作，具体精细化为增添、修改、删除以及查询这些功能。

首先是界面的展示，需要将数据库中的图书信息和读者信息展现在窗口上，为此而设计各种UI界面，其次就是功能的展示了。这些功能都需要连接数据库来实现，为此设计出了Dao类来在数据库与UI展示架构桥梁。Dao类中有多种方法，而且这些方法根据所要实现的功能来设计，所以就需要有不同的返回值与参数。这两个模块中的主界面展现出所有读者和图书信息，将其打印在表格中，在最下面有增添、修改、删除这三个按钮，当用户点击增添时，会弹出一个增添界面，只需用户在对话框中填写即可，确认后隐藏此界面，并刷新主界面显示添加的信息。当用户修改时，需要将鼠标移动到需要修改信息的那一列并点击，然后点击修改，弹出修改界面，确认后隐藏此界面，并刷新主界面显示出修改后的信息，删除功能较为简单，并不需要写出界面。总之每回操作完成后，查询界面就会更新。

本文详细阐述了系统的设计开发与实现，本文共5章，第1章主要介绍了图书管理系统的功能与作用，第2章是对图书管理系统的分析，第3章是分析本系统数据库的设计，第4章是对系统功能的具体实现进行阐述，第5章介绍在实践完成过程中的一些心得体会，以及对此次实践成果的评价，进行总结。

关键词：读者信息；图书信息；数据库

**目录**

[第1章 概述 1](#_Toc503969934)

[1.1 图书管理系统的功能 1](#_Toc503969935)

[1.2 图书管理系统的作用 1](#_Toc503969936)

[第2章 系统分析 2](#_Toc503969937)

[2.1问题描述 2](#_Toc503969938)

[2.2技术分析 2](#_Toc503969939)

[2.3需求分析 2](#_Toc503969940)

[第3章 数据库分析与设计 3](#_Toc503969941)

[3.1 实体分析 3](#_Toc503969942)

[3.2 关系设计 4](#_Toc503969943)

[3.3数据库设计 4](#_Toc503969944)

[第4章 系统实现 6](#_Toc503969945)

[4.1 图书馆管理系统登陆界面 6](#_Toc503969946)

[4.2 图书信息管理 7](#_Toc503969947)

[4.3 读者信息管理 8](#_Toc503969948)

[第5章 结束语 10](#_Toc503969949)

[附录: 主要源程序 12](#_Toc503969950)

[指导老师评语 29](#_Toc503969951)

# 第1章 概述

本章主要阐述图书管理系统的功能与作用。

## 1.1 图书管理系统的功能

第一部分，你得先登录之后才能进行一下的操作。管理员登录后，就可以管理图书、管理读者，对图书和读者进行增删改差。

## 1.2 图书管理系统的作用

方便图书管理，实现图书智能管理，为图书馆提供快速的图书信息检索功能，快捷的图书借阅和归还流程。

# 第2章 系统分析

本章主要是对图书管理系统项目进行合理性的分析，通过问题描述、技术分析、需求分析这三个方面连描述这个项目。

## 2.1问题描述

图书管理系统是图书馆必不可少的软件系统。为方便图书管理，实现图书智能管理，为图书馆提供快速的图书信息检索功能，快捷的图书借阅和归还流程。经过分析，决定设计一个图书管理系统。

## 2.2技术分析

系统主要运用Eclipse，它同时也能开发跨平台的应用程序，如开发使用微软操作系统的手机的程序等。总体来说是一个非常庞大的软件，甚至包含代码测试功能。同时还需要MySQL创建数据库 ,和java连接数据库的操作。

## 2.3需求分析

方便图书管理，实现图书智能管理，为图书馆提供快速的图书信息检索功能，快捷的图书借阅和归还流程。

该系统主要包括以下功能：

（1）管理员、读者登陆。

（2）查询图书信息。

（3）增添读者信息、修改指定读者信息、删除指定读者信息

（4）增添图书信息、修改指定图书信息、删除指定图书信息

# 第3章 数据库分析与设计

数据库拥有很强的功能，可以建立一个数据库把我们所需要的数据给存储起来，并可以进行增删改查的操作，还可以对数据库进行保存和他的安全性等等多的工能。我们采用MySQL数据库来保存会员的信息及储值记录、消费记录。

## 3.1 实体分析

本系统涉及的实体包括管理员、读者、图书等。经过分析，管理员实体E-R图如图3.1，读者实体E-R图如图3.2，图书实体E-R图如图3.3。

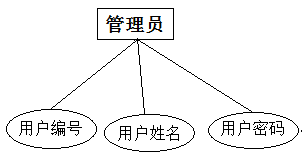


图3.1 管理员实体E-R图

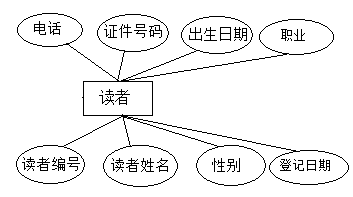


图3.2 读者实体E-R图

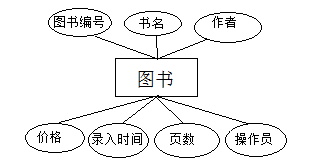


图3.3 图书实体E-R图

## 3.2 关系设计

这里有三个实体，分别是：管理员、读者、图书，它们之间的关系分别是：管理员管理读者，一对多；管理员管理图书，一对多；读者借阅图书，多对多。系统E-R图如图3.4所示。

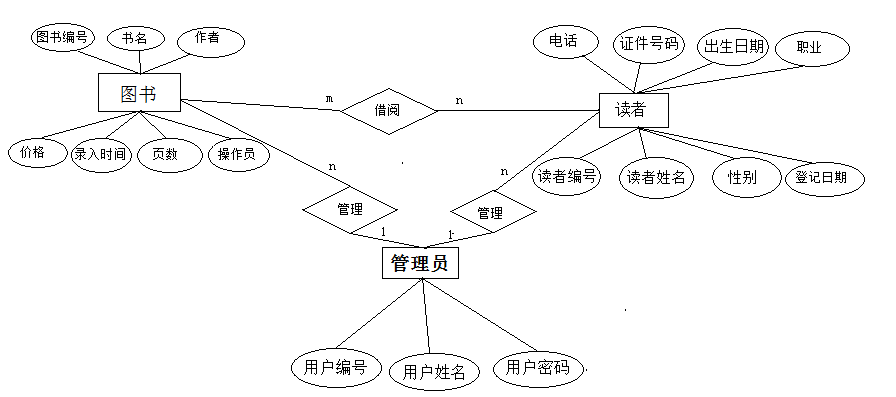


图3.4 系统E-R图

## 3.3数据库设计

图书管理系统的数据库设计总共有四个表：管理员表、读者表、图书表。表的具体结构见表3.1、表3.2和表3.3。

表3.1 管理员表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **描述** | **数据类型及长度** | **是否为空** | **主键** |
| id | 用户编号 | int(11) | 否 | 是 |
| usename | 用户姓名 | varchar(30) | 是 |  |
| pwd | 用户密码 | varchar(30) | 是 |  |

表3.2 读者表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **描述** | **数据类型及长度** | **是否为空** | **主键** |
| id | 读者编号 | int(11) | 否 | 是 |
| name | 读者姓名 | varchar(20) | 是 |  |
| sex | 性别 | varchar(4) | 是 |  |
| vocation | 职业 | varchar(50) | 是 |  |
| birthday | 出生日期 | date | 是 |  |
| paperNO | 证件号码 | varchar(20) | 是 |  |
| tel | 电话 | varchar(20) | 是 |  |
| createDate | 登记日期 | date | 是 |  |

表3.3 图书表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **描述** | **数据类型及长度** | **是否为空** | **主键** |
| id | 图书编号 | int(11) | 否 | 是 |
| bookname | 书名 | varchar(70) | 是 |  |
| author | 作者 | varchar(30) | 是 |  |
| price | 价格 | double(8) | 是 |  |
| inTime | 录入时间 | date | 是 |  |
| page | 页数 | int(11) | 是 |  |
| operator | 操作员 | int(11) | 是 |  |

# 第4章 系统实现

## 4.1 图书馆管理系统登陆界面

为方便图书管理，实现图书智能管理，为图书馆提供快速的图书信息检索功能，快捷的图书借阅和归还流程。

4.1.1图书管理系统登陆界面

图书管理系统的界面设计是整个程序的入口，管理员和读者的登陆都是在此开始的，所有用户必须通过此登录界面来实现登陆，未登录或登录失败的都将无法进行下一步操作。除了针对不同的用户来呈现不同的登录后的界面外，还有就是丰富界面的细节，为此添加书本图片美化界面。

4.1.2**表图格式说明**

图书管理系统登陆界面表如表4.1所示

表4.1 图书管理系统登陆界面表

|  |  |  |
| --- | --- | --- |
| 帐号 | 密码 | 背景 |
| 读者帐号登陆 | 读者密码 | 书本图片 |
| 管理员帐号登陆 | 管理员密码 | 图书管理、读者管理 |

图书管理系统的登陆界面如图4.1所示

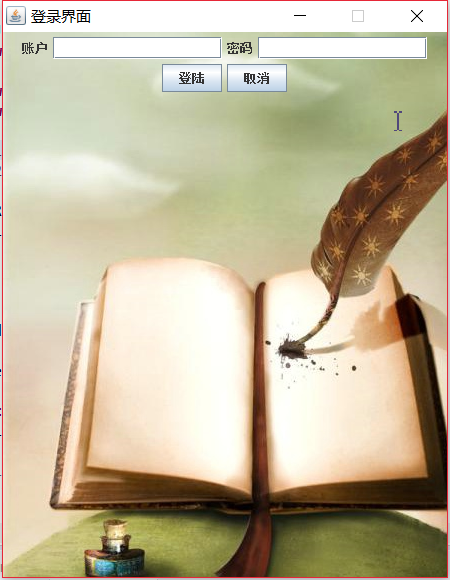


图4.1 图书管理系统的登陆界面

## 4.2 图书信息管理

图书信息管理模块就是为了方便管理员的对图书信息的管理。

4.2.1 图书信息管理界面意义

这个界面实现管理员与数据库之间的交互。将数据库中图书信息的数据打印到界面上展示，管理员也可以通过界面中的几个操作功能实现数据库中信息的修改。

4.2.2 图书信息管理界面内容

图书信息管理界面表如表4.2所示

表4.2图书信息管理界面表

|  |  |  |
| --- | --- | --- |
| 增加 | 修改 | 删除 |
| 增添图书信息 | 修改指定图书信息 | 删除指定图书信息 |

图书信息管理界面如图4.2所示

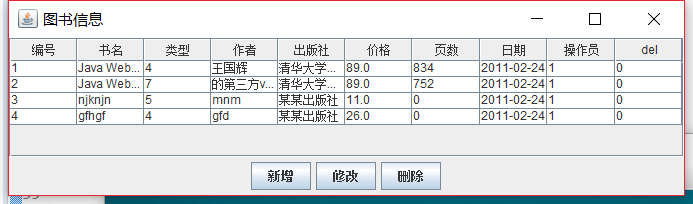


图4.2 图书信息管理界面

## 4.3 读者信息管理

读者信息管理模块就是为了方便管理员对读者信息的管理。

4.3.1 读者信息管理界面意义

这个界面实现管理员与数据库之间的交互。将数据库中读者信息的数据打印到界面上展示，管理员也可以通过界面中的几个操作功能实现数据库中读者信息的修改。

4.3.2 读者信息管理界面内容

读者信息管理界面表如表4.3所示

表4.3 读者信息管理界面表

|  |  |  |
| --- | --- | --- |
| 增加 | 修改 | 删除 |
| 增添读者信息 | 修改指定读者信息 | 删除指定读者信息 |

读者信息管理界面如图4.3所示

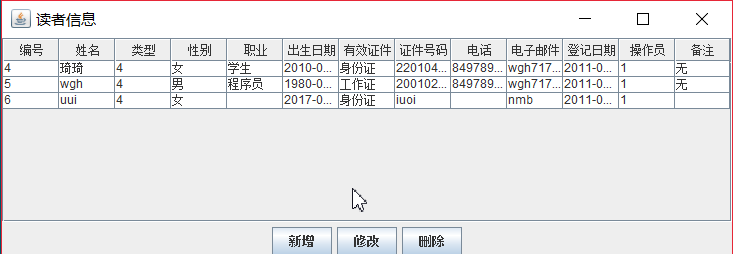


图4.3 读者信息管理界面

# 第5章 结束语

本章主要是对实践成果的总结，首先介绍课题完成情况和心得、然后就是这个系统的不足和改进。

本次实践课题在韩玉民老师的悉心指导和严格要求下已完成，经过两个月的努力，最终较好地完成的课题开发，实现了要求的系统功能。

两个月以来忙碌紧张而又有条不紊的实践使我有机会对本专业的基本理论、专业知识和基本技术有了更深入的了解和体会，使我在这学期中所学到的知识得到了系统和升华，真正达到了学以致用。

（1）从心态方面来说，在整个课程设计的过程中我学到了做任何事情所要有的态度经过这项二级课题的亲身研究，我从中学习到了很多。在整个课程设计的过程中我学到了做任何事情所要有的态度和心态，首先我明白了做学问要一丝不苟，对于出现的任何问题和偏差都不要轻视，要通过正确的途径去解决，在做事情的过程中要有耐心和毅力，不要一遇到困难就打退堂鼓，只要坚持下去就可以找到思路去解决问题的。在工作中要学会与人合作的态度，认真听取别人的意见，这样做起事情来就可以事倍功半。

（2）从实践方面来说，我充分的体会到了“实践出真知”这一点，书本上的知识是不够的，只有把理论与实践相结合才能够真正的学到知识。在一个系统开发的过程中编码不是重要的，重要的是分析系统、建立起系统模型。经过三周的努力，个人通讯录管理系统终于完成了。这个课题不像以往所做的课题，这里不仅仅包括了书本上的知识，这里大部分运用了课外知识，所以在做课题之前，我从图书馆借了许多关于系统开发的书本，在做课题之中，这些课本对我的帮助很大，这里有课本以外的知识，所以我认为学习不仅仅是书本上的知识，只有你通过实践和训练才能把书本上的知识扩展和应用 ，才能学到课本以外的知识，实践出真知这句话真的不错。还有就是做课题并不是那么的容易，在成功之前我经过了许多次的更改，每一次都是一种提高。我就得在错误中学到的知识更加的有用。在做课题之中，我也遇到了许多的问题，有问题是好事，有问题才有提高，解决问题找老师，在韩老师的指导下，问题解决了。

（3）系统的不足之处：登陆权限设置有待改进，毕竟时间有限，操作员、管理员、读者这三个群体之间的权限并未优化太深

（4）刚开始时，为了在做课题就对这个课题开始研究，借了许多的课外书，有做了一些想法，但是在做课题之中，还是遇到了一些问题，虽说这个会员管理系统不是多么的麻烦，但是还是遇到了问题，还好问题不是多大。而且一个成功的系统要符合用户的需要，要以用户为中心，开发出用户想要的系统，而不能自以为是，想当然。

总之，这次课题实践让对所学知识有了更深的利用，学会了应用，实践能力有了很大提高。

在此非常感谢韩老师的指导和同学们的帮助。

# 附录: 主要源程序

**登录界面代码：**

package com.libsys.main;

import java.awt.BorderLayout;

import java.awt.Graphics;

import java.awt.Image;

import java.awt.Toolkit;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.ImageIcon;

import javax.swing.JApplet;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTextField;

import javax.swing.border.Border;

import javax.tools.Tool;

import com.libsys.manager.dao.Managerdao;

public class LoginUI {

public static void main(String[] args) {

JFrame jf=new JFrame("登录界面");

JLabel jl1= new JLabel("账户");

JLabel jl2=new JLabel("密码");

JButton jb1=new JButton("登陆");

JButton jb2=new JButton("取消");

BjPanel bj=new BjPanel("E://j.jpg");

JTextField jtf1=new JTextField(15);

JTextField jtf2=new JTextField(15);

jf.setSize(450, 580);

jf.setLocationRelativeTo(null);

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

bj.add(jl1);

bj.add(jtf1);

bj.add(jl2);

bj.add(jtf2);

bj.add(jb1,BorderLayout.SOUTH);

bj.add(jb2);

jf.setResizable(false);

jf.add(bj);

jf.setVisible(true);

jb1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent )

Managerdao md=new Managerdao();

if (md.log(jtf1.getText(), jtf2.getText())){

JOptionPane.showMessageDialog(null, "登陆成功");

jf.setVisible(false);

new MainUI();}

else{

if(!md.log(jtf1.getText())){

JOptionPane.showMessageDialog(null, "没有此账号");

}

else JOptionPane.showMessageDialog(null, "密码错误");

}

}

});

jb2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

System.exit(0);

}

});

}

}

**菜单界面代码：**

package com.libsys.main;

import java.awt.BorderLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JFrame;

import javax.swing.JMenu;

import javax.swing.JMenuBar;

import javax.swing.JMenuItem;

import com.libsys.book.ui.BookinfoUI;

import com.libsys.book.ui.BooktypeUI;

import com.libsys.reader.ui.ReaderUI;

public class MainUI extends JFrame{

private JMenuBar jmb=new JMenuBar();

private JMenu jm1=new JMenu("图书管理");

private JMenu jm2= new JMenu("读者管理");

private JMenuItem jmt1 =new JMenuItem("图书信息");

private JMenuItem jmt2 =new JMenuItem("图书类型");

private JMenuItem jmt3=new JMenuItem("读者信息");

public MainUI(){

this.setTitle("图书管理系统");

this.setSize(760,480);

this.setLocationRelativeTo(null);

BjPanel bj=new BjPanel("E://j2.jpg");

jm1.add(jmt1);

jm1.add(jmt2);

jm2.add(jmt3);

jmb.add(jm1);

jmb.add(jm2);

this.add(bj);

this.add(jmb,BorderLayout.NORTH);

this.setResizable(false);

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

this.setVisible(true);

addActionListener();

}

public void addActionListener(){

jmt1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

new BookinfoUI();

}

});

jmt2.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

new BooktypeUI();

}

});

jmt3.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

new ReaderUI();

}

});

}

}

package com.libsys.main;

import java.awt.Dimension;

import java.awt.Graphics;

import java.awt.Image;

import java.awt.LayoutManager;

import java.awt.Toolkit;

import javax.swing.JPanel;

public class BjPanel extends JPanel {

private Image im;

public BjPanel(String s) {

// TODO Auto-generated constructor stub

im=Toolkit.getDefaultToolkit().getImage(s);

}

public void paintComponent(Graphics g){

g.drawImage(im, 0, 0,this);

}

}

**数据库操作代码：**

package com.libsys.book.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

import com.libsys.book.pojo.Booktype;

import com.libsys.util.DB;

public class Booktypedao {

private DB db=new DB();

private Connection conn;

private ResultSet rs;

private Statement stmt;

private PreparedStatement pstmt;

public boolean delBooktype(int id){

conn=db.getConnection();

String sql="delete from tb\_booktype where id=?";

try {

pstmt=conn.prepareStatement(sql);

pstmt.setInt(1, id);

int a=pstmt.executeUpdate();

return a>0;

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally{

db.colse(pstmt, conn, rs);;

}

return false;

}

//查询单条记录的方法

public Booktype getReaderTypeById(int id){

Booktype bt=null;

String sql="select id,typename from tb\_booktype where id=?";

conn=db.getConnection();

try {

pstmt=conn.prepareStatement(sql);

pstmt.setInt(1, id);

rs=pstmt.executeQuery();

//遍历结果集

if(rs.next()){

bt=new Booktype();

bt.setId(rs.getInt("id"));

bt.setTypename(rs.getString("typename"));

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally{

db.colse(pstmt, conn, rs);;

}

return bt;

}

//修改的方法

public boolean updateReaderType(Booktype bt){

conn=db.getConnection();

String sql="update tb\_booktype set typename =? where id=?";

try {

pstmt=conn.prepareStatement(sql);

pstmt.setString(1,bt.getTypename());

pstmt.setInt(2, bt.getId());

int a=pstmt.executeUpdate();

return a>0;

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally{

db.colse(pstmt, conn, rs);

}

return false;

}

public boolean addReaderType(Booktype bt){

conn=db.getConnection();

String sql="insert into tb\_booktype (typename) values(?)";

try {

pstmt=conn.prepareStatement(sql);

pstmt.setString(1, bt.getTypename());

int a=pstmt.executeUpdate();

return a>0;

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally{

db.colse(pstmt, conn, rs);

}

return false;

}

//查询的方法必须有返回值

public List<Booktype> queryAll(){

List<Booktype> list=new ArrayList<Booktype>();

String sql="select id,typename from tb\_booktype";

conn=db.getConnection();

try {

pstmt=conn.prepareStatement(sql);

rs=pstmt.executeQuery();

//遍历结果集

while(rs.next()){

Booktype bt=new Booktype();

bt.setId(rs.getInt("id"));

bt.setTypename(rs.getString("typename"));

//将对象放置到集合中

list.add(bt);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally{

db.colse(pstmt, conn, rs);

}

return list;

}

}

package com.libsys.book.pojo;

import java.util.Date;

public class Bookinfo {

private int id;

private String bookname;

private int typeid;

private String author;

private String ISBN;

private Double price;

private int page;

private Date inTime;

private int operator;

private int del;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getBookname() {

return bookname;

}

public void setBookname(String bookname) {

this.bookname = bookname;

}

public int getTypeid() {

return typeid;

}

public void setTypeid(int typeid) {

this.typeid = typeid;

}

public String getAuthor() {

return author;

}

public void setAuthor(String author) {

this.author = author;

}

public String getISBN() {

return ISBN;

}

public void setISBN(String iSBN) {

ISBN = iSBN;

}

public Double getPrice() {

return price;

}

public void setPrice(Double price) {

this.price = price;

}

public int getPage() {

return page;

}

public void setPage(int page) {

this.page = page;

}

public Date getInTime() {

return inTime;

}

public void setInTime(Date inTime) {

this.inTime = inTime;

}

public int getOperator() {

return operator;

}

public void setOperator(int operator) {

this.operator = operator;

}

public int getDel() {

return del;

}

public void setDel(int del) {

this.del = del;

}

}

**图书修改界面代码：**

package com.libsys.book.ui;

import java.awt.BorderLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.List;

import java.util.Vector;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.table.DefaultTableModel;

import javax.swing.table.TableModel;

import com.libsys.book.dao.Bookdao;

import com.libsys.book.pojo.Bookinfo;

public class BookinfoUI extends JFrame {

private JPanel jp=new JPanel();

private JButton jb1=new JButton("新增");

private JButton jb2=new JButton("修改");

private JButton jb3=new JButton("删除");

private JTable jtable=new JTable();

private JScrollPane jsp=new JScrollPane(jtable);

public BookinfoUI(){

this.setTitle("图书信息");

this.setSize(1800,1000);

this.setLocationRelativeTo(null);

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

jp.add(jb1);

jp.add(jb2);

jp.add(jb3);

this.add(jp,BorderLayout.SOUTH);

this.add(jsp);

addActionLiserter();

loadtable();

this.setVisible(true);

}

private void addActionLiserter() {

// TODO Auto-generated method stub

//添加

jb1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

new BookinfoUISave(BookinfoUI.this);

}

});

//修改

jb2.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

int index=jtable.getSelectedRow();

if(index==-1){

//如果值为-1说明没有选中

JOptionPane.showMessageDialog(null, "请选择你要修改的数据");

return; //结束方法

}

String sid= (String) jtable.getValueAt(index, 0);

//将字符串转换成 int类型

int id=Integer.parseInt(sid);

//创建一个修改窗体的对象

new BookinfoUIEdit(BookinfoUI.this,id);

}

});

//删除

jb3.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

int index=jtable.getSelectedRow();

if(index==-1){

//如果值为-1说明没有选中

JOptionPane.showMessageDialog(null, "请选择你要修改的数据");

return; //结束方法

}

String sid= (String) jtable.getValueAt(index, 0);

//将字符串转换成 int类型

int id=Integer.parseInt(sid);

//创建一个修改窗体的对象

new BookinfoUIDel(BookinfoUI.this,id);

}

});

}

public void loadtable(){

Bookdao bookdao=new Bookdao();

List<Bookinfo> list=bookdao.queryBookinfo();

Vector<Vector<String>> data=new Vector<Vector<String>>();

for(int i=0;i<list.size();i++){

Bookinfo bki=list.get(i);

Vector<String> v1=new Vector<String>();

v1.add(String.valueOf(bki.getId()));

v1.add(bki.getBookname());

v1.add(String.valueOf(bki.getTypeid()));

v1.add(bki.getAuthor());

v1.add(bki.getISBN());

v1.add(bki.getPrice().toString());

v1.add(String.valueOf(bki.getPage()));

v1.add(String.valueOf(bki.getInTime()));

v1.add(String.valueOf(bki.getOperator()));

v1.add(String.valueOf(bki.getDel()));

data.add(v1);

}

Vector<String> head=new Vector<String>();//{"id","c","类型","作者","出版社","价格","页数","日期","操作员","del"};

head.add("编号");

head.add("书名");

head.add("类型");

head.add("作者");

head.add("出版社");

head.add("价格");

head.add("页数");

head.add("日期");

head.add("操作员");

head.add("del");

//创建一个表格模型

TableModel tm=new DefaultTableModel(data, head);

//将模型中的数据设置到表格中

jtable.setModel(tm);

}

}

package com.libsys.book.ui;

import javax.swing.JOptionPane;

import com.libsys.book.dao.Bookdao;

public class BookinfoUIDel {

private BookinfoUI bui;

private int id;

public BookinfoUIDel(BookinfoUI bui, int id) {

this.bui=bui;

this.id=id;

int a=JOptionPane.showConfirmDialog(null, "确认要删除吗？");

if(a==0){

Bookdao bookdao=new Bookdao();

if(bookdao.delBookinfo(id)){

JOptionPane.showMessageDialog(null, "成功删除");

bui.loadtable();

}

else JOptionPane.showMessageDialog(null, "删除失败");

}

else bui.loadtable();

}

}

**图书修改界面代码：**

package com.libsys.book.ui;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.Vector;

import javax.swing.JButton;

import javax.swing.JComboBox;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTextField;

import com.libsys.book.dao.Bookdao;

import com.libsys.book.pojo.Bookinfo;

public class BookinfoUIEdit extends JFrame{

private JPanel jp=new JPanel();

private JPanel jp1=new JPanel();

private JPanel jp2=new JPanel();

private JPanel jp3=new JPanel();

private JPanel jp4=new JPanel();

private JPanel jp5=new JPanel();

private JPanel jp6=new JPanel();

private JPanel jp7=new JPanel();

private JLabel jl1=new JLabel("图书名称");

private JLabel jl2=new JLabel("图书作者");

private JLabel jl3=new JLabel("图书类型");

private JLabel jl4=new JLabel("价格");

private JLabel jl5=new JLabel("图书页数");

private JLabel jl6=new JLabel("出版社");

private JTextField jtf1=new JTextField(20);

private JTextField jtf2=new JTextField(20);

private JTextField jtf4=new JTextField(20);

private JTextField jtf5=new JTextField(20);

private JTextField jtf6=new JTextField(20);

private JButton jb1=new JButton("保存");

private JButton jb2=new JButton("取消");

private Vector<String> v=new Vector<String>();

JComboBox<String> jbb=new JComboBox<String>(v);

private BookinfoUI bui;

private int id;

public BookinfoUIEdit(BookinfoUI bui,int id){

this.id=id;

this.bui=bui;

this.setTitle("修改图书");

this.setSize(400,500);

this.setLocationRelativeTo(null);

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

jp.setLayout(new GridLayout(7, 2));

addcomponent();

addActionListener();

load();

this.setVisible(true);

}

public void addcomponent(){

v.add("历史");

v.add("地理");

v.add("语文");

v.add("科学");

v.add("数学");

v.add("自然");

v.add("计算机");

jbb.setSelectedIndex(1);

jp1.add(jl1);

jp1.add(jtf1);

jp2.add(jl2);

jp2.add(jtf2);

jp3.add(jl3);

jp3.add(jbb);

jp4.add(jl4);

jp4.add(jtf4);

jp5.add(jl5);

jp5.add(jtf5);

jp6.add(jl6);

jp6.add(jtf6);

jp7.add(jb1);

jp7.add(jb2);

jp.add(jp1);

jp.add(jp2);

jp.add(jp3);

jp.add(jp4);

jp.add(jp5);

jp.add(jp6);

jp.add(jp7);

this.add(jp);

}

public void addActionListener(){

jb1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

String name=jtf1.getText().trim();

if(name.length()<=1 || name.equals("")){

JOptionPane.showMessageDialog(null,"请输入书名");

return;

}

String author=jtf2.getText().trim();

if(author.length()<=1 || author.equals("")){

JOptionPane.showMessageDialog(null,"请输入作者");

return;

}

String price=jtf4.getText().trim();

String page =jtf5.getText().trim();

String ISBN=jtf6.getText().trim();

Bookinfo book =new Bookinfo();

book.setBookname(name);

book.setAuthor(author);

book.setTypeid(jbb.getItemCount());

book.setPrice(Double.parseDouble(price));

book.setPage(Integer.parseInt(page));

book.setISBN(ISBN);

Bookdao bookdao=new Bookdao();

if(bookdao.editBookinfo(book,id)){

JOptionPane.showMessageDialog(null, "修改成功");

// 添加成功后 添加页面要隐藏

BookinfoUIEdit.this.setVisible(false);

// 刷新表格的方法

bui.loadtable();

}

else System.out.println("gg");

}

});

jb2.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

load();

}

});

}

public void load(){

Bookdao bookdao=new Bookdao();

Bookinfo bk=bookdao.getBookinfoid(id);

jtf1.setText(bk.getBookname());

jtf2.setText(bk.getAuthor());

jbb.setSelectedIndex(bk.getTypeid());

jtf4.setText(String.valueOf(bk.getPrice()));

jtf5.setText(String.valueOf(bk.getPage()));

jtf6.setText( bk.getISBN());

}

}

**图书添加界面代码：**

package com.libsys.book.ui;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.Vector;

import javax.swing.JButton;

import javax.swing.JComboBox;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTextField;

import javax.swing.tree.DefaultTreeCellEditor.EditorContainer;

import com.libsys.book.dao.Bookdao;

import com.libsys.book.pojo.Bookinfo;

public class BookinfoUISave extends JFrame{

private BookinfoUI bui;

private JPanel jp=new JPanel();

private JPanel jp1=new JPanel();

private JPanel jp2=new JPanel();

private JPanel jp3=new JPanel();

private JPanel jp4=new JPanel();

private JPanel jp5=new JPanel();

private JPanel jp6=new JPanel();

private JPanel jp7=new JPanel();

private JLabel jl1=new JLabel("图书名称");

private JLabel jl2=new JLabel("图书作者");

private JLabel jl3=new JLabel("图书类型");

private JLabel jl4=new JLabel("价格");

private JLabel jl5=new JLabel("图书页数");

private JLabel jl6=new JLabel("出版社");

private JTextField jtf1=new JTextField(20);

private JTextField jtf2=new JTextField(20);

private JTextField jtf4=new JTextField(20);

private JTextField jtf5=new JTextField(20);

private JTextField jtf6=new JTextField(20);

private JButton jb1=new JButton("保存");

private JButton jb2=new JButton("取消");

private Vector<String> v=new Vector<String>();

private JComboBox<String> jbb=new JComboBox<String>(v);

public BookinfoUISave(BookinfoUI bui){

this.bui=bui;

this.setTitle("添加图书");

this.setSize(400,500);

this.setLocationRelativeTo(null);

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

jp.setLayout(new GridLayout(7, 2));

addcomponent();

addActionListener();

this.setVisible(true);

}

public void addcomponent(){

v.add("历史");

v.add("地理");

v.add("语文");

jbb.setSelectedIndex(1);

jp1.add(jl1);

jp1.add(jtf1);

jp2.add(jl2);

jp2.add(jtf2);

jp3.add(jl3);

jp3.add(jbb);

jp4.add(jl4);

jp4.add(jtf4);

jp5.add(jl5);

jp5.add(jtf5);

jp6.add(jl6);

jp6.add(jtf6);

jp7.add(jb1);

jp7.add(jb2);

jp.add(jp1);

jp.add(jp2);

jp.add(jp3);

jp.add(jp4);

jp.add(jp5);

jp.add(jp6);

jp.add(jp7);

this.add(jp);

}

public void addActionListener(){

jb1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

JOptionPane.showConfirmDialog(null, "确认保存？");

Bookinfo book =new Bookinfo();

book.setBookname(jtf1.getText());

book.setAuthor(jtf2.getText());

book.setTypeid(jbb.getSelectedIndex());

book.setPrice(Double.parseDouble(jtf4.getText()));

book.setPage(Integer.parseInt(jtf5.getText()));

book.setISBN(jtf6.getText());

Bookdao bookdao=new Bookdao();

if(bookdao.addBookinfo(book)){

JOptionPane.showMessageDialog(null, "添加成功");

BookinfoUISave.this.setVisible(false);

// 刷新表格的方法

bui.loadtable();

}

else {JOptionPane.showMessageDialog(null, "添加失败");

}

}

});

jb2.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

BookinfoUISave.this.setVisible(false);

}

});

}

**图书的修改删除界面代码：**

package com.libsys.book.ui;

import java.awt.BorderLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.List;

import java.util.Vector;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.table.DefaultTableModel;

import javax.swing.table.TableModel;

import com.libsys.book.dao.Booktypedao;

import com.libsys.book.pojo.Booktype;

public class BooktypeUI extends JFrame{

JButton save =new JButton("新增");

JButton edit =new JButton("修改");

JButton del =new JButton("删除");

// 一个表格

JTable jtable = new JTable();

//滚动条

JScrollPane jsp=new JScrollPane(jtable);

//一个面板

JPanel jp = new JPanel();

public BooktypeUI(){

this.setTitle("图书类型");

this.setSize(300,300);

this.setLocationRelativeTo(null);

jp.add(save);

jp.add(edit);

jp.add(del);

this.add(jp,BorderLayout.SOUTH);

this.add(jsp);

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

addActionLiserter();

loadtable(); // 给表格添加数据

this.setVisible(true);

}

// 给按钮添加事件的方法

public void addActionLiserter(){

//给添加数据的按钮添加事件

save.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

new BooktypeUIAdd(BooktypeUI.this);

}

});

//给修改的按钮添加事件

edit.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// 先判断用户选择了那条数据

//返回用户先择的是第几条记录

int index=jtable.getSelectedRow();

if(index==-1){

//如果值为-1说明没有选中

JOptionPane.showMessageDialog(null, "请选择你要修改的数据");

return; //结束方法

}

//获取用户选中记录的编号

String sid= (String) jtable.getValueAt(index, 0);

//将字符串转换成 int类型

int id=Integer.parseInt(sid);

//创建一个修改窗体的对象

new BooktypeUIEdit(BooktypeUI.this, id);

}

});

//给删除的按钮添加事件

del.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

//判断用户是否选择到记录

int index=jtable.getSelectedRow();

if(index==-1){

//如果值为-1说明没有选中

JOptionPane.showMessageDialog(null, "请选择你要修改的数据");

return; //结束方法

}

int a= JOptionPane.showConfirmDialog(null, "你是真的真的要删除这个记录吗 ");

if(a==0){ //a的值是0 说明用户是确定删除

//或用户删除的id

String sid= (String) jtable.getValueAt(index, 0);

//将id转换成 int类型

int id=Integer.parseInt(sid);

//创建dao层 从数据库中删除数据

Booktypedao btd=new Booktypedao();

if(btd.delBooktype(id)){

JOptionPane.showMessageDialog(null, "删除成功");

}else{

JOptionPane.showMessageDialog(null, "删除失败");

}

//刷新表格

loadtable();

}

}

});

}

// 给表格添加数据

public void loadtable(){

//从数据库中查询到所有的数据

Booktypedao btd=new Booktypedao();

List<Booktype> list=btd.queryAll();

//将list数据放置到一个Vertor集合中

Vector<Vector<String>> vdata=new Vector<Vector<String>>();

for(Booktype bt:list){

Vector<String> v1=new Vector<String>();

//将集合中的每一个对象都放置到新的集合中

v1.add(String.valueOf(bt.getId()));

v1.add(bt.getTypename());

vdata.add(v1);

}

Vector<String> head=new Vector<String>();

head.add("编号");

head.add("类型");

//创建一个表格模型

TableModel tm=new DefaultTableModel(vdata, head);

//将模型中的数据设置到表格中

jtable.setModel(tm);

}

}

# 指导老师评语

成绩：

指导老师签字：

年 月 日