界面



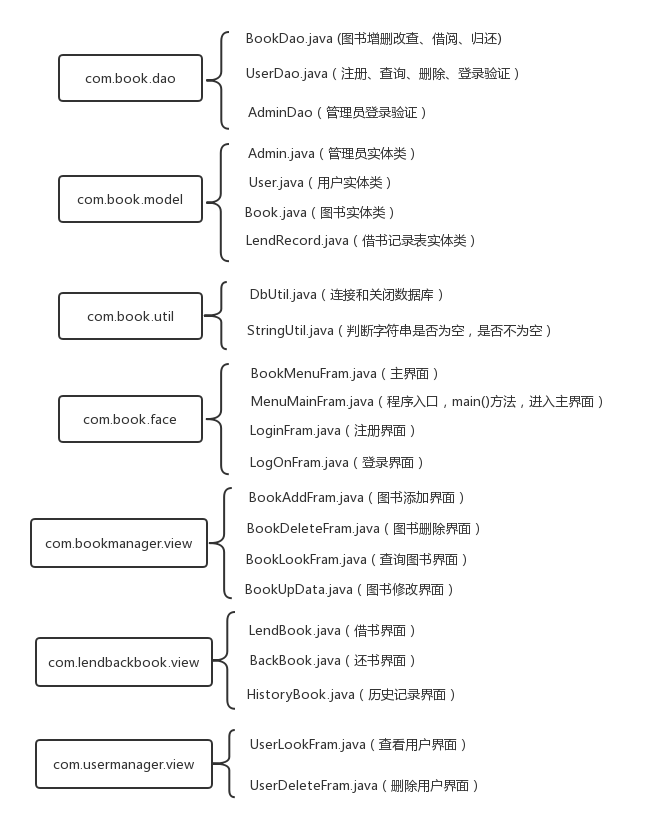




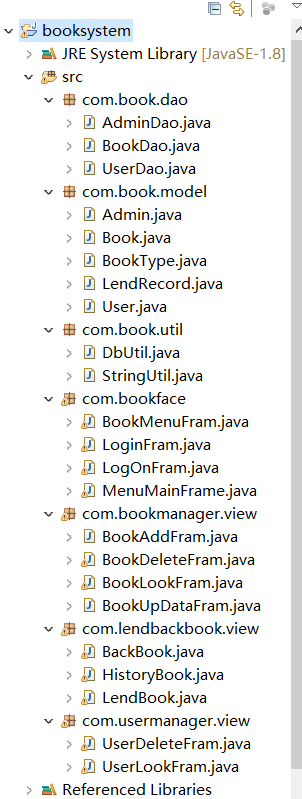
等.......

Java层次分析

1. 逻辑图



2.包结构



3.重要代码分析

a、Dao类（以BookDao为例）

package com.book.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import com.book.model.Book;

import com.book.model.User;

import com.book.util.StringUtil;

public class BookDao {

/\*\*

\* 图书添加

\* @param con

\* @param book

\* @return

\* @throws Exception

\*/

public int add(Connection con, Book book) throws Exception {

//插入图书前 检查bookId是否已经存在，若存在，返回0

String sqlid = "select \* from book where bookId = ?";

PreparedStatement pstmtid = (PreparedStatement) con.prepareStatement(sqlid);

pstmtid.setString(1, book.getBookId());

ResultSet rs = pstmtid.executeQuery();

if(rs.next()) {

return 0;

}

String sql = "insert into book values(?,?,?,?,?,?)";

PreparedStatement pstmt = (PreparedStatement) con.prepareStatement(sql);

pstmt.setString(1, book.getBookId());

pstmt.setString(2, book.getBookName());

pstmt.setString(3, book.getPress());

pstmt.setString(4, book.getAuthor());

pstmt.setString(5, book.getTypeId());

pstmt.setInt(6, book.getLend());

return pstmt.executeUpdate();

}

/\*\*

\* 图书查询

\* @param con

\* @param bookMessage

\* @return

\* @throws Exception

\*/

public ResultSet list(Connection con, Book bookMessage) throws Exception {

//将book表和bookType表连接

//StringBuffer sb = new StringBuffer("select \* from book,booktype where book.typeId=booktype.typeId");

StringBuffer sb = new StringBuffer("select \* from book where bookId = bookId ");

//向sb中添加，并进行模糊查询

if(StringUtil.isNotEmpty(bookMessage.getBookId())) {

sb.append(" and bookId like '%" + bookMessage.getBookId() + "%'");

}

//按书名模糊查询

if(StringUtil.isNotEmpty(bookMessage.getBookName())) {

sb.append(" and book.bookName like '%" + bookMessage.getBookName() + "%'");

}

//按作者模糊查询

if(StringUtil.isNotEmpty(bookMessage.getAuthor())) {

sb.append(" and book.author like '%" + bookMessage.getAuthor() + "%'");

}

//按图书类型模糊查询

if(StringUtil.isNotEmpty(bookMessage.getTypeId())) {

sb.append(" and book.typeId like '%" + bookMessage.getTypeId() + "%'");

}

PreparedStatement pstmt=(PreparedStatement) con.prepareStatement(sb.toString());

return pstmt.executeQuery();

}

//历史记录查询

public ResultSet listHistory(Connection con, User userMessage) throws Exception {

String sql = "select recordId,userName, bookName, (case back when 0 then '否' when 1 then '是' end) as back from lendrecord where userName = ?";

PreparedStatement pstmt = (PreparedStatement) con.prepareStatement(sql);

pstmt.setString(1, userMessage.getUsername());

return pstmt.executeQuery();

}

/\*\*

\* 图书删除

\* @param con

\* @param id

\* @return

\* @throws Exception

\*/

public int delete(Connection con, String id) throws Exception {

String sql = "delete from book where bookId = ?";

PreparedStatement pstmt = (PreparedStatement) con.prepareStatement(sql);

pstmt.setString(1, id);

return pstmt.executeUpdate();

}

/\*\*

\* 图书修改

\* @param con

\* @param bookMessage1

\* @return

\* @throws Exception

\*/

public int update(Connection con, Book bookMessage1) throws Exception {

String sql = "update book set bookName=?, press=?, author=? where bookId=?";

PreparedStatement pstmt = (PreparedStatement) con.prepareStatement(sql);

pstmt.setString(1, bookMessage1.getBookName());

pstmt.setString(2, bookMessage1.getPress());

pstmt.setString(3, bookMessage1.getAuthor());

pstmt.setString(4, bookMessage1.getBookId());

return pstmt.executeUpdate();

}

//检查该id的图书是否存在,若存在，返回0，否则返回1

public ResultSet bookCheck(Connection con, String id) throws Exception{

//插入图书前 检查bookId是否已经存在，若存在，返回0

String sqlid = "select \* from book where bookId = ?";

PreparedStatement pstmtid = (PreparedStatement) con.prepareStatement(sqlid);

pstmtid.setString(1, id);

ResultSet rs = pstmtid.executeQuery();

return rs;

}

//

/\*\*

\* 图书借阅

\* 修改book表中的lend字段和 借阅记录表（lendrecord）中的back字段 lend=1,back=0

\* @param con

\* @param textid

\* @param bookName

\* @param userMessage

\* @return

\* @throws Exception

\*/

public int lend (Connection con, String textid, String bookName, User userMessage) throws Exception {

//先修改book表中图书的状态

String sql = "update book set lend=? where bookId=?";

PreparedStatement pstmt = (PreparedStatement) con.prepareStatement(sql);

pstmt.setString(1, "1");

pstmt.setString(2, textid);

pstmt.executeUpdate();

//插入借阅记录表(lendrecord) 信息

String insql = "insert into lendrecord (userId,userName,bookId,bookName,back)values(?,?,?,?,?)";

PreparedStatement lendpstmt = (PreparedStatement) con.prepareStatement(insql);

lendpstmt.setString(1, userMessage.getUserId());

lendpstmt.setString(2, userMessage.getUsername());

lendpstmt.setString(3, textid);

lendpstmt.setString(4, bookName);

lendpstmt.setString(5, "0");

lendpstmt.executeUpdate();

return 0;//成功

}

//图书归还

/\*\*

\* 修改book表中的lend字段和 借阅记录表（lendrecord）中的back字段 lend=0,back=1

\* @param con

\* @param textid

\* @param userMessage

\* @return

\* @throws Exception

\*/

public int back (Connection con, String textid, User userMessage) throws Exception {

//先修改book表中图书的状态

String sql = "update book set lend=? where bookId=?";

PreparedStatement pstmt = (PreparedStatement) con.prepareStatement(sql);

pstmt.setString(1, "0");

pstmt.setString(2, textid);

pstmt.executeUpdate();

//修改借阅记录表(lendrecord) 信息 ，back=1,表示已经归还

String insql = "update lendrecord set back=? where bookId=? and userName=? ";

PreparedStatement lendpstmt = (PreparedStatement) con.prepareStatement(insql);

lendpstmt.setString(1, "1");

lendpstmt.setString(2, textid);

lendpstmt.setString(3, userMessage.getUsername());

lendpstmt.executeUpdate();

return 0;//成功

}

}

b.Model类（以Book为例）

package com.book.model;

public class Book {

private String bookId;//图书id

private String bookName;//图书名

private String press;//图书出版社

private String author;//图书作者

private String typeId;//图书类别id

private int lend;//图书是否被借, 是为1, 默认为0

public String getBookId() {

return bookId;

}

public void setBookId(String bookId) {

this.bookId = bookId;

}

public String getBookName() {

return bookName;

}

public void setBookName(String bookName) {

this.bookName = bookName;

}

public String getPress() {

return press;

}

public void setPress(String press) {

this.press = press;

}

public String getAuthor() {

return author;

}

public void setAuthor(String author) {

this.author = author;

}

public String getTypeId() {

return typeId;

}

public void setTypeId(String typeId) {

this.typeId = typeId;

}

public int getLend() {

return lend;

}

public void setLend(int lend) {

this.lend = lend;

}

}

c.Util类  
连接和关闭数据库

package com.book.util;

import java.sql.Connection;

import java.sql.DriverManager;

public class DbUtil {

private String url = "jdbc:mysql://localhost:3306/work?serverTimezone=UTC";

private String username = "root";

private String password = "123456";

//连接数据库方法

public Connection getCon() throws Exception {

//Class.forName("com.mysql.jdbc.Driver");

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con = (Connection) DriverManager.getConnection(url, username, password);

return con;

}

//关闭数据库方法

public void closeCon(java.sql.Connection con) throws Exception {

if(con != null) {

con.close();

}

}

/\* public static void main(String[] args) {

DbUtil dbUtil = new DbUtil();

try {

dbUtil.getCon();

System.out.println("数据库连接成功");

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace(); //在命令行打印异常信息在程序中出错的位置及原因。

System.out.println("数据库连接错误");

}

} \*/

}

d、工具类  
判断字符串是否为空，是否不为空

package com.book.util;

public class StringUtil {

/\*\*

\* 判断字符串是否为空

\* @param str

\* @return

\*/

public static boolean isEmpty(String str) {

if(str == null || "".equals(str.trim())) {//trim()的作用是去掉字符串两端的多余的空格

return true; //注意，是两端的空格，且无论两端的空格有多少个都会去掉，

}else { // 当然，中间的那些空格不会被去掉

return false;

}

}

/\*\*

\* 判断不为空

\* @param str

\* @return

\*/

public static boolean isNotEmpty(String str) {

if(str != null && !"".equals(str.trim())) {

return true;

} else {

return false;

}

}

}

e、Fram类（以BookMenuFram为例）  
各种界面

package com.bookface;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.ImageIcon;

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 com.book.util.StringUtil;

import com.bookmanager.view.BookAddFram;

import com.bookmanager.view.BookDeleteFram;

import com.bookmanager.view.BookLookFram;

import com.bookmanager.view.BookUpDataFram;

import com.lendbackbook.view.BackBook;

import com.lendbackbook.view.HistoryBook;

import com.lendbackbook.view.LendBook;

import com.usermanager.view.UserDeleteFram;

import com.usermanager.view.UserLookFram;

public class BookMenuFram extends JFrame {

static String loginId = null;

static String loginName = null;//若是管理员登录，存管理员的名字，否则，存普通用户的名字

//标记用户类别，若是0代表普通用户，1代表管理员，默认是0

static int flag = 0;

//系统主界面

public static void BookMenuFram() {

JFrame frame = new JFrame();//主菜单窗口

frame.setLayout(null);

//开6个面板，方便设置位置

JPanel pan1 = new JPanel();//系统名字

JPanel pan2 = new JPanel();//图书管理

JPanel pan3 = new JPanel();//图书管理的操作按钮

JPanel pan4 = new JPanel();//借还书

JPanel pan5 = new JPanel();//借还书的操作按钮

JPanel pan6 = new JPanel();//用户管理

JPanel pan7 = new JPanel();//用户管理的操作按钮

JPanel pan8 = new JPanel();//放登录退出按钮

JPanel pan9 = new JPanel();//放置图片

JTextField text = new JTextField(); // 文本框，可删除

text.setText("未登录！");//设置提示未登录

// JLabel text = new JLabel();//提示框，不可删除

// text.setText("未登录！");//设置提示未登录

//提示框

JLabel label1 = new JLabel("小型图书管理系统");

JLabel label2 = new JLabel("图书管理");

JLabel label3 = new JLabel("借还书");

JLabel label4 = new JLabel("用户管理");

//添加图片

ImageIcon im = new ImageIcon("imge/1.jpg");

JLabel pac = new JLabel(im);

pac.setBounds(355,125, im.getIconWidth(), im.getIconHeight());

pan9.add(pac);

pan9.setBounds(355,125, 932, 630);

//按钮

JButton button1 = new JButton("登录");

JButton button2 = new JButton("图书查询");

JButton button3 = new JButton("图书添加");

JButton button4 = new JButton("图书修改");

JButton button5 = new JButton("图书删除");

JButton button6 = new JButton("办理借书");

JButton button7 = new JButton("办理还书");

JButton button8 = new JButton("历史查询");

JButton button9 = new JButton("查询用户");

JButton button10 = new JButton("删除用户");

JButton button11 = new JButton("退出");

//设置颜色格式

Color blacka = new Color(30,144,255);

Color blackb = new Color(0,255,255);

Color blackc = new Color(255,69,0);

Color blackd = new Color(255,215,0);

//设置字体大小对象

Font font = new Font("宋体",Font.BOLD,80);//标题字体大小

Font f = new Font("宋体",Font.BOLD,30);//提示框字体大小

Font f1 = new Font("宋体",Font.BOLD,20);//text

//设置按钮的大小

button2.setPreferredSize(new Dimension(200,65));

button3.setPreferredSize(new Dimension(200,65));

button4.setPreferredSize(new Dimension(200,65));

button5.setPreferredSize(new Dimension(200,65));

button6.setPreferredSize(new Dimension(200,65));

button7.setPreferredSize(new Dimension(200,65));

button8.setPreferredSize(new Dimension(200,65));

button9.setPreferredSize(new Dimension(200,65));

button10.setPreferredSize(new Dimension(200,65));

text.setPreferredSize(new Dimension(170,35));//

//设置字体大小

button2.setFont(f);

button3.setFont(f);

button4.setFont(f);

button5.setFont(f);

button6.setFont(f);

button7.setFont(f);

button8.setFont(f);

button9.setFont(f);

button10.setFont(f);

text.setFont(f1);

label1.setFont(font);//设置标题字体

label2.setFont(f);

label3.setFont(f);

label4.setFont(f);

//向面板中添加组件

pan1.add(label1);

pan1.setBackground(blacka);

pan1.setBounds(0, 0, 1100, 120);//大标题

//pan8定位在右上角，放登录、退出和显示框

pan8.add(text);

pan8.add(button1);

pan8.add(button11);

pan8.setBackground(blacka);

pan8.setBounds(1100, 0, 190, 120);

pan2.add(label2);

pan2.setBackground(blackb);

pan2.setBounds(0, 120, 150, 284);//图书管理

pan3.add(button2);

pan3.add(button3);

pan3.add(button4);

pan3.add(button5);

pan3.setBounds(150, 120, 200, 284);

pan4.add(label3);

pan4.setBackground(blackc);

pan4.setBounds(0, 404, 150, 213);//借还书

pan5.add(button6);

pan5.add(button7);

pan5.add(button8);

pan5.setBounds(150, 404, 200, 213);

pan6.add(label4);

pan6.setBackground(blackd);

pan6.setBounds(0, 617, 150, 142);//用户管理

pan7.add(button9);

pan7.add(button10);

pan7.setBounds(150, 617, 200, 200);

String noLogin = "未登录！";//JVM先查看常量池中有没有，如有，地址指向它，若没有，创建新对象保存

//获取登录后的用户名

if(StringUtil.isNotEmpty(LogOnFram.userName())) {

loginName = LogOnFram.userName();

text.setText("欢迎您，" + loginName);

}

if(StringUtil.isNotEmpty(LogOnFram.adminName())) {

loginName = LogOnFram.adminName();

text.setText("欢迎您，" + loginName);

flag = 1;

}

//获取登录后的id

if(StringUtil.isNotEmpty(LogOnFram.id())) {

loginId = LogOnFram.id();

}

//登录监听

button1.addActionListener(new ActionListener() {//登录监听

public void actionPerformed(ActionEvent e) {

String getText = text.getText().toString();//1

if(getText.equals(noLogin)) {

LogOnFram.LogOnFram();

frame.dispose();//关闭当前登录界面

} else {

JOptionPane.showMessageDialog(null, "请您先退出！");

}

}

});

//各项操作监听,button2--button10

button2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String getText = text.getText().toString();//1

if(getText.equals(noLogin)) { //equals()方法如果不重写，就是比较的字符串内容。而'=='比较的是地址

JOptionPane.showMessageDialog(null, "请您先登录！");

return;

} else {

BookLookFram.BookLookFram();//图书查询

}

}

});

button3.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String getText = text.getText().toString();//1

if(getText.equals(noLogin)) {

JOptionPane.showMessageDialog(null, "请您先登录！");

return;

} else {

if(flag == 1) {//如果flag是1，这可以添加图书

BookAddFram.bookAddFram();//图书添加

} else {

JOptionPane.showMessageDialog(null, "管理员才可以执行此操作！");

}

}

}

});

button4.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String getText = text.getText().toString();//1

if(getText.equals(noLogin)) {

JOptionPane.showMessageDialog(null, "请您先登录！");

return;

} else {

if(flag == 1) {//如果flag是1，这可以添加图书

BookUpDataFram.bookUpDataFram();//修改图书

} else {

JOptionPane.showMessageDialog(null, "管理员才可以执行此操作！");

}

}

}

});

button5.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String getText = text.getText().toString();//1

if(getText.equals(noLogin)) {

JOptionPane.showMessageDialog(null, "请您先登录！");

return;

} else {

if(flag == 1) {//如果flag是1，这可以添加图书

BookDeleteFram.BookDeleteFram();//删除图书

} else {

JOptionPane.showMessageDialog(null, "管理员才可以执行此操作！");

}

}

}

});

button6.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String getText = text.getText().toString();//1

if(getText.equals(noLogin)) {

JOptionPane.showMessageDialog(null, "请您先登录！");

return;

} else { //办理借书

LendBook.LendBook();

}

}

});

button7.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String getText = text.getText().toString();//1

if(getText.equals(noLogin)) {

JOptionPane.showMessageDialog(null, "请您先登录！");

return;

} else { //办理还书

BackBook.BackBook();

}

}

});

button8.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String getText = text.getText().toString();//1

if(getText.equals(noLogin)) {

JOptionPane.showMessageDialog(null, "请您先登录！");

return;

} else { //历史查询

HistoryBook.HistoryBook();

}

}

});

button9.addActionListener(new ActionListener() {//查询用户

public void actionPerformed(ActionEvent e) {

String getText = text.getText().toString();//1

if(getText.equals(noLogin)) {

JOptionPane.showMessageDialog(null, "请您先登录！");

return;

} else {

if(flag == 1) {//如果flag是1，这可以添加图书

UserLookFram.UserLookFram();//用户查询

} else {

JOptionPane.showMessageDialog(null, "管理员才可以执行此操作！");

}

}

}

});

button10.addActionListener(new ActionListener() {//删除用户

public void actionPerformed(ActionEvent e) {

String getText = text.getText().toString();//1

if(getText.equals(noLogin)) {

JOptionPane.showMessageDialog(null, "请您先登录！");

return;

} else {

if(flag == 1) {//如果flag是1，这可以添加图书

UserDeleteFram.UserDeleteFram();//用户删除

} else {

JOptionPane.showMessageDialog(null, "管理员才可以执行此操作！");

}

}

}

});

button11.addActionListener(new ActionListener() {//退出登录

public void actionPerformed(ActionEvent e) {

String getText = text.getText().toString();//1

if(getText.equals(noLogin)) {

JOptionPane.showMessageDialog(null, "请您先登录！");

return;

} else {

// frame.dispose();

// LogOnFram.LogOnFram();

System.exit(0);

}

}

});

//向容器中添加JPanel面板

frame.add(pan1);

frame.add(pan2);

frame.add(pan3);

frame.add(pan4);

frame.add(pan5);

frame.add(pan6);

frame.add(pan7);

frame.add(pan8);

frame.add(pan9);

//窗口设置

frame.setBounds(310, 100, 1300, 800);

frame.setResizable(false);//设置窗口不能扩大

frame.setVisible(true);

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

}

public static String loginName() {

return loginName;

}

public static String loginId() {

return loginId;

}

}