

南京信息工程大学

滨江学院

软件体系结构课程设计



题 目 流动图书馆

学生姓名毛济洲

学 号 20182344050

院 系 物联网工程学院

专 业 软件工程

指导教师王玉祥

1. 项目背景

软件系统名字：小型图书管理系统

本项目任务开发者：毛济洲

本项目用户：图书管理员、用户

2. 参考资料

《java2 实用教程》、

《软件体系结构与设计》、

百度、

GitHub、

CSDN。

3. 任务概述

设计一个程序实现流动 图书馆管理,通过连接数据库实现图书的查询,加购,添加删除用户、图书。锻炼我们设计 java 程序的整体思维和能力,通过动手实践巩固上学期在课堂上学到的知识。同时在编写程序的过程中也会引发新的思考,再通过找资料等途径去解决问题。强化学生的创新能力和自主解决问题的能力。

4.编写目的

- 1、对“在线书店概要设计的阶段任务成果形成文档,以便阶段验收、评审,最终的文档验收。
- 2、对需求阶段的文档再次确认过程,对前一阶段需求没有做充分或错误的提出修改。
- 3、明确整个系统的功能框架和数据库结构,为下一阶段的详细设计、编码、和测试提供参考依据。
- 4、明确编码规范和命名规范,统一程序界面。

4. 实现语言

系统管理员必须具备一定的网络及数据库的操作和管理知识,并具有高度的责任感和强烈的安全意识。

一般用户除了具有一定的计算机应用能力外,还必须各司其职,不得越权操作,不得随意泄露口令,以共同维护整个系统的安全和正常运行。

服务器操作系统: Windows10、macOS

客户端操作系统: Windows10、macOS

数据库: mysql-8.0.16-winx64

JDK: 1.8.0_202JDBC: 8.0.16

开发使用的是集成开发环境软件 eclipse。

版本列表:

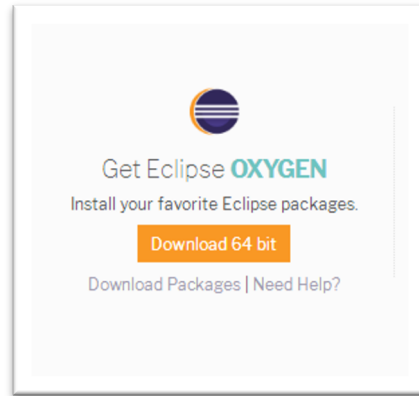
从 2006 年起, Eclipse 基金会每年都会安排同步发布。至今, 同步发布主要在 6 月进行, 并且会在接下来的 9 月及 2 月释放出 SR1 及 SR2 版本。

在 Neon 版本之后, 每个主版本一般在 6 月发布, 后续更新版在九月, 十二月, 和 3 月发布。

版本代号	平台版本	主要版本发行日期	SR1 发行日期	SR2 发行日期
Callisto	3.2	2006 年 6 月 26 日	N/A	N/A
Europa	3.3	2007 年 6 月 27 日	2007 年 9 月 28 日	2008 年 2 月 29 日
Ganymede	3.4	2008 年 6 月 25 日	2008 年 9 月 24 日	2009 年 2 月 25 日
Galileo	3.5	2009 年 6 月 24 日	2009 年 9 月 25 日	2010 年 2 月 26 日
Helios	3.6	2010 年 6 月 23 日	2010 年 9 月 24 日	2011 年 2 月 25 日
Indigo	3.7	2011 年 6 月 22 日	2011 年 9 月 23 日	2012 年 2 月 24 日
Juno	3.8 及 4.2	2012 年 6 月 27 日	2012 年 9 月 28 日	2013 年 3 月 1 日
Kepler	4.3	2013 年 6 月 26 日	2013 年 9 月 27 日	2014 年 2 月 28 日
Luna	4.4	2014 年 6 月 25 日	2014 年 9 月 25 日	N/A
Mars	4.5	2015 年 6 月 24 日	N/A	N/A
Neon	4.6	2016 年 6 月 22 日	N/A	N/A
Oxygen	4.7	2017 年 6 月 28 日	N/A	N/A
Photon	4.8	2018 年 6 月	N/A	N/A

获取途径:

去官网下载: <http://www.eclipse.org/downloads/>



5. 安装配置过程

需要安装 jdk 和配置环境变量，然后再安装集成开发工具 eclipse 即可。

安装配置 jdk 过程：

(1) 下载 jdk 和 jre，安装在 java 根目录下不同文件夹里

(2) 配置环境变量：

新建 JAVA_HOME 变量：jdk 的安装目录

添加 PATH 变量：%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;

新建 CLASSPATH 变量：.;%JAVA_HOME%\lib;%JAVA_HOME%\lib\tools.jar

(3) 检验是否配置成功

运行 cmd，输入 java，显示命令帮助信息

输入 javac，显示命令帮助信息

输入 java -version，显示 java 版本信息

输入 javac -version，显示 java 编译器版本信息

此时配置成功。

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.19041.264]
(c) 2019 Microsoft Corporation. 保留所有权利。

C:\Users\Harry>java -version
java version "1.8.0_251"
Java(TM) SE Runtime Environment (build 1.8.0_251-b08)
Java HotSpot(TM) 64-Bit Server VM (build 25.251-b08, mixed mode)

C:\Users\Harry>
```

6. 用户界面

界面界面要求美观、大方，操作方便，符合用户习惯。

7. 软件系统主要目标

该系统有普通用户和管理员，如果是用户登录，则只能对图书进行查询、借阅、归还等基础操作，如果是管理员登录，还可以对图书进行增删改查等各项操作以及对用户进行查询和删除操作。

8.设计思路

- 1、用户登录，有管理员和普通用户
- 2、普通用户可以借阅，归还，查询
- 3、管理员可以管理图书和用户

①.设计数据库

(1) 用户表(user):

用户 id, 用户名, 用户密码

(2) 管理员表(admin):

管理员 id, 管理员名, 管理员密码

(3) 图书信息表(book):

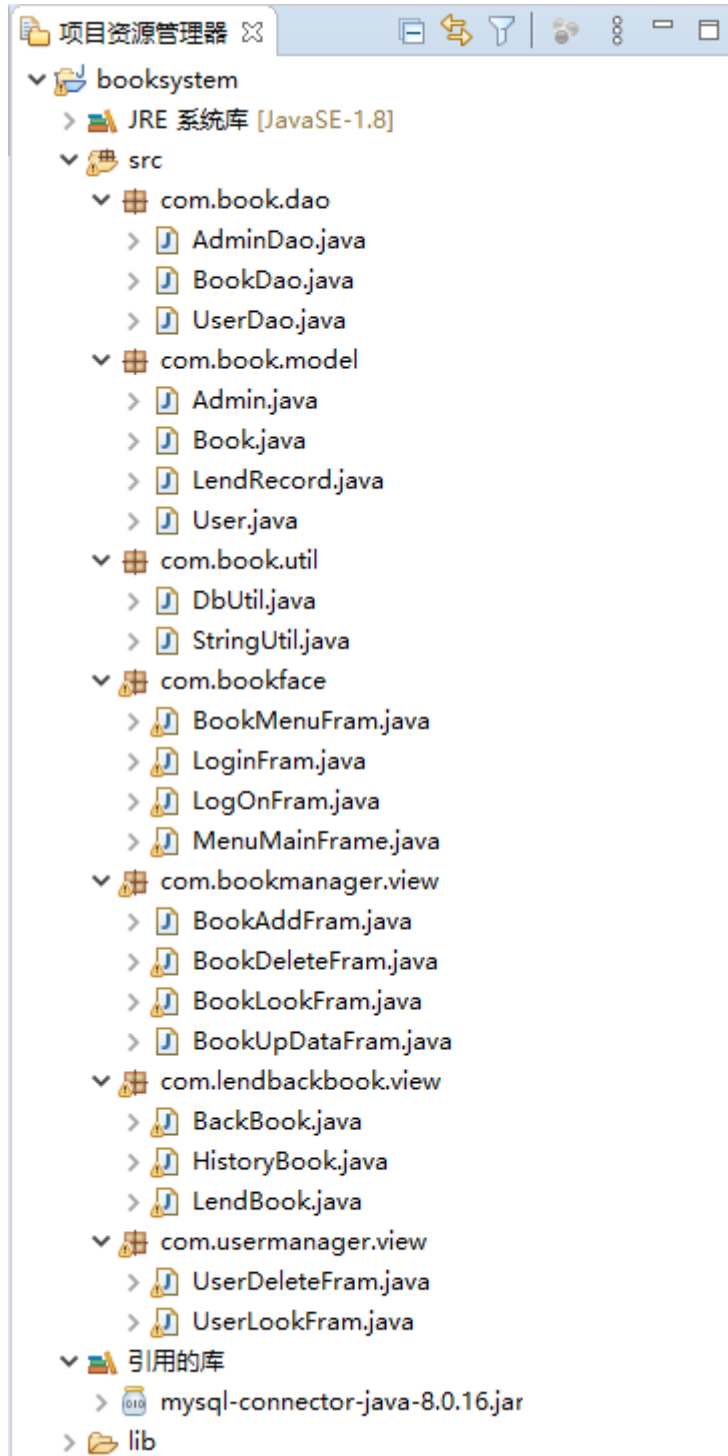
图书 id, 图书名, 图书出版社, 图书作者, 图书状态

(4) 借阅记录表(lendrecord) :

借阅记录 id, 用户 id, 用户名, 图书 id, 图书名, 是否删除

②. 代码设计

包结构:



9. 方案分析比较

除了当前使用的方案，当时我还设想了如下几种方案：

(1) 用类似 C 语言面向过程编程的方法，只编写一个主类，将所有功能写成主类的成员方法，然后在 `main` 函数中按过程顺序调用这些功能以及文件的读取和保存功能。

方案分析：面向过程编程因为符合人类通常的思维，并且因为我们上学期有

C 语言的基础，初学者编程和阅读起来将更加容易。但是将所有方法以及数据写进一个类会使代码看起来很臃肿，并且这样相当于丢弃了 java 面向对象编程的一些很好的特性和思想。对提高自身技术水平的意义不大。

(2) 将每个功能单独编写为一个类，他们继承同一个父类。在主类中根据用户输入选择调用。

方案分析：这样方便定位程序的错误，更易编写和调试。然而定义的类过多也会使程序看起来很奇怪，同时考虑到各功能之间其实并没有太多相同之处，继承的意义不大，故弃用此方案。

10.方案特点

(1) 模块简洁，清晰分明

模块之间通过一个或一组参数相互调用，实现了 java 语言封装的特色。

(2) 系统健壮性良好

程序的每个输入都有异常检查和合理性约束，并且被伪死循环包围。不满足条件或者格式的输入将无法跳出循环以至必须重新输入。这样将不会出现程序无法识别用户输入的情况。

(3) 安全性得以保障

记录类被定义为终结类，类中的变量和数据结点都被 private 修饰符所修饰。只能通过 public 方法来调用，同时类中不提供修改变量的方法，从而确保记录中的数据不会被恶意修改。保障了数据正确性和安全性。但是这里有一个缺陷就是数据可以被恶意删除，系统没有权限控制和操作日志记录功能。

11. 界面设计要求

用户界面采用 Windows 风格，便于用户进行操作。其主要界面，如下图所示。



用户界面设计规则

1. 字体

宋体，正常体，黑色

2. 控件

a). 尺寸

在合理的布局下尽可能多的显示控件内的内容。

b).布局

按照操作流程或浏览顺序自左至右、由上而下的排放各种控件，使界面整体协调、美观大方。

c).自适应父对象的尺寸改变

控件应具有自适应父对象的尺寸改变的能力，当父对象的尺寸发生变化时，控件应能自动改变自己的尺寸并使界面保持整体协调，尽量减少因父对象的尺寸改变而带来的操作或浏览上的不便。

12. 数据库表设计

1、用户表：

用户 id，用户名，用户密码

2、管理员表：

管理员 id，管理员名，管理员密码

3、图书信息表：

图书 id，图书名，图书出版社，图书作者，图书状态

4、借阅记录表：

借阅记录 id，用户 id，用户名，图书 id，图书名，是否归还

1、 用户表(user)

列名	数据类型	是否主键	是否为空	长度	备注
userId	varchar	是	否	20	用户 id
username	varchar	否	否	20	用户名
password	varchar	否	否	20	用户密码

2、管理员表(admin)

列名	数据类型	是否主键	是否为空	长度	备注
adminId	varchar	是	否	20	管理员 id
adminname	varchar	否	否	20	管理员名
password	varchar	否	否	20	管理员密码

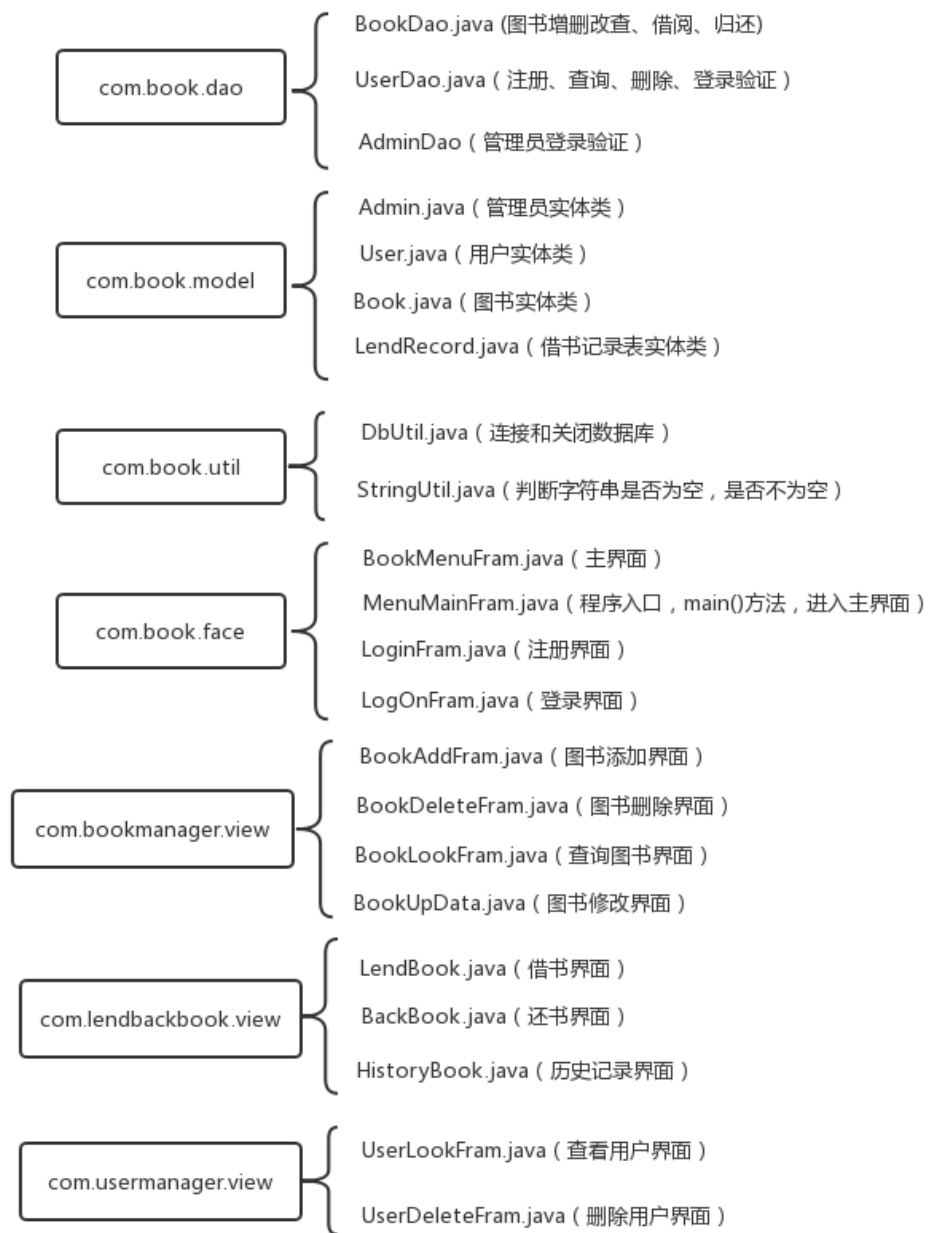
3、图书表(book)

列名	数据类型	是否主键	是否为空	长度	备注
bookId	varchar	是	否	20	图书 id
bookName	varchar	否	否	20	图书名
press	varchar	否	否	20	图书出版社
author	varchar	否	否	20	图书作者
lend	int	否	否	5	图书是否被借出 是为 1 默认为 0

4、加购记录表(lendrecord)

列名	数据类型	是否主键	是否为空	长度	备注
recordId	int	是	否	20	借阅记录 id
userId	varchar	否	否	20	用户 id
userName	varchar	否	否	20	用户名
bookId	varchar	否	否	20	图书 id
bookName	varchar	否	否	20	图书名
back	int	否	否	1	是否归还 是为 1 默认为 0

13 包结构的功能



https://blog.csdn.net/qq_42524288

14. 重要代码分析

a、Dao 类 (以 BookDao 为例)

Dao 类中写的是对数据库的一些基本操作

```
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）

Model 类，主要是 get, set 方法

get 和 set 方法可以自动生成，快捷键：

Shift+Alt+S -> Generate Getters and Setters ->选择你需要的 get, set 参数

```

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 isEmpty(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;
}
}

```

15. 图片实例

登录界面

流动图书馆

身 份

管理员

账 号

123

密 码

...

登录

注册

消息

管理员登录成功

确定

登录界面

流动图书馆

身 份

普通用户

普通用户

管理员

账 号

密 码

登录

注册

归还图书

归还图书

图书帐号

归还

借出图书

图书帐号

添加

图书

图书

图书

图书

图书

借阅历

消息

借书成功

确定

图书删除

图书帐号

删除

消息
删除成功
确定

添加界面

图书编号

图书名字

作者

出版社

提交

消息
添加成功
确定

图书查询

图书号

1

书名

作者

查询

图书id	图书名	出版社	作者	是否已借出
1	数据结构	清华大学...	李四	是

图书查询

图书号

书名

作者

查询

图书id	图书名	出版社	作者	是否已借出
1	数据结构	清华大学...	李四	是
2	java入门	升达	张三	否
3	四级通关	火星教育...	马走日	否
4	六级通关	火星教育...	马走日	否
5	计算机网络	人民教育...	象飞田	否
6	高数上	马飞飞出...	卢本伟	否
7	高数下	马飞飞出...	卢本伟	否

查看用户

用户查询

用 户 号

用 户 名

用户id	用户名
1234	user1

查询

删除用户

用户删除

用 户 号

删除



流动图书馆

欢迎您, admin

登录

退出

图书管理

图书查询

图书添加

图书修改

图书删除

借阅管理

图书借阅

图书归还

借阅历史查询

用户管理

查询用户

删除用户