

Java 程序设计报告

一、小组简介

1512 班 33490

11510352 李子强

11510369 贾彦文

11510372 朱雨萌

二、本小组程序的基本功能

可实现基本自主式图书管理系统的功能

读者部分：读者登陆，查询，查询已借书目，借书，还书，续借，逾期图书处理，修改密码；

管理员部分：读者信息管理，图书信息管理，修改密码；

系统初始化；

三、突出创意及创新

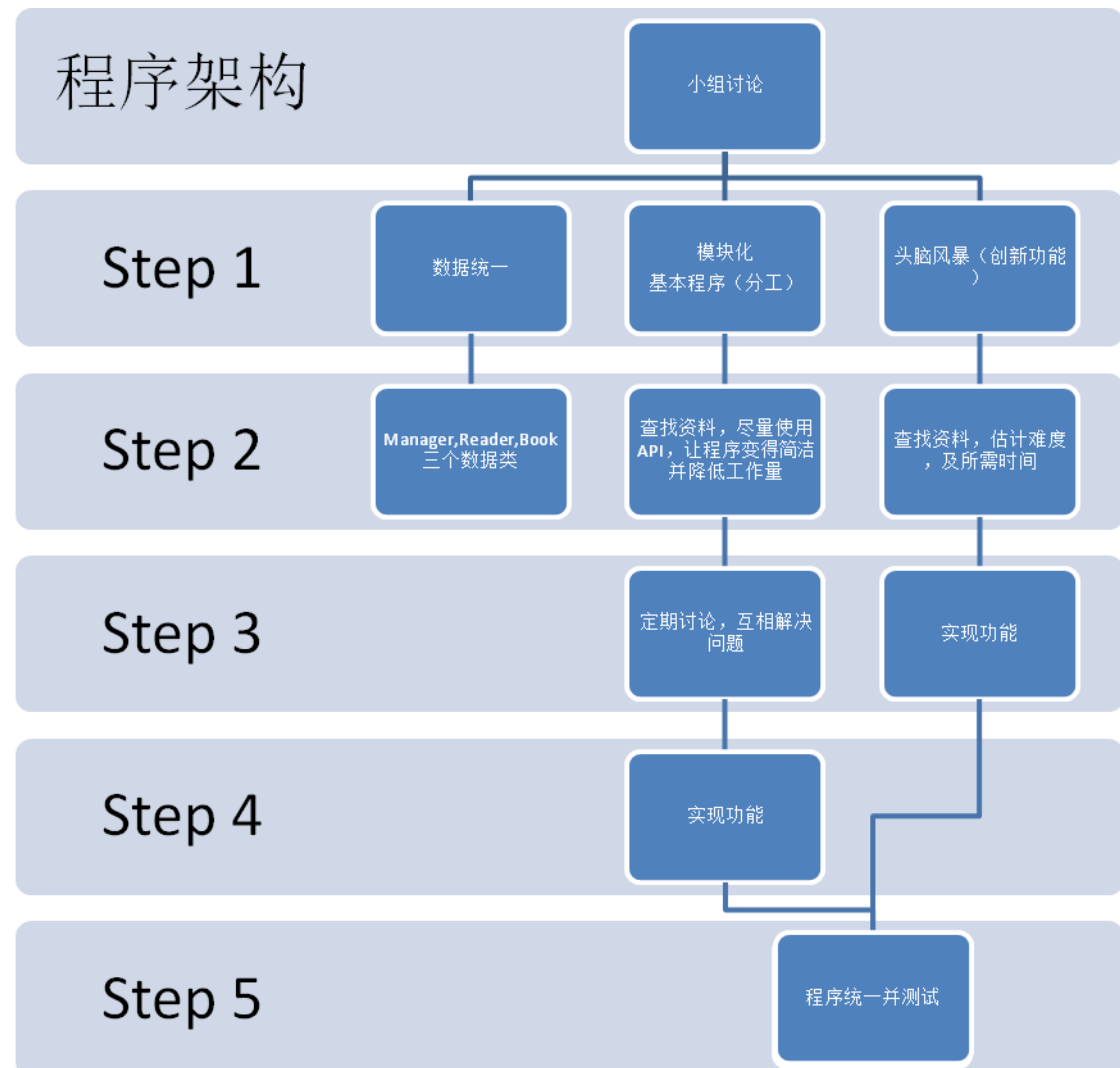
1、将图书、读者、管理员的数据，保存为文件，下次运行可以读取。

2、修改读者、管理员密码，管理员可以找回读者密码

3、支持初始化系统

4、读者、图书列表支持输出 TXT 文件

四、描述程序的基本设计思路



1. Manager 使用一维数组储存（限制大小为 10），Book 和 Reader 使用动态数组储存
 - 3 个数据类都使用 Serializable 接口使数据序列化方便储存
2. 日期使用 java.util.Date 类，此类自带时间前后判断方法，和获取系统当前时间。
 - 续借所需要的日期变换是自建 date2 类，使天数+1，并循环 31 次。
 - 日期格式化使用 java.text.SimpleDateFormat 类的方法。
 - 计算间隔天数使用 java.util.Calendar 类实现。
3. 读者登录的密码在初始化时已经设定，默认为学号后 6 位，不足 6 位则是全部输入
 - 读者密码在读者界面可以找回，也可以忘记密码可通过管理员界面重新设定
 - 新读者账号必须有管理员录入才能使用
4. 管理员密码也可以修改，但忘记了只能系统初始化，全部信息丢失。
5. 所有列表展示是遍历储存数据的数组实现的。
6. 数据及文本输入和读取使用 java.io 类
 - 数据类使用 Serializable 接口后可以使用 ObjectOutputStream 和 FileOutputStream 输出为*.dat 文件，三个数据类，分别输出。里面储存：账号、密码、读者信息、图书信息.....
 - 下次运行时相应的可以读取数据。
 - 文本输出使用 String 类下的 format 方法将字符串格式化，用 StringBuffer 类 append 方法连接字符串，速度提高。
7. 图书按关键字查找使用 String 类下 contains 方法只要包含 CharSequence 类型的值就会返回 true。遍历图书数组即可。
8. 输出系统初始化，内置 3 个读者，5 本书，1 个管理员账号。

五、程序运行截图

1、读者部分

登入展示：

```
=====|| 欢迎使用图书馆管理系统 ||=====
请选择登录类型：
1.读者 2.管理员 3.系统初始化1
请输入读者证号（学号）： 11510369
请输入密码（默认为学号后6位）： 510369
登录成功！
```

查询和借书：

```
=====|| 读者功能 ||=====

请输入功能编号：
1.查询可借阅图书 2.查询已借图书 3.借书 4.还书 5.逾期图书处理 6.续借 7.修改密码 8.退出登录
```

```
1
请输入书名关键词：盗墓
```

```
图书编号 书名      书架编号 在馆数量
10001    盗墓笔记 A01      5
```

```
=====|| 读者功能 ||=====

请输入功能编号：
1.查询可借阅图书 2.查询已借图书 3.借书 4.还书 5.逾期图书处理 6.续借 7.修改密码 8.退出登录
3
```

图书编号	书名	入馆日期	书架编号	读者	图书状态	可续借次数
10001	盗墓笔记	1993-02-04	A01	0	在馆	0
10002	围城	1996-05-04	B02	0	在馆	0
10003	平凡的心	2001-10-03	C03	0	在馆	0
10004	哈利波特	1997-07-14	D04	0	在馆	0
10005	花千骨	2006-12-09	E05	0	在馆	0

```
请输入图书编号：10001
《盗墓笔记》借书完成。还书日期：2016-02-08，还可续借1次。
```

续借和查询：

=====|读者功能|=====

请输入功能编号：
1.查询可借阅图书 2.查询已借图书 3.借书 4.还书 5.逾期图书处理 6.续借 7.修改密码 8.退出登录

6

读者证号：11510369

图书编号	书名	借阅日期	到期日期	可续借次数
10001	盗墓笔记	2016-01-08	2016-02-08	1

请输入图书编号：10001

《盗墓笔记》续借完成,还书日期：2016-03-10

=====|读者功能|=====

请输入功能编号：
1.查询可借阅图书 2.查询已借图书 3.借书 4.还书 5.逾期图书处理 6.续借 7.修改密码 8.退出登录

2

读者证号：11510369

图书编号	书名	借阅日期	到期日期	可续借次数
10001	盗墓笔记	2016-01-08	2016-03-10	0

逾期处理：（将系统日期更改为 3 月 11 日，来测试逾期）

=====|读者功能|=====

请输入功能编号：
1.查询可借阅图书 2.查询已借图书 3.借书 4.还书 5.逾期图书处理 6.续借 7.修改密码 8.退出登录

5

读者证号：11510369

图书编号	书名	借阅日期	到期日期	可续借次数
10001	盗墓笔记	2016-01-08	2016-03-10	0

需要缴纳罚款1元。

是否现在处理？(Y/N) y

处理完成，逾期图书已经归还。

还书：（重新借书）

=====|读者功能|=====

请输入功能编号：
1.查询可借阅图书 2.查询已借图书 3.借书 4.还书 5.逾期图书处理 6.续借 7.修改密码 8.退出登录

4

请输入图书编号：10002

《围城》还书完成。归还日期：2016-01-08。

修改密码：

=====|读者功能|=====

请输入功能编号：
1.查询可借阅图书 2.查询已借图书 3.借书 4.还书 5.逾期图书处理 6.续借 7.修改密码 8.退出登录

7

请输入读者证号（学号）：11510369

请输入新密码：123456

重新登入：

=====||欢迎使用图书馆管理系统||=====

请选择登录类型：
1.读者 2.管理员 3.系统初始化1

请输入读者证号（学号）：11510369

请输入密码（默认为学号后6位）：123456

登录成功！

退出登录：

=====|读者功能|=====

请输入功能编号：
1.查询可借阅图书 2.查询已借图书 3.借书 4.还书 5.逾期图书处理 6.续借 7.修改密码 8.退出登录

8

已退出登录。

2、管理员部分

登入展示：

```
=====|| 欢迎使用图书馆管理系统 ||=====
请选择登录类型：
1.读者  2.管理员 3.系统初始化2
请输入管理员名：admin
请输入密码：123456
登录成功！
```

①读者信息管理

录入新读者：

```
=====|| 管理员功能 ||=====
请输入功能编号：
1.读者信息管理  2.图书信息管理  3.修改密码  4.退出登录1
=====|| 读者信息 ||=====
请输入功能编号：
1.录入新读者  2.修改读者密码  3.所有读者信息列表展示  4.退出 读者信息管理1

姓名:程庆沙

性别:1.男 2.女1

学号:11111007

是否继续输入？(Y/N) n
```

修改读者密码：

```
=====|| 读者信息 ||=====
请输入功能编号：
1.录入新读者  2.修改读者密码  3.所有读者信息列表展示  4.退出 读者信息管理
2
请输入读者证号（学号）：11111007

请输入新密码：007007
```

重新登入：

```
=====|| 欢迎使用图书馆管理系统 ||=====
请选择登录类型：
1.读者  2.管理员 3.系统初始化1
请输入读者证号（学号）：11111007
请输入密码（默认为学号后6位）：007007
登录成功！
```

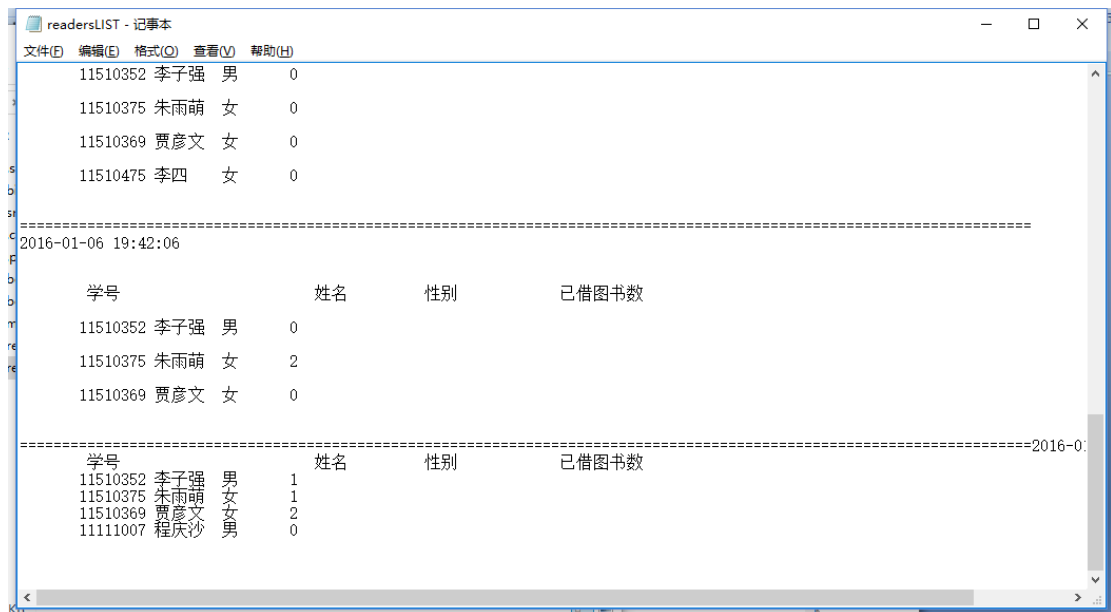
所有读者信息列表展示：

```
=====|| 读者信息 ||=====
请输入功能编号：
1.录入新读者  2.修改读者密码  3.所有读者信息列表展示  4.退出 读者信息管理
3

学号      姓名      性别      已借图书数
11510352  李子强    男        1
11510375  朱雨萌    女        1
11510369  贾彦文    女        2
11111007  程庆沙    男        0

是否需要输出结果列表为txt？(Y/N) y
输出完成。
```

输出的 txt 文档展示：



5

②图书信息管理

新书入馆：

```
=====| 管理员功能 |=====
请输入功能编号：
1. 读者信息管理    2. 图书信息管理    3. 修改密码    4. 退出登录
2

=====| 图书信息管理 |=====
请输入功能编号：
1. 新书入馆    2. 编辑图书信息    3. 所有图书信息列表展示    4. 退出 图书信息管理1

图书编号: 20001

书名: 量子Java

价格: 99

是否继续输入? (Y/N) n
```

编辑图书信息：

```
=====| 图书信息管理 |=====
请输入功能编号：
1. 新书入馆    2. 编辑图书信息    3. 所有图书信息列表展示    4. 退出 图书信息管理2

图书编号    书名    入馆日期    书架编号    读者    图书状态    可续借次数
10001    盗墓笔记    1993-02-04    A01    11510369    借出    1
10002    围城    1996-05-04    B02    11510375    借出    1
10003    平凡的心    2001-10-03    C03    0    在馆    0
10004    哈利波特    1997-07-14    D04    11510369    借出    1
10005    花千骨    2006-12-09    E05    11510352    借出    0
20001    量子Java    1900-01-01    Z01    0    在馆    0

请输入图书编号: 20001

请输入要修改的属性：
1. 图书编号    2. 书名    3. 入馆日期    4. 书架编号    5. 图书状态4

请输入新书架编号: A02

新书架编号设定完成。
是否继续编辑其它图书信息? (Y/N) n
```

所有图书信息列表展示：

```

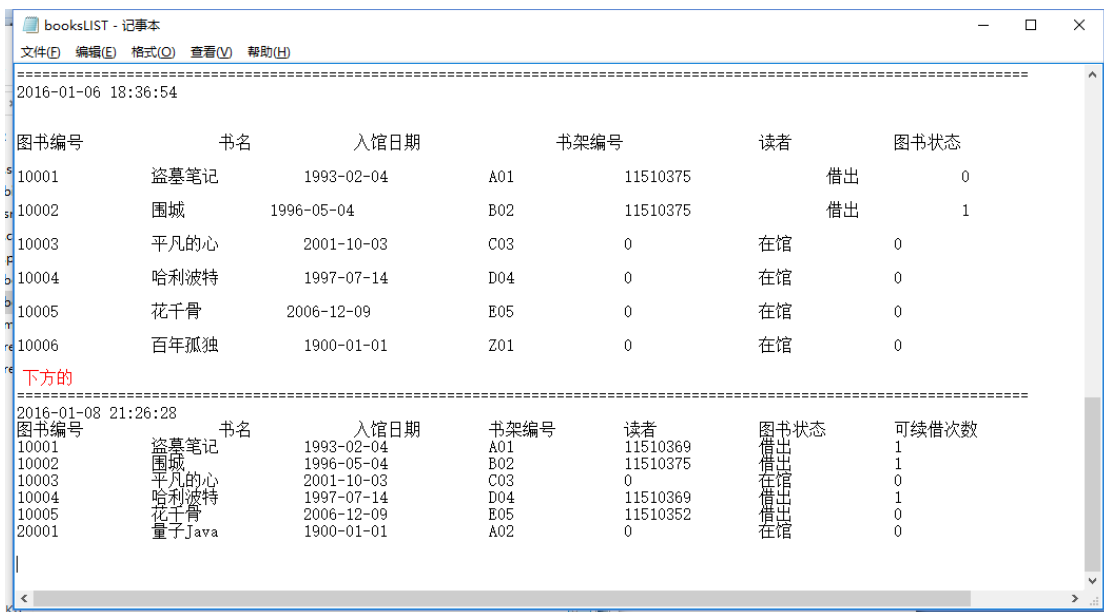
=====| 图书信息管理 |=====
请输入功能编号：
1.新书入馆      2.编辑图书信息      3.所有图书信息列表展示      4.退出 图书信息管理 3

图书编号      书名      入馆日期      书架编号      读者      图书状态      可续借次数
10001      盗墓笔记      1993-02-04      A01      11510369      借出      1
10002      围城      1996-05-04      B02      11510375      借出      1
10003      平凡的心      2001-10-03      C03      0      在馆      0
10004      哈利波特      1997-07-14      D04      11510369      借出      1
10005      花千骨      2006-12-09      E05      11510352      借出      0
20001      量子Java      1900-01-01      A02      0      在馆      0

是否需要输出结果列表为txt? (Y/N) y
输出完成。

```

输出的 txt 文档展示：



③修改密码

```

=====| 管理员功能 |=====
请输入功能编号：
1.读者信息管理      2.图书信息管理      3.修改密码      4.退出登录 3
请输入原密码： 123456
原密码正确！

请输入新密码： 654321
新密码设定成功，请重新登录。

```

重新登入：

```

=====| 欢迎使用图书馆管理系统 |=====
请选择登录类型：
1.读者      2.管理员      3.系统初始化 2
请输入管理员名： admin
请输入密码： 654321
登录成功！

```

④登出界面展示：

```

=====| 管理员功能 |=====
请输入功能编号：
1.读者信息管理      2.图书信息管理      3.修改密码      4.退出登录
4
已退出登录。

```

3、系统初始化

将目录下的两个 txt 文档删除，再使用系统初始化功能。

名称	修改日期	类型	大小
.settings	2016/1/5 21:46	文件夹	
bin	2016/1/5 21:57	文件夹	
src	2016/1/5 21:57	文件夹	
.classpath	2015/12/16 16:15	CLASSPATH 文件	1 KB
.project	2015/12/16 16:15	PROJECT 文件	1 KB
books.dat	2016/1/6 9:48	DAT 文件	1 KB
booksLIST	2016/1/6 9:44	文本文档	3 KB
managers.dat	2016/1/6 9:48	DAT 文件	1 KB
readers.dat	2016/1/6 9:48	DAT 文件	1 KB
readersLIST	2016/1/6 9:38	文本文档	1 KB

初始状态

名称	修改日期	类型	大小
.settings	2016/1/5 21:46	文件夹	
bin	2016/1/5 21:57	文件夹	
src	2016/1/5 21:57	文件夹	

下方为空白，已被删除

*****欢迎使用图书馆管理系统*****

请选择登录类型：

1. 读者 2. 管理员 3. 系统初始化

3

系统已经初始化完成。

名称	修改日期	类型	大小
.settings	2016/1/5 21:46	文件夹	
bin	2016/1/5 21:57	文件夹	
src	2016/1/5 21:57	文件夹	
books.dat	2016/1/6 9:52	DAT 文件	2 KB
managers.dat	2016/1/6 9:52	DAT 文件	1 KB
readers.dat	2016/1/6 9:52	DAT 文件	1 KB

初始化结束后

六、源代码：

1、data 文件夹

①Book

```
package data;
```

```
import java.io.Serializable;
```

```
import java.text.SimpleDateFormat;
```

```
import java.util.Date;
```

```
import data.date2;
```

```
public class Book implements Serializable {
    // 基本信息
    private int bookID;// 图书编号
    private String bookName;// 书名
    private double bookPrice;// 单价
    private Date bookIndate = new Date(0, 0, 1);// 入馆时间
    // 二次编辑信息
    private String bookLoc;// 书架编号
    private int stock;// 在馆数量
    // 借出信息
    private String bookState;// 图书状态：借出 在馆
    private int renewtimes;// 可续借次数
    private int keeper;// ， 目前借出人,0 为没有借出
    private Date Outdate;// 借书日期
    private Date Duedate;// 应还日期
    // 构造函数

    public Book(int bookID, String name, double price) {
        // 新书入馆构造方法
        setBookID(bookID);
        setBookName(name);
        setBookPrice(price);
        setBookState("在馆");// 默认在馆
        setBookLoc("Z01");// 暂存书架
    }

    public Book() {
    }

    public int getBookID() {
        return bookID;
    }

    public void setBookID(int bookID) {
        this.bookID = bookID;
    }

    public String getBookName() {
        return bookName;
    }

    public void setBookName(String bookName) {
        this.bookName = bookName;
    }
}
```



```
}

public double getBookPrice() {
    return bookPrice;
}

public void setBookPrice(double bookPrice) {
    if (bookPrice < 0) {
        this.bookPrice = 0;
    } else {
        this.bookPrice = bookPrice;
    }
}

public String SgetBookIndate() {
    SimpleDateFormat d1 = new SimpleDateFormat("yyyy-MM-dd");
    return d1.format(bookIndate);
}

public Date getBookIndate() {
    return bookIndate;
}

public void setBookIndate(int year, int month, int date) {
    bookIndate = new Date(year - 1900, month - 1, date);
}

public String getBookLoc() {
    return bookLoc;
}

public void setBookLoc(String bookLoc) {
    this.bookLoc = bookLoc;
}

public String getBookState() {
    return bookState;
}

public void setBookState(String bookState) {
    this.bookState = bookState;
}

public int getRenewtimes() {
```

```
        return renewtimes;
    }

    public void setRenewtimes(int renewtimes) {
        this.renewtimes = renewtimes;
    }

    public int getKeeper() {
        return keeper;
    }

    public void setKeeper(int keeper) {
        this.keeper = keeper;
    }

    public String SgetOutdate() {
        SimpleDateFormat d1 = new SimpleDateFormat("yyyy-MM-dd");
        return d1.format(Outdate);
    }

    public Date getOutdate() {
        return Outdate;
    }

    public void setOutdate() {
        Outdate = new Date();
    }

    public String SgetDuedate() {
        SimpleDateFormat d1 = new SimpleDateFormat("yyyy-MM-dd");
        return d1.format(Duedate);
    }

    public Date getDuedate() {
        return Duedate;
    }

    public void setDuedate() {
        Duedate = (Date) getOutdate().clone();
        date2 temp = new date2(Duedate.getYear() + 1900, Duedate.getMonth() + 1,
Duedate.getDate());
        temp.next31Day();
        Duedate.setDate(temp.getDay());
        Duedate.setMonth(temp.getMonth() - 1);
    }
}
```

```

        Duedate.setYear(temp.getYear() - 1900);
    }

    public void duedateRenew() {
        Duedate = (Date) getDuedate().clone();
        date2 temp = new date2(Duedate.getYear() + 1900, Duedate.getMonth() + 1,
Duedate.getDate());
        temp.next31Day();
        Duedate.setDate(temp.getDay());
        Duedate.setMonth(temp.getMonth() - 1);
        Duedate.setYear(temp.getYear() - 1900);
    }

    public int getStock() {
        return stock;
    }

    public void setStock(int stock) {
        this.stock = stock;
    }
}

```

②date2

package data;

```

public class date2 {
    private int month; // 1-12
    private int day; // 1-31 based on month
    private int year; // any year

    private static final int[] daysPerMonth = // days in each month
    { 0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

    // constructor: call checkMonth to confirm proper value for month;
    // call checkDay to confirm proper value for day
    public date2(int theYear, int theMonth, int theDay) {
        month = checkMonth(theMonth); // validate month
        year = checkYear(theYear); // validate year
        day = checkDay(theDay); // validate day
    } // end Date constructor

    public int getMonth() {
        return month;
    }
}

```

```
public int getDay() {
    return day;
}

public int getYear() {
    return year;
}

private int checkYear(int theYear) {
    if (theYear >= 1900 && theYear <= 9999)
        return theYear;
    else
        throw new IllegalArgumentException("Year must be 1900-9999");
}

// utility method to confirm proper month value
private int checkMonth(int testMonth) {
    if (testMonth > 0 && testMonth <= 12) // validate month
        return testMonth;
    else // month is invalid
        throw new IllegalArgumentException("month must be 1-12");
} // end method checkMonth

// utility method to confirm proper day value based on month and year
private int checkDay(int testDay) {
    // check if day in range for month
    if (testDay > 0 && testDay <= daysPerMonth[month])
        return testDay;

    // check for leap year
    if (month == 2 && testDay == 29 && (year % 400 == 0 || (year % 4 == 0 && year %
100 != 0)))
        return testDay;

    throw new IllegalArgumentException("day out-of-range for the specified month and
year");
} // end method checkDay

// return a String of the form month/day/year
public String toString() {
    return String.format("%d/%d/%d", year, month, day);
} // end method toString
```

```
public void nextDay() {
    int tempDay = day + 1;
    if (tempDay > daysPerMonth[month]) {
        if (month == 2 && (year % 400 == 0 || (year % 4 == 0 && year % 100 != 0))) {
            day = tempDay;
        } else {
            int tempMonth = month + 1;
            day = 1;
            if (tempMonth > 12) {
                year += 1;
                month = 1;
            } else
                month = tempMonth;
        }

    } else
        day = tempDay;
}

public void next31Day() {
    for (int i = 1; i <= 31; i++) {
        nextDay();
    }
}

public static boolean NcheckYear(int theYear) {
    boolean flag;
    if (theYear >= 1900 && theYear <= 9999)
        flag = true;
    else
        flag = false;
    return flag;
}

public static boolean NcheckMonth(int testMonth) {
    boolean flag;
    if (testMonth > 0 && testMonth <= 12)
        flag = true;
    else
        flag = false;
    return flag;
}

public static boolean NcheckDay(int theYear, int testMonth, int testDay) {
```

```
        boolean flag = false;
        if (testDay > 0 && testDay <= daysPerMonth[testMonth])
            flag = true;
        else
            flag = false;
        if (testMonth == 2 && testDay == 29 && (theYear % 400 == 0 || (theYear % 4 == 0 &&
theYear % 100 != 0))) {
            flag = true;
        }
        return flag;
    }
}
```

③Manager

```
package data;

import java.io.Serializable;

public class Manager implements Serializable {
    private String userName = "";
    private int usercode;

    public Manager() {
    }

    public Manager(String name, int pwd) {
        setUserName(name);
        setUsercode(pwd);
    }

    public String getUserName() {
        return userName;
    }

    public void setUserName(String userName) {
        this.userName = userName;
    }

    public int getUsercode() {
        return usercode;
    }

    public void setUsercode(int usercode) {
        this.usercode = usercode;
    }
}
```

```

    }
}

```

④Reader

```
package data;
```

```
import java.io.Serializable;
```

```

public class Reader implements Serializable{
    private int ID;//学号&借书卡号
    private String Name;//姓名
    private String Gender;//性别
    private int keepingbooks;//已借图书数，最多借 5 本
    private int[] bookIndex = new int[5];
    private int password;//密码
    //构造函数
    public Reader(){
    public Reader(String name,String gender,int id){
        setName(name);
        setGender(gender);
        setID(id);
        setKeepingbooks(0);
        //默认密码设定
        String code =Integer.toString(id);
        String password;
        if(code.length()>=6){// 判断是否长度大于等于 6
            password =code.substring(code.length()- 6);//一个参数表示截取传递的序号之
后的部分
        }
        else
            password=Integer.toString(id);
        setPassword(Integer.parseInt(password));
    }

    public int getPassword() {
        return password;
    }
    public void setPassword(int password) {
        this.password = password;
    }
    public int[] getBookIndex() {
        return bookIndex;
    }
    public void setBookIndex(int[] bookIndex) {

```

```
        this.bookIndex = bookIndex;
    }
    public int getID() {
        return ID;
    }
    public void setID(int iD) {
        ID = iD;
    }
    public String getName() {
        return Name;
    }
    public void setName(String name) {
        Name = name;
    }
    public String getGender() {
        return Gender;
    }
    public void setGender(String gender) {
        Gender = gender;
    }
    public int getKeepingbooks() {
        return keepingbooks;
    }
    public void setKeepingbooks(int keepingbooks) {
        this.keepingbooks = keepingbooks;
    }
}
```

2、Manager 文件夹

①MyFile

```
package Manager;
import java.io.*;

public class MyFile {

    public static boolean creatTxtFile(String name) throws IOException {
        boolean flag = false;
        File filename = new File(name);
        if (!filename.exists()) {
            filename.createNewFile();
            flag = true;
        }
        return flag;
    }
}
```



```
}

/**
 * 写文件
 *
 * @param newStr
 *      新内容
 * @throws IOException
 */
public static boolean writeTxtFile(String newStr,String name) throws IOException {
    // 先读取原有文件内容，然后进行写入操作
    boolean flag = false;
    String filein = newStr + "\r\n";
    String temp = "";

    FileInputStream fis = null;
    InputStreamReader isr = null;
    BufferedReader br = null;

    FileOutputStream fos = null;
    PrintWriter pw = null;
    try {
        // 文件路径
        File file = new File(name);
        // 将文件读入输入流
        fis = new FileInputStream(file);
        isr = new InputStreamReader(fis);
        br = new BufferedReader(isr);
        StringBuffer buf = new StringBuffer();

        // 保存该文件原有的内容
        for (int j = 1; (temp = br.readLine()) != null; j++) {
            buf = buf.append(temp);
            // System.getProperty("line.separator")
            // 行与行之间的分隔符 相当于"\n"
            buf = buf.append(System.getProperty("line.separator"));
        }
        buf.append(filein);

        fos = new FileOutputStream(file);
        pw = new PrintWriter(fos);
        pw.write(buf.toString().toCharArray());
        pw.flush();
        flag = true;
    }
}
```

```

    } catch (IOException e1) {
        // TODO 自动生成 catch 块
        throw e1;
    } finally {
        if (pw != null) {
            pw.close();
        }
        if (fos != null) {
            fos.close();
        }
        if (br != null) {
            br.close();
        }
        if (isr != null) {
            isr.close();
        }
        if (fis != null) {
            fis.close();
        }
    }
    return flag;
}
}

```

②saveTXT

```

package Manager;

import java.util.ArrayList;
import data.Book;
import data.Reader;

public class saveTXT {
    // 转换所有图书列表为字符串
    public static String books2TXT(ArrayList a1) {
        StringBuffer result = new StringBuffer();
        result.append(String.format("\n%-6s\t%-10s%-15s\t%-8s\t%-8s\t%-8s\t%-8s", "图书编号", "书名", "入馆日期", "书架编号", "读者", "图书状态", "可续借次数"));
        result.append(System.getProperty("line.separator"));
        int counter = -1;
        for (int i = 0; i < a1.size(); i++) {
            result.append(String.format("\n%-6s\t%-10s%-15s\t%-8s%-18s%-16s%-8s", ((Book)

```

```

a1.get(i)).getBookID(),
                ((Book) a1.get(i)).getBookName(), ((Book) a1.get(i)).getBookIndate(),
                ((Book) a1.get(i)).getBookLoc(), ((Book) a1.get(i)).getKeeper(), ((Book)
a1.get(i)).getBookState(),
                ((Book) a1.get(i)).getRenewtimes());
        result.append(System.getProperty("line.separator"));
        counter = i;
    }

    if (counter == -1) {
        result.append(String.format("\n 图书列表目前没有图书。"));
        result.append(System.getProperty("line.separator"));
    }
    return result.toString();
}

// 转换所有读者列表为字符串
public static String readers2TXT(ArrayList a1) {
    StringBuffer result = new StringBuffer();
    result.append(String.format("\n%-19s%-5s%-2s\t%-5s", "学号", "姓名", "性别", "已借
图书数"));
    result.append(System.getProperty("line.separator"));
    int counter = -1;
    for (int i = 0; i < a1.size(); i++) {
        result.append(String.format("\n%-10d%-5s%-2s\t%-5s", ((Reader) a1.get(i)).getID(),
((Reader) a1.get(i)).getName(),
                ((Reader) a1.get(i)).getGender(),
                ((Reader) a1.get(i)).getKeepingbooks()));
        result.append(System.getProperty("line.separator"));
        counter = i;
    }
    if (counter == -1) {
        result.append(String.format("\n 读者列表目前没有读者。"));
        result.append(System.getProperty("line.separator"));
    }
    return result.toString();
}

```

③show

```

package Manager;

import java.util.ArrayList;
import data.Reader;

```

```

import data.Book;

public class show {
    // 展示所有图书
    public static void showbooks(ArrayList a1) {
        System.out.printf("\n%-6s\t%-10s%-15s\t%-8s\t%-8s\t%-8s", "图书编号", "书名",
            "入馆日期", "书架编号", "读者", "图书状态",
            "可续借次数");

        int counter = -1;
        for (int i = 0; i < a1.size(); i++) {
            System.out.printf("\n%-6s\t%-10s%-15s\t%-8s%-18s%-16s%-8s", ((Book)
a1.get(i)).getBookID(),
                ((Book) a1.get(i)).getBookName(), ((Book) a1.get(i)).getBookIndate(),
                ((Book) a1.get(i)).getBookLoc(), ((Book) a1.get(i)).getKeeper(), ((Book)
a1.get(i)).getBookState(),
                ((Book) a1.get(i)).getRenewtimes());
            counter = i;
        }

        if (counter == -1) {
            System.out.print("\n 图书列表目前没有图书。");
        }

    }
    // 展示所有读者
    public static void showreaders(ArrayList a1) {
        System.out.printf("\n%-19s%-5s%-2s\t%-5s", "学号", "姓名", "性别", "已借图书数");
        int counter = -1;
        for (int i = 0; i < a1.size(); i++) {
            System.out.printf("\n%-10d%-5s%-2s\t%-5s", ((Reader) a1.get(i)).getID(), ((Reader)
a1.get(i)).getName(),
                ((Reader) a1.get(i)).getGender(), ((Reader) a1.get(i)).getKeepingbooks());
            counter = i;
        }
        if (counter == -1) {
            System.out.print("\n 读者列表目前没有读者。");
        }
    }
}

```

3、reader 文件夹

①BR

```
package reader;
```

```

import data.Book;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;

public class BR {
    // 借书方法
    public static void borrow(int stuID, int bookID, ArrayList a1) {
        if (stuOverdue(bookID, a1) == true) {
            if (search.bookID2State(bookID, a1).equals("在馆")) {
                ((Book) a1.get(search.bookID2index(bookID, a1))).setBookState("
借出");
                ((Book) a1.get(search.bookID2index(bookID,
a1))).setKeeper(stuID);
                ((Book) a1.get(search.bookID2index(bookID, a1))).setOutdate();
                ((Book) a1.get(search.bookID2index(bookID, a1))).setDuedate();
                ((Book) a1.get(search.bookID2index(bookID,
a1))).setRenewtimes(1);

                System.out.printf("《%s》借书完成。还书日期: %s, 还可续借 1 次。\\n",
                    ((Book) a1.get(search.bookID2index(bookID,
a1))).getBookName(),
                    ((Book) a1.get(search.bookID2index(bookID,
a1))).SgetDuedate());
            } else {
                if (search.bookID2State(bookID, a1).equals("借出")) {
                    System.out.printf("《%s》已被借出。\\n", ((Book)
a1.get(search.bookID2index(bookID, a1))).getBookName());
                } else {
                    System.out.print(search.bookID2State(bookID, a1)+"\\n");
                    // "您输入的这本书不存在。"
                }
            }
        } else
            System.out.print("\\n 您有逾期未还图书, 现在不可借书。");
    }

    // 还书方法
    public static void back(int stuID, int bookID, ArrayList a1) {
        if (search.bookID2State(bookID, a1).equals("在馆")) {

```

```

        System.out.printf(" 《 %s 》 已为在馆状态。 \n", ((Book)
a1.get(search.bookID2index(bookID, a1))).getBookName());
    } else {
        if (search.bookID2State(bookID, a1).equals("借出")) {
            Date tempTime = new Date();
            if (dueCheck(bookID, a1, tempTime) == true) {
                ((Book) a1.get(search.bookID2index(bookID,
a1))).setBookState("在馆");
                ((Book) a1.get(search.bookID2index(bookID,
a1))).setKeeper(0);
                SimpleDateFormat d1 = new SimpleDateFormat("yyyy-MM-dd");
                System.out.printf(" 《%s》 还书完成。归还日期: %s。 \n",
                    ((Book) a1.get(search.bookID2index(bookID,
a1))).getBookName(), d1.format(tempTime));
            } else
                System.out.printf(" 《%s》 已经逾期, 请缴纳罚金后方可还书。 \n",
                    ((Book) a1.get(search.bookID2index(bookID,
a1))).getBookName());
        } else {
            System.out.print(search.bookID2State(bookID, a1)+"\n");
            // "您输入的这本书不存在。"
        }
    }
}

// 根据图书编号检查是否逾期
public static boolean dueCheck(int bookID, ArrayList a1, Date checkTime) {
    boolean flag = ((Book) a1.get(search.bookID2index(bookID,
a1))).getDuedate().after(checkTime);
    // true 是没有逾期
    return flag;
}

/**
 * 计算两个日期之间相差的天数 一天罚款一元
 *
 * @param smdate
 *      较小的时间
 * @param bdate
 *      较大的时间
 * @return 相差天数
 * @throws ParseException
 */
public static int daysBetween(Date smdate, Date bdate) throws ParseException

```

```

{
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    smdate = sdf.parse(sdf.format(smdate));
    bdate = sdf.parse(sdf.format(bdate));
    Calendar cal = Calendar.getInstance();
    cal.setTime(smdate);
    long time1 = cal.getTimeInMillis();
    cal.setTime(bdate);
    long time2 = cal.getTimeInMillis();
    long between_days = (time2 - time1) / (1000 * 3600 * 24);

    return Integer.parseInt(String.valueOf(between_days));
}

// 根据书的持有人检查是否逾期
public static boolean stuOverdue(int stuID, ArrayList a1) {
    Date tempTime = new Date();
    boolean check = true;
    for (int i = 0; i < a1.size(); i++) {
        if (((Book) a1.get(i)).getKeeper() == stuID) {
            boolean flag = dueCheck(((Book) a1.get(i)).getBookID(), a1,
tempTime);
            if (flag == false)
                check = false;
            break;
        }
    }
    return check;
    // true 是该学生没有逾期, false 是逾期。
}

// 续借图书方法
public static void renew(int bookID, ArrayList a1) {
    String state = search.bookID2State(bookID, a1);
    if (state.equals("您输入的这本书不存在。")) {
        System.out.print("\n 您输入的这本书不存在。");
    } else {
        if (state.equals("在馆")) {
            System.out.printf("\n  《 %s 》 未 借 出 。 ", ((Book)
a1.get(search.bookID2index(bookID, a1))).getBookName());
        } else {
            Date tempTime = new Date();
            if (dueCheck(bookID, a1, tempTime) == true) {
                if (((Book) a1.get(search.bookID2index(bookID,

```

```

a1))).getRenewtimes() == 1) {
    ((Book) a1.get(search.bookID2index(bookID,
a1))).setRenewtimes(0);
    ((Book) a1.get(search.bookID2index(bookID,
a1))).duedateRenew();
    System.out.printf("《%s》续借完成, 还书日期: %s\n",
    ((Book) a1.get(search.bookID2index(bookID,
a1))).getBookName(),
    ((Book) a1.get(search.bookID2index(bookID,
a1))).SgetDuedate());
    } else {
        System.out.printf("《%s》续借次数已用完。 \n",
        ((Book) a1.get(search.bookID2index(bookID,
a1))).getBookName());
    }
    } else
        System.out.printf("《%s》已经逾期, 不可续借。 \n",
        ((Book) a1.get(search.bookID2index(bookID,
a1))).getBookName());
    }
}

// 获取罚款方法 (逾期)
public static int fine(int stuID, ArrayList a1) throws ParseException {
    int total = 0;
    if (stuOverdue(stuID, a1) == true) {
        total = 0;
    } else {
        int fine = 0;
        Date tempTime = new Date();
        for (int i = 0; i < a1.size(); i++) {
            if (((Book) a1.get(i)).getKeeper() == stuID || ((Book)
a1.get(i)).getBookState().equals("借出")
            || ((Book) a1.get(i)).getDuedate().before(tempTime))
                fine += daysBetween(((Book) a1.get(i)).getDuedate(),
tempTime);
        }
        total = fine;
    }
    return total;
}
}

```


②

25

```

package reader;

import data.Book;
import data.Manager;
import data.Reader;

import java.util.ArrayList;

public class search {
    // 根据学生编号展示列出所有已借图书
    public static void keeper(int stuID, ArrayList a1) {
        System.out.printf("\n%-10s  %-15s %-8s %-8s %-8s", "图书编号", "书名", "借阅日期",
"到期日期", "可续借次数");
        int counter = -1;
        for (int i = 0; i < a1.size(); i++) {
            if (((Book) a1.get(i)).getKeeper() == stuID) {
                System.out.printf("\n%-10s  %-15s  %-8s  %-8s  %-8s", ((Book)
a1.get(i)).getBookID(),
                                ((Book) a1.get(i)).getBookName(), ((Book) a1.get(i)).SgetOutdate(),
                                ((Book) a1.get(i)).SgetDuedate(), ((Book)
a1.get(i)).getRenewtimes());
                counter = i;
            }
        }
        if (counter == -1) {
            System.out.print("\n 您还没有借书。 \n");
        }
    }

    /*
    * public static void keeping(reader.getID(index)){ System.out.printf(
    * "\n%-10s  %-15s %-8s %-8s", "图书编号", "书名", "借阅日期", "到期日期"); for (Book
    * element:Book.value()){ if(Book.getkeeper()==reader.getID())
    * System.out.print("\n%-10s  %-15s %-8s %-8s", Book.getID(), Book.getName(),
    * Book.getBorrow(), Book.getDue()); } }
    */
    // 根据图书编号确定图书状态
    public static String bookID2State(int bookID, ArrayList a1) {
        int counter = -1;
        String s1;
        for (int i = 0; i < a1.size(); i++) {

```

```

        if (((Book) a1.get(i)).getBookID() == bookID) {
            counter = i;
            break;
        }
    }
    if (counter == -1) {
        s1 = "您输入的这本书不存在。";
    } else {
        s1 = ((Book) a1.get(counter)).getBookState();
    }
    return s1;
}

```

// 根据图书编号查找 ArrayList 的索引号

```

public static int bookID2index(int bookID, ArrayList a1) {
    int counter = -1; // -1 意味着该图书不存在
    for (int i = 0; i < a1.size(); i++) {
        if (((Book) a1.get(i)).getBookID() == bookID) {
            counter = i;
            break;
        }
    }
    return counter;
}

```

// 根据关键字查找可借图书

```

public static void query(CharSequence name, ArrayList a1) {
    System.out.printf("\n%-10s  %-15s %-8s %-8s", "图书编号", "书名", "书架编号", "在
    馆数量");
    int counter = -1;
    for (int i = 0; i < a1.size(); i++) {
        if (((Book) a1.get(i)).getBookName().contains(name)) {
            System.out.printf("\n%-10s      %-15s      %-8s      %-8s", ((Book)
a1.get(i)).getBookID(),
                                ((Book) a1.get(i)).getBookName(), ((Book) a1.get(i)).getBookLoc(),
                                ((Book) a1.get(i)).getStock());
            counter = i;
        }
    }
    if (counter == -1) {
        System.out.print("\n 无查询结果。");
    }
}

```

```
// 根据学号编号查找 ArrayList 的索引号
public static int stuID2index(int stuID, ArrayList a1) {
    int counter = -1;// -1 意味着该学号不存在
    for (int i = 0; i < a1.size(); i++) {
        if (((Reader) a1.get(i)).getID() == stuID) {
            counter = i;
            break;
        }
    }
    return counter;
}

// 根据管理员名称查找 Array 的索引号
public static int admin2index(String name, Manager[] m1) {
    int counter = -1;// -1 意味着该管理员名称不存在
    for (int i = 0; i < m1.length; i++) {
        if (m1[i].getUserName().equals(name)) {
            counter = i;
            break;
        }
    }
    return counter;
}
}
```

③test

```
package reader;

import java.util.ArrayList;

import data.Book;

public class test {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        ArrayList< Book > books = new ArrayList< Book >();
        Book b1= new Book();
        books.add(b1);
        b1.setBookState("在馆");
        System.out.print(b1.getBookState());

    }
}
```

```
}
```

4、load 文件夹(test)

```
package jia_load;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.io.*;
import java.util.*;

import Manager.MyFile;
import Manager.saveTXT;
import Manager.show;
import data.Book;
import data.Manager;
import data.Reader;
import data.date2;
import reader.search;

public class user {
    // 将 Book 指定的对象写入至指定的文件
    public static void writeObjectsToFile(ArrayList<Book> books, String filename) {
        File file = new File(filename); // 创建，并初始化 File 对象 file
        try {
            // 创建，并初始化 ObjectOutputStream 对象 objOutputStream
            ObjectOutputStream objOutputStream = new ObjectOutputStream(new
FileOutputStream(file));
            for (Object obj : books) {
                // 将对象写入文件
                objOutputStream.writeObject(obj);
            }
            // 关闭流 objOutputStream
            objOutputStream.close();
        } catch (IOException e) { // 捕获异常 IOException
            e.printStackTrace(); // 异常信息输出
        }
    }

    // 将 Book 指定文件中的对象数据读回
    public static ArrayList<Book> readObjectsFromFile(String filename) throws
FileNotFoundException {
        File file = new File(filename);
```

```

// 如果文件不存在就丢出异常
if (!file.exists())
    throw new FileNotFoundException();
// 使用 List 先存储读回的对象
ArrayList<Book> list = new ArrayList<Book>();
try {
    FileInputStream fileInputStream = new FileInputStream(file); // 创建，并初始化
FileInputStream
    // 创建，并初始化 ObjectInputStream 对象 objInputStream
    ObjectInputStream objInputStream = new ObjectInputStream(fileInputStream);
    while (fileInputStream.available() > 0) {
        list.add((Book) objInputStream.readObject()); // 读取一个对象，并添加与列
表中
    }
    objInputStream.close(); // 关闭 objInputStream
} catch (ClassNotFoundException e) { // 捕获 ClassNotFoundException 异常
    e.printStackTrace(); // 异常信息输出
} catch (IOException e) { // 捕获 IOException 异常
    e.printStackTrace(); // 异常信息输出
}
return list;
}

// 将 Reader 指定的对象写入至指定的文件
public static void RwriteObjectsToFile(ArrayList<Reader> readers, String filename) {
    File file = new File(filename); // 创建，并初始化 File 对象 file
    try {
        // 创建，并初始化 ObjectOutputStream 对象 objOutputStream
        ObjectOutputStream objOutputStream = new ObjectOutputStream(new
FileOutputStream(file));
        for (Object obj : readers) {
            // 将对象写入文件
            objOutputStream.writeObject(obj);
        }
        // 关闭流 objOutputStream
        objOutputStream.close();
    } catch (IOException e) { // 捕获异常 IOException
        e.printStackTrace(); // 异常信息输出
    }
}

// 将 Reader 指定文件中的对象数据读回
public static ArrayList<Reader> RreadObjectsFromFile(String filename) throws
FileNotFoundException {

```

```

        File file = new File(filename);
        // 如果文件不存在就丢出异常
        if (!file.exists())
            throw new FileNotFoundException();
        // 使用 List 先存储读回的对象
        ArrayList<Reader> list = new ArrayList<Reader>();
        try {
            FileInputStream fileInputStream = new FileInputStream(file); // 创建，并初始化
FileInputStream
            // 创建，并初始化 ObjectInputStream 对象 objInputStream
            ObjectInputStream objInputStream = new ObjectInputStream(fileInputStream);
            while (fileInputStream.available() > 0) {
                list.add((Reader) objInputStream.readObject()); // 读取一个对象，并添加与
列表中
            }
            objInputStream.close(); // 关闭 objInputStream
        } catch (ClassNotFoundException e) { // 捕获 ClassNotFoundException 异常
            e.printStackTrace(); // 异常信息输出
        } catch (IOException e) { // 捕获 IOException 异常
            e.printStackTrace(); // 异常信息输出
        }
        return list;
    }

    // 将 Manager 指定的对象写入至指定的文件
    public static void MwriteObjectsToFile(Object[] objs, String filename) {
        File file = new File(filename); // 创建，并初始化 File 对象 file
        try {
            // 创建，并初始化 ObjectOutputStream 对象 objOutputStream
            ObjectOutputStream objOutputStream = new ObjectOutputStream(new
FileOutputStream(file));
            for (Object obj : objs) {
                // 将对象写入文件
                objOutputStream.writeObject(obj);
            }
            // 关闭流 objOutputStream
            objOutputStream.close();
        } catch (IOException e) { // 捕获异常 IOException
            e.printStackTrace(); // 异常信息输出
        }
    }

    // 将 Manager 指定文件中的对象数据读回
    public static Manager[] MreadObjectsFromFile(String filename) throws

```

```

FileNotFoundException {
    File file = new File(filename);
    // 如果文件不存在就丢出异常
    if (!file.exists())
        throw new FileNotFoundException();
    // 使用 List 先存储读回的对象
    List<Manager> list = new ArrayList<Manager>();
    try {
        FileInputStream fileInputStream = new FileInputStream(file); // 创建，并初始化
        // 创建，并初始化 ObjectInputStream 对象 objInputStream
        ObjectInputStream objInputStream = new ObjectInputStream(fileInputStream);
        while (fileInputStream.available() > 0) {
            list.add((Manager) objInputStream.readObject()); // 读取一个对象，并添加
        }
        objInputStream.close(); // 关闭 objInputStream
    } catch (ClassNotFoundException e) { // 捕获 ClassNotFoundException 异常
        e.printStackTrace(); // 异常信息输出
    } catch (IOException e) { // 捕获 IOException 异常
        e.printStackTrace(); // 异常信息输出
    }
    Manager[] managers = new Manager[list.size()]; // 创建，并初始化 Manager 数组
    return list.toArray(managers);
}

// 主函数
public static void main(String[] args) throws IOException, ClassNotFoundException {
    String Bfile = "books.dat";
    String Rfile = "readers.dat";
    String Mfile = "managers.dat";
    ArrayList<Book> books = new ArrayList<Book>();
    ArrayList<Reader> readers = new ArrayList<Reader>();
    Manager[] managers = new Manager[10];
    boolean flag; // 判断变量
    int stuID; // 学号
    int choice = 0; // 用户选择
    // 载入初始数据

    Scanner input = new Scanner(System.in);

    do {
        do {
            System.out.println("***** 欢迎使用图书馆管理系统

```

```
*****");
    System.out.println("请选择登录类型:");
    System.out.print("1.读者\t2.管理员\t3.系统初始化"); // 验证使用者是否
    是用户还是管理员，当然这里是用户还是管理员必须选一个
    choice = input.nextInt();
    if (choice != 1 && choice != 2 && choice != 3) {
        System.out.println("请重新选择读者或管理员！");
    } else {
        break;
    }
} while (choice != 1 && choice != 2 && choice != 3);

switch (choice) {
// 初始化
case 3:
    // books 文件初始化
    Book book1 = new Book(10001, "盗墓笔记", 56.6); // 构造对象 book 来存储
    图书的所有属性
    book1.setBookIndate(1993, 2, 4);
    book1.setStock(5);
    book1.setBookLoc("A01");
    books.add(book1);

    Book book2 = new Book(10002, "围城", 45.8);
    book2.setBookIndate(1996, 5, 4);
    book2.setStock(5);
    book2.setBookLoc("B02");
    books.add(book2);

    Book book3 = new Book(10003, "平凡的心", 45.0);
    book3.setBookIndate(2001, 10, 3);
    book3.setStock(5);
    book3.setBookLoc("C03");
    books.add(book3);

    Book book4 = new Book(10004, "哈利波特", 89.0);
    book4.setBookIndate(1997, 7, 14);
    book4.setStock(5);
    book4.setBookLoc("D04");
    books.add(book4);

    Book book5 = new Book(10005, "花千骨", 50.5);
    book5.setBookIndate(2006, 12, 9);
    book5.setStock(5);
```



```

book5.setBookLoc("E05");
books.add(book5);
writeObjectsToFile(books, Bfile);

// readers 文件初始化
Reader stu1 = new Reader("李子强", "男", 11510352);
readers.add(stu1);
Reader stu2 = new Reader("朱雨萌", "女", 11510375);
readers.add(stu2);
Reader stu3 = new Reader("贾彦文", "女", 11510369);
readers.add(stu3);
RwriteObjectsToFile(readers, Rfile);

// managers 文件初始化
managers[0] = new Manager("admin", 123456);
for (int i = 1; i < managers.length; i++) {
    managers[i] = new Manager();
}
MwriteObjectsToFile(managers, Mfile);
System.out.print("系统已经初始化完成。 \n");
break;
// 读者登陆
case 1:
    books = readObjectsFromFile(Bfile);
    readers = RreadObjectsFromFile(Rfile);
    flag = false;
    do {
        System.out.print("请输入读者证号（学号）: ");
        stuID = input.nextInt();
        if (search.stuID2index(stuID, readers) == -1) {
            System.out.print("您输入的学号不存在，请重新输入！ \n");
            continue;
        }
        System.out.print("请输入密码（默认为学号后 6 位）: ");
        int pwd = input.nextInt();
        flag = (readers.get(search.stuID2index(stuID, readers)).getPassword() ==
pwd) ? true : false;
        if (flag) {
            System.out.println("登录成功！");
            break;
        } else {
            System.out.print("密码错误，请重新输入！ \n"); // 用循环保证用
户输入正确的用户名和密码
        }
    }

```

```

    } while (flag == false);
    int function1;
    do { // 读者功能循环
        function1 = -1;
        do {
            System.out.print("***** 读 者 功 能
*****");
            System.out.printf("\n  请 输 入 功 能 编 号  :
\n%-10s %-10s %-10s %-10s \n%-10s %-10s %-10s %-10s\n",
                "1.查询可借阅图书", "2.查询已借图书", "3.借书", "4.还
书", "5.逾期图书处理", "6.续借", "7.修改密码", "8.退出登录");
            function1 = input.nextInt();
            flag = (function1 >= 1 && function1 <= 8) ? true : false;
            if (flag == true) {
            } else
                System.out.print("您输入的功能不存在，请重新输入\n");
        } while (flag == false);
        switch (function1) {
        case 1: // 查询可借阅图书
            System.out.print("请输入书名关键词: ");
            CharSequence name1 = input.next();
            reader.search.query(name1, books);
            break;
        case 2: // 查询已借图书
            System.out.printf("读者证号: %d", stuID);
            reader.search.keeper(stuID, books);
            System.out.println();// 换行
            break;
        case 3: // 借书
            System.out.print("请输入图书编号: ");
            int LbookID = input.nextInt();
            reader.BR.borrow(stuID, LbookID, books);
            writeObjectsToFile(books, Bfile);
            break;
        case 4: // 还书
            System.out.print("请输入图书编号: ");
            int RbookID = input.nextInt();
            reader.BR.back(stuID, RbookID, books);
            writeObjectsToFile(books, Bfile);
            break;
        case 5: // 逾期图书处理
            System.out.printf("读者证号: %d", stuID);
            reader.search.keeper(stuID, books);
            try {

```

```

        if (reader.BR.fine(stuID, books) == 0) {
            System.out.print("\n 您没有逾期图书,不需要缴纳罚款。
");
        } else {
            System.out.printf("\n 需要缴纳罚款 %d 元。 ",
reader.BR.fine(stuID, books));

            System.out.print("\n 是否现在处理? (Y/N)");
            char choice5 = 'N';
            choice5 = input.next().charAt(0);
            Date tempTime = new Date();
            if
(String.valueOf(choice5).equalsIgnoreCase(String.valueOf("Y"))) {
                for (int i = 0; i < books.size(); i++) {
                    if (((Book) books.get(i)).getKeeper() == stuID
                        || ((Book)
books.get(i)).getBookState().equals("借出")
                        || ((Book)
books.get(i)).getDuedate().before(tempTime)) {
                        ((Book) books.get(i)).setKeeper(0);
                        ((Book) books.get(i)).setBookState("在馆
");
                    }
                }
                System.out.print("\n 处理完成,逾期图书已经归还。
");
            } else
                System.out.print("\n 暂时不处理。");
        }
    } catch (ParseException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    writeObjectsToFile(books, Bfile);
    break;
case 6:// 续借
    System.out.printf("读者证号: %d", stuID);
    reader.search.keeper(stuID, books);
    System.out.print("\n 请输入图书编号: ");
    int bookID6 = input.nextInt();
    reader.BR.renew(bookID6, books);
    writeObjectsToFile(books, Bfile);
    break;
case 7:// 修改密码
    do {

```

```

        System.out.print("请输入读者证号（学号）：");
        stuID = input.nextInt();
        if (search.stuID2index(stuID, readers) == -1) {
            System.out.print("您输入的学号不存在，请重新输入！\n");
        }
    } while (search.stuID2index(stuID, readers) == -1);
    System.out.print("\n 请输入新密码：");
    int pwd = input.nextInt();
    readers.get(search.stuID2index(stuID, readers)).setPassword(pwd);
    RwriteObjectsToFile(readers, Rfile);
    break;
case 8:// 退出登录
    writeObjectsToFile(books, Bfile);
    System.out.print("\n 已退出登录。");
    break;
}
} while (function1 != 8);
break;// 读者函数部分退出

// 管理员登陆
case 2:
    books = readObjectsFromFile(Bfile);
    readers = RreadObjectsFromFile(Rfile);
    managers = MreadObjectsFromFile(Mfile);
    String userName = "";// 当前用户名
    flag = false;
    do {
        System.out.print("请输入管理员名：");
        String name = input.next();
        if (search.admin2index(name, managers) == -1) {
            System.out.print("您输入的管理员名不存在，请重新输入！\n");
            continue;
        }
        System.out.print("请输入密码：");
        int pwd = input.nextInt();
        flag = (managers[search.admin2index(name, managers)].getUsercode()
== pwd) ? true : false;
    } while (!flag);
    if (flag) {
        System.out.println("登录成功！");
        userName = name;
        break;
    }
    System.out.print("密码错误，请重新输入!\n"); // 用循环保证用户输

```

入正确的用户名和密码

```

    } while (flag == false);

    int function2 = -1;
    do { // 管理员功能循环
        do {
            System.out.print("***** 管 理 员 功 能
*****");
            System.out.printf("\n  请 输 入 功 能 编 号 :
\n%-10s %-10s %-10s %-10s", "1.读者信息管理", "2.图书信息管理", "3.修改密码",
                                "4.退出登录");
            function2 = input.nextInt();
            flag = (function2 >= 1 && function2 <= 4) ? true : false;
            if (flag == true) {
            } else
                System.out.print("您输入的功能不存在，请重新输入\n");
        } while (flag == false);
        switch (function2) {
        case 1: // 读者信息管理
            int function21 = -1;
            do {
                do {
                    System.out.print("*****读者信息管
理*****");
                    System.out.printf("\n  请 输 入 功 能 编 号 :
\n%-10s %-10s %-10s %-10s", "1.录入新读者", "2.修改读者密码",
                                    "3.所有读者信息列表展示", "4.退出 读者信
息管理");

                    function21 = input.nextInt();
                    flag = (function21 >= 1 && function21 <= 4) ? true : false;
                    if (flag == true) {
                    } else
                        System.out.print("您输入的功能不存在，请重新输
入\n");

                } while (flag == false);
                switch (function21) {
                case 1: // 录入新读者
                    char choice11;
                    do {
                        System.out.print("\n 姓名:");
                        String name = input.next();
                        System.out.print("\n 性别:1.男 2.女");
                        String gender;
                        int choice11g;

```

```

do {
    choice11g = input.nextInt();
    if (choice11g != 1 && choice11g != 2)
        System.out.print("您输入的选项不存在，
请重新输入\n");

    } while (choice11g != 1 && choice11g != 2);
    if (choice11g == 1)
        gender = "男";
    else
        gender = "女";
    System.out.print("\n 学号:");
    int ID = input.nextInt();
    Reader readertemp = new Reader(name, gender, ID);
    readers.add(readertemp);
    System.out.print("\n 是否继续输入？（Y/N）");
    choice11 = 'N';
    choice11 = input.next().charAt(0);
} while
(String.valueOf(choice11).equalsIgnoreCase(String.valueOf("Y")));
RwriteObjectsToFile(readers, Rfile);
break;
case 2:// 修改读者密码
do {
    System.out.print("请输入读者证号（学号）：");
    stuID = input.nextInt();
    if (search.stuID2index(stuID, readers) == -1) {
        System.out.print("您输入的学号不存在，请重
新输入！ \n");
    }
} while (search.stuID2index(stuID, readers) == -1);
System.out.print("\n 请输入新密码：");
int pwd = input.nextInt();
readers.get(search.stuID2index(stuID,
readers)).setPassword(pwd);

RwriteObjectsToFile(readers, Rfile);
break;
case 3:// 所有读者信息列表展示
    // 读取文件数据
    readers = RreadObjectsFromFile(Rfile);
    // 读取的对象
    show.showreaders(readers);
    System.out.println();// 换行

    char choice212;

```

```

        do {
            System.out.print("\n 是否需要输出结果列表为
txt? (Y/N) ");

            choice212 = 'N';
            choice212 = input.next().charAt(0);

        } while
(String.valueOf(choice212).equalsIgnoreCase(String.valueOf("N"))&&!String.valueOf(choice212).e
qualsIgnoreCase(String.valueOf("Y")));

        if(String.valueOf(choice212).equalsIgnoreCase(String.valueOf("Y"))){
            Date now =new Date();
            SimpleDateFormat d1 = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
            StringBuffer result = new StringBuffer();

            result.append("=====
=====\\n");

            result.append(d1.format(now)+"\\n");
            result.append(System.getProperty("line.separator"));
            result.append(saveTXT.readers2TXT(readers)+"\\n");
            String readerTXT="readersLIST.txt";
            MyFile.creatTxtFile(readerTXT);
            MyFile.writeTxtFile(result.toString(),readerTXT);
            System.out.print("输出完成。\\n");
        }
        break;
    case 4:// 退出读者信息管理
        RwriteObjectsToFile(readers, Rfile);
        System.out.print("已退出读者信息管理。\\n");
        break;
    }
    } while (function21 != 4);
    break;
    case 2:// 图书信息管理
        int function22 = -1;
        do {
            do {
                System.out.print("***** 图书信息管
理*****");

                System.out.printf("\\n 请 输 入 功 能 编 号 :
\\n%-10s %-10s %-10s %-10s", "1.新书入馆", "2.编辑图书信息",
"3.所有图书信息列表展示", "4.退出 图书信
息管理");

```

```

        function22 = input.nextInt();
        flag = (function22 >= 1 && function22 <= 4) ? true : false;
        if (flag == true) {
        } else
            System.out.print("您输入的功能不存在，请重新输入\n");

    } while (flag == false);
    switch (function22) {
    case 1:// 新书入馆
        char choice21;
        do {
            System.out.print("\n 图书编号:");
            int id = input.nextInt();
            System.out.print("\n 书名:");
            String name = input.next();
            System.out.print("\n 价格:");
            double price = input.nextDouble();
            Book booktemp = new Book(id, name, price);
            books.add(booktemp);
            System.out.print("\n 是否继续输入？（Y/N）");
            choice21 = 'N';
            choice21 = input.next().charAt(0);
        } while
        (String.valueOf(choice21).equalsIgnoreCase(String.valueOf("Y")));
        writeObjectsToFile(books, Bfile);

        break;
    case 2:// 编辑图书信息
        books = readObjectsFromFile(Bfile);// 读取文件数据
        char choice22;
        do {
            show.showbooks(books);// 读取的对象
            System.out.println();// 换行

            int index = -1;// 图书索引号
            do { // 获取图书编号
                System.out.print("\n 请输入图书编号: ");
                int bookID = input.nextInt();
                index = search.bookID2index(bookID, books);
                if (index == -1)
                    System.out.print("您输入的图书编号不存在，请重新输入。 \n");
            } while (index == -1);

```



```

        int edit;
        do { // 获取修改的属性
            System.out.print("\n 请输入要修改的属性: ");

            System.out.printf("\n%-6s\t%-10s%-15s\t%-8s\t%-8s", "1.图书编号", "2.书名", "3.入馆日期",
                "4.书架编号", "5.图书状态");

            edit = input.nextInt();
            flag = (edit >= 1 && edit <= 5) ? true : false;
            if (flag == true) {
            } else
                System.out.print("您输入的要修改属性不存在，请重新输入\n");

        } while (flag == false);
        switch (edit) {
        case 1: // 图书编号
            System.out.print("\n 请输入新图书编号: ");
            int id = input.nextInt();
            books.get(index).setBookID(id);
            System.out.print("\n 新图书编号设定完成。");
            break;
        case 2: // 书名
            System.out.print("\n 请输入新书名: ");
            String name = input.next();
            books.get(index).setBookName(name);
            System.out.print("\n 新书名设定完成。");
            break;
        case 3: // 入馆日期
            flag=false;
            do{
                System.out.print("\n 请输入新入馆日期（YYYY
MM DD，年月日用空格分开）: ");

                int year=input.nextInt();
                int month=input.nextInt();
                int day=input.nextInt();

                flag=(date2.NcheckYear(year)&&date2.NcheckMonth(month));
                if(flag==false){
                    System.out.print("您输入的日期有误，请重新输入\n");
                }
            } else
            {
                if(date2.NcheckDay(year, month,

```

```

day)==false)

        {
            System.out.print(" 您输入的日期有
误，请重新输入\n");

            flag=false;
        }
        else

books.get(index).setBookIndate(year,month,day);
    }
    }while(flag==false);
    System.out.print("\n 新入馆日期设定完成。");

    break;
case 4:// 书架编号
    System.out.print("\n 请输入新书架编号： ");
    String bookLoc = input.next();
    books.get(index).setBookLoc(bookLoc);
    System.out.print("\n 新书架编号设定完成。");
    break;
case 5:// 图书状态
    int function225 = -1;
    do {
        System.out.print("\n 请选择图书状态： 1.
在馆\t2.借出");

        function225 = input.nextInt();
        flag = (function225 >= 1 && function225
<= 2) ? true : false;

        if (flag == true) {
        } else
            System.out.print("您输入的图书状态
不存在，请重新输入\n");

    } while (flag == false);
    if (function225 == 1)
        books.get(index).setBookState("在馆");
    else
        books.get(index).setBookState("借出");
    break;
}

System.out.print("\n 是否继续编辑其它图书信息？
(Y/N) ");

choice22 = 'N';
choice22 = input.next().charAt(0);

```

```

        }
        while
        (String.valueOf(choice22).equalsIgnoreCase(String.valueOf("Y")));
        writeObjectsToFile(books, Bfile);

        break;
    case 3:// 所有图书信息列表展示
        // 读取文件数据
        books = readObjectsFromFile(Bfile);
        // 读取的对象
        show.showbooks(books);
        System.out.println();// 换行

        char choice223;
        do {
            System.out.print("\n 是否需要输出结果列表为
txt? (Y/N) ");

            choice223 = 'N';
            choice223 = input.next().charAt(0);
        }
        while
        (String.valueOf(choice223).equalsIgnoreCase(String.valueOf("N"))&&!String.valueOf(choice223).e
qualsIgnoreCase(String.valueOf("Y")));

        if(String.valueOf(choice223).equalsIgnoreCase(String.valueOf("Y"))){
            Date now =new Date();
            SimpleDateFormat d1 = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
            StringBuffer result = new StringBuffer();

            result.append("=====
=====\\n");

            result.append(d1.format(now)+"\\n");
            result.append(System.getProperty("line.separator"));
            result.append(saveTXT.books2TXT(books)+"\\n");
            String bookTXT="booksLIST.txt";
            MyFile.creatTxtFile(bookTXT);
            MyFile.writeTxtFile(result.toString(),bookTXT);
            System.out.print("输出完成。\\n");
        }

        break;
    case 4:// 退出图书信息管理
        System.out.print("已退出图书信息管理。\\n");
        break;

```

```

        }
    } while (function22 != 4);
    break;
case 3:// 修改密码
    do {
        System.out.print("请输入原密码: ");
        int rawPwd = input.nextInt();
        flag = (managers[search.admin2index(userName,
managers)].getUsercode() == rawPwd) ? true
            : false;
        if (flag) {
            System.out.println("原密码正确!");
            break;
        }
        System.out.print("密码错误, 请重新输入!\n");
    } while (flag == false);

    System.out.print("\n 请输入新密码: ");
    int newPwd = input.nextInt();
    managers[search.admin2index(userName,
managers)].setUserCode(newPwd);
    System.out.print("新密码设定成功, 请重新登录.\n");
    MwriteObjectsToFile(managers, Mfile);
    break;
case 4:// 退出管理员登录
    writeObjectsToFile(books, Bfile);
    RwriteObjectsToFile(readers, Rfile);
    System.out.print("已退出登录.\n");
    break;
    }
} while (function2 != 4);

    break;// 管理员函数部分退出
    }
} while (choice != 1 && choice != 2);
}
}

```

六、每位组员完成的列表

1、11510352 李子强

Student Name	李子强		
Student ID	11510352		
Task Name	Start	End	Duration (days)
数据	2015.12.19	2015.12.20	1
人员	2015.12.19	2015.12.20	1
借书	2015.12.20	2015.12.23	3
还书	2015.12.20	2015.12.23	3
续借	2015.12.23	2015.12.24	1
逾期图书处理	2015.12.24	2015.12.26	2
关键字查找	2015.12.26	2015.12.28	2
查询已借图书	2015.12.28	2015.12.29	1
读者功能统一	2015.12.29	2016.1.3	5
管理员功能统一	2015.12.30	2016.1.3	4
登陆界面整合	2015.12.31	2016.1.3	3
test1	2015.12.26	2015.12.26	1
test2	2015.12.30	2015.12.30	1
test3	2016.1.5	2016.1.5	1

2、11510369 贾彦文

Student Name	贾彦文		
Student ID	11510369		
Task Name	Start	End	Duration (days)
数据部分	2015/12/16	2015/12/20	4
人员	2015/12/18	2015/12/19	1
系统登陆部分	2015/12/19	2015/12/23	4
test1	2015/12/26	2015/12/26	1
数据部分更改	2015/12/26	2015/12/30	4
test2	2015/12/30	2015/12/30	1
登陆系统更改	2015/12/31	2016/1/5	5
test3	2016/1/5	2016/1/5	1

3、11510369 朱雨萌

Student Name	朱雨萌		
Student ID	11510375		
Task Name	Start	End	Duration (days)
图书数据	2015/12/12	2015/12/21	9
录入新图书	2015/12/22	2015/12/24	2
读者数据	2015/12/26	2015/12/28	2
录入新读者	2015/12/27	2015/12/29	2
编辑读者信息	2015/12/27	2015/12/29	2
编辑图书信息	2015/12/26	2015/12/30	4
图书信息列表显示	2016/1/1	2016/1/5	4
读者信息列表显示	2016/1/1	2016/1/5	4