**Java程序设计**

**大作业报告**

**题目：**

**学院：**

**姓名：**

**学号：**

**专业班级：**

**任课教师：**

**截止时间：**

**大作业要求：**

请根据本学期所学的Java和面向对象的思想实现一个完整的系统。开发环境和工具不限，大作业**题目和内容自定**，代码总行数不低于800行（含注释）。

具体要求如下：

1. 采用**面向对象程序设计，**要运用继承、多态等机制；
2. 代码规范需遵循Java通用编码规范要求；
3. 具备**文件读写**功能，程序中的数据要以文件方式保存；
4. 程序中要求具备**异常处理**；
5. 使用第三方类库或者插件需在报告和程序中注明来源和版权信息，并详细说明使用了该第三方类库或者插件哪部分功能。
6. 要求具备**菜单功能**，根据用户的输入，执行相应的操作。在**菜单选项**中，必须包含“关于”一项，该菜单项用来显示本作业的基本信息，包括程序名称、版本、完成时间，姓名、学号、班级等信息；
7. 程序要求做到界面友好，对于用户的输入操作要有明确的提示信息，并对输入的数据进行合法性和正确性的检查；
8. 每个文件、类、方法的前面都必须有**规范的注释**，注释中要说明作者是谁、主要功能、编写时间等信息。变量、常量、关键语句等都必须有注释说明。
9. 所有源程序代码(含注释)都要拷贝到大作业报告的【3.3 程序实现】中。
10. 源代码为多文件项目，合理分配好各个源代码文件内容。

**模板格式说明：**

1. 正文模板中的**红色文字部分为说明文字，请仔细阅读后**撰写正式的内容。
2. 排版要求：

页面设置：上2.54cm，下2.54cm，左3.67cm，右2.67cm，页眉1.5cm,页脚1.75cm，行间距1.35倍。

全文中文字体采用宋体，英文字体采用Times New Roman字体，一级标题使用小三号字体，二级、三级标题均使用四号字体，正文内容采用小四号字体，代码字体大小为五号。**目 录**

一、 需求分析 5

1.1、开发环境 5

1.2、功能需求 5

二、 系统分析与设计 1

2.1、本程序需解决的关键技术问题 1

2.2、总体功能图 1

2.3、类图 2

2.4、程序流程图 3

2.4.1 用户流程图 3

2.4.2 管理员流程图 4

2.4.3 登录模块图 4

2.4.4 查询功能模块图 5

2.4.5借阅/归还功能模块图 5

2.4.6显示功能模块图 6

2.4.7添加功能模块图 6

2.4.8删除功能模块图 7

2.4.9 书库信息功能模块图 7

三、 程序设计与实现 8

3.1、项目文件组织结构 8

3.2、程序设计 9

3.2.1 Book类 9

3.2.2 Library类 9

3.2.3 Person类 10

3.2.4 Root类 10

3.2.5 User类 10

3.2.6 Main类 11

3.3、程序实现 11

3.3.1查询功能模块 11

3.3.2借阅、归还功能模块 13

3.3.3 图书整理功能模块 15

3.3.4 图书添加功能模块 15

3.3.5 图书删除功能模块 16

3.3.6保存书籍信息模块（txt） 16

3.3.8书库信息更新模块（dat） 17

3.3.9书库信息获取模块（dat） 18

3.3.10 登录功能模块 19

四、 系统测试 23

1. 登录测试 23

2. 普通用户登录测试 24

2.1 显示书籍列表 24

2.2 查询书籍信息 25

2.2.1 通过书名查询 25

2.2.2 通过作者查询 25

2.2.3 通过种类查询 25

3. 管理员登录测试 27

3.1 书库信息获取（文件读取） 28

3.2 增加书籍测试 29

3.3 整理书籍测试 30

3.4 书库信息更新测试 30

3.5 删除书籍测试 32

五、 总结与体会 33

# 需求分析

## **1.1、开发环境**

操作系统及版本：

Win10

开发工具及版本：

JDK 16.0.1

IntelliJ IDEA 2021.1 x64

## **1.2、功能需求**

《图书信息管理系统》主要针对的用户是小型图书馆或者个人，书库的数量和种类较少，读者的数量也比较少。相应的功能需求有：

1、 能够储存一定数量的书籍及其信息，并且有效的进行相应的图书信息操作。

2、 用户可对书籍进行借阅和返还，并且可以将书库的信息保存txt文件出来。

3、 管理员可对该系统进行书籍信息的删除、添加，对书库信息进行文件导入、导出操作。

4、 用户和管理员都可以查看书库里的书本信息，也可以通过书名、作者、种类等进行书籍的具体查询。

# 系统分析与设计

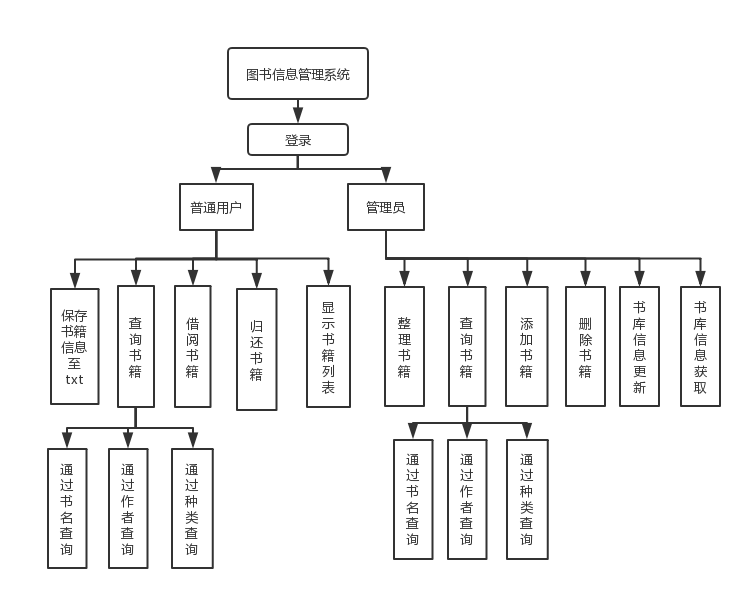
## **2.1、本程序需解决的关键技术问题**

通过集合来储存图书的信息，需要解决集合在类中的传递和应用等关键技术。

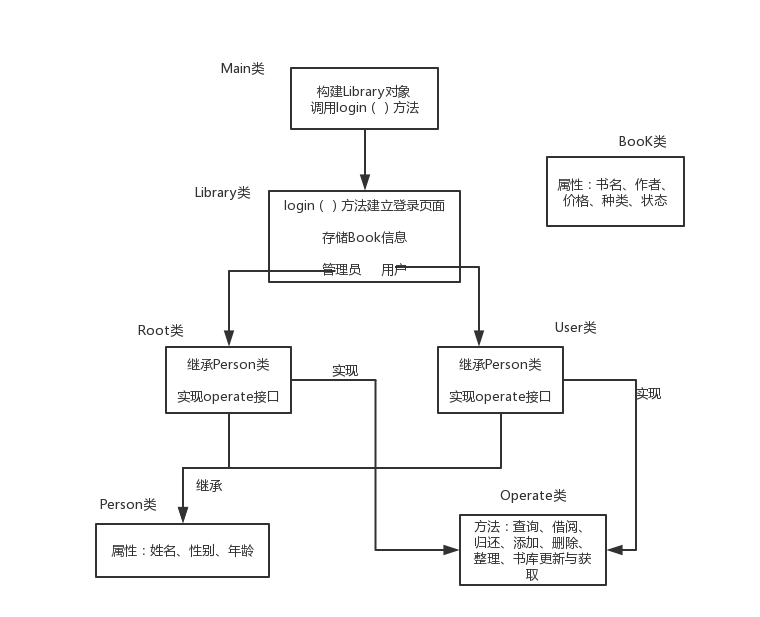
分为用户和管理员，含有共同的操作功能，需要解决接口、多态、继承等问题。

要想实现书籍信息的文件操作，需要有文件读写技术，还需要有异常处理。

## **2.2、总体功能图**

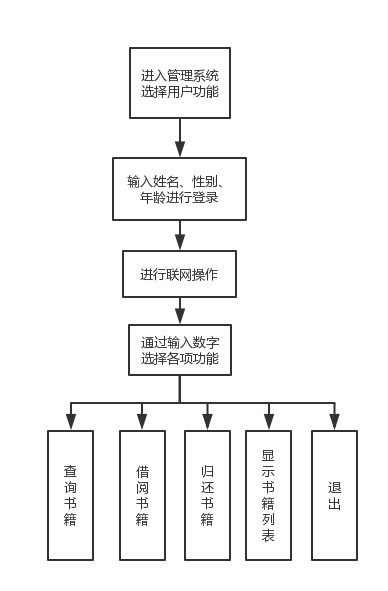


## **2.3、类图**

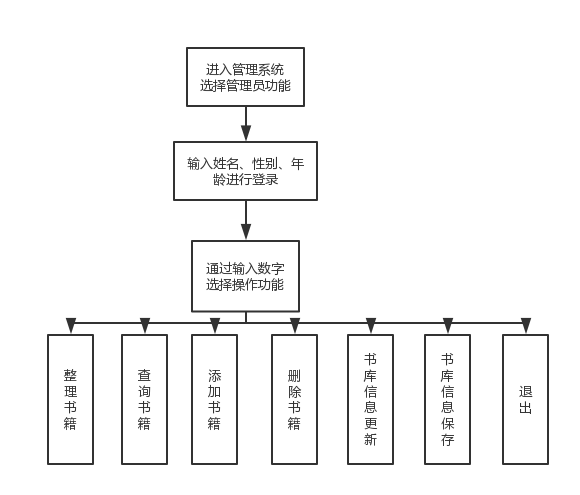


## **2.4、程序流程图**

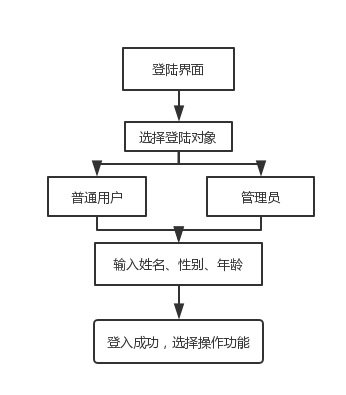
### 2.4.1 用户流程图



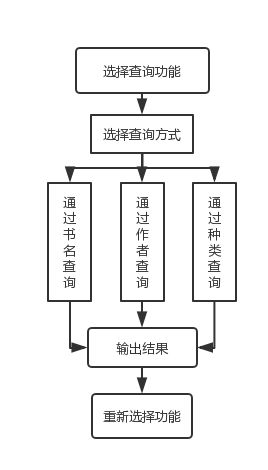
### 2.4.2 管理员流程图



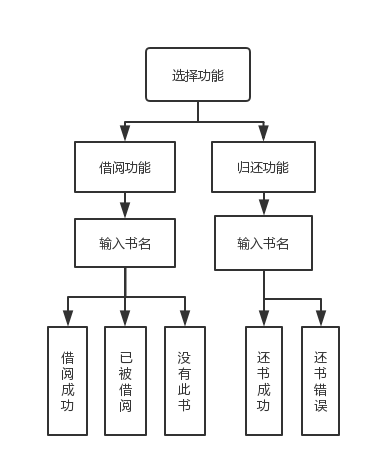
### 2.4.3 登录模块图



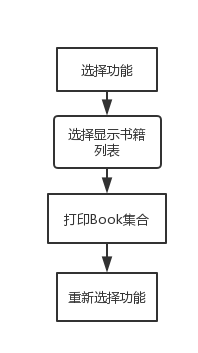
### 2.4.4 查询功能模块图



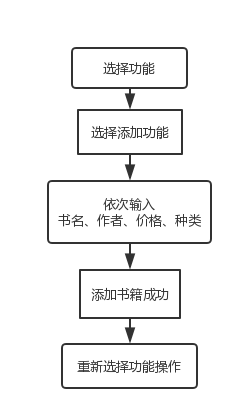
### 2.4.5借阅/归还功能模块图



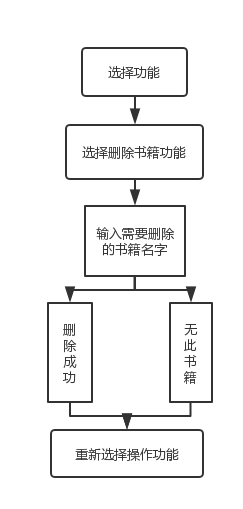
### 2.4.6显示功能模块图



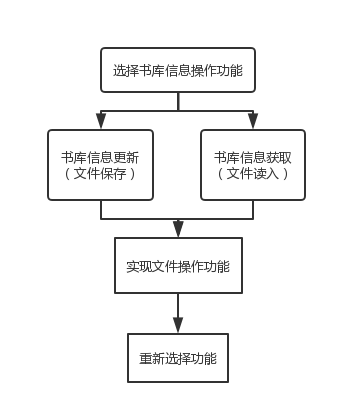
### 2.4.7添加功能模块图



### 2.4.8删除功能模块图

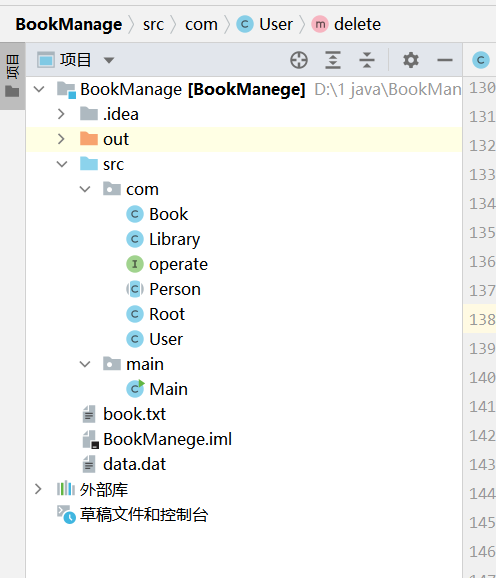


### 2.4.9 书库信息功能模块图



# 程序设计与实现

## **3.1、项目文件组织结构**



项目名为BookManage,其中有com包和main包。

com包有Book类，Library类，operate接口，Person类，Root类，User类。

main包中有Main类。文件有book.txt和data.dat。

Book类定义了书的属性（书名name、作者author、价格price、分类category、状态state）并且定义属性的get、set方法。Library类构建了图书集合books，写了登录函数，记录Book数据。

Person类定义了人的属性（姓名name、性别sex、年龄age）并且有属性的get、set方法。Operate接口，为操作的接口，里面有各种功能的方法名（以便于User、Root中进行重写）。Root类（实现管理员功能）和User类（实现普通用户功能）继承Person类并实现operate接口。

Main类的主函数开启项目。

book.txt是普通用户保存出来的书籍信息文件，data.dat是书库信息，帮助管理员实现读写功能。

## **3.2、程序设计**

### 3.2.1 Book类

该类继承Serializable，存储对象在存储介质中，以便在下次使用的时候，可以很快捷的重建一个副本，便于数据传输，尤其是在远程调用的时候。通过构造函数给出书的属性，并且定义了各种关于书的属性的set、get方法。便于后续对书中信息的操作。部分代码如下：

**private** String **name**;*//书名***private** String **author**;*//作者***private int price**;*//价格***private** String **category**;*//种类***private boolean state**;

*//状态 true-未借出// false-已借出*

Book(String name,String author,**int** price,String category,**boolean** state){  
 **this**.**name** = name;  
 **this**.**author** = author;  
 **this**.**price** = price;  
 **this**.**category** = category;  
 **this**.**state** = state;  
}

### 3.2.2 Library类

创新的地方是我在该类中构建了图书集合books，用集合来保存图书的信息。并且定义了登陆界面的login（）方法，有普通用户、管理员、关于、退出操作选择。里面有图书信息管理系统的登陆界面，以及普通用户和管理员的操作界面，通过获取用户的选择的功能数字，来调用User类或者Root类中的operate（）方法。部分代码如下：

**static** List<Book> *books* = **new** ArrayList<>();

### 3.2.3 Person类

该类为抽象类，定义了Person的属性，姓名、性别、年龄。同时还有各种关于属性的set、get方法。部分代码如下：

**private** String **name**;*//姓名***private** String **sex**;*//性别***private int age**;*//年龄*

Person(String name,String sex,**int** age) {  
 **this**.**age** = age;  
 **this**.**name** = name;  
 **this**.**sex** = sex;  
}

### 3.2.4 Root类

该类是管理员类，里面有各种关于管理员的操作方法。Root类继承Person类同时实现operate接口。

Operate接口中有各种方法，为了保证简洁性，在Root类和User类中对delete（）方法和add（）方法重写后的功能是不同的。delete（）在Root类中重写后是删除图书功能，而在User类中重写后是借阅功能。Add（）在Root类中是添加图书功能，在User类中则是归还功能。单独写了sort（）方法，实现对集合books中的图书信息按照价格的高低进行排序。select（）方法实现对图书信息查询功能，可选择书名、作者或种类进行查询。list（）方法是输出books集合中的信息。在该类中还有FileSave（）和FileRead（）方法，以data.dat文件运用对象流实现文件读写功能，同时使用异常处理操作。实现将预先存储在data.dat文件中关于图书的各种信息读入集合books中，或者将books中的信息保存在data.dat文件中，保证每次运行程序时都可以将书籍信息保留下来。

### 3.2.5 User类

该类是普通用户类，同样和Root类一样，继承了Person类并实现operate接口。

这里创新的是在该类中也有FileRead（）方法，用来实现一开始的“联网操作”，其实就是让普通用户在使用系统前将data.dat中最新的图书信息读入books集合，以保证普通用户可以获取最新的图书信息，进行相应的操作。该类中还有FileSaveTxt（）方法，方便普通用户将books集合中的信息保存到txt文件中。select（）方法是对图书进行查询，可以按照书名、作者和种类进行查询。delecte（）方法是对图书进行借阅，add（）方法是对图书进行归还，list（）方法是显示集合中的图书信息。

### 3.2.6 Main类

该类的主函数main中构建了Library对象，并且调用对象中的login（）方法启动登陆界面，开启整个项目。

## **3.3、程序实现**

### 3.3.1查询功能模块

查询功能在Root类中和User类中都有，该方法是实现operate接口后的方法重写。通过switch判断用户选择的功能，从而进入相应的查询功能。通过for循环遍历books集合中的书籍信息，通过equals判断是否相等，从而输出查询结果，若有此书籍则可以将信息从Book对象中通过get方法输出。

部分代码如下：

*//书籍查找功能* @Override  
 **public void** select(List<Book> books) {  
 System.***out***.println(**"请选择你要查询的方式："**);  
 System.***out***.println(**"1.通过书名查询"**);  
 System.***out***.println(**"2.通过作者查询"**);  
 System.***out***.println(**"3.通过种类查询"**);  
 **int** k = **sc**.nextInt();  
  
 **switch** (k) {  
  
 *// 根据书名进行查询* **case** 1:  
 System.***out***.println(**"请输入你要查询书籍的书名："**);  
 String name1 = **sc**.next();  
 **if** (name1 != **null**) {  
 **for** (Book book : books) {  
 **if** (book.getName().equals(name1)) {  
 System.***out***.println(**"有此书籍！"**);  
 System.***out***.println(**"书名："** + book.getName() + **" 作者："** + book.getAuthor() + **" 价格："** + book.getPrice() + **" 分类："** + book.getCategory() + **" 状态："** + book.isState());  
 **return**;  
 }  
 }  
 System.***out***.println(**"暂时没有此书！"**);  
 }  
 **break**;  
  
 *// 根据作者进行查询* **case** 2:  
 System.***out***.println(**"请输入你要查询书籍的作者："**);  
 String author1 = **sc**.next();  
 **if** (author1 != **null**) {  
 **for** (Book book : books) {  
 **if** (book.getAuthor().equals(author1)) {  
 System.***out***.println(**"有此书籍！"**);  
 System.***out***.println(**"书名："** + book.getName() + **" 作者："** + book.getAuthor() + **" 价格："** + book.getPrice() + **" 分类："** + book.getCategory() + **" 状态："** + book.isState());  
 **return**;  
 }  
 }  
 System.***out***.println(**"暂时没有此书！"**);  
 }  
 **break**;  
  
 *// 根据种类进行查询* **case** 3:  
 System.***out***.println(**"请输入你要查询书籍的种类："**);  
 String cater1 = **sc**.next();  
 **if** (cater1 != **null**) {  
 **for** (Book book : books) {  
 **if** (book.getCategory().equals(cater1)) {  
 System.***out***.println(**"有此书籍！"**);  
 System.***out***.println(**"书名："** + book.getName() + **" 作者："** + book.getAuthor() + **" 价格："** + book.getPrice() + **" 分类："** + book.getCategory() + **" 状态："** + book.isState());  
  
 }  
 }  
 System.***out***.println(**"查询结束!"**);  
 }  
 **break**;  
  
 **default**:  
 System.***out***.println(**"输入有误！"**);  
 }  
 }

### 3.3.2借阅、归还功能模块

该两种方法在user类中，通过获取普通用户输入的书名，通过for循环遍历books集合中的信息，equals（）判断到书名相同时，则通过book对象中的setState（）改变书籍的状态为false（或true）表示借阅成功（或还书成功）。

部分代码如下：

*//借书功能的实现*@Override  
**public void** delete(List<Book> books) {  
 System.***out***.println(**"请输入要借阅的书名:"**);  
 String name1 = **sc**.next();  
 **if** (name1 != **null**) {  
 **for** (Book book : books) {  
 **if** (book.getName().equals(name1)) {  
 **if** (book.isState()) {  
 book.setState(**false**);  
 System.***out***.println(**"借阅成功！"**);  
 **break**;  
 } **else** {  
 System.***out***.println(**"已被借阅！"**);  
 **break**;  
 }}} }}

*//还书功能的实现*@Override  
**public void** add(List<Book> books) {  
 System.***out***.println(**"请输入要还的书名："**);  
 String name = **sc**.next();  
 *//通过书名进行还书* **if** (name != **null**) {  
 **for** (Book book : books) {  
 **if** (book.getName().equals(name) && !book.isState()) {  
 book.setState(**true**);  
 System.***out***.println(**"还书成功！"**);  
 **return**;} }  
 System.***out***.println(**"还书错误！"**);  
 }}

### 3.3.3 图书整理功能模块

该方法在管理员Root类中，实现对图书按照价格从大到小进行排序。BookComparator直接在调用sort接口时穿一个Comparator的对象，即匿名内部类，实现compare接口。调用Collections.sort()返回的排序方式为自然排序。

部分代码如下：

**public void** sort(List<Book> books) {  
 System.***out***.println(**"图书馆中的书按价格整理:"**);  
 **class** BookComparator **implements** Comparator<Book>{  
 @Override  
 **public int** compare(Book o1, Book o2) {  
 **return** (**int**)(o1.getPrice() - o2.getPrice());  
 }  
 }  
 Collections.*sort*(books, **new** BookComparator());  
 System.***out***.println(**"整理结束！"**);}

### 3.3.4 图书添加功能模块

该方法位于Root类中，实现对图书集合books进行添加操作。通过对输入的书名、作者、价格、种类的获取，将信息通过add（）调用Book类的构造方法添加入books集合中。

部分代码如下：

*//对书进行添加操作*@Override  
**public void** add (List < Book > books) {  
 System.***out***.println(**"请输入书名："**);  
 String name = **sc**.next();  
 System.***out***.println(**"请输入作者："**);  
 String author = **sc**.next();  
 System.***out***.println(**"请输入价格："**);  
 **int** price = **sc**.nextInt();  
 System.***out***.println(**"请输入种类："**);  
 String category = **sc**.next();  
 books.add(**new** Book(name, author, price, category, **true**));  
 *//添加书籍的默认状态是ture - 未被借出* System.***out***.println(**"添加书籍成功！"**);}

### 3.3.5 图书删除功能模块

该方法在Root类中，实现管理员对图书的删除功能。通过获取输入的书名信息，for循环通过equals（）判断书名，再通过remove（）将books集合中关于这本书的信息进行移除。

部分代码如下：

*//对书进行删除操作*@Override  
 **public void** delete (List<Book> books){  
 System.***out***.println(**"请输入要删除的书名:"**);  
 String str = **sc**.next();  
 **if** (str != **null**) {  
 **for** (Book book : books) {  
 **if** (book.getName().equals(str)) {  
 books.remove(book);  
 System.***out***.println(**"删除成功！"**);  
 **return** } }  
 System.***out***.println(**"无此书籍！"**); } }

### 3.3.6保存书籍信息模块（txt）

该方法位于User类中，将文件保存为book.txt ,实现普通用户将books集合中的书籍信息保存至book.txt的文件写入操作。使用BufferedWriter实现，效率比Outputstream高。先将数据读取到缓存中，再从缓存中读取写入文件。同时采用try-catch异常处理。@Override  
**public void** FileSaveTxt(List<Book> books) {  
 **try** {  
 *//将文件保存为book.txt* File file = **new** File(**"book.txt"**);  
 FileWriter fileWriter = **new** FileWriter(file);  
 *//将写入转化为流的形式* BufferedWriter bufferedWriter = **new** BufferedWriter(fileWriter);  
 **for** (Book book : books) {  
 bufferedWriter.write(String.*valueOf*(book));  
 bufferedWriter.newLine();  
 bufferedWriter.flush();  
 *//清空缓冲区* }  
 bufferedWriter.close();  
 *//关闭读写流* } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
}

### 3.3.8书库信息更新模块（dat）

该模块位于Root类中，采用对象流，通过FileOutPutStream实现管理员将books集合中的书籍信息写入data.dat文件中功能，既实现书库信息的更新。当管理员在添加或删除书籍后，通过本功能可以对data.dat信息进行及时更新，在下次程序启动后，可保证上一次对图书信息进行的操作。运用异常处理。

部分代码如下：

*// 书库信息更新（文件保存）  
// 书库通过管理员添加新书籍后，可进行文件保存  
//通过文件保存功能,保存到 date.dat ，实现书库信息更新* **public void** FileSave (List < Book > books) {  
 *//异常处理* **try** {  
 *//创建文件输出流对象* FileOutputStream fout=**new** FileOutputStream(*file*);  
 *//创建缓冲区输出流对象* BufferedOutputStream bout=**new** BufferedOutputStream(fout);*//加速  
 //创建对象输出流* ObjectOutputStream objOut=**new** ObjectOutputStream(bout);  
 objOut.writeObject(books);*//将学图书集合对象输出到文件保存* objOut.flush();  
 objOut.close();  
 System.***out***.println(**"图书信息写出到文件成功！"**);  
 System.***out***.println(**"书库更新成功！"**);  
 } **catch** (FileNotFoundException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 } **catch** (IOException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 } }

### 3.3.9书库信息获取模块（dat）

该模块功能在Root类和User类中，在User类中时实现“联网”操作，既通过对象流ObjectInputStream将data.dat读取到books集合中，保证图书信息管理系统的实时性和连续性。在Root中实现书库信息data.dat的获取，实现文件读取功能。

部分代码如下：

*//书库信息获取（文件读取）***public** List<Book> FileRead () {  
 *//异常处理* **try** {  
 *//创建文件输入流对象* FileInputStream fin=**new** FileInputStream(*file*);  
 *//创建缓冲区输入流对象* BufferedInputStream bin=**new** BufferedInputStream(fin);*//加速  
 //创建对象输入流* ObjectInputStream objin=**new** ObjectInputStream(bin);  
 *//将对象输入流的中的对象读给List集合* **return** Library.*books*=(List<Book>) objin.readObject();  
  
 } **catch** (FileNotFoundException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 } **catch** (IOException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 } **catch** (ClassNotFoundException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }  
 System.***out***.println(**""**);  
 System.***out***.println(**"书库信息获取成功！"**);  
 **return** Library.*books*;  
}

### 3.3.10 登录功能模块

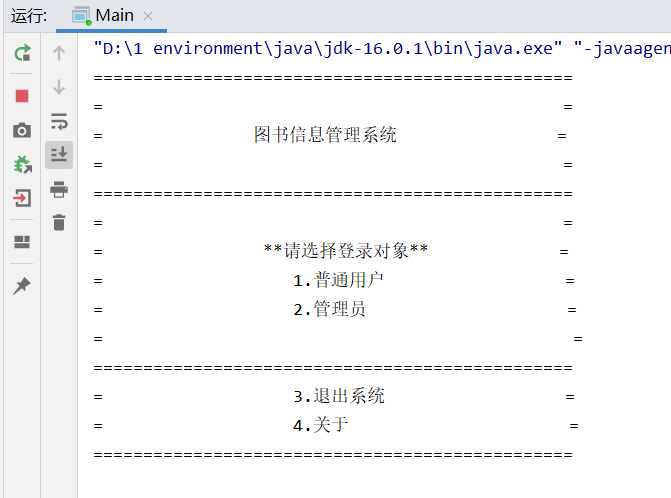
login（）方法位于Library类中，用于将操作信息提供给用户和管理员。同时获取用户和管理员功能的选择。

部分代码如下：

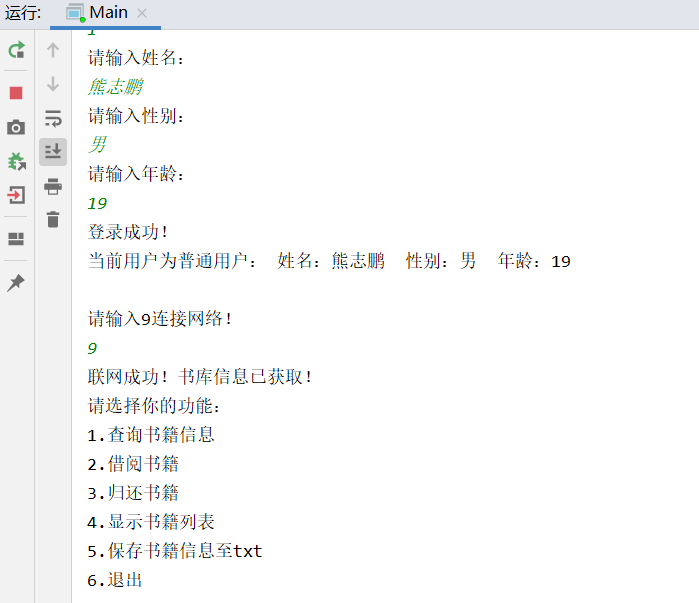
**public void** login(){  
 **int** count = 0;  
 Person p = **null**;  
 **while**(count<2){  
 System.***out***.println(**"================================================"**);  
System.***out***.println(**"= ="**);  
 System.***out***.println(**"= 图书信息管理系统 ="**);  
System.***out***.println(**"= ="**); System.***out***.println(**"================================================"**);  
System.***out***.println(**"= ="**);  
System.***out***.println(**"= \*\*请选择登录对象\*\* ="**);  
System.***out***.println(**"= 1.普通用户 ="**);  
 System.***out***.println(**"= 2.管理员 ="**);  
ystem.***out***.println(**"= ="**);  
System.***out***.println(**"================================================"**);  
System.***out***.println(**"= 3.退出系统 ="**);  
 System.***out***.println(**"= 4.关于 ="**); System.***out***.println(**"================================================"**);  
 **int** n = **sc**.nextInt();  
 **if**(n == 3){  
 **return**; }  
 **switch**(n){  
 *//进入普通用户操作界面* **case** 1:  
 System.***out***.println(**"请输入姓名："**);  
 String Uname = **sc**.next();  
 *//获取用户的姓名* System.***out***.println(**"请输入性别："**);  
 String Usex = **sc**.next();  
 *//获取性别* System.***out***.println(**"请输入年龄："**);  
 **int** Uage = **sc**.nextInt();  
 *//获取年龄  
 //构造普通用户对象* p = **new** User(Uname,Usex,Uage);  
 System.***out***.println(**"登录成功！"**);  
 System.***out***.println(**"当前用户为普通用户："**+**" 姓名："** + p.getName() + **" "**+**" 性别："** + p.getSex() + **" "**+**" 年龄："**+ p.getAge());  
 System.***out***.println(**" "**);  
 System.***out***.println(**"请输入9连接网络！"**);  
 **int** i = **sc**.nextInt();  
 p.operate(*books*,i);  
 *//普通用户的循环操作* **while** (**true**) {  
 System.***out***.println(**"请选择你的功能："**);  
 System.***out***.println(**"1.查询书籍信息"**);  
 System.***out***.println(**"2.借阅书籍"**);  
 System.***out***.println(**"3.归还书籍"**);  
 System.***out***.println(**"4.显示书籍列表"**);  
 System.***out***.println(**"5.保存书籍信息至txt"**);  
 System.***out***.println(**"6.退出"**);  
 i = **sc**.nextInt();  
 **if** (i == 6) {  
 System.***out***.println(**"您已成功退出！"**);  
 **break**;  
 }**else** {  
 p.operate(*books*,i);  
 *//调用普通用户的操作方法* } }  
 **break**;  
 *//进入管理员操作界面* **case** 2:  
 System.***out***.println(**"请输入管理员姓名："**);  
 String Rname = **sc**.next();  
 System.***out***.println(**"请输入性别："**);  
 String Rsex = **sc**.next();  
 System.***out***.println(**"请输入年龄："**);  
 **int** Rage = **sc**.nextInt();  
 *//构造管理员对象，并传递管理员信息* p = **new** Root(Rname,Rsex,Rage);  
 System.***out***.println(**"登录成功！"**);  
 System.***out***.println(**"当前管理员："**+**" 姓名："** + p.getName() + **" "**+**" 性别："** + p.getSex() + **" "**+**" 年龄："** + p.getAge());  
 **while** (**true**) {  
 System.***out***.println(**"请选择你的功能："**);  
 System.***out***.println(**"1.整理书籍"**);  
 System.***out***.println(**"2.查询书籍信息"**);  
 System.***out***.println(**"3.增加书籍"**);  
 System.***out***.println(**"4.删除书籍"**);  
 System.***out***.println(**"5.显示书籍列表"**);  
 System.***out***.println(**"6.书库信息更新（文件保存）"**);  
 System.***out***.println(**"7.书库信息获取（文件读取）"**);  
 System.***out***.println(**"8.退出"**);  
 **int** j = **sc**.nextInt();  
 **if** (j == 8) {  
 System.***out***.println(**"您已成功退出！"**);  
 **break**;  
 }**else**{  
 *//调用管理员的操作方法* p.operate(*books*,j);  
 }  
 } **break**;  
  
 **case** 4:  
 System.***out***.println(**"==================================="**);  
System.***out***.println(**"============\*\*关于\*\*==============="**);  
 System.***out***.println(**"==================================="**);  
System.***out***.println(**"= 程序名称：图书信息管理系统 ="**);  
 System.***out***.println(**"= 版本V2.0 ="**);  
 System.***out***.println(**"= 完成时间：2021.6.17 ="**);  
System.***out***.println(**"= 完成人：熊志鹏 8008120400 ="**);  
System.***out***.println(**"= 计算机Ⅱ类2014班 ="**); System.***out***.println(**"==================================="**);  
 **break**;  
 } } }}

# 系统测试

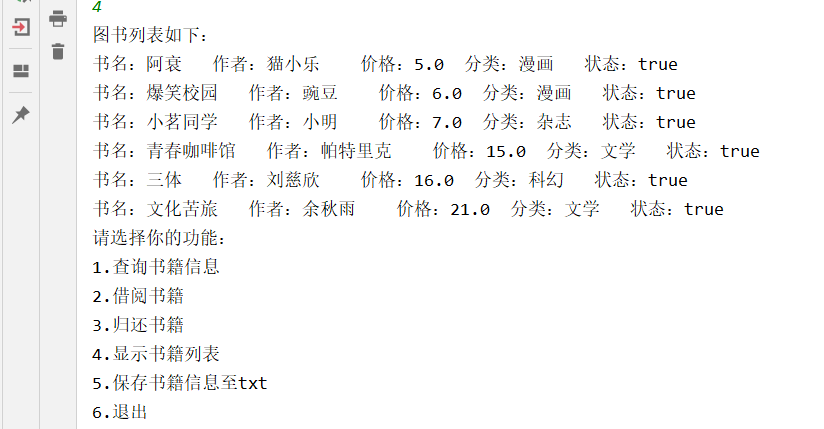
### 登录测试



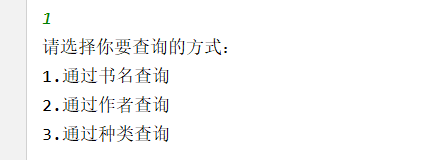
### 普通用户登录测试



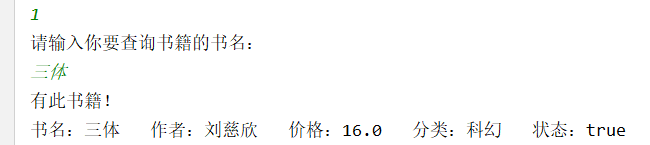
### 显示书籍列表



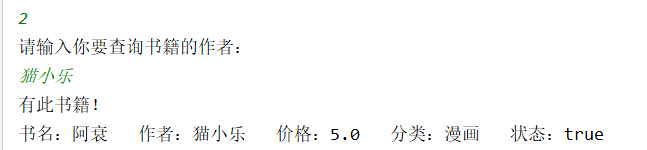
### 查询书籍信息

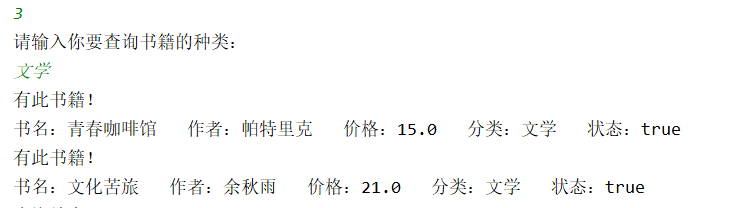


### 通过书名查询

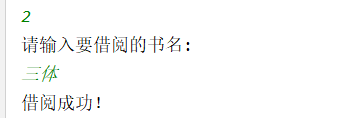


### 通过作者查询

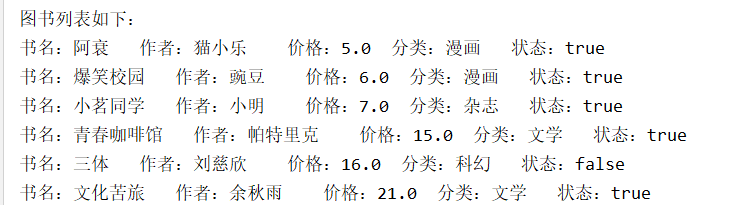


* + 1. 通过种类查询

2.3.借阅功能测试

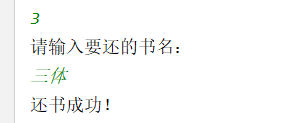
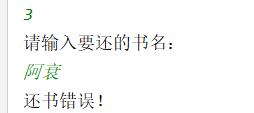


对《三体》借阅后，图书信息中《三体》的状态由true变为false表示已经借出。

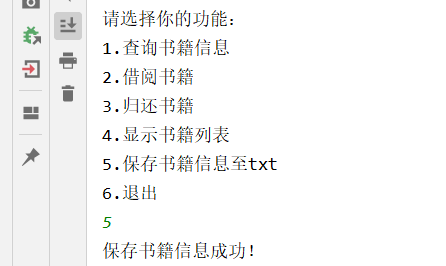


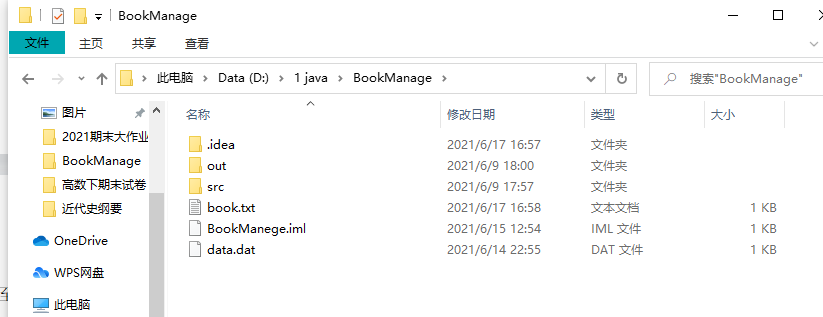
2.4.归还功能测试

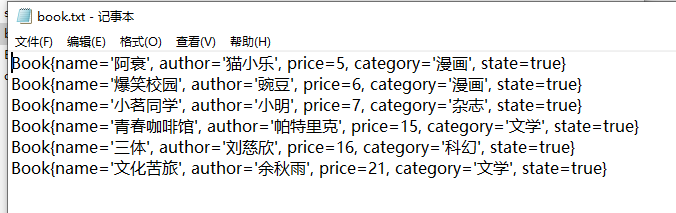
因为《阿衰》并未被借出，所以输入阿衰后输出“还书错误！”。而在重新输入《三体》后，显示还书成功。

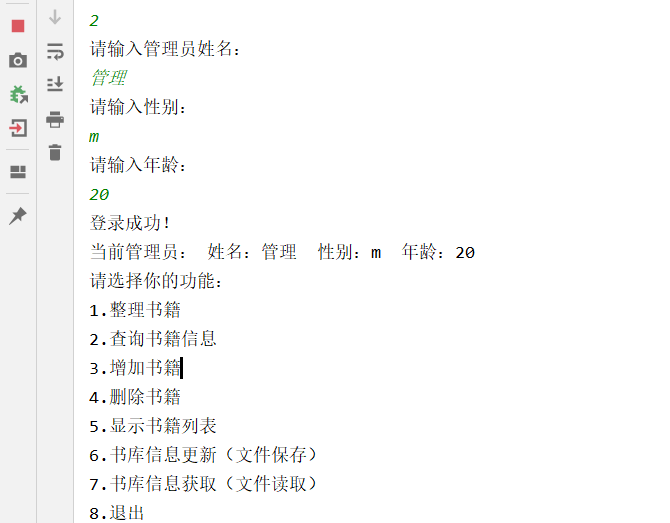


2.5.保存书籍信息至txt测试

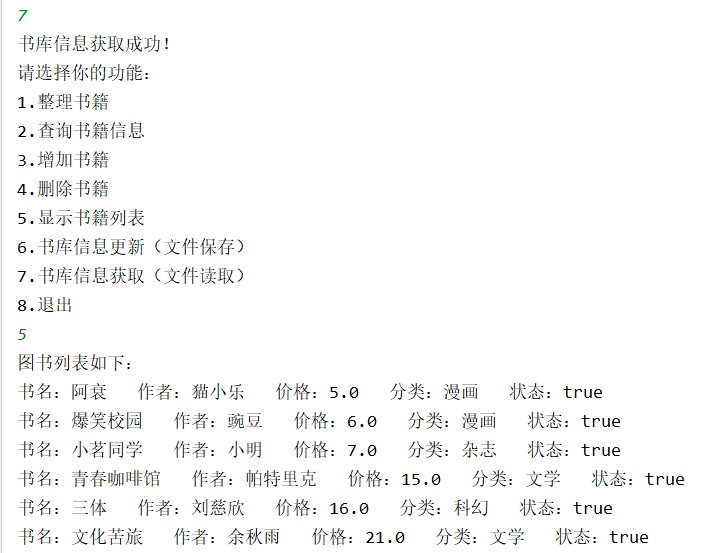
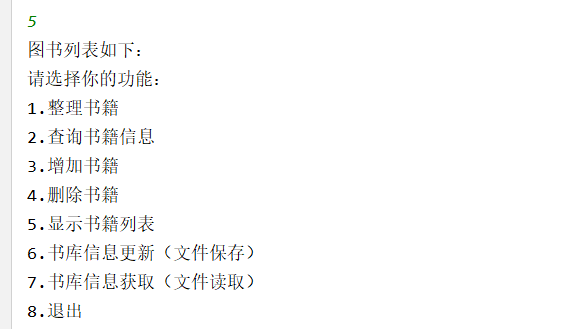




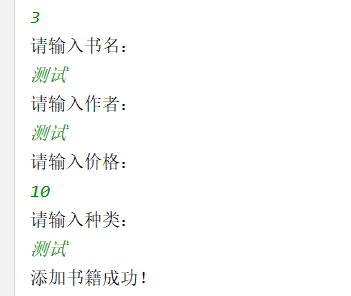


1. 管理员登录测试
   1. 书库信息获取（文件读取）

一开始通过5显示书籍列表功能，可以看到此时没有书籍。通过7书库信息获取功能，将data.dat文件中的图书信息读取进入图书管理系统后，获取到了书库信息。

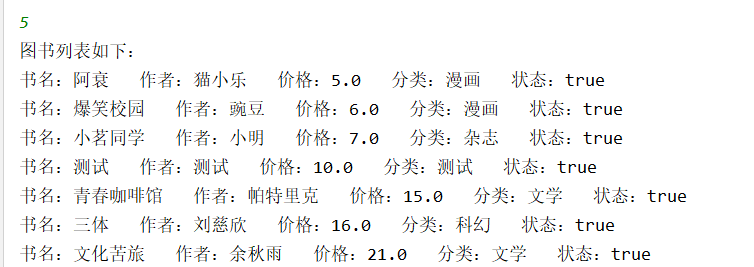
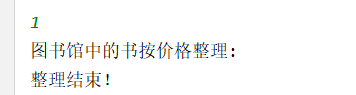


### 增加书籍测试



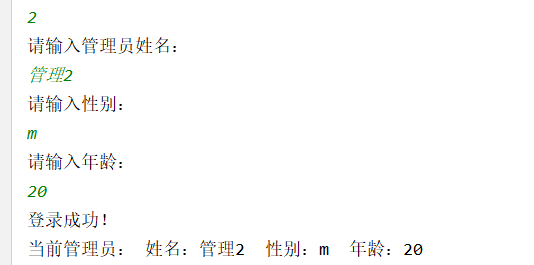
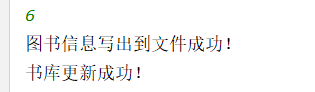
### 整理书籍测试

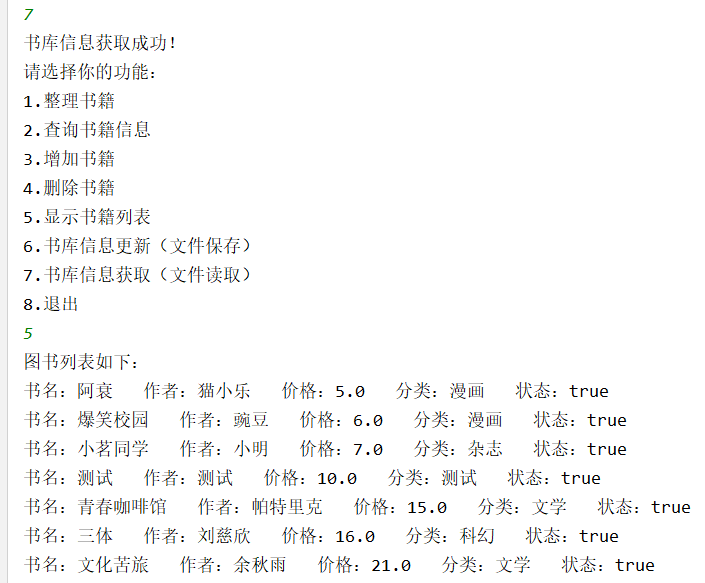
先通过5显示出书籍列表信息，刚刚添加的书籍在列表的最后。再通过整理功能，已经按照价格的高低进行了相应的排序。



### 书库信息更新测试

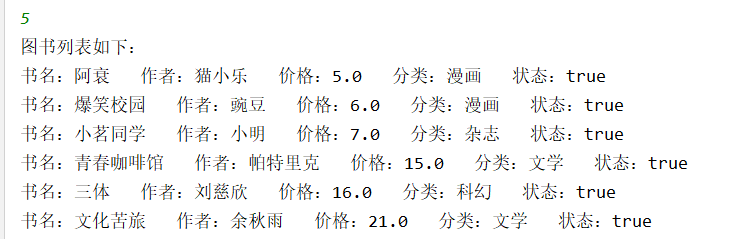
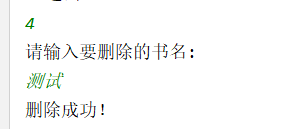
因为data.dat文件内的内容无法进行查看，所以通过重启程序后，进入管理员功能界面选择书库信息获取功能后，查看在书籍列表中是否有上一次运行程序时添加的《测试》书籍。可以看到列表中有《测试》这本书，说明书库信息更新功能运行成功。





### 删除书籍测试

删除《测试》这本书。



# 总结与体会

经过两个多星期的探索与努力，我的图书信息管理系统也终于完成了。一开始确定好这个项目后，就着手开始确定好项目里具体要实现的功能和实现这些功能的方法。在进行的过程中，一开始的图书管理系统对于普通用户来说，是没有“联网”这个功能的，也就是每次在运行程序后，上一次对于图书信息管理系统的操作是无法保存下来的。而在文件读写功能的添加时，一开始用FileReader在读取价格信息强制转换时一直报错，后来在向同学询问和上网查找相关读写功能的实现后，改用了ObjectInputStream对象流的方法，成功实现了对于data.dat文件的读取功能。我也意识到了自己在平时学习java的过程中，还是投入的时间太少，平常敲代码的时间也比较少，单单只是上课、看视频和完成老师布置的实验报告书时会去实际操作。这一点我觉得我做的不够好，还需要加强java的学习。

我认为我这个图书信息管理系统闪光之处，就在于使用了集合来存储图书的信息。同时在每次进入程序后，通过FileRead（）将data.dat的数据向管理系统进行更新，于是添加了FileRead（）功能，通过ObjectInputStream每一次运行程序时进行文件读取功能，实现图书信息管理系统的连续性和有效性。不足就是没有使用到GUI界面，交互性可能不太好。通过大作业的探索和努力，我深入了解了多态和继承，也另外学习了关于集合的知识与应用。俗话说的好啊，学无止境，我们对于各种知识的学习都是永无止境的。对于java来说，还有很多很多的内容自己并没有完全掌握，还需要自己加把劲，好好学习。当然，以后要加强代码能力，不能光说不练，还是要加强实践，自己动手练习，才能好好掌握。总而言之，还有很长的路要走！加油！