

课 程 设 计 报 告

课程名称 计算机程序设计基础 2

班 级 ***
学 号 *****
姓 名 ***

2024 年 9 月 6 日

一、设计内容与设计要求

1. 课程设计目的

面向对象程序设计课程设计是集中实践性环节之一，是学习完《面向对象程序设计》课程后进行的一次全面的综合练习。要求学生达到熟练掌握 C++语言的基本知识和技能；基本掌握面向对象程序设计的思想和方法；能够利用所学的基本知识和技能，解决简单的面向对象程序设计问题，从而提高动手编程解决实际问题的能力。**尤其重视创新思维培养。**

2. 课题题目

1) 图书管理系统

3. 文档设计要求

3. 1 设计课题题目：每个同学都单独完成 1 道课题。后面有范题，仅供同学们参考，不列入本次课程设计的课题。

3. 2 对于程设题目，按照范题的格式。自行虚构软件需求。并按照第 4 点要求，编写设计文档。基本要求系统中设计的类的数目不少于 4 个，每个类中要有各自的属性（多于 3 个）和方法（多于 3 个）；需要定义一个抽象类，采用继承方式派生这些类。并设计一个多重继承的派生类。在程序设计中，引入虚函数的多态性、运算符重载等机制。

4. 程序设计的基本要求：

4. 1 要求利用面向对象的方法以及 C++的编程思想来完成系统的设计；

4. 2 要求在设计的过程中，建立清晰的类层次；

4. 3 根据课题完成以下主要工作：①完成系统需求分析：包括系统设计目的与意义；系统功能需求（系统流程图）；输入输出的要求。②完成系统总体设计：包括系统功能分析；系统功能模块划分与设计（系统功能模块图）。③完成系统详细设计：数据文件；类层次图；界面设计与各功能模块实现。④系统调试：调试出现的主要问题，编译语法错误及修改，重点是运行逻辑问题修改和调整。⑤使用说明书及编程体会：说明如何使用你编写的程序，详细列出每一步的操作步骤。⑥关键源程序（带注释）

4. 4 图书管理系统**应至少包含以下功能：**

- ① 用户信息管理：存储用户信息，如用户名、借阅历史等。支持对用户的信息进行查询、增加、删除、修改等操作；
 - ② 图书信息管理：存储图书信息，如书名、作者、类别、关键字、简介、借阅状态等。支持对图书的信息进行查询、增加、删除、修改等操作，且要求图书查询功能具有一定模糊查询的能力，例如用户输入一个字，可查询到包含该字的图书；
 - ③ 借还书记录管理：记录用户借还书的时间等信息、查看用户或图书的借阅历史等；
 - ④ 统计分析功能：根据借还书记录进行简单的统计分析，比如最受欢迎的书籍、借阅最多的用户、用户借阅量随时间的变化趋势等。可参考某些 app 的年度报告；
 - ⑤ 特殊情况处理：对非法输入等特殊情况进行合适的处理；
 - ⑥ 你认为需要的其他功能。
-

- 4.5 自己设计测试数据，将将测试数据存在文件中，通过文件来进行数据读写来测试；
- 4.6 按规定格式完成课程设计报告，并在网络学堂上按时提交；
- 4.7 不得抄袭他人程序、课程设计报告，每个人应独立完成，在程序和设计报告中体现自己的个性设计。

5. 进度安排

小学期 第 2 周	7.22	需求调研	
	7.23	类的定义	
	7.24	函数设置	
	7.25	代码调试	
	7.25	编写说明	

注： 1、一定要保留自己那个课题的完整任务书在课程设计报告里面。 2、“评分表”放在“附录：源程序清单”的后面。

图书管理系统

目 录

1.	系统需求分析.....	1
2.	总体设计.....	4
3.	详细设计.....	7
4.	系统调试.....	21
5.	测试结果与分析.....	22
6.	总结.....	31
	附录 1：源程序清单	32
	附录 2：评分表	79

1. 系统需求分析

图书管理系统记录了图书、借书记录和与书有关的人的信息。图书信息包括图书的名称、图书的作者、类型、关键词、简介和借阅状态（0表示在架，1表示外借）。借书记录信息包括借书开始日期、借书结束日期、书名、借书人名和借阅状态（0表示已归还，1表示未归还）。与书有关的人包括使用者（借书者）和管理员，其中使用者信息包括姓名、登录密码、年龄、性别、学生证或工作证号、使用者的身份（学生/教师/工作人员）、借书权限开始日期和借书权限结束日期。管理员信息包括姓名、登录密码、年龄、性别、学生证或工作证号、使用者的身份（学生/教师/工作人员）、借书权限开始日期、借书权限结束日期、管理系统中的职位（图书管理员等）、管理员权限开始日期和管理员权限结束日期。经过上述分析，可以设计能提供以下功能的考勤管理系统：

1). 录入图书信息：从键盘输入信息，输入格式为：书名|作者|类别|关键字|简介|借阅状态（0表示在架，1表示外借），所有信息均可以包含空格，每行一条记录。

例如：

C/C++程序设计教程|黄维通 孙甲松|大学教材|C语言 程序设计|本书是作者根据长期教学实践编写而成的。全书内容由浅入深，逐步介绍C/C++语言中的基本概念和语法，使读者全面而系统地理解和掌握用C/C++语言进行程序设计的方法。|0
（在程序中为一行）

电子电路与系统基础|李国林|大学教材|电子 电路|本课程是对原“电路原理”“模拟电路”“通信电路”和“数字电路”等课程重构形成的新电路原理课程，体系架构为一条主干四个分支。|1
（在程序中为一行）

另外还可以选择逐项输入书名、作者、类别、关键字、简介和借阅状态（0表示在架，1表示外借）信息，所有信息也可以包含空格。

2). 修改图书信息：可以对图书信息的任意部分进行修改，有单行修改和逐项修改两种方式。单行修改从键盘输入一整行图书数据，输入格式为：书名|作者|类别|关键字|简介|借阅状态（0表示在架，1表示外借）。

例如：

C/C++程序设计教程|黄维通 孙甲松|大学教材|C语言 程序设计|本书是作者根据长期教学实践编写而成的。全书内容由浅入深，逐步介绍C/C++语言中的基本概念和语法，使读者全面而系统地理解和掌握用C/C++语言进行程序设计的方法。|1
（在程序中为一行）

另外还可以选择逐项修改书名、作者、类别、关键字、简介和借阅状态（0表示在架，1表示外借）信息。

- 3). 查询图书信息：可以按照书名和书的全部信息（书名、作者、类别、关键字、简介和借阅状态）进行模糊查找，并将所有包含查找结果显示在屏幕上。
- 4). 删除图书信息：可以按照书名进行查找，并将查找到的书删除。
- 5). 列出图书信息：可以列出系统内全部图书的信息。
- 6). 列出图书书名：可以列出系统内全部图书的书名。
- 7). 录入借书记录：从键盘输入信息，输入格式为：借阅状态（0表示已归还，1表示未归还）借阅人 书名 借书年月日 还书年月日（未归还不填写还书年月日）

例如：

1 李华 电子电路与系统基础 2024 6 22

0 小美 论语 2024 5 2 2024 7 18

另外还可以选择逐项输入借阅状态（0表示已归还，1表示未归还）、借阅人、书名、借书年月日和还书年月日（未归还不填写还书年月日）。

- 8). 还书更新记录：当读者归还书籍更改借阅状态为0，并更新还书年月日。
 - 9). 查询借书记录：可以按照书名或人名进行模糊查找，并列出所有相关的借书记录。
 - 10). 查询最受欢迎的书：可以统计所有读书记录，选出被借阅次数最多的书籍，支持并列排名功能。
 - 11). 查询常借阅的读者：可以统计所有读书记录，选出借阅次数最多的读者，支持并列排名功能。
 - 12). 查询借书数目随时间的变化：可以统计所有读书记录，按日期给出每日的借书数目，并给出借书数目最多的日期。
 - 13). 录入人员信息：可以选择增加使用者或管理员的信息，使用者的输入格式为使用者学号（工号）姓名 密码 年龄 性别（男“m”，女“f”）身份 借书权限开始年月日 借书权限结束年月日，每行一条记录。
- 例如：
- 2021010322 小红 abcd1234 21 f 学生 2021 8 31 2025 7 31
- 另外还可以选择逐项输入使用者学号（工号）、姓名、密码、年龄、性别（男“m”，女“f”）、身份、借书权限开始年月日和借书权限结束年月日。
- 管理员的输入格式为管理员学号（工号）姓名 密码 年龄 性别（男“m”，女“f”）身份 借

书权限开始年月日 借书权限结束年月日 管理员职位 管理员权限开始年月日 管理员权限结束年月日，每行一条记录。

例如：

2012004267 李华 aaa111 30 m 工作人员 2012 8 31 2032 7 31 图书管理员 2012 8 31 2032 7
31

另外还可以选择逐项输入管理员学号（工号）、姓名、密码、年龄、性别（男“m”，女“f”）、身份、借书权限开始年月日、借书权限结束年月日、管理员权限开始年月日和管理员权限结束年月日。

14). 修改人员信息：可以选择修改使用者或管理员的信息，输入使用者或管理员的名字后系统就能重新输入其信息，达到修改的目的。

15). 删除人员信息：可以选择删除使用者或管理员的信息，输入使用者或管理员的名字后系统就能删除其对应的信息。

16). 列出人员信息：可以列出系统内所有使用者和管理员的信息。

17). 查询人员信息：向系统内输入用户名，就可以进行查找，如果该用户存在，就显示其对应的信息，如该用户不存在则显示不存在。

18). 用户注册：为方便程序使用，本程序设计了用户在系统中进行注册的功能。在程序开始运行时，用户需要用户的信息，进而可以注册使用者账号，通过运行登录程序可进入图书管理系统内部使用其功能。

19). 用户登录：为方便程序使用，本程序设计了用户登录系统的功能。在程序开始运行时，用户需要输入用户名和与之对应的密码，进而登录程序进入图书管理系统内部使用其功能。

20). 退出账号：当用户不想再使用程序时，他可以选择退出账号但不退出程序，让下一位使用者登录程序进行运行。

21). 系统以菜单方式工作。（所谓菜单指用户可以自由选择所要执行的功能。学生可以通过以上功能注册账号、登录系统、录入信息、修改信息、查询信息、整理统计出所要了解的信息并退出账号或程序，除了要实现上述的基本功能之外，本系统还在细节上下工夫，使用户使用方便，在使用的过程中保持愉快的心情。图书管理系统有广大的用户群，这其中既有图书管理人员、老师、学生、也可以面向一般用户使用。总之，该系统可以满足用户需求，实现对其管理的人员的考勤工作。

2. 总体设计

图书管理系统主要包含三方面共二十个主要功能。第一方面是图书管理方面，主要包括录入图书信息、修改图书信息、查询图书信息、删除图书信息、列出图书信息和列出图书书名这六大功能。其中管理员用户可以使用全部功能，而普通用户只能进行查询图书信息、列出图书信息和列出图书书名的操作。

在录入图书信息时，可以选择单行添加图书信息或逐项添加图书信息。如果选择单行添加，用户可将图书信息按格式输入。如果选择逐项添加，用户可按程序提示输入图书信息。输入完毕后，程序会提示数据已成功写入，并返回主界面，以进行下一次操作。

在修改图书信息时，可选择单行修改图书信息或逐项修改图书信息。如果选择单行修改，用户可将图书信息按格式输入进行修改。如果选择逐项添加，用户可按程序提示输入图书信息进行修改。输入完毕后，程序会提示数据已成功写入，并返回主界面，进行下一次操作。

在查询图书信息时，可以选择按书名查找或全部信息查找。该程序具备模糊查找功能，如果选择按书名查找，则用户可按程序提示输入要查找的书名，之后程序会显示出所有包含输入书名的书籍信息。如果选择全部信息查找，用户可输入要查找的信息，之后程序会显示出所有信息中包含输入字符的书籍信息。之后程序会返回主界面，以进行下一次操作。

在删除图书信息时，用户可按要求输入要删除的书籍名称，之后程序会删除对应名称的书籍。删除完毕后，程序会提示数据已成功删除，并返回主界面，以进行下一次操作。

在列出图书信息时，程序会显示出系统中所有的图书信息。随后程序会返回主界面，以进行下一次操作。

在列出图书书名时，程序会显示出系统中所有的图书书名。随后程序会返回主界面，以进行下一次操作。

第二方面是借书记录管理方面，主要包括录入借书记录、还书更新记录、查询借书记录、查询最受欢迎的书、查询常借阅的读者和查询借书数目随时间的变化这六大功能。其中管理员用户可以使用全部功能，而普通用户只能进行列出借书记录、查询借书记录、查询最受欢迎的书、查询常借阅的读者和查询借书数目随时间变化的操作。

在录入借书记录时，可以选择单行添加借书记录或逐项添加借书记录。如果选择单行添加，用户可将借书记录按格式输入。如果选择逐项添加，用户可按程序提示输入借书记录。输入完毕后，程序会提示数据已成功写入，并返回主界面，以进行下一次操作。

在还书更新记录时，程序会提示输入归还书籍的名称、还书名称和还书日期，之后程序会提示记录已修改完成。随后程序会返回主界面，以进行下一次操作。

在查询借书记录时，可以选择按书名查找或按人名查找。该程序具备模糊查找功能，如果选择按书名查找，则用户可按程序提示输入要查找的书名，之后程序会显示出所有包含输入书名字符的书籍的借阅记录。如果选择按人名查找，用户可按程序提示输入要查找的人名，之后程序会显示出所有包含输入人名的借阅记录。随后程序会返回主界面，以进行下一次操作。

在查询最受欢迎的书时，程序会显示出被借阅次数最多的书籍以及其借阅次数，并支持并列显示多本被借阅次数最多的书籍，随后程序会返回主界面，以进行下一次操作。

在查询常借阅的读者时，程序会显示出借阅次数最多的读者以及其借阅次数，并支持并列显示多个借阅次数最多的读者，随后程序会返回主界面，以进行下一次操作。

在查询借书数目随时间的变化时，程序会按时间顺序显示出每天被借出的图书数目，随后程序会返回主界面，以进行下一次操作。

第三方面是用户管理方面。主要包括录入人员信息、修改人员信息、删除人员信息、列出人员信息、查找人员信息、用户注册、用户登录和退出账号这八大功能。其中管理员用户可以使用全部功能，而普通用户只能进行用户注册、用户登录和退出账号的操作。

在录入人员信息时，可以选择单行添加人员信息或逐项添加人员信息。如果选择单行添加，用户可将人员信息按格式输入。如果选择逐项添加，用户可按程序提示输入图书信息。输入完毕后，程序会提示人员信息已成功写入，并返回主界面，以进行下一次操作。

在修改人员信息时，可以选择单行修改人员信息或逐项修改人员信息。如果选择单行修改，用户可将人员信息按格式输入进行修改。如果选择逐项添加，用户可按程序提示输入人员信息进行修改。输入完毕后，程序会提示人员信息已成功写入，并返回主界面，以进行下一次操作。

在删除人员信息时，用户可按要求输入要删除的人员名称，之后程序会删除对应的人员信息。删除完毕后，程序会提示人员信息已成功删除，并返回主界面，以进行下一次操作。

在列出人员信息时，程序会显示出系统中所有的图书信息。随后程序会返回主界面，以进行下一次操作。

在查询人员信息时，用户可输入要查找的人员名称，之后程序会显示出所有信息中包含输入人员名称的人员信息。之后程序会返回主界面，以进行下一次操作。

在用户注册时，用户可选择单行添加人员信息或逐项添加人员信息。如果选择单行添加，用户可将人员信息按格式输入。如果选择逐项添加，用户可按程序提示输入图书信息。输入完毕后，程序会提示人员已成功注册，并返回主界面，以进行下一次操作。

在用户登录时，用户可输入自己的用户名和密码，之后就能登录系统进行各项操作。如果用户名或密码不正确，程序会提示重新输入。

在退出账号时，程序会退出登录，之后可以使用新的账号登录系统。

图书管理系统的功能模块图：

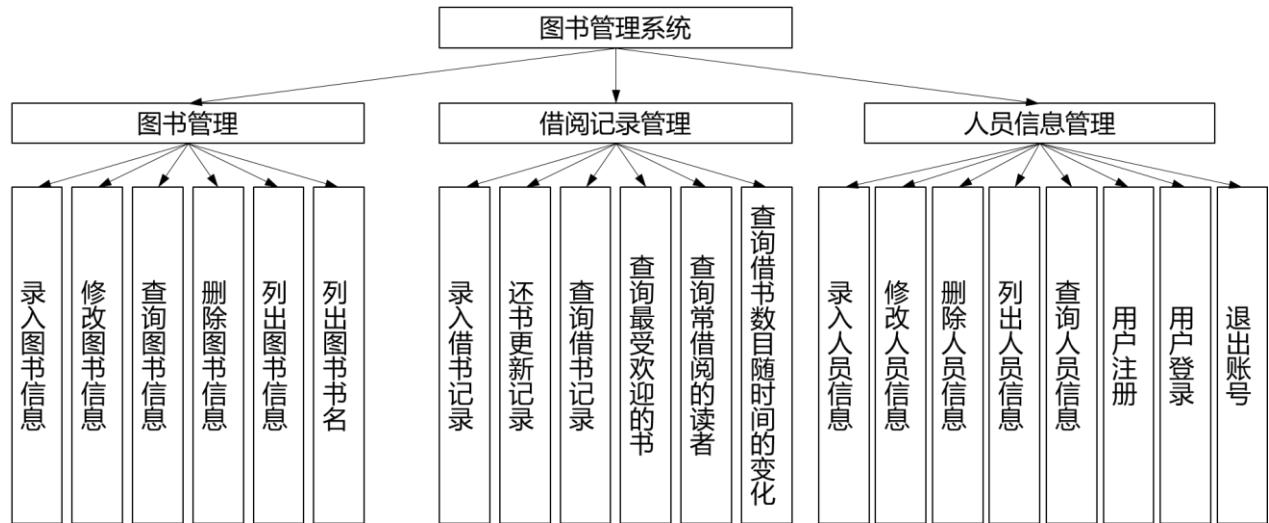


图1 图书管理系统的功能模块图

3. 详细设计

图书管理系统中类的类层次图为：



图2 图书管理系统类的类层次图

学生考勤管理系统中各功能模块的实现（管理员）：

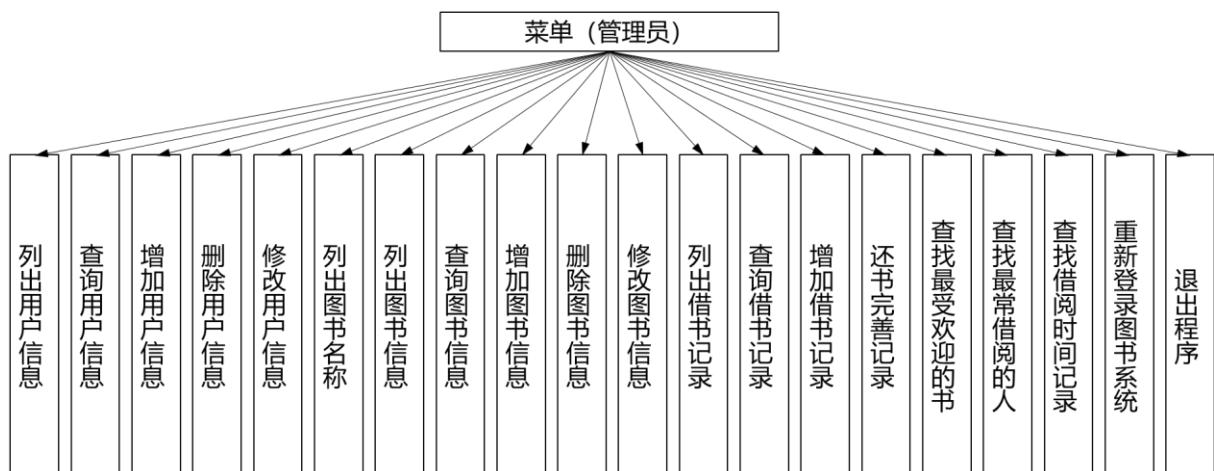


图3 图书管理系统中各功能模块的实现（管理员）

学生考勤管理系统中各功能模块的实现（使用者）：

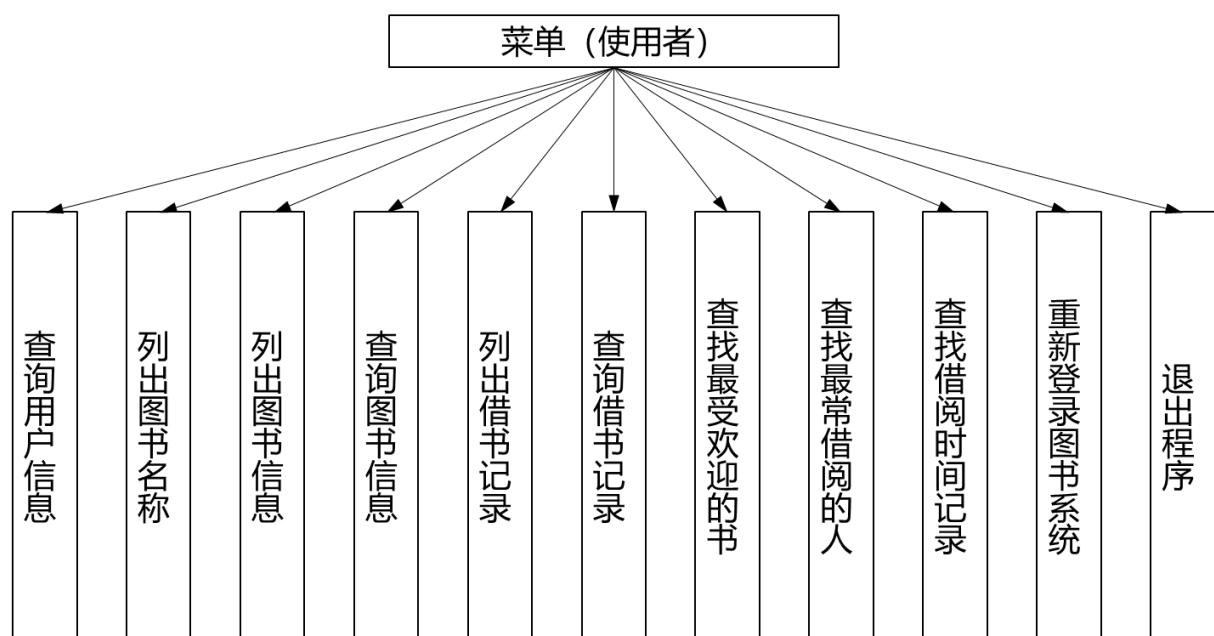


图4 图书管理系统中各功能模块的实现（使用者）

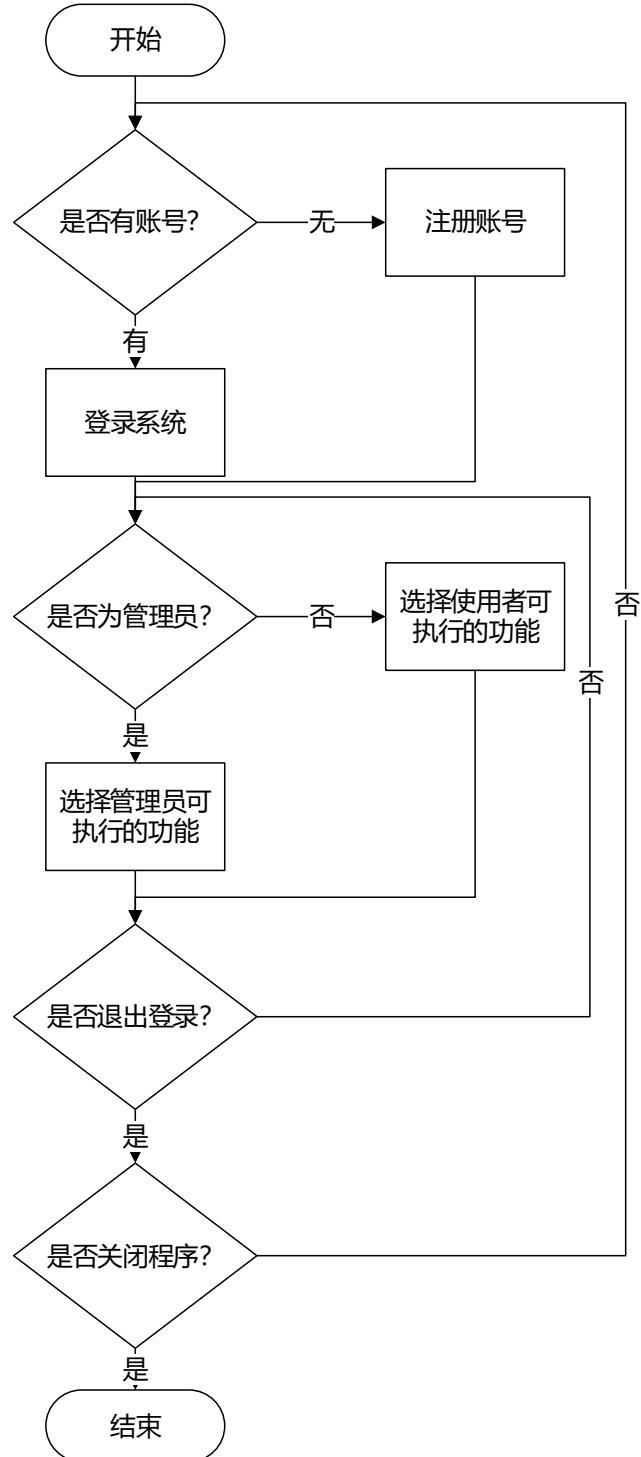


图5 图书管理系统运行流程图

1. 录入图书信息模块

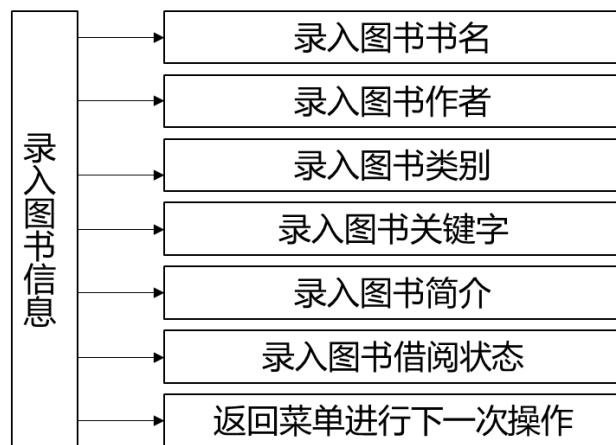


图6 录入图书信息模块

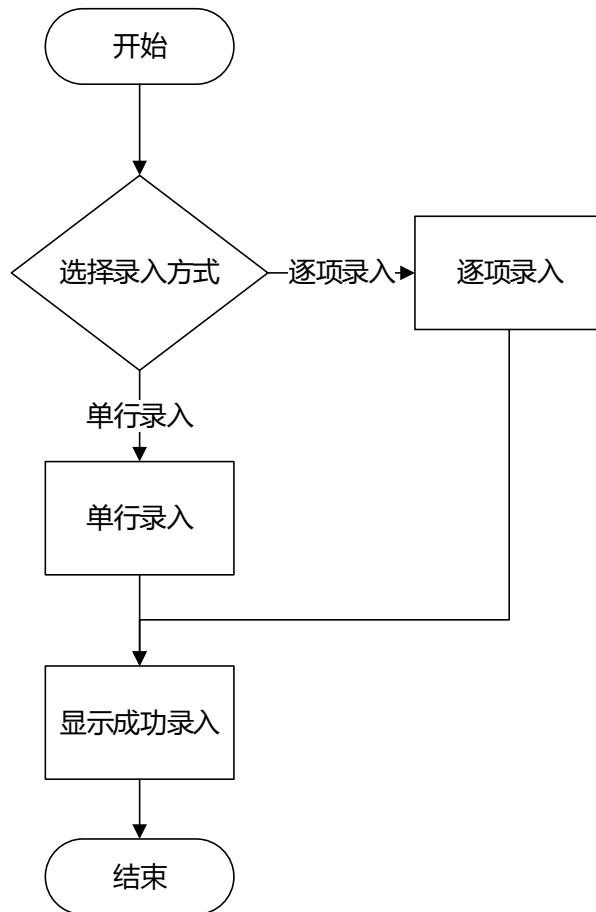


图7 录入图书信息模块运行流程图

2. 修改图书信息模块



图8 修改图书信息模块

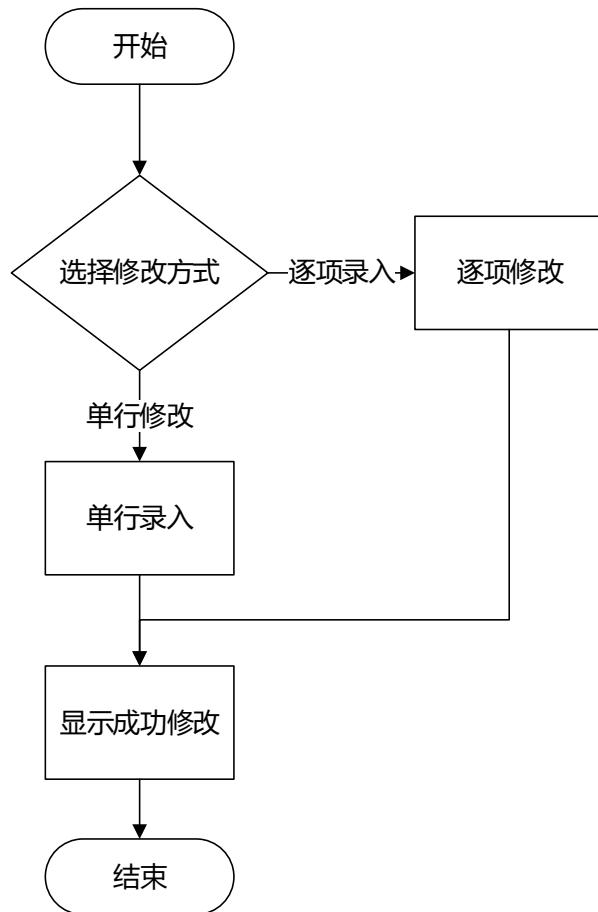


图9 修改图书信息模块流程图

3. 查询图书信息模块

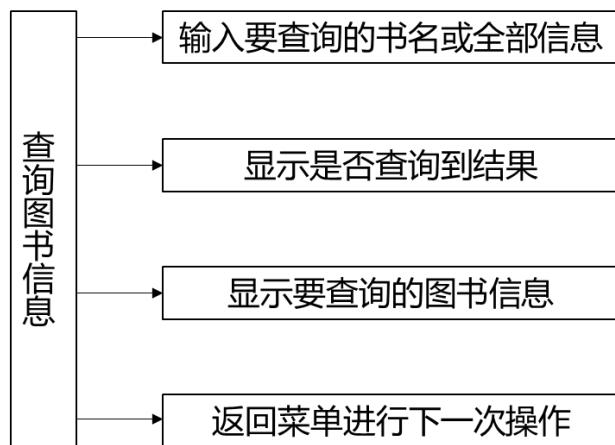


图10 查询图书信息模块

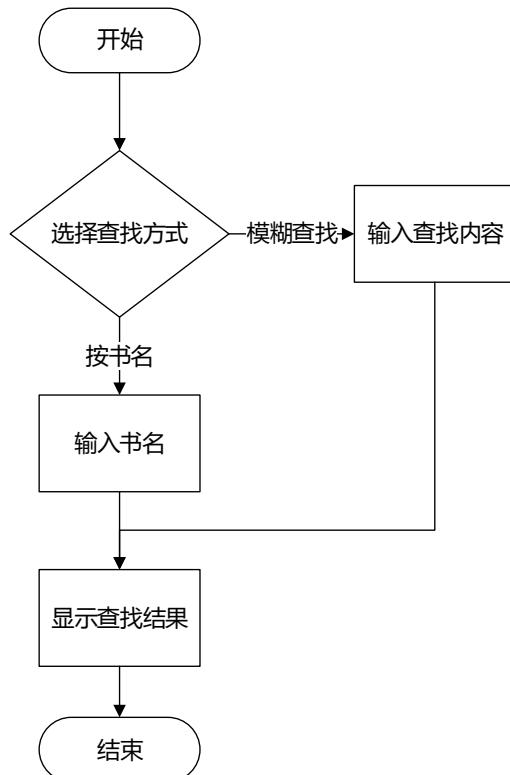


图11 查询图书信息模块流程图

4. 删 除 图 书 信 息 模 块

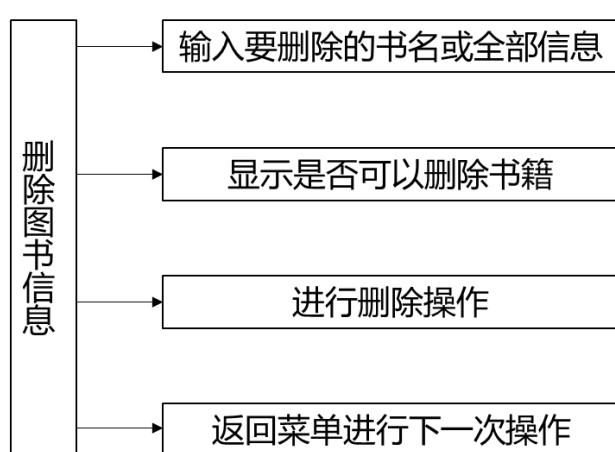


图12 删除图书信息模块

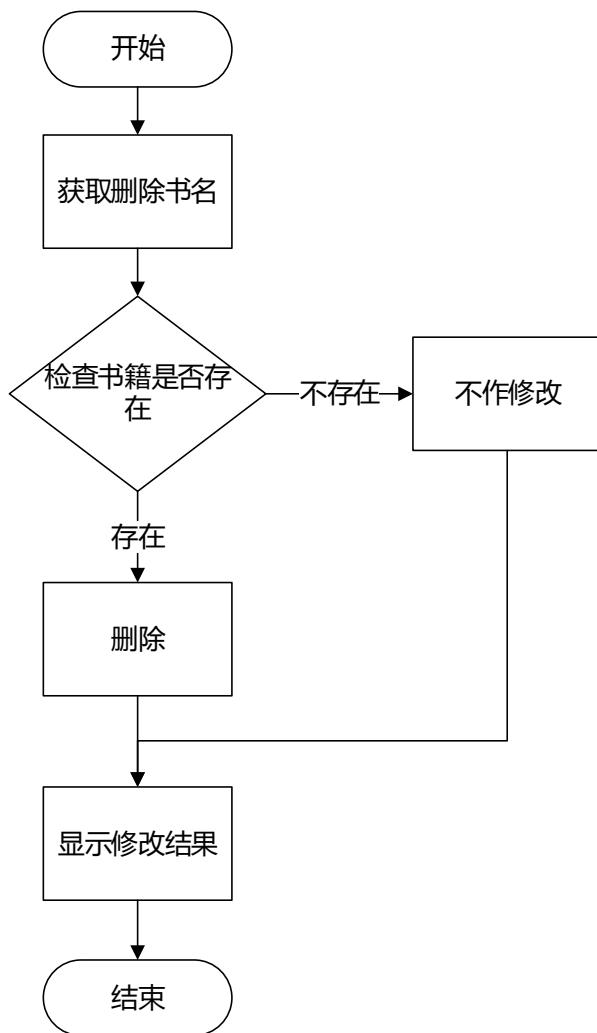


图13 删除图书信息模块流程图

5. 列出图书信息模块

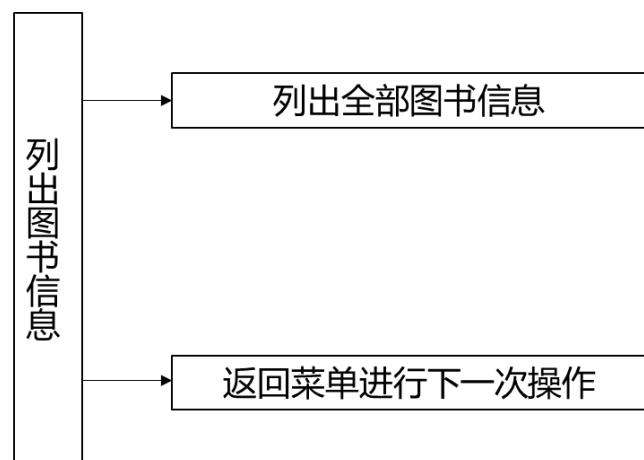


图14 列出图书信息模块

6. 列出图书书名模块

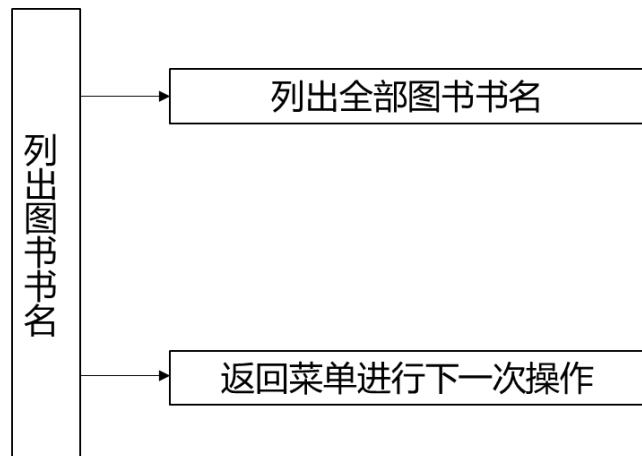


图15 列出图书书名模块

7. 录入借书记录模块

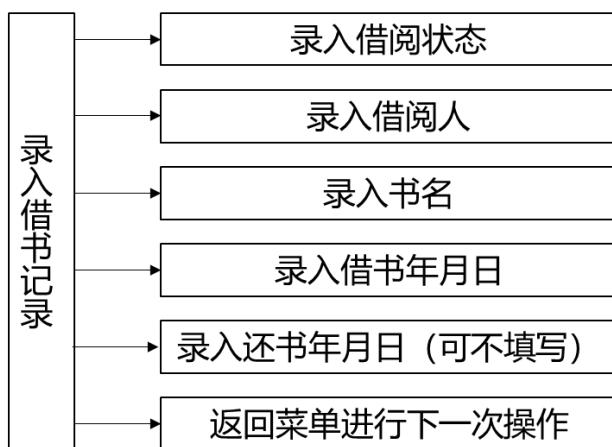


图16 录入借书记录模块

8. 录入借书记录模块

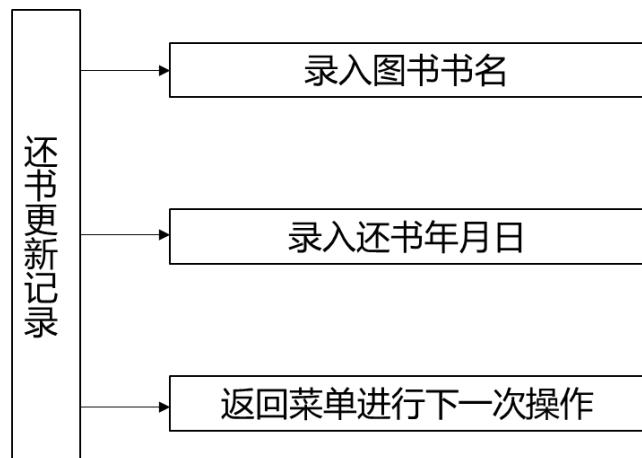


图17 录入借书记录模块

9. 查询借书记录模块

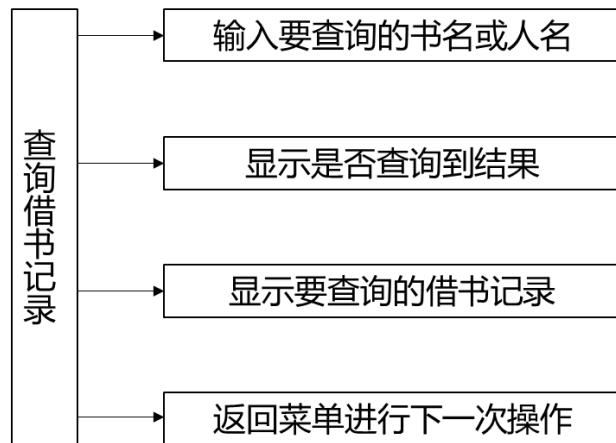


图18 查询借书记录模块

10. 查询最受欢迎的书模块

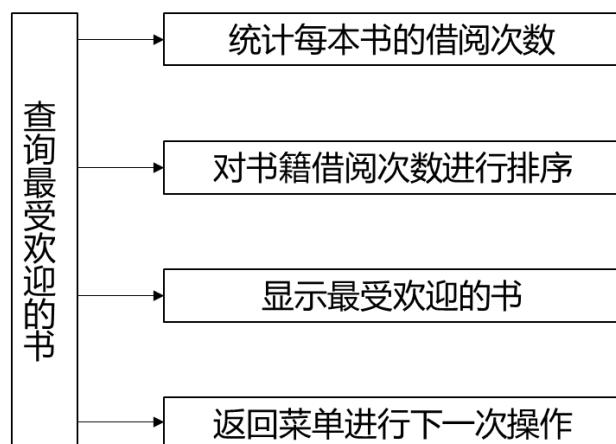


图19 查询最受欢迎的书模块

11. 查询常借阅的读者模块

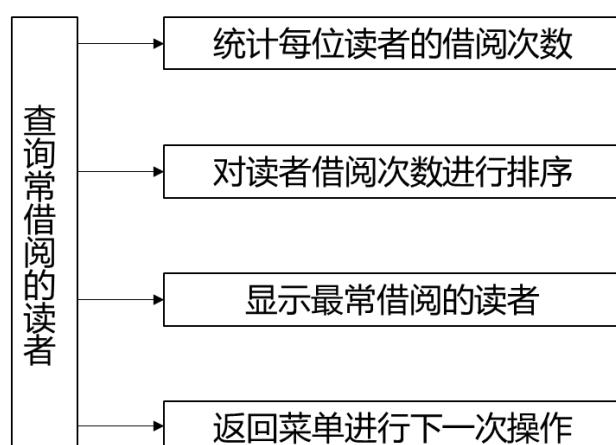


图20 查询常借阅的读者模块

12. 统计分析查询借书数目随时间的变化模块



图21 查询借书数目随时间的变化模块

13. 录入人员信息模块



图22 录入人员信息模块 (管理员)

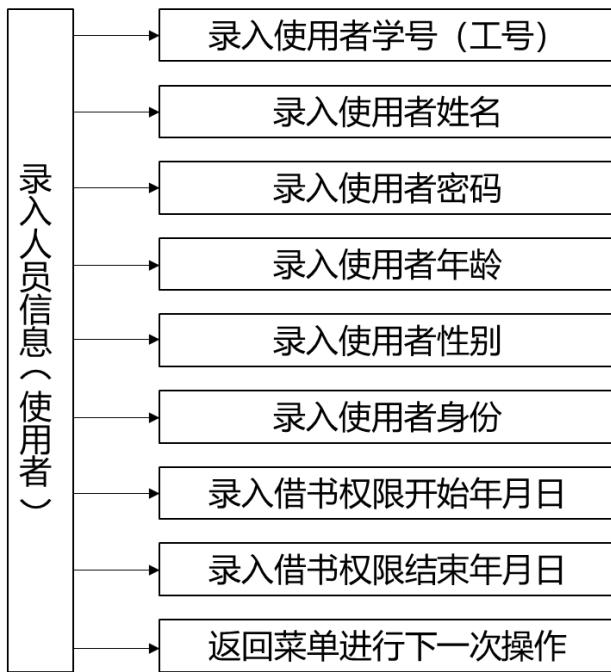


图23 录入人员信息模块（使用者）

14. 修改人员信息模块

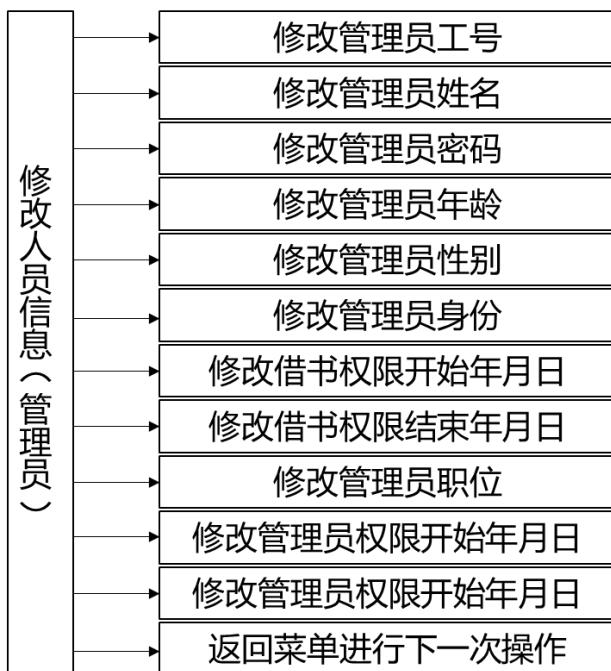


图24 修改人员信息模块（管理员）

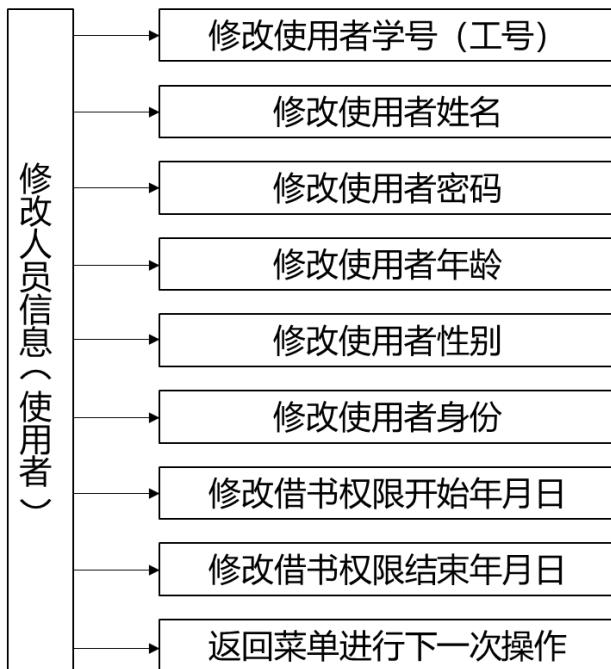


图25 修改人员信息模块（使用者）

15. 删除人员信息模块

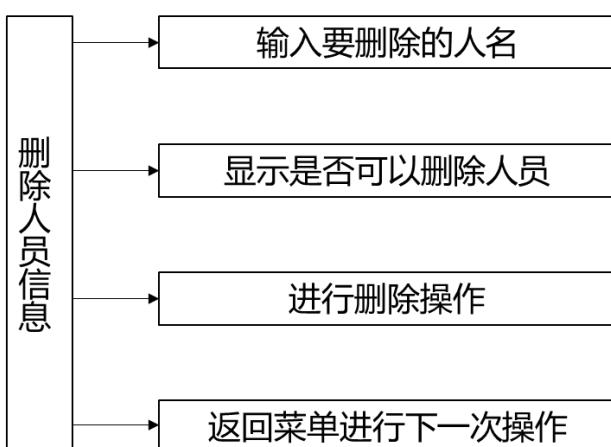


图26 删除人员信息模块

16. 列出人员信息模块

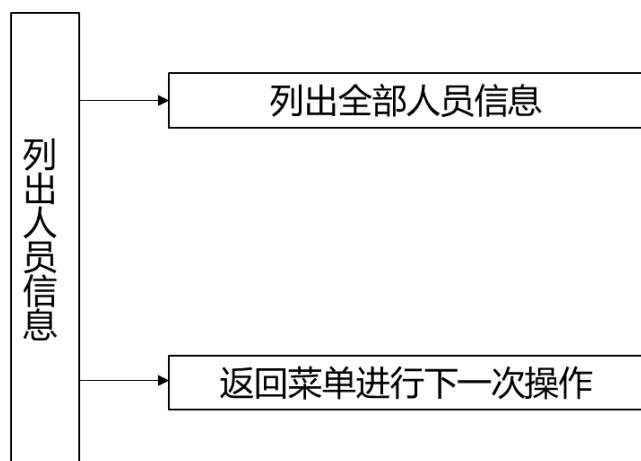


图27 列出人员信息模块

17. 查询人员信息模块

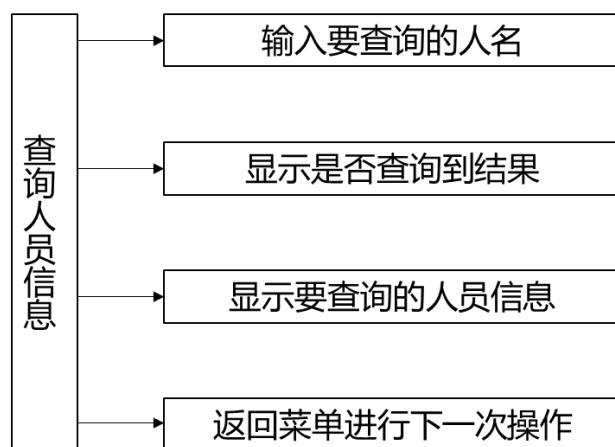


图28 查询人员信息模块

18. 用户注册模块

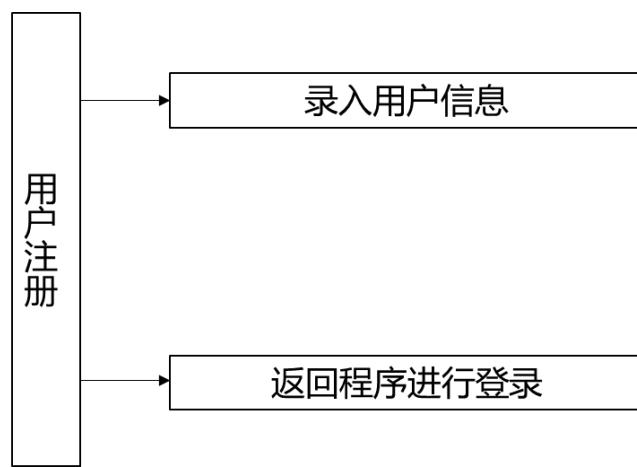


图29 用户注册模块

19. 用户登录模块

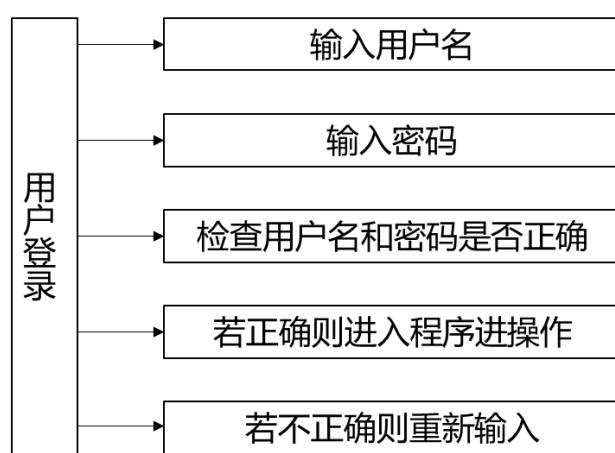


图30 用户登录模块

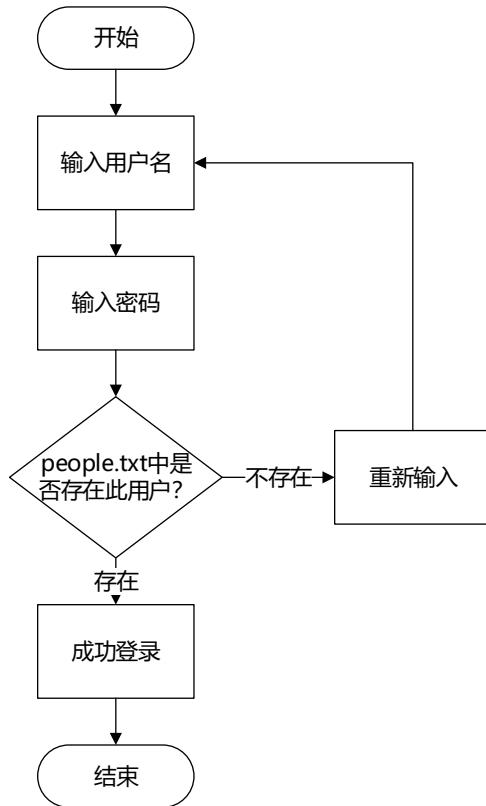


图31 用户登录模块流程图

20. 退出程序模块

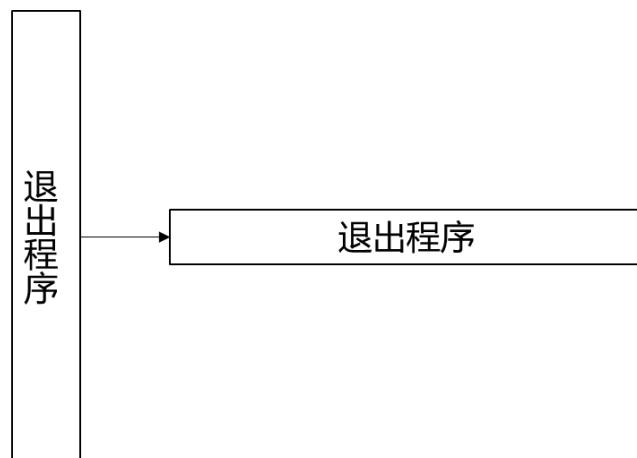


图32 退出程序模块

系统的UML图为：

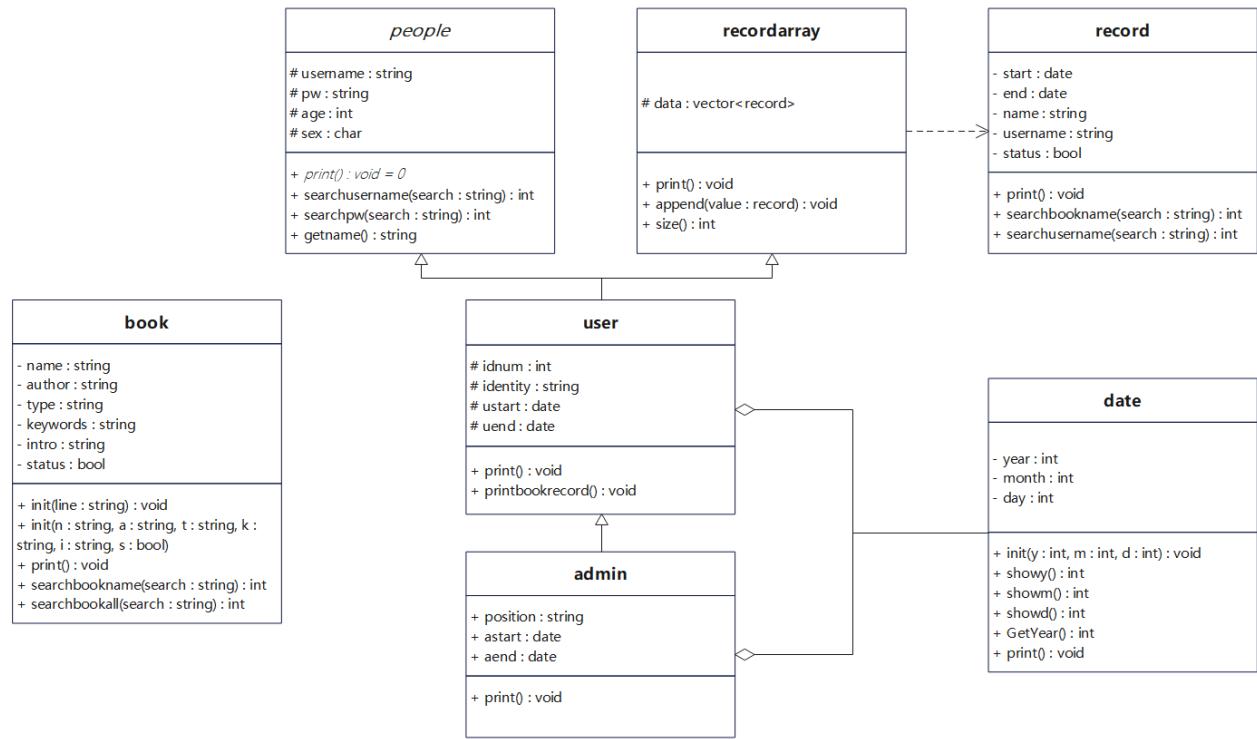


图33 系统的UML图

4. 系统调试

程序编写完成后，我进行了程序调试，调试过程中出现了一些问题，但经过不断改进，程序已经可以正常运行，并顺利实现了预期功能。

第一个问题是允许输入含有空格的书籍信息。理论上，书籍的书名信息和简介中可能出现空格，但传统的C++输入和按空格区分元素的储存方法对其不适用。因此，我在调试时修改了数据的存储方式，在单行输入和存储书籍信息时用“|”进行分割，这就可以输入含有空格的书籍信息了。

第二个问题是避免错误的数据输入。程序中经常有输入数字的环节，若使用者输入了字符，程序就可能崩溃，若输入了不在范围内的数字，程序也不能正常运转。因此，在调试中，我使用int `getint(int arr[], int size)`函数解决这一问题，`arr[]`由可输入的数字构成的数组，`size`是该数组的大小。该函数先使用`cin.fail()`判断输入是否成功，若成功则进一步判断输入数字是否在数组中。如果在，则返回输入值，否则则会使用`cin.clear()`和`cin.ignore(std::numeric_limits<streamsize>::max(), '\n')`清空缓冲区重新输入。通过这一改进，可有效避免数据的错误输入。

第三个问题是逐项输入用户信息时产生的问题。我发现在调试过程中程序不能准确录入输入在键盘上的信息。因此，我重新研读代码，修改了与“<<”和“>>”运算符相关的重载语句，并调整了程序逻辑，最终解决了这一问题。

第四个问题是在运行时发现程序难以有效进行对书名和人名的统计分析。经过仔细研究，我使用C++中STL的一个关联容器`map`构建了`wordCount`，构建起了书名或人名关键字与借书次数的对应关系，接着将`wordCount`中的信息转移到`vector`类型的`sortedWordCount`中，再对其进行排序，即可解决这一问题，得到最受欢迎的书和最常借阅的人信息了。同时，程序也具有一定的扩展能力，可以进一步实现对全部书籍和人员借阅次数的排序。

第五个问题是打开文件时易发生错误，而在运行时又难以迅速判断错误发生原因。因此，我在程序中添加了检查文件是否打开和成功关闭的模块，并在出现异常时显示异常信息。通过这种改进，我逐步排查出了代码中的错误，使程序能正常进行读写。

第六个问题是程序的运行逻辑问题。最初的程序，执行完给定的功能后就会退出程序，要想运行其余功能还要重新进入程序，较为繁琐。经过调试，我决定在程序中加入一个平台来实现这些功能。改进后的程序通过输入数字决定要实现的功能，在输入完之后返回平台等待下一次操作。使用者可以选择退出程序或切换账号进行下一次登录，较为方便，提高了程序运行效率。

第七个问题是关于指针的深拷贝问题。在用户登录函数中，开始时我在存储读入的用户信息时忽略了指针深拷贝的问题，导致用户名和密码不能正常对应，运行经常不能得到正确结果。经过深入思考，我对程序进行优化，采用深度拷贝函数解决了这一问题，实现了程序中用户的正常登录使用功能。

通过对程序的不懈调试，我找到了程序在运行中的问题并加以解决，实现了图书管理系统对书籍、借书记录和人员进行管理的设计目标。

5. 测试结果与分析

本程序的测试数据文件是book.txt、record.txt和user.txt。其中程序运行前的book.txt如下所示。

book.txt

电子电路与系统基础|李国林|大学教材|电子电路|本课程是对原“电路原理”“模拟电路”“通信电路”和“数字电路”等课程重构形成的新电路原理课程，体系架构为一条主干四个分支。|1

三国演义（上）|罗贯中|章回小说|三国演义|《三国演义》全名《三国志通俗演义》，其成书在元末明初，再经修改润色而在清代定型。它演述汉末晋初百年风云，开启了以演义小说诠释解构历史、进而与史书分庭抗礼的局面。|0

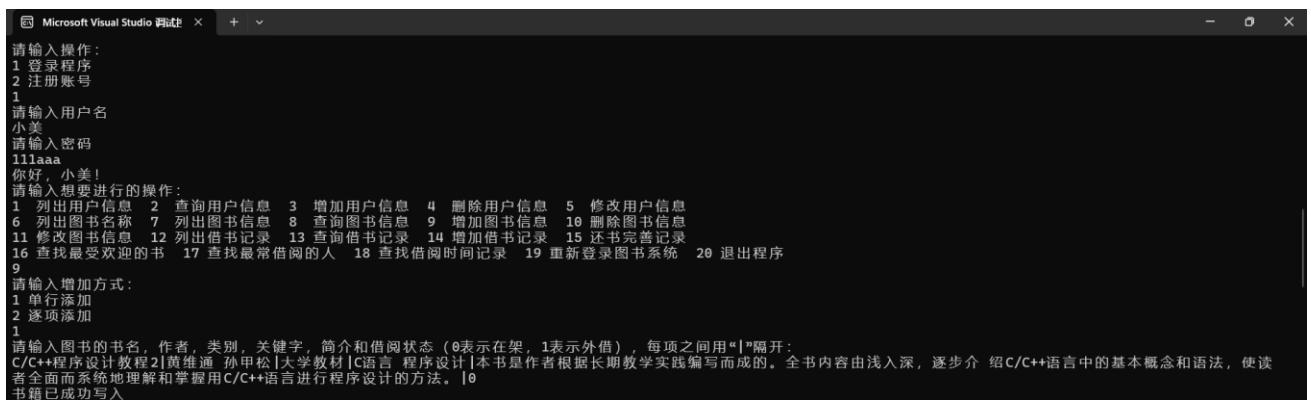
资本论（第一卷）|马克思|马克思主义|资本论|《资本论》是马克思毕生研究的成果和最主要的著作，第1卷于1867年由马克思亲自编辑出版。|0

红楼梦|曹雪芹 无名氏|章回小说|红楼梦|《红楼梦》诞生于清中叶，描写了封建末世勋戚显贵之家贾府由盛而衰的命运，以及贾宝玉与黛玉、宝钗的爱情婚姻悲剧，其思想和艺术都体现了中国古代小说的最高成就。|1

C/C++程序设计教程|黄维通 孙甲松|大学教材|C语言 程序设计|本书是作者根据长期教学实践编写而成的。全书内容由浅入深，逐步介绍C/C++语言中的基本概念和语法，使读者全面而系统地理解和掌握用C/C++语言进行程序设计的方法。|0

（每列为一行）

录入图书信息功能：



The screenshot shows a Microsoft Visual Studio window titled "Microsoft Visual Studio 调试" (Debug). The code editor contains the following text:

```
请输入操作：
1 登录程序
2 注册账号
3 ...
请输入用户名
小美
请输入密码
111aaa
你好，小美！
请输入想要进行的操作：
1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序
9 ...
请输入增加方式：
1 单行添加
2 逐项添加
1 ...
请输入图书的书名，作者，类别，关键字，简介和借阅状态（0表示在架，1表示外借），每项之间用“|”隔开：
C/C++程序设计教程2|黄维通 孙甲松|大学教材|C语言 程序设计|本书是作者根据长期教学实践编写而成的。全书内容由浅入深，逐步介绍C/C++语言中的基本概念和语法，使读者全面而系统地理解和掌握用C/C++语言进行程序设计的方法。|0
书籍已成功写入
```

列出图书信息功能：

```
Microsoft Visual Studio 调试 x + - x

请输入想要进行的操作:
1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序
7

name 电子电路与系统基础
author 李国林
type 大学教材
keywords 电子电路
intro 本课程是对原“电路原理”“模拟电路”“通信电路”和“数字电路”等课程重构形成的新电路原理课程，体系架构为一条主干四个分支。
status 外借

-----
name 三国演义（上）
author 罗贯中
type 章回小说
keywords 三国演义
intro 《三国演义》全名《三国志通俗演义》，其成书在元末明初，再经修改润色而在清代定型。它演述汉末晋初百年风云，开启了以演义小说诠释解构历史、进而与史书分庭抗礼的局面。
status 在架

-----
name 资本论（第一卷）
author 马克思
type 马克思主义
keywords 资本论
intro 《资本论》是马克思毕生研究的成果和最主要的著作，第1卷于1867年由马克思亲自编辑出版。
status 在架

-----
name 红楼梦
author 曹雪芹 无名氏
type 章回小说
keywords 红楼梦
intro 《红楼梦》诞生于清中叶，描写了封建末世勋戚显贵之家贾府由盛而衰的命运，以及贾宝玉与黛玉、宝钗的爱情婚姻悲剧，其思想和艺术都体现了中国古代小说的最高成就。
status 外借

-----
name C/C++程序设计教程
author 黄维通 孙甲松
type 大学教材
keywords C语言 程序设计
intro 本书是作者根据长期教学实践编写而成的。全书内容由浅入深，逐步介绍C/C++语言中的基本概念和语法，使读者全面而系统地理解和掌握用C/C++语言进行程序设计的方法。
status 外借

-----
name C/C++程序设计教程2
author 黄维通 孙甲松
type 大学教材
keywords C语言 程序设计
intro 本书是作者根据长期教学实践编写而成的。全书内容由浅入深，逐步介绍C/C++语言中的基本概念和语法，使读者全面而系统地理解和掌握用C/C++语言进行程序设计的方法。
status 在架
```

修改图书信息功能：

```
Microsoft Visual Studio 调试 x + - x

请输入想要进行的操作:
1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序
11
请输入修改书籍名称:
C/C++程序设计教程
请输入修改方式:
1 单行修改
2 逐项修改
1
请输入图书的书名, 作者, 类别, 关键字, 简介和借阅状态(0表示外借, 1表示在架), 每项之间用“|”隔开:
C/C++程序设计教程|黄维通|孙甲松|大学教材|C语言 程序设计|本书是作者根据长期教学实践编写而成的。全书内容由浅入深，逐步介绍C/C++语言中的基本概念和语法，使读者全面而系统地理解和掌握用C/C++语言进行程序设计的方法。|0
书籍已成功修改
请输入想要进行的操作:
1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序
7

name 电子电路与系统基础
author 李国林
type 大学教材
keywords 电子电路
intro 本课程是对原“电路原理”“模拟电路”“通信电路”和“数字电路”等课程重构形成的新电路原理课程，体系架构为一条主干四个分支。
status 外借

-----
name 三国演义（上）
author 罗贯中
type 章回小说
keywords 三国演义
intro 《三国演义》全名《三国志通俗演义》，其成书在元末明初，再经修改润色而在清代定型。它演述汉末晋初百年风云，开启了以演义小说诠释解构历史、进而与史书分庭抗礼的局面。
status 在架

-----
name 资本论（第一卷）
```

```

author 马克思
type 马克思主义
keywords 资本论
intro 《资本论》是马克思毕生研究的成果和最主要的著作，第1卷于1867年由马克思亲自编辑出版。
status 在架

-----
name 红楼梦
author 曹雪芹 无名氏
type 章回小说
keywords 红楼梦
intro 《红楼梦》诞生于清中叶，描写了封建末世勋戚显贵之家贾府由盛而衰的命运，以及贾宝玉与黛玉、宝钗的爱情婚姻悲剧，其思想和艺术都体现了中国古代小说的最高成就。
status 外借

-----
name C/C++程序设计教程
author 黄维通 孙甲松
type 大学教材
keywords C语言 程序设计
intro 本书是作者根据长期教学实践编写而成的。全书内容由浅入深，逐步介绍C/C++语言中的基本概念和语法，使读者全面而系统地理解和掌握用C/C++语言进行程序设计的方法。
status 在架

-----
name C/C++程序设计教程2
author 黄维通 孙甲松
type 大学教材
keywords C语言 程序设计
intro 本书是作者根据长期教学实践编写而成的。全书内容由浅入深，逐步介绍C/C++语言中的基本概念和语法，使读者全面而系统地理解和掌握用C/C++语言进行程序设计的方法。
status 在架

```

删除图书信息功能：

```

Microsoft Visual Studio 调试 x + v

请输入想要进行的操作：
1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序
10
请输入删除书籍名称：
C/C++程序设计教程
书籍已成功删除
请输入想要进行的操作：
1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序
7

name 电子电路与系统基础
author 李国林
type 大学教材
keywords 电子电路
intro 本课程是对原“电路原理”“模拟电路”“通信电路”和“数字电路”等课程重构形成的新电路原理课程，体系架构为一条主干四个分支。
status 外借

-----
name 三国演义（上）
author 罗贯中
type 章回小说
keywords 三国演义
intro 《三国演义》全名《三国志通俗演义》，其成书在元末明初，再经修改润色而在清代定型。它演述汉末晋初百年风云，开启了以演义小说诠释解构历史、进而与史书分庭抗礼的局面。
status 在架

-----
name 资本论（第一卷）
author 马克思
type 马克思主义
keywords 资本论
intro 《资本论》是马克思毕生研究的成果和最主要的著作，第1卷于1867年由马克思亲自编辑出版。
status 在架

-----
name 红楼梦
author 曹雪芹 无名氏
type 章回小说
keywords 红楼梦
intro 《红楼梦》诞生于清中叶，描写了封建末世勋戚显贵之家贾府由盛而衰的命运，以及贾宝玉与黛玉、宝钗的爱情婚姻悲剧，其思想和艺术都体现了中国古代小说的最高成就。
status 外借

-----
name C/C++程序设计教程2
author 黄维通 孙甲松
type 大学教材
keywords C语言 程序设计
intro 本书是作者根据长期教学实践编写而成的。全书内容由浅入深，逐步介绍C/C++语言中的基本概念和语法，使读者全面而系统地理解和掌握用C/C++语言进行程序设计的方法。
status 在架

```

列出图书书名：

请输入想要进行的操作：
1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序
6

电子电路与系统基础
三国演义（上）
资本论（第一卷）
红楼梦
C/C++程序设计教程2

程序运行前的record.txt如下面所示（随程序运行记录会发生变化）。

record.txt

0 小红 战争与和平 2024 3 8 2024 5 23
0 小美 论语 2024 4 3 2024 4 17
0 小明 水浒传 2024 6 22 2024 7 17
0 小亮 朝花夕拾 2024 5 30 2024 6 5
0 小美 论语 2024 5 2 2024 7 18
0 李华 论语 2024 7 19 2024 7 31
0 小明 朝花夕拾 2024 5 30
0 李华 西游记 2024 6 21
0 小美 西游记 2024 6 21
1 李华 电子电路与系统基础 2024 6 22

录入借书记录：

请输入想要进行的操作：
1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序
14
请输入增加方式：
1 单行添加
2 逐项添加
1
请输入借阅状态（0表示已归还，1表示未归还）、借阅人、书名、借书年月日和还书年月日（未归还不填写还书年月日），每项之间用“ ”隔开：
0 小红 论语 2024 1 3 2024 3 17
书籍已成功写入

还书更新记录（自动更新时间）：

```
请输入想要进行的操作:  
1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息  
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息  
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录  
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序  
15  
请输入归还书籍名称:  
西游记  
-----  
name      李华  
bookname  西游记  
status    未归还  
start date 2024.6.21  
-----  
是否完善此借书记录?  
1 是  
2 否  
2  
-----  
name      小美  
bookname  西游记  
status    未归还  
start date 2024.6.21  
-----  
是否完善此借书记录?  
1 是  
2 否  
1  
请输入还书日期输入方式:  
1 自动添加  
2 手动添加  
1  
书籍记录已更改完成
```

还书更新记录（手动更新时间）：

```
请输入归还书籍名称:  
西游记  
-----  
name      李华  
bookname  西游记  
status    未归还  
start date 2024.6.21  
-----  
是否完善此借书记录?  
1 是  
2 否  
1  
请输入还书日期输入方式:  
1 自动添加  
2 手动添加  
2  
请输入还书年月日:  
2024 9 1  
书籍记录已更改完成
```

查询借书记录:

```
请输入想要进行的操作:  
1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息  
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息  
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录  
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序  
13  
请输入查找方式:  
1 按书名查找  
2 按人名查找  
1  
请输入要查找的书名:  
论  
-----  
name      小美  
bookname  论语  
status    已归还  
start date 2024.4.3  
end date  2024.4.17  
-----  
name      小美  
bookname  论语  
status    已归还  
start date 2024.5.2  
end date  2024.7.18  
-----  
name      李华  
bookname  论语  
status    已归还  
start date 2024.7.19  
end date  2024.7.31  
-----  
共找到3个包含书名论的结果
```

列出借书记录:

```

请输入想要进行的操作:
1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序
12

-----
name    小红
bookname  战争与和平
status   已归还
start date 2024.3.8
end date  2024.5.23
-----


-----
name    小美
bookname  论语
status   已归还
start date 2024.4.3
end date  2024.4.17
-----


-----
name    小明
bookname  水浒传
status   已归还
start date 2024.6.22
end date  2024.7.17
-----


-----
name    小亮
bookname  朝花夕拾
status   已归还
start date 2024.5.30
end date  2024.6.5
-----


-----
name    小美
bookname  论语
status   已归还
start date 2024.5.2
end date  2024.7.18
-----


-----
name    李华
bookname  论语
status   已归还
start date 2024.7.19
end date  2024.7.31
-----


-----
name    小明
bookname  朝花夕拾
status   已归还
start date 2024.5.30
end date  2024.6.5
-----


-----
name    李华
bookname  西游记
status   已归还
start date 2024.6.22
end date  2024.7.20
-----


-----
name    李华
bookname  电子电路与系统基础
status   未归还
start date 2024.6.22
-----


-----
name    小红
bookname  论语
status   已归还
start date 2024.3.17
end date  2024.3.17
-----
```

查询最受欢迎的书

```

请输入想要进行的操作:
1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序
16

最受欢迎的书籍是 论语，被借阅了4 次。
```

查询最常借阅的读者

```

请输入想要进行的操作:
1 查询用户信息 2 修改用户信息 3 列出图书名称 4 列出图书信息 5 查询图书信息
6 列出借书记录 7 查询借书记录 8 查找最受欢迎的书 9 查找最常借阅的人
10 查找借阅时间记录 11 重新登录图书系统 12 退出程序
9

借阅最多的读者是 李华，小红，借阅了 3 次。
```

进行时间统计，查询借书数目随时间的变化：

请输入想要进行的操作：
1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序
18

日期 借阅数量
2024.1.3 1
2024.3.8 1
2024.4.3 1
2024.5.2 1
2024.5.30 2
2024.6.22 3
2024.7.19 1

借阅最多的日期是 2024.6.22，借阅了 3 本书。

程序运行前的user.txt如下面所示。

user.txt

0 2021010322 小红 abcd1234 21 f 学生 2021 8 31 2025 7 31
0 2022011231 小明 123456 20 m 学生 2022 8 31 2026 7 31
0 2022011231 小亮 123456 20 m 学生 2022 8 31 2026 7 31
1 2016019431 小美 111111 26 f 工作人员 2016 8 31 2036 7 31 图书归纳员 2016 8 31 2036 7 31

录入人员信息：

请输入想要进行的操作：
1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序
3
请输入增加内容：
1 使用者
2 管理员
2
请输入增加方式：
1 单行添加
2 逐项添加
1
请输入管理员工号、姓名、密码、年龄、性别（男“m”，女“f”）、身份、借书权限开始年月日、结束年月日、职位、管理员权限开始年月日和结束年月日，中间用“ ”隔开：
2012004267 李华 aaa111 30 m 工作人员 2012 8 31 2032 7 31 图书管理员 2012 8 31 2032 7 31
信息已成功写入

修改人员信息：

请输入想要进行的操作：
1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序
5
请输入要修改的人名：
小亮

type 使用者
name 小亮
age 22
sex 男
id number 2020013211
identity 学生
start date 2020.8.31
end date 2024.7.31
借书记录：
朝花夕拾 已归还, 2024.5.30 借出, 2024.6.5 归还
共有1条借书结果

请输入修改方式：
1 单行修改
2 逐项修改
1
请输入使用者工号（工号）、姓名、密码、年龄、性别（男“m”，女“f”）、身份、借书权限开始年月日和结束年月日，中间用“ ”隔开：
0 2822011231 小亮 123456 20 m 学生 2022 8 31 2026 7 31
输入错误，请重新输入
2022011231 小亮 123456 20 m 学生 2022 8 31 2026 7 31
用户小亮已成功修改

删除人员信息：

请输入想要进行的操作：
1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序
4
请输入要删除的人名：
李华
用户李华已成功删除

列出人员信息：

请输入想要进行的操作：

```

1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序
1
-----
```

type 使用者
name 小红
age 21
sex 女
id number 2021010322
identity 学生
start date 2021.8.31
end date 2025.7.31
借书记录：
战争与和平 已归还, 2024.3.8 借出, 2024.5.23 归还
论语 已归还, 2024.1.3 借出, 2024.3.17 归还
共有2条借书结果

type 使用者
name 小明
age 20
sex 男
id number 2022011231
identity 学生
start date 2022.8.31
end date 2026.7.31
借书记录：
水浒传 已归还, 2024.6.22 借出, 2024.7.17 归还
朝花夕拾 已归还, 2024.5.30 借出, 2024.6.5 归还
共有2条借书结果

type 使用者
name 小亮
age 20
sex 男
id number 2022011231
identity 学生
start date 2022.8.31
end date 2026.7.31
借书记录：
朝花夕拾 已归还, 2024.5.30 借出, 2024.6.5 归还
共有1条借书结果

type 管理员
name 小美
age 26
sex 女
id number 2016019431
identity 工作人员
user start 2016.8.31
user end 2036.7.31
position 图书归纳员
admin start 2016.8.31
admin end 2036.7.31
借书记录：
论语 已归还, 2024.4.3 借出, 2024.4.17 归还
论语 已归还, 2024.5.2 借出, 2024.7.18 归还
共有2条借书结果

查询人员信息：

请输入想要进行的操作：

```

1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序
2
请输入要查找的人名：
小美
-----
```

type 管理员
name 小美
age 26
sex 女
id number 2016019431
identity 工作人员
user start 2016.8.31
user end 2036.7.31
position 图书归纳员
admin start 2016.8.31
admin end 2036.7.31
借书记录：
论语 已归还, 2024.4.3 借出, 2024.4.17 归还
论语 已归还, 2024.5.2 借出, 2024.7.18 归还
共有2条借书结果

找到1条包含人名小美的结果

用户注册：

```

-----  

欢迎使用图书管理系统！  

2024年9月6日 10:22:17
-----
```

```
请输入操作：
1 登录程序
2 注册账号
2
请输入增加内容：
1 使用者
2 管理员
2
请输入增加方式：
1 单行添加
2 逐项添加
1
请输入管理员工号、姓名、密码、年龄、性别（男“m”，女“f”）、身份、借书权限开始年月日、结束年月日、职位、管理员权限开始年月日和结束年月日，中间用“ ”隔开：
20120904267 李华 aaa111 38 m 工作人员 2012 8 31 2032 7 31 图书管理员 2012 8 31 2032 7 31
信息已成功写入。
请输入想要进行的操作：
1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序
```

用户登录：

```
欢迎使用图书管理系统！
2024年9月6日 10:28:50
-----
请输入操作：
1 登录程序
2 注册账号
1
请输入用户名
小美
请输入密码
111111
你好，小美！
```

退出程序：

```
请输入想要进行的操作：
1 列出用户信息 2 查询用户信息 3 增加用户信息 4 删除用户信息 5 修改用户信息
6 列出图书名称 7 列出图书信息 8 查询图书信息 9 增加图书信息 10 删除图书信息
11 修改图书信息 12 列出借书记录 13 查询借书记录 14 增加借书记录 15 还书完善记录
16 查找最受欢迎的书 17 查找最常借阅的人 18 查找借阅时间记录 19 重新登录图书系统 20 退出程序
20
谢谢使用，再见，李华！
```

6. 总结

感谢老师和助教学长学姐们在暑期的小学期中给出的上机实践题目，让我在编写程序的过程中对C++面向对象的编程思想有了更加深入的认识。题目中涉及的图书管理系统对C++知识考察较为全面，抽象类、多重继承、虚函数的多态性和运算符重载等课程中的重点在本题中都有涉及。

在此次实验中，我成功搭建了图书管理系统，用人员抽象类和借书记录类派生出了使用者类和管理员类，通过抽象类和多重继承实现了对人员信息的管理。在录入和显示人员、图书以及借书记录信息时，程序利用了print()函数的多态性，并对流操作符“<<”和“>>”进行了重载，成功实现了信息的输入文件和屏幕显示。经过调试，程序可以实现对书籍信息、借书记录和人员信息的增加、删除、显示和修改操作，并可根据登录程序的用户类别（使用者或管理员）匹配相应的可执行操作。

在创新方面，本程序主要有以下五个创新点：①创造性设计了登录系统的功能。用户可以注册使用者账号，也可以通过已有账号登录系统。登录之后，系统会读取用户是使用者还是管理员的信息，并匹配与之对应的操作。管理员可以对书、借书记录和人员信息进行增加、删除、查询和修改，但使用者只能进行查看操作。②引入切换用户名的机制。当一个用户使用完程序之后，他可以选择退出账号而不退出程序，之后程序可以使用下一位用户的信息进行登录。这样更符合用户的操作习惯，提升程序的运行效率。③在录入信息时提供了逐项录入功能，该功能可以避免用户在整行输入时带来的错误风险，提高输入的准确性。④提供了详尽的异常处理功能。例如程序通过用户输入数字进行对应操作，如果用户输入数字错误，则程序会提示重新输入。另外如果程序在打开文件出现异常时，也会进行提示并返回界面。这些功能提升了程序的稳健性，使其更加适合使用。⑤在程序中引入了系统时间功能，可以在程序运行时实时显示系统时间，并可选择在还书时自动输入还书时间。

总而言之，这次的程设实践让我对C++语言有了更为深入的认识，对进行较大规模的程序写作也有了更多的经验。我相信，这次实践一定会为我之后学习编程语言筑牢基础，对我用编程语言解决实际问题提供更大帮助。在此对老师和学长学姐们诚挚地说一声：您辛苦了！

附录 1：源程序清单

main.cpp

主函数所在文件

```
#include "action.h"
#include "book.h"
#include "bookdate.h"
#include "people.h"
#include "record.h"
#include <iostream>
#include <chrono>
#include <ctime>
#pragma warning(disable:4996)
using namespace std;

int main()
{
    int flag1 = -1, flag2 = 0, action;
    string username;
    start:
    cout << "-----" << endl;
    cout << "欢迎使用图书管理系统！" << endl;
    auto now = chrono::system_clock::now(); // 获取当前时间点
    time_t now_c = chrono::system_clock::to_time_t(now); // 转换为time_t类型
    tm* parts = localtime(&now_c); // 使用localtime获取本地时间
    cout << 1900 + parts->tm_year << "年" << 1 + parts->tm_mon << "月" <<
parts->tm_mday << "日" // 月份中的日期
        << parts->tm_hour << ":" << parts->tm_min << ":" << parts-
>tm_sec << std::endl; // 秒
    cout << "-----" << endl; // 打印日期和时间
    while (flag1 == -1) // 登录界面
    {
        flag1 = userlogin(username);
    }
    while (!flag2) // 分析用户行为
    {
        if (flag1 == -2)
        {
            flag2 = 1;
        }
        else if (flag1 == 1) // admin
```

```

{
    action = useraction(1);
    if (action == 1)listuser();
    else if (action == 2)searchuser();
    else if (action == 3)adduser();
    else if (action == 4)deleteuser();
    else if (action == 5)changeuser();
    else if (action == 6)listbookname();
    else if (action == 7)listbook();
    else if (action == 8)searchbook();
    else if (action == 9)addbook();
    else if (action == 10)deletebook();
    else if (action == 11)changebook();
    else if (action == 12)listrecord();
    else if (action == 13)searchrecord();
    else if (action == 14)addrecord();
    else if (action == 15)returnrecord();
    else if (action == 16)popular(0);
    else if (action == 17)popular(1);
    else if (action == 18)recordtendency();
    else if (action == 19) { flag1 = -1;flag2 = 0;goto
start; }
    else flag2 = 1;
}
else if (flag1 == 0)//user
{
    action = useraction(0);
    if (action == 1)_searchuser(username);
    else if (action == 2)_changeuser(username);
    else if (action == 3)listbookname();
    else if (action == 4)listbook();
    else if (action == 5)searchbook();
    else if (action == 6)listrecord();
    else if (action == 7)searchrecord();
    else if (action == 8)popular(0);
    else if (action == 9)popular(1);
    else if (action == 10)recordtendency();
    else if (action == 11) { flag1 = -1;flag2 = 0;goto
start; }
    else flag2 = 1;
}
cout << "谢谢使用，再见，" << username << "!" << endl;

```

```

    return 0;
}

action.h
获取数字和信息的函数
#pragma once
#include <iostream>
#ifndef __ACTION__
#define __ACTION__
using namespace std;

//获取用户输入数字
int getint(int arr[], int size);

//获取用户想要进行的操作，0代表user，1代表admin
int useraction(int type); //type为0或1
#endif // !__ACTION__

action.cpp
获取数字和信息的函数
#pragma once
#ifndef __ACTION__
#include "action.h"
#endif // !__ACTION__

#include <iostream>
using namespace std;

int getint(int arr[], int size)
{
    int i, j, flag = 0;
    cin >> i;
    if (cin.fail()) flag = 0;
    else
    {
        for (j = 0; j < size; j++)
        {
            if (i == arr[j]) flag = 1;
        }
    }
    while (!flag)
    {

```

```

        cin.clear();
        cin.ignore(std::numeric_limits<streamsize>::max(), '\n');
        cout << "输入错误, 请重新输入" << endl;
        cin >> i;
        if (cin.fail())flag = 0;
        else
        {
            for (j = 0;j < size;j++)
            {
                if (i == arr[j])flag = 1;
            }
        }
    }
    return i;
}

//获取用户想要进行的操作, 0代表user, 1代表admin
int useraction(int type)//type为0或1
{
    int i = 0;
    if (type)//admin
    {
        int arr[] =
{ 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20 };
        cout << "请输入想要进行的操作: " << endl;
        cout << "1 列出用户信息" << " "; //listuser
        cout << "2 查询用户信息" << " "; //searchuser
        cout << "3 增加用户信息" << " "; //adduser
        cout << "4 删除用户信息" << " "; //deleteuser
        cout << "5 修改用户信息" << endl; //changeuser
        cout << "6 列出图书名称" << " "; //listbookname
        cout << "7 列出图书信息" << " "; //listbook
        cout << "8 查询图书信息" << " "; //searchbook
        cout << "9 增加图书信息" << " "; //addbook
        cout << "10 删除图书信息" << endl; //deletebook
        cout << "11 修改图书信息" << " "; //changebook
        cout << "12 列出借书记录" << " "; //listrecord
        cout << "13 查询借书记录" << " "; //searchrecord
        cout << "14 增加借书记录" << " "; //addrecord
        cout << "15 还书完善记录" << endl; //returnrecord
        cout << "16 查找最受欢迎的书" << " "; //popular(0)
    }
}

```

```

        cout << "17 查找最常借阅的人" << " ";//popular(1)
        cout << "18 查找借阅时间记录" << " ";//recordtendency();
        cout << "19 重新登录图书系统" << " ";
        cout << "20 退出程序" << endl;
        i = getInt(arr, 20);
    }
} else//user
{
    int arr[] = { 1,2,3,4,5,6,7,8,9,10,11,12 };
    cout << "请输入想要进行的操作: " << endl;
    cout << "1 查询用户信息" << " ";//_searchuser(username)
    cout << "2 修改用户信息" << " ";//_changeuser(username)
    cout << "3 列出图书名称" << " ";//listbookname
    cout << "4 列出图书信息" << " ";//listbook
    cout << "5 查询图书信息" << endl;//searchbook
    cout << "6 列出借书记录" << " ";//listrecord
    cout << "7 查询借书记录" << " ";//searchrecord
    cout << "8 查找最受欢迎的书" << " ";//popular(0)
    cout << "9 查找最常借阅的人" << endl;//popular(1)
    cout << "10 查找借阅时间记录" << " ";//recordtendency();
    cout << "11 重新登录图书系统" << " ";
    cout << "12 退出程序" << endl;

    i = getInt(arr, 12);
}
return i;
}


```

book.h

书籍类相关函数

```

#include <iostream>
#include <fstream>
#include <sstream>

#ifndef __BOOK__
#define __BOOK__
using namespace std;

```

```
class book
```

```
{
```

```
public:
```

```

//book() {}
//~book() {}
void init(string line)
{
    istringstream iss(line);
    string b;
    if (!getline(iss, name, '|')) { cout << "Not enough tokens."
<< endl; }
    if (!getline(iss, author, '|')) { cout << "Not enough tokens."
<< endl; }
    if (!getline(iss, type, '|')) { cout << "Not enough tokens."
<< endl; }
    if (!getline(iss, keywords, '|')) { cout << "Not enough
tokens." << endl; }
    if (!getline(iss, intro, '|')) { cout << "Not enough tokens."
<< endl; }
    if (!getline(iss, b, '|')) { cout << "Not enough tokens." <<
endl; }
    if (b == "1")status = true;
    else status = false;
};
void init(string n, string a, string t, string k, string i, bool s)
{
    if (!n.empty()) { name = n; }
    if (!a.empty()) { author = a; }
    if (!t.empty()) { type = t; }
    if (!k.empty()) { keywords = k; }
    if (!i.empty()) { intro = i; }
    status = s;
};
void print()//打印
{
    cout << "-----" << endl;
    cout << "name      " << name << endl;
    cout << "author     " << author << endl;
    cout << "type      " << type << endl;
    cout << "keywords   " << keywords << endl;
    cout << "intro      " << intro << endl;
    if (status)cout << "status      " << "外借" << endl;
    else cout << "status      " << "在架" << endl;
    cout << "-----" << endl;
}

```

```

int searchbookname(string search) const//查找书名
{
    int flag = 0;
    size_t found = name.find(search);
    if (found != std::string::npos)
    {
        flag = 1;
    }
    return flag;
}
int searchbookall(string search) const//查找书籍全部内容
{
    int flag = 0;
    size_t found_n = name.find(search), found_a =
author.find(search), found_t = type.find(search), found_k =
type.find(search), found_i = intro.find(search);
    if (found_n != std::string::npos || found_a != std::string::npos || found_t != std::string::npos || found_k != std::string::npos || found_i != std::string::npos)
    {
        flag = 1;
    }
    return flag;
}
friend ostream& operator << (ostream& output, book& B); //输出算符重载
private:
    string name;//书名
    string author;//作者
    string type;//类型
    string keywords;//关键词
    string intro;//简介
    bool status;//借阅状态（0表示在架，1表示外借）
};

//查找书目
void searchbook();

//增加书目
void addbook();

//删除数目
void deletebook();

```

```

//列出书名
void listbookname();

//列出书籍
void listbook();

//改变书籍信息
void changebook();

#endif
// !__BOOK__

book.cpp
书籍类相关函数
#pragma once
#ifndef __BOOK__
#include "book.h"
#endif // !__BOOK__
#ifndef __ACTION__
#include "action.h"
#endif // !__ACTION__
#include <fstream>
#include <iostream>
#include <sstream>
using namespace std;

int recordtype = 0;//0表示正常输出，1表示在查询人时输出
int usertype = 0;//0表示从文件中向计算机中读取，1表示从键盘读取

ostream& operator << (ostream& output, book& B)
{
    output << "-----" << endl;
    output << "name" << B.name << endl;
    output << "author" << B.author << endl;
    output << "type" << B.type << endl;
    output << "keywords" << B.keywords << endl;
    output << "intro" << B.intro << endl;
    if (B.status)output << "status" << "外借" << endl;
    else output << "status" << "在架" << endl;
    output << "-----" << endl;
    return output;
}

```

```
}

//查找书目
void searchbook()
{
    book* b = new book;
    string line, search;
    int count = 0, i, arr[] = { 1,2 };
    cout << "请输入查找方式: " << endl;
    cout << "1 书名查找" << endl;
    cout << "2 全部信息查找" << endl;
    i = getInt(arr, 2);
    ifstream infile("book.txt", ios::in | ios::_Nocreate);
    if (i == 1)
    {
        cout << "请输入要查找的书名: " << endl;
        cin >> search;
        while (!infile.eof())
        {
            getline(infile, line);
            if (line.empty()) break;
            b->init(line);
            if (b->searchbookname(search))
            {
                count++;
                cout << *b;
            }
        }
    }
    else//i==2
    {
        cout << "请输入要查找的全部信息: " << endl;
        cin >> search;
        while (!infile.eof())
        {
            getline(infile, line);
            if (line.empty()) break;
            b->init(line);
            if (b->searchbookall(search))
            {
                count++;
                cout << *b;
            }
        }
    }
}
```

```

        }
    }
    if (count)cout << "共找到" << count << "个包含" << search << "的结果"
<< endl;
    else cout << "未找到包含" << search << "的结果" << endl;
    infile.close();
    delete b;
}
//增加书目
void addbook()
{
    int i, arr[] = { 1,2 };
    cout << "请输入增加方式: " << endl;
    cout << "1 单行添加" << endl;
    cout << "2 逐项添加" << endl;
    i = getInt(arr, 2);
    while (i != 1 && i != 2)
    {
        cout << "输入错误, 请重新输入" << endl;
        cin >> i;
    }
    cin.clear();
    cin.ignore(std::numeric_limits<streamsize>::max(), '\n');
    if (i == 1)
    {
        string line;
        cout << "请输入图书的书名, 作者, 类别, 关键字, 简介和借阅状态(0表示在架, 1表示外借), 每项之间用“|”隔开: " << endl;
        getline(cin, line);
        ofstream outfile("book.txt", ios::app | ios::ate |
ios::_Nocreate);
        outfile << line << endl;
        outfile.close();
    }
    else//i==2
    {
        string name;//书名
        string author;//作者
        string type;//类型
        string keywords;//关键词
    }
}

```

```

    string intro;//简介
    int status;//借阅状态
    cout << "请输入图书的书名: " << endl;
    getline(cin, name);
    cout << "请输入图书的作者: " << endl;
    getline(cin, author);
    cout << "请输入图书的类别: " << endl;
    getline(cin, type);
    cout << "请输入图书的关键字: " << endl;
    getline(cin, keywords);
    cout << "请输入图书的简介: " << endl;
    getline(cin, intro);
    cout << "请输入图书的借阅状态(0表示在架, 1表示外借): " << endl;
    cin >> status;
    while (cin.fail() || status != 0 || status != 1)
    {
        cout << "借阅状态输入错误, 请重新输入" << endl;
        cin.clear();
        cin.ignore(std::numeric_limits<streamsize>::max(), '\n');
        cin >> status;
    }
    ofstream outfile("book.txt", ios::app | ios::ate |
ios::_Nocreate);
    outfile << name << "|" << author << "|" << type << "|" <<
keywords << "|" << intro << "|" << status << endl;
    outfile.close();
}
cout << "书籍已成功写入" << endl;
}

//删除数目
void deletebook()
{
    string line, bookname, name;
    int flag = 0, flag2 = 0;
    ifstream infile("book.txt", ios::in | ios::_Nocreate);
    ofstream outfile("_book_.txt", ios::out | ios::app);
    cout << "请输入删除书籍名称: " << endl;
    cin >> name;
    while (!infile.eof())
    {
        getline(infile, line);
        if (line.empty())break;

```

```

    istringstream iss(line);
    if (!getline(iss, bookname, '|')) {
        cout << "Not enough tokens." << endl;
    }
    if (bookname == name) { flag = 1; }
    else { outfile << line << endl; }
}
infile.close();
outfile.close();
remove("book.txt");
flag2 = rename("_book_.txt", "book.txt");
if (flag && !flag2) cout << "书籍已成功删除" << endl;
else cout << "抱歉，不存在该书籍，无法删除" << endl;
}

void listbookname()
{
    ifstream infile("book.txt", ios::in | ios::_Nocreate);
    string line, name;
    cout << "-----" << endl;
    while (!infile.eof())
    {
        getline(infile, line);
        if (line.empty())break;
        istringstream iss(line);
        if (!getline(iss, name, '|')) {
            cout << "Not enough tokens." << endl;
        }
        cout << name << endl;
    }
    cout << "-----" << endl;
    infile.close();
}

void listbook()
{
    book* b = new book;
    ifstream infile("book.txt", ios::in | ios::_Nocreate);
    string line;
    while (!infile.eof())
    {
        getline(infile, line);
        if (line.empty())break;

```

```

        b->init(line);
        cout << *b;
    }
    infile.close();
    delete b;
}

//string修改到了这里
void changebook()
{
    string line, line2, bookname, name;
    int flag = 0, flag2 = 0;
    ifstream infile("book.txt", ios::in | ios::_Nocreate);
    ofstream outfile("_book_.txt", ios::out | ios::app);
    cout << "请输入修改书籍名称: " << endl;
    cin >> bookname;
    while (!infile.eof())
    {
        getline(infile, line);
        if (line.empty())break;
        istringstream iss(line);
        if (!getline(iss, name, '|')) { cout << "Not enough tokens."
<< endl; }
        if (bookname == name)
        {
            flag = 1;
            int i, a[] = { 1,2 };
            cout << "请输入修改方式: " << endl;
            cout << "1 单行修改" << endl;
            cout << "2 逐项修改" << endl;
            i = getInt(a, 2);
            if (i == 1)
            {
                cout << "请输入图书的书名, 作者, 类别, 关键字, 简介和借
阅状态 (0表示外借, 1表示在架), 每项之间用“|”隔开: " << endl;
                cin.clear();
                cin.ignore(std::numeric_limits<streamsize>::max(),
'\n');
                getline(cin, line);
                outfile << line << endl;
            }
            else//i==2

```

```

{
    string name;//书名
    string author;//作者
    string type;//类型
    string keywords;//关键词
    string intro;//简介
    int status;//借阅状态
    cout << "请输入图书的书名: " << endl;
    cin >> name;
    cout << "请输入图书的作者: " << endl;
    cin >> author;
    cout << "请输入图书的类别: " << endl;
    cin >> type;
    cout << "请输入图书的关键字: " << endl;
    cin >> keywords;
    cout << "请输入图书的简介: " << endl;
    cin >> intro;
    cout << "请输入图书的借阅状态 (0表示在架, 1表示外借) : "
<< endl;
    cin >> status;
    while (cin.fail() || status != 0 || status != 1)
    {
        cout << "借阅状态输入错误, 请重新输入" << endl;
        cin.clear();
        cin.ignore(std::numeric_limits<streamsize>::max(), '\n');
        cin >> status;
    }
    ofstream outfile("book.txt", ios::app | ios::ate |
ios::nocreate);
    outfile << name << "|" << author << "|" << type <<
"|" << keywords << "|" << intro << "|" << status << endl;
}
}
else
{
    outfile << line << endl;
}
}
infile.close();
outfile.close();

```

```

remove("book.txt");
rename("_book_.txt", "book.txt");
if (flag) cout << "书籍已成功修改" << endl;
else cout << "抱歉，不存在书籍" << bookname << ", 故未作修改" << endl;
}

```

bookdate.h

日期类相关函数

```

#pragma once
#include <iostream>

#ifndef __BOOKDATE__
#define __BOOKDATE__
using namespace std;

class date//日期类
{
public:
    void init(int y, int m, int d) { year = y;month = m;day = d; } //初始化
    int showy() { return year; }
    int showm() { return month; }
    int showd() { return day; }
    int GetYear() { return year; }
    void print() const//打印
    {
        bool i = month < 10, j = day < 10;
        cout << year << "." << month << "." << day;
        if (i && j)cout << " ";
        else if (i || j)cout << " ";
        else;
    }
    friend istream& operator >> (istream& input, date& D); //输入算符重载
    friend ostream& operator << (ostream& output, date& D); //输出算符重载
    const bool operator> (const date& D) const
    {
        if (year > D.year) return true;
        else if (year < D.year) return false;
        else if (month > D.month) return true;
        else if (month < D.month) return false;
        else return day > D.day;
    }
}

```

```

    const bool operator<(const date& D) const
    {
        if (year < D.year) return true;
        else if (year > D.year) return false;
        else if (month < D.month) return true;
        else if (month > D.month) return false;
        else return day < D.day;
    }
private:
    int year, month, day;
};

#endif // !_BOOKDATE_

```

bookdate.cpp

日期类相关函数

```

#pragma once
#ifndef _BOOKDATE_
#include "bookdate.h"
#endif // !_BOOKDATE_

```

```

#include <iostream>
using namespace std;

//date
istream& operator >> (istream& input, date& D)
{
    input >> D.year >> D.month >> D.day;
    return input;
}

```

//日期中间有空格，两端没有

```

ostream& operator << (ostream& output, date& D)
{
    output << D.year << " " << D.month << " " << D.day;
    return output;
}

```

people.h

人类、使用者类和管理

员类相关函数

```
#pragma once
```

```

#ifndef __RECORD__
#include "record.h"
#endif // !__RECORD__

#include <iostream>
#include <vector>

#ifndef __PEOPLE__
#define __PEOPLE__
using namespace std;
class people//人类
{
public:
    virtual void print()const = 0;//打印
    int searchusername(string search) const;//按人名查找
    int searchpw(string search) const;//查找密码
    string getname() const { return username; };//获取姓名
    string getpw() const { return pw; };//获取姓名
protected:
    string username;//姓名
    string pw;//密码
    int age;//年龄
    char sex;//性别 m-male f-female
};

class recordarray
{
public:
    virtual void print();
    recordarray() {};
    recordarray(const vector<record>& init) : data(init) {}
    void append(record value) { data.push_back(value); }const// 获取数组
大小
    int size() const { return data.size(); }// 获取数组元素
    record operator[](int index) const { return data[index]; }
protected:
    vector<record> data;// 私有成员变量，用于存储整数
};

class user :public people, public recordarray//使用者类，记录数组类
{

```

```

public:
    virtual void print()const;//打印
    void printbookrecord()const;//打印借书记录
    friend istream& operator >> (istream& input, user& U);//输入算符重载
    friend ostream& operator << (ostream& output, user& U);//输出算符重载
protected:
    int idnum;//学生证或工作证号idnumber
    string identity;//使用者的身份 学生/教师/工作人员 至多13个汉字
    date ustart;//借书权限开始日期
    date uend;//借书权限结束日期
};

class admin :public user//管理员类
{
public:
    virtual void print()const;
    friend istream& operator >> (istream& input, admin& A);//输入算符重载
    friend ostream& operator << (ostream& output, admin& A);//输出算符重载
protected:
    string position;//管理系统中的职位 至多13个汉字
    date astart;//管理员权限开始日期
    date aend;//管理员权限结束日期
};

//列出用户信息
void listuser();

//1 单行添加,2 逐项添加
void _adduser(int type, ofstream& outfile);

void _addadmin(int type, ofstream& outfile);

//增加用户信息
int adduser();

//查找record.txt中用户信息
void _searchuser(string username);

//查找record.txt中用户信息
void searchuser();

```

```
//删除用户信息
void deleteuser();

void _changeuser(string username);

//改变用户
void changeuser();

//用户登录和注册
int userlogin(string& n);

#endif // !_PEOPLE_

people.cpp
人类、使用者类和管理
员类相关函数
#pragma once
#ifndef __PEOPLE__
#include "people.h"
#endif // !_PEOPLE__

#ifndef __ACTION__
#include "action.h"
#endif // !_ACTION__

#include <iostream>
using namespace std;
extern int recordtype;
extern int usertype;

//people
int people::searchpw(string search) const
{
    int flag = 0;
    if (pw == search) { flag = 1; }
    return flag;
}

int people::searchusername(string search) const
{
    int flag = 0;
```

```

        size_t found = username.find(search);
        if (found != std::string::npos) { flag = 1; }
        return flag;
    }

//recordarray
void recordarray::print()
{
    int i = 0, size = data.size();
    for (i = 0;i < size;i++) { cout << data[i]; }
}
//user
void user::print()const
{
    cout << "-----" << endl;
    cout << "type      使用者" << endl;
    cout << "name      " << username << endl;
    cout << "age       " << age << endl;
    if (sex == 'm')cout << "sex      男" << endl;
    else cout << "sex      女" << endl;
    cout << "id number  " << idnum << endl;
    cout << "identity   " << identity << endl;
    cout << "start date  ";
    ustart.print();
    cout << endl;
    cout << "end date   ";
    uend.print();
    cout << endl;
    printbookrecord();
    cout << "-----" << endl;
}

void user::printbookrecord() const
{
    record* r = new record;
    recordarray ra;
    string line, uname = username;
    int count = 0;
    ifstream infile("record.txt", ios::in | ios::_Nocreate);
    cout << "借书记录: " << endl;
    while (!infile.eof())
    {

```

```

getline(infile, line);
if (line.empty())break;
istringstream iss(line);
iss >> *r;
if (r->searchusername(uname))
{
    count++;
    recordtype = 1;
    ra.append(*r);
}
ra.print();
recordtype = 0;
infile.close();
if (count)cout << "共有" << count << "条借书结果" << endl;
else cout << "未找到借书结果" << endl;
delete r;
}

istream& operator >> (istream& input, user& U)
{
    if (usertype)input >> U.idnum >> U.username >> U.pw >> U.age >>
U.sex >> U.identity >> U.ustart >> U.uend;
    else
    {
        char junk;
        input >> junk >> U.idnum >> U.username >> U.pw >> U.age >>
U.sex >> U.identity >> U.ustart >> U.uend;
    }
    return input;
}

ostream& operator << (ostream& output, user& U)
{
    output << "0 " << U.idnum << " " << U.username << " " << U.pw << " "
<< U.age << " " << U.sex << " " << U.identity << " " << U.ustart << " "
<< U.uend;
    return output;
}
//admin

void admin::print()const

```

```

{
    cout << "-----" << endl;
    cout << "type      管理员" << endl;
    cout << "name      " << username << endl;
    cout << "age       " << age << endl;
    if (sex == 'm') cout << "sex      男" << endl;
    else cout << "sex      女" << endl;
    cout << "id number  " << idnum << endl;
    cout << "identity   " << identity << endl;
    cout << "user start  ";
    ustart.print();
    cout << endl;
    cout << "user end    ";
    uend.print();
    cout << endl;
    cout << "position    " << position << endl;
    cout << "admin start ";
    astart.print();
    cout << endl;
    cout << "admin end    ";
    aend.print();
    cout << endl;
    printbookrecord();
    cout << "-----" << endl;
}

```

```

istream& operator >> (istream& input, admin& A)
{
    if (usertype)input >> A.idnum >> A.username >> A.pw >> A.age >>
A.sex >> A.identity >> A.ustart >> A.uend >> A.position >> A.astart >>
A.aend;
    else
    {
        char junk;
        input >> junk >> A.idnum >> A.username >> A.pw >> A.age >>
A.sex >> A.identity >> A.ustart >> A.uend >> A.position >> A.astart >>
A.aend;
    }
    return input;
}

ostream& operator << (ostream& output, admin& A)

```

```

{
    output << "1 " << A.idnum << " " << A.username << " " << A.pw << " "
<< A.age << " " << A.sex << " " << A.identity << " " << A.ustart << " "
<< A.uend << " " << A.position << " " << A.astart << " " << A.aend;
    return output;
}

void listuser()
{
    ifstream infile("user.txt", ios::in | ios::_Nocreate);
    string line;
    while (!infile.eof())
    {
        getline(infile, line);
        if (line.empty())break;
        istringstream iss(line);
        if (line[0] == '0')//使用者
        {
            user u;
            people* p = &u;
            iss >> u;
            p->print();
        }
        else//管理员
        {
            admin a;
            people* p = &a;
            iss >> a;
            p->print();
        }
    }
    infile.close();
}

//1 单行添加,2 逐项添加
void _adduser(int type, ofstream& outfile)
{
    if (type == 1)//先扫入再写入，避免输入格式不对
    {
        string line;
        int flag = 0;
        user* u = new user;

```

```

cout << "请输入使用者学号(工号)、姓名、密码、年龄、性别(男“m”，女
“f”)、身份、借书权限开始年月日和结束年月日，中间用“ ”隔开: " << endl;
    cin.clear();
    cin.ignore(std::numeric_limits<streamsize>::max(), '\n');
    getline(cin, line);
    istringstream iss(line);
    usertype = 1;
    iss >> *u;
    if (iss.fail())flag = 1;
    while (flag)
    {
        iss.clear();
        iss.ignore(std::numeric_limits<streamsize>::max(), '\n');
        cout << "输入错误，请重新输入" << endl;
        getline(cin, line);
        istringstream iss(line);
        iss >> *u;
        if (iss.fail())flag = 1;
        else flag = 0;
    }
    usertype = 0;
    outfile << *u << endl;
    delete u;
}
else
{
    int idnum, age;//学生证或工作证号idnumber
    char sex;
    string username, pw, identity;//使用者的身份 学生/教师/工作人员
至多13个汉字
    date ustard, uend;
    cout << "请输入使用者学号(工号): " << endl;
    cin >> idnum;
    cout << "请输入使用者姓名: " << endl;
    cin >> username;
    cout << "请输入使用者密码: " << endl;
    cin >> pw;
    cout << "请输入使用者年龄: " << endl;
    cin >> age;
    cout << "请输入使用者性别(男“m”，女“f”): " << endl;
    cin >> sex;
    cout << "请输入使用者身份: " << endl;
}

```

```

    cin >> identity;
    cout << "请输入使用者借书权限开始年月日: " << endl;
    cin >> ustart;
    cout << "请输入使用者借书权限结束年月日: " << endl;
    cin >> uend;
    outfile << "0 " << idnum << " " << username << " " << pw << "
" << age << " " << sex << " " << identity << " " << ustart << " " << uend
<< endl;
}
}

void _addadmin(int type, ofstream& outfile)
{
    if (type == 1)
    {
        string line;
        int flag = 0;
        admin* a = new admin;
        cout << "请输入管理员工号、姓名、密码、年龄、性别（男“m”，女“f”）、身份、借书权限开始年月日、结束年月日、职位、管理员权限开始年月日和结束年月日，中间用“ ”隔开: " << endl;
        cin.clear();
        cin.ignore(std::numeric_limits<streamsize>::max(), '\n');
        getline(cin, line);
        istringstream iss(line);
        usertype = 1;
        iss >> *a;
        if (iss.fail())flag = 1;
        while (flag)
        {
            iss.clear();
            iss.ignore(std::numeric_limits<streamsize>::max(), '\n');
            cout << "输入错误，请重新输入" << endl;
            getline(cin, line);
            istringstream iss(line);
            iss >> *a;
            if (iss.fail())flag = 1;
            else flag = 0;
        }
        usertype = 0;
        outfile << *a << endl;
        delete a;
    }
}

```

```

    }
else
{
    int idnum, age;//学生证或工作证号idnumber
    char sex;
    string username, pw, identity, position; // 使用者的身份 学生 /
教师 / 工作人员
    date ustart, uend, astart, aend;//借书权限开始日期
    cout << "请输入管理员学号(工号): " << endl;
    cin >> idnum;
    cout << "请输入管理员姓名: " << endl;
    cin >> username;
    cout << "请输入管理员密码: " << endl;
    cin >> pw;
    cout << "请输入管理员年龄: " << endl;
    cin >> age;
    cout << "请输入管理员性别(男“m”，女“f”): " << endl;
    cin >> sex;
    cout << "请输入管理员身份: " << endl;
    cin >> identity;
    cout << "请输入管理员借书权限开始年月日: " << endl;
    cin >> ustart;
    cout << "请输入管理员借书权限结束年月日: " << endl;
    cin >> uend;
    cout << "请输入管理员职位: " << endl;
    cin >> position;
    cout << "请输入管理员管理权限开始年月日: " << endl;
    cin >> astart;
    cout << "请输入管理员管理权限结束年月日: " << endl;
    cin >> aend;
    outfile << "1 " << idnum << " " << username << " " << pw <<
    " << age << " " << sex << " " << identity << " " << ustart << " " << uend
    << " " << position << " " << astart << " " << aend << endl;
}
}

int adduser()
{
    int i, j, a[] = { 1,2 };
    cout << "请输入增加内容: " << endl;
    cout << "1 使用者" << endl;

```

```

cout << "2 管理员" << endl;
i = getInt(a, 2);
cout << "请输入增加方式: " << endl;
cout << "1 单行添加" << endl;
cout << "2 逐项添加" << endl;
j = getInt(a, 2);
ofstream outfile("user.txt", ios::app | ios::ate | ios::_Nocreate);
if (i == 1) { _adduser(j, outfile); }
else//i==2
{
    _addadmin(j, outfile);
}
outfile.close();
cout << "信息已成功写入" << endl;
return i-1;
}

//查找record.txt中用户信息
void _searchuser(string username)
{
    int type = -1, count=0;
    string line;
    ifstream infile("user.txt", ios::in | ios::_Nocreate);
    while (!infile.eof())
    {
        getline(infile, line);
        if (line.empty())break;
        istringstream iss(line);
        if (line[0] == '0')//使用者
        {
            user u;
            people* p = &u;
            iss >> u;
            if (p->searchusername(username)) { p->print();type =
1;count++; }
        }
        else//管理员
        {
            admin a;
            people* p = &a;
            iss >> a;
            if (p->searchusername(username)) { p->print();type =

```

```

0;count++;}
    }
}
infile.close();
if (type != -1)cout << "找到" << count << "条包含人名" << username << "的结果" << endl;
else cout << "未找到包含人名" << username << "的结果" << endl;
}

//查找record.txt中用户信息
void searchuser()
{
    string username;
    cout << "请输入要查找的人名: " << endl;
    cin >> username;
    _searchuser(username);
}

void deleteuser()
{
    string username, line;
    int count = 0, flag = 0, flag2 = 0;
    cout << "请输入要删除的人名: " << endl;
    cin >> username;
    ifstream infile("user.txt", ios::in | ios::_Nocreate);
    ofstream outfile("_user_.txt", ios::out | ios::app);
    while (!infile.eof())
    {
        getline(infile, line);
        if (line.empty())break;
        istringstream iss(line);
        if (line[0] == '0')//使用者
        {
            user u;
            people* p = &u;
            iss >> u;
            if (p->searchusername(username)) flag = 1;
            else outfile << u << endl;
        }
        else//管理员
        {
            admin a;
            people* p = &a;
        }
    }
}

```

```

        iss >> a;
        if (p->searchusername(username)) flag = 1;
        else outfile << a << endl;
    }
}

infile.close();
outfile.close();
remove("user.txt");
flag2 = rename("_user_.txt", "user.txt");
if (flag && !flag2) cout << "用户" << username << "已成功删除" <<
endl;
else cout << "抱歉，不存在用户" << username << "，无法删除" << endl;
}

void _changeuser(string username)
{
    string line;
    int count = 0, flag = 0, flag2 = 0, j, arr[] = { 1,2 };
    ifstream infile("user.txt", ios::in | ios::Nocreate);
    ofstream outfile("_user_.txt", ios::out | ios::app);
    while (!infile.eof())
    {
        getline(infile, line);
        if (line.empty())break;
        istringstream iss(line);
        if (line[0] == '0')//使用者
        {
            user u;
            people* p = &u;
            iss >> u;
            if (p->searchusername(username))
            {
                p->print();
                cout << "请输入修改方式：" << endl;
                cout << "1 单行修改" << endl;
                cout << "2 逐项修改" << endl;
                j = getint(arr, 2);
                _adduser(j, outfile);
                flag = 1;
            }
            else outfile << u << endl;
        }
    }
}

```

```

else//管理员
{
    admin a;
    people* p = &a;
    iss >> a;
    if (p->searchusername(username))
    {
        p->print();
        cout << "请输入修改方式: " << endl;
        cout << "1 单行修改" << endl;
        cout << "2 逐项修改" << endl;
        j = getint(arr, 2);
        _addadmin(j, outfile);
        flag = 1;
    }
    else outfile << a << endl;
}
infile.close();
outfile.close();
remove("user.txt");
flag2 = rename("_user_.txt", "user.txt");
if (flag && !flag2) cout << "用户" << username << "已成功修改" <<
endl;
else cout << "抱歉, 不存在用户" << username << ", 无法修改" << endl;
}

//改变用户
void changeuser()
{
    string username, line;
    int count = 0, flag = 0, flag2 = 0, arr[] = { 1,2 };
    cout << "请输入要修改的人名: " << endl;
    cin >> username;
    _changeuser(username);
}

//用户登录和注册, 使用者返回0, 管理员返回1
int userlogin(string& n)
{
    string username, username2, pw, pw2, line;
    int flag = 0, i, arr[] = { 1,2 };

```

```

people* p = NULL, * q = NULL;
user u;
admin a;
cout << "请输入操作: " << endl;
cout << "1 登录程序" << endl;
cout << "2 注册账号" << endl;
i = getint(arr, 2);
if (i == 1)
{
    cout << "请输入用户名" << endl;
    cin >> username;
    ifstream infile("user.txt", ios::in | ios::_Nocreate);
    while (!infile.eof())
    {
        getline(infile, line);
        if (line.empty())break;
        istringstream iss(line);
        if (line[0] == '0')//使用者
        {
            p = &u;
            iss >> u;
            if (p->searchusername(username))
            {
                flag = 1;extern int recordtype;
                extern int usertype;
                username2 = p->getname();
                pw2 = p->getpw();
            }
        }
        else//管理员
        {
            p = &a;
            iss >> a;
            if (p->searchusername(username))
            {
                flag = 2;
                username2 = p->getname();
                pw2 = p->getpw();
            }
        }
    }
    infile.close();
}

```

```

while (!flag)
{
    p = NULL;
    cout << "没有此用户, 请重新输入用户名" << endl;
    cin >> username;
    ifstream infile("user.txt", ios::in | ios::_Nocreate);
    while (!infile.eof())
    {
        getline(infile, line);
        if (line.empty())break;
        istringstream iss(line);
        if (line[0] == '0')//使用者
        {
            p = &u;
            iss >> u;
            if (p->searchusername(username))
            {
                flag = 1;extern int recordtype;
                extern int usertype;
                username2 = p->getname();
                pw2 = p->getpw();
            }
        }
        else//管理员
        {
            p = &a;
            iss >> a;
            if (p->searchusername(username))
            {
                flag = 2;
                username2 = p->getname();
                pw2 = p->getpw();
            }
        }
    }
    infile.close();
}
cout << "请输入密码" << endl;
cin >> pw;
if (pw==pw2)
{
    n = username2;
}

```

```

        cout << "你好, " << username2 << "!" << endl;
        if (flag == 1) { return 0; } //user
        else { return 1; } //admin
    }
    else
    {
        cout << "请输入操作: " << endl;
        cout << "1 重新输入" << endl;
        cout << "2 退出程序" << endl;
        i = getint(arr, 2);
        if (i == 1) return -1; //reboot
        else return -2; //exit
    }
}
else//i==2
{
    i = adduser();
    return i;
}
}
record.h

```

借书记录类相关函数

```

#pragma once
#include <algorithm>
#include <fstream>
#include <iomanip>
#include <iostream>
#include <map>
#include <sstream>
#include <vector>
#ifndef __BOOKDATE__
#include "bookdate.h"
#endif // __BOOKDATE__

```

```

#ifndef __RECORD__
#define __RECORD__

```

```

class record//借书记录类
{
public:
    void print()//打印
    {

```

```

cout << "-----" << endl;
cout << "username      " << username << endl;
cout << "bookname      " << name << endl;
if (status)
{
    cout << "status      " << "未归还" << endl;
    cout << "start date   ";
    start.print();
    cout << endl;
}
else
{
    cout << "status      " << "已归还" << endl;
    cout << "start date   ";
    start.print();
    cout << endl;
    cout << "end date     ";
    end.print();
    cout << endl;
}
cout << "-----" << endl;
}

int searchbookname(string search) const//查找书名
{
    int flag = 0;
    size_t found = name.find(search);
    if (found != std::string::npos) {
        flag = 1;
    }
    return flag;
}

int searchusername(string search) const//查找用户名
{
    int flag = 0;
    size_t found = username.find(search);
    if (found != std::string::npos) {
        flag = 1;
    }
    return flag;
}

friend istream& operator >> (istream& input, record& R); //输入算符重载

```

```

    friend ostream& operator << (ostream& output, record& R); //输出算符重
载
private:
    date start; //借书开始日期
    date end; //借书结束日期
    string name; //书名
    string username; //人名
    bool status; //借阅状态 (0表示已归还, 1表示未归还)
};

//增加借书记录
void addrecord();

//列出借书记录
void listrecord();

//查询用户借阅历史
void searchuserbookrecord();

//查询图书借阅历史
void searchbookrecord();

//查找记录
void searchrecord();

//还书更新记录
void returnrecord();

vector<string> splitname(const string& str, int type);

//查找受欢迎的书和常借阅的人, 0为受欢迎的书, 1为常借阅的人
void popular(int type);

vector<date> splitdate(const string& str);

//查找借书数目随时间的变化
void recordtendency();

#endif // !_RECORD_
record.cpp
借书记录类相关函数

```

```

#pragma once
#ifndef __RECORD__
#include "record.h"
#endif // !__RECORD__

#ifndef __BOOKDATE__
#include "bookdate.h"
#endif // !__BOOKDATE__

#ifndef __ACTION__
#include "action.h"
#endif // !__ACTION__

#include <algorithm>
#include <fstream>
#include <iomanip>
#include <iostream>
#include <map>
#include <sstream>
#include <vector>
#include <chrono>
#include <ctime>
#pragma warning(disable:4996)
using namespace std;
extern int recordtype;
//record

istream& operator >> (istream& input, record& R)
{
    int s;
    input >> s >> R.username >> R.name >> R.start;
    if (!s)
    {
        input >> R.end;
        R.status = 0;
    }
    else
    {
        R.end.init(0, 0, 0);
        R.status = 1;
    }
    return input;
}

```

```

ostream& operator << (ostream& output, record& R)//输出算符重载
{
    if (recordtype)
    {
        output << left << setw(11) << R.name << " ";
        if (R.status)
        {
            output << "未归还, ";
            R.start.print();
            output << "借出" << endl;
        }
        else
        {
            output << "已归还, ";
            R.start.print();
            output << "借出, ";
            R.end.print();
            output << "归还" << endl;
        }
    }
    else
    {
        output << "-----" << endl;
        output << "name" << R.username << endl;
        output << "bookname" << R.name << endl;
        if (R.status)
        {
            output << "status" << "未归还" << endl;
            output << "start date" ;
            R.start.print();
            output << endl;
        }
        else
        {
            output << "status" << "已归还" << endl;
            output << "start date" ;
            R.start.print();
            output << endl;
            output << "end date" ;
            R.end.print();
            output << endl;
        }
    }
}

```

```

        }
        output << "-----" << endl;
    }
    return output;
}

//增加借书记录
void addrecord()
{
    int i, arr[] = { 1,2 };
    cout << "请输入增加方式: " << endl;
    cout << "1 单行添加" << endl;
    cout << "2 逐项添加" << endl;
    i = getint(arr, 2);
    if (i == 1)
    {
        string line;
        cout << "请输入借阅状态(0表示已归还, 1表示未归还)、借阅人、书名、借
书年月日和还书年月日(未归还不填写还书年月日), 每项之间用“ ”隔开: " << endl;
        cin.clear();
        cin.ignore(std::numeric_limits<streamsize>::max(), '\n');
        getline(cin, line);
        ofstream outfile("record.txt", ios::app | ios::ate |
ios::nocreate);
        outfile << line << endl;
        outfile.close();
    }
    else//i==2
    {
        date start;//借书开始日期
        date end;//借书结束日期
        string name;//书名
        string username;//人名
        int status;//借阅状态
        cout << "请输入图书的借阅状态(0表示已归还, 1表示未归还): " << endl;
        cin >> status;
        while (cin.fail() || status != 0 || status != 1)
        {
            cout << "借阅状态输入错误, 请重新输入" << endl;
            cin.clear();
            cin.ignore(std::numeric_limits<streamsize>::max(), '\n');
    }
}

```

```

        cin >> status;
    }
    cout << "请输入借阅人名: " << endl;
    cin >> username;
    cout << "请输入图书书名: " << endl;
    cin >> name;
    cout << "请输入借书年月日: " << endl;
    cin >> start;
    cout << "请输入还书年月日: " << endl;
    cin >> end;
    ofstream outfile("record.txt", ios::app | ios::ate |
ios::_Nocreate);
    outfile << status << " " << username << " " << name << " " <<
start << " " << end << endl;
    outfile.close();
}
cout << "书籍已成功写入" << endl;
}

//列出借书记录
void listrecord()
{
    record* r = new record;
    ifstream infile("record.txt", ios::in | ios::_Nocreate);
    string line;
    while (!infile.eof())
    {
        getline(infile, line);
        if (line.empty())break;
        istringstream iss(line);
        iss >> *r;
        cout << *r;
    }
    infile.close();
    delete r;
}

//查询用户借阅历史
void searchuserbookrecord()
{
    record* r = new record;
    string line, name;

```

```

int count = 0;
ifstream infile("record.txt", ios::in | ios::_Nocreate);
cout << "请输入要查找的书名: " << endl;
cin >> name;
while (!infile.eof())
{
    getline(infile, line);
    if (line.empty())break;
    istringstream iss(line);
    iss >> *r;
    if (r->searchbookname(name))
    {
        count++;
        cout << *r;
    }
}
infile.close();
if (count)cout << "共找到" << count << "个包含书名" << name << "的结果"
" << endl;
else cout << "未找到包含书名" << name << "的结果" << endl;
}

//查询图书借阅历史
void searchbookrecord()
{
    string username, line;
    record* r = new record;
    int count = 0;
    cout << "请输入要查找的人名: " << endl;
    cin >> username;
    ifstream infile("record.txt", ios::in | ios::_Nocreate);
    while (!infile.eof())
    {
        getline(infile, line);
        if (line.empty())break;
        istringstream iss(line);
        iss >> *r;
        if (r->searchusername(username))
        {
            count++;
            cout << *r;
        }
    }
}

```

```

    }
    infile.close();
    if (count)cout << "共找到" << count << "个包含人名" << username << "的
结果" << endl;
    else cout << "未找到包含人名" << username << "的结果" << endl;
    delete r;
}

//查找记录
void searchrecord()
{
    record* r = new record;
    string line;
    int count = 0, i, arr[] = { 1,2 };
    cout << "请输入查找方式: " << endl;
    cout << "1 按书名查找" << endl;
    cout << "2 按人名查找" << endl;
    i = getInt(arr, 2);
    ifstream infile("record.txt", ios::in | ios::_Nocreate);
    if (i == 1)
    {
        string name;
        cout << "请输入要查找的书名: " << endl;
        cin >> name;
        while (!infile.eof())
        {
            getline(infile, line);
            if (line.empty())break;
            istringstream iss(line);
            iss >> *r;
            if (r->searchbookname(name))
            {
                count++;
                cout << *r;
            }
        }
        if (count)cout << "共找到" << count << "个包含书名" << name << "
的结果" << endl;
        else cout << "未找到包含书名" << name << "的结果" << endl;
    }
    else//i==2
{

```

```

        string username;
        cout << "请输入要查找的人名: " << endl;
        cin >> username;
        ifstream infile("record.txt", ios::in | ios::_Nocreate);
        while (!infile.eof())
        {
            getline(infile, line);
            if (line.empty())break;
            istringstream iss(line);
            iss >> *r;
            if (r->searchusername(username))
            {
                count++;
                cout << *r;
            }
        }
        if (count)cout << "共找到" << count << "个包含人名" << username
        << "的结果" << endl;
        else cout << "未找到包含人名" << username << "的结果" << endl;
    }
    infile.close();
    delete r;
}

//还书更新记录
void returnrecord()
{
    record* r = new record;
    string line, name;
    int flag = 0, flag2 = 0, count = 0, i, arr[] = { 1,2 };
    ifstream infile("record.txt", ios::in | ios::_Nocreate);
    ofstream outfile("_record_.txt", ios::out | ios::app);
    cout << "请输入归还书籍名称: " << endl;
    cin >> name;
    while (!infile.eof())
    {
        getline(infile, line);
        if (line.empty())break;
        istringstream iss(line);
        iss >> *r;
        if (r->searchbookname(name) && line[0] == '1')
        {

```

```

cout << *r;
cout << "是否完善此借书记录? " << endl;
cout << "1 是" << endl;
cout << "2 否" << endl;
i = getInt(arr, 2);
if (i == 1)
{
    flag = 1;
    int y, m, d;
    count = 1;
    cout << "请输入还书日期输入方式: " << endl;
    cout << "1 自动添加" << endl;
    cout << "2 手动添加" << endl;
    i = getInt(arr, 2);
    if (i == 1)
    {
        auto now = chrono::system_clock::now(); // 获取
当前时间点
        time_t now_c =
chrono::system_clock::to_time_t(now); // 转换为time_t类型
        tm* parts = localtime(&now_c); // 使用localtime
获取本地时间
        y = 1900 + parts->tm_year;
        m = 1 + parts->tm_mon;
        d = parts->tm_mday;
        line[0] = '0';
        outfile << line << " " << y << " " << m << " "
<< d << endl;
    }
    else
    {
        cout << "请输入还书年月日: " << endl;
        cin >> y >> m >> d;
        line[0] = '0';
        outfile << line << " " << y << " " << m << " "
<< d << endl;
    }
}
else
{
    outfile << line << endl;
}

```

```

        }
        else
        {
            outfile << line << endl;
        }
    }
    infile.close();
    outfile.close();
    remove("record.txt");
    flag2 = rename("_record_.txt", "record.txt");
    if (count && !flag2) cout << "书籍记录已更改完成" << endl;
    else
    {
        if (flag) cout << "书籍" << name << "已经归还，无需更改" << endl;
        else if (flag2) cout << "文件损坏，无法更改" << endl;
        else cout << "书籍" << name << "不存在，未做更改" << endl;
    }
    delete r;
}

```

//分割，0为受欢迎的书分割，1为按常借阅的人分割

```

vector<string> splitname(const string& str, int type) {
    vector<string> tokens;
    istringstream iss(str);
    string junk, name, username;
    iss >> junk >> username >> name;
    if (type)tokens.push_back(username);
    else tokens.push_back(name);
    return tokens;
}

```

//查找受欢迎的书和常借阅的人，0为受欢迎的书，1为常借阅的人

```

void popular(int type)
{
    int max = 0, flag = 0;
    ifstream infile("record.txt");
    if (!infile.is_open()) {
        cerr << "无法打开文件:record.txt" << endl;
        exit(1);
    }
    map<string, int> wordCount;
    string line;

```

```

while (getline(infile, line)) {
    // 对每一行进行分词
    vector<string> words = splitname(line, type);
    for (const string& word : words) {
        // 统计每个单词出现的次数
        wordCount[word]++;
    }
}
infile.close();

// 将wordCount的内容转移到vector中
vector<pair<string, int>> sortedWordCount(wordCount.begin(),
wordCount.end());

// 对vector进行排序，按照单词出现的次数降序排序
sort(sortedWordCount.begin(), sortedWordCount.end(),
[](const pair<string, int>& a, const pair<string, int>& b) {
    return a.second > b.second; // 降序
});

// 打印排序后的单词及其出现次数，可在此改进显示所有人的借阅情况和所有书的被
借阅情况
for (const auto& pair : sortedWordCount) {
    if (!flag)
    {
        max = pair.second;
        flag = 1;
        if (type)cout << "借阅最多的读者是 ";
        else cout << "最受欢迎的书籍是 ";
    }
    if (pair.second == max) { cout << pair.first << ", ";}
}
if (type)cout << "借阅了 " << max << " 次。" << endl;
else cout << "被借阅了" << max << " 次。" << endl;
}

// 分割，0为受欢迎的书分割，1为按常借阅的人分割
vector<date> splitdate(const string& str) {
    vector<date> tokens;
    istringstream iss(str);
    string junk1, junk2, junk3;

```

```

date d;
iss >> junk1 >> junk2 >> junk3 >> d;
tokens.push_back(d);
return tokens;
}

//查找借书数目随时间的变化
void recordtendency()
{
    int max = 0, flag = 0;
    ifstream infile("record.txt");
    if (!infile.is_open()) {
        cerr << "无法打开文件:record.txt" << endl;
        exit(1);
    }
    map<date, int> dateCount;
    string line;
    while (getline(infile, line)) {
        // 对每一行进行分词
        vector<date> dates = splitdate(line);
        for (const date& date : dates) {
            // 统计每个单词出现的次数
            dateCount[date]++;
        }
    }
    infile.close();

    // 将wordCount的内容转移到vector中
    vector<pair<date, int>> sortedDateCount(dateCount.begin(),
dateCount.end());

    // 对vector进行排序，按照单词出现的次数降序排序
    sort(sortedDateCount.begin(), sortedDateCount.end(),
        [] (const pair<date, int>& a, const pair<date, int>& b) {
            return a.first < b.first; // 升序
        });
    cout << "-----" << endl;
    cout << "日期" << "借阅数量" << endl;
    // 打印借阅数量
    for (const auto& pair : sortedDateCount) {
        if (!flag)
        {

```

```
        max = pair.second;
        flag = 1;
    }
    else max = pair.second > max ? pair.second : max;
    pair.first.print();
    cout << " " << pair.second << endl;
}
cout << "-----" << endl;
cout << "借阅最多的日期是 ";
for (const auto& pair : sortedDateCount) {
    if (pair.second == max) { pair.first.print(); cout << ", "; }
}
cout << "借阅了 " << max << " 本书。" << endl;
}
```

附录 2：评分表

第 1 题评分标准

项 目	评 价	
设计方案的合理性与创新性	3	
设计与调试结果	4	
设计说明书的质量	1	
程序基本要求涵盖情况	4	
程序代码编写素养情况	2	
课程设计周表现情况	1	
综合成绩	15	