图书管理系统

**中期考核**

姓名：张海森

目录

**[一、 项目简介](#_Toc10840_WPSOffice_Level1)** **[1](#_Toc10840_WPSOffice_Level1)**

**[二、总体设计](#_Toc581_WPSOffice_Level1)** **[1](#_Toc581_WPSOffice_Level1)**

[(一)界面](#_Toc581_WPSOffice_Level2) [1](#_Toc581_WPSOffice_Level2)

[(二)程序设计](#_Toc2360_WPSOffice_Level2) [1](#_Toc2360_WPSOffice_Level2)

[(三)容错性](#_Toc5296_WPSOffice_Level2) [2](#_Toc5296_WPSOffice_Level2)

[(四)记录](#_Toc30212_WPSOffice_Level2) [2](#_Toc30212_WPSOffice_Level2)

**[三、详细设计](#_Toc2360_WPSOffice_Level1)** **[3](#_Toc2360_WPSOffice_Level1)**

[(一)采编入库](#_Toc22724_WPSOffice_Level2) [3](#_Toc22724_WPSOffice_Level2)

[(二)清除库存](#_Toc9852_WPSOffice_Level2) [3](#_Toc9852_WPSOffice_Level2)

[(三)书籍信息](#_Toc31960_WPSOffice_Level2) [4](#_Toc31960_WPSOffice_Level2)

[(四)借阅](#_Toc5867_WPSOffice_Level2) [5](#_Toc5867_WPSOffice_Level2)

[(五)归还](#_Toc7726_WPSOffice_Level2) [6](#_Toc7726_WPSOffice_Level2)

[(六)预约借书](#_Toc16672_WPSOffice_Level2) [7](#_Toc16672_WPSOffice_Level2)

**[四、数据结构与数据处理](#_Toc5296_WPSOffice_Level1)** **[7](#_Toc5296_WPSOffice_Level1)**

[(一)数据结构](#_Toc17035_WPSOffice_Level2) [7](#_Toc17035_WPSOffice_Level2)

[(二)数据处理](#_Toc567_WPSOffice_Level2) [7](#_Toc567_WPSOffice_Level2)

**[五、程序亮点](#_Toc30212_WPSOffice_Level1)** **[8](#_Toc30212_WPSOffice_Level1)**

**[六、程序兼容性说明](#_Toc22724_WPSOffice_Level1)** **[8](#_Toc22724_WPSOffice_Level1)**

[(一)输入输出](#_Toc12124_WPSOffice_Level2) [8](#_Toc12124_WPSOffice_Level2)

[(二)显示输入有误](#_Toc31650_WPSOffice_Level2) [8](#_Toc31650_WPSOffice_Level2)

[(三)数据库功能](#_Toc4892_WPSOffice_Level2) [8](#_Toc4892_WPSOffice_Level2)

**[七、总结与体会](#_Toc9852_WPSOffice_Level1)** **[8](#_Toc9852_WPSOffice_Level1)**

1. 项目简介

作为一名当代大学生，我们深知图书馆在我们的日常生活和学习中所具有的重要的作用。



图 1 广工图书馆

而随着时代的进步，图书馆中的书籍种类和数量正在变得越来越多、也越来越复杂。在这种情况下，建立合理的图书管理系统便会给我们管理、借还图书馆的书籍带来很大的方便。同时，由于书籍种类和数量的庞杂，凭借人手工建立、管理、记录图书馆内的藏书、借阅归还情况是不可行的，因此，这个以C语言为基础的图书管理系统便有了自己的用武之地。

该图书管理系统可以完成图书的采编入库、清除库存、借阅与归还（包含预约借阅）、查看某著者全部著作、查看某书籍状态信息的功能。



图 2 程序主界面

通过使用图书管理系统，可以为大量的图书管理服务提供一定的方便。

二、总体设计

(一)界面

不采用图形界面，但是对于大部分的信息输入操作和信息输出显示采用表格的方式，整体位置选取居中的区域，这样方便查看所需信息和输入有用信息。

采用菜单选项的方式引导使用者完成对应的操作，操作过程中有每一步操作的提示，同时在桌面显示输入的信息和错误信息，方便使用者查看。

(二)程序设计

链表是这个图书管理系统的基础。因为无论从对书籍的管理上，还是对借阅者信息的管理上，都符合链表的特性。

对书籍而言，链表的一个节点可以用来存储一本书籍的信息，而链表之间的链接又为在所有书籍中搜索需要的书籍提供了途径，使用链表不仅可以不需考虑图书馆中藏书量的实际多少，而且可以方便完成书籍的增删、管理操作。

对借阅者而言，一个节点可以存储一个借阅者的信息，而链表之间的链接又反映出这些借阅者之间的共同特点（即都借阅了这本书），书籍节点又可以指向这本书的借阅者链表，这样，管理一本书的借阅者就显得很方便了。由于借阅者本身未必按时还书，所以链表较之队列更具合理性。

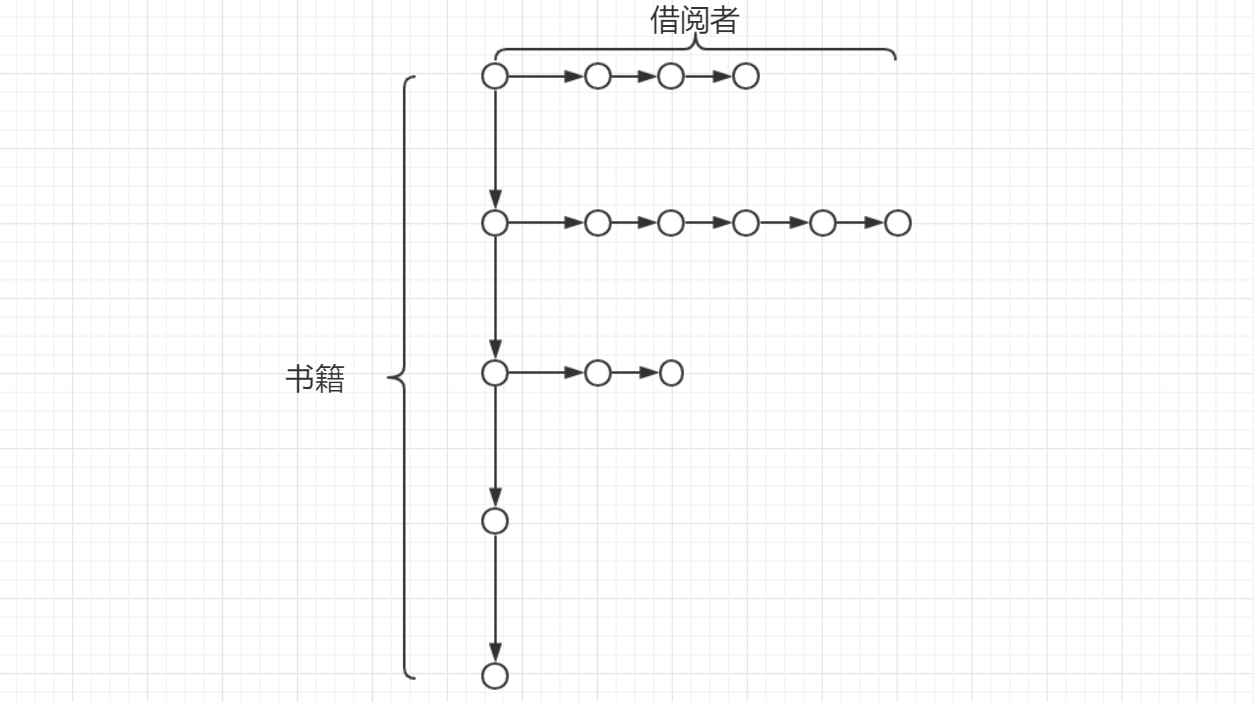


图 3 总体数据结构设计

(三)容错性

首先是对于总菜单选项下的输入，能够识别非法输入并自动重新进入选择中(直到按下代表退出的按键x)，对于书籍信息选项下的两个输入项，一般的非法输入使得程序不执行其所对应的两个操作中的任意一个操作，直接进入到主菜单的选择中。

其次是对于那些需要输入信息的操作，如果由于失误而造成的输入信息错误，可以在对于这些信息的确认环节中选择放弃该次操作，而不必麻烦地再次执行上述操作的反操作。

最后，合理的容错性要求对于不正确的操作要能够指出并不加以执行，比如借阅所输入的书号在图书馆藏书中找不到，以及清除出库不存在于图书馆馆藏中的书籍。

(四)记录

记录其一是为保存上一次程序运行后的结果而服务的，这样使得程序可以多次反复使用，

而不必始终保持运行(这样的记录就相当于这个程序的数据库，当程序需要获取上一次运行后的结果时，可以直接从数据库提取信息，而在程序本次运行即将结束的时候，又对这一次运行后的结果进行记录)。

记录其二是作为程序的log文件来使用的，记录程序中的每一次会话，只需要将程序每一次运行的全部过程全部记录就可以了。



图 4 数据库txt

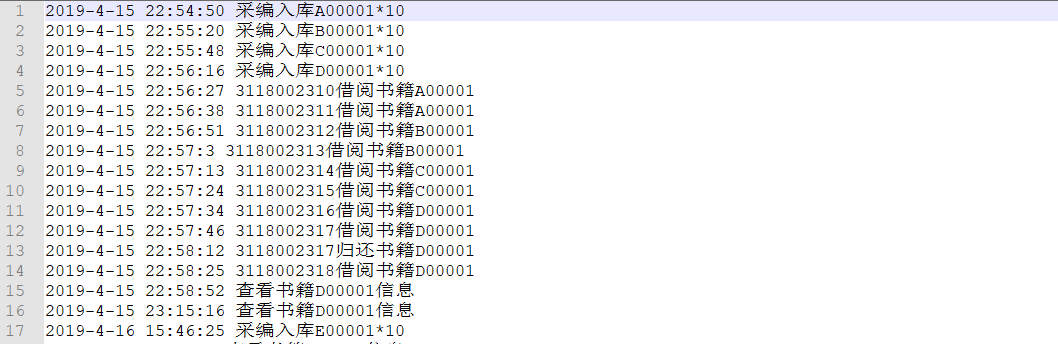


图 5 log txt

三、详细设计

(一)采编入库

采编入库即为对一本新购进的图书进行分类操作后将它存储在书籍链表中以供将来使用，可以认为，采编入库可以分为分类、录入书籍信息、增加到书籍链表中、赋值相关信息和确认信息五个步骤。

分类操作是为了得到这本书的书号，在本程序中考虑为手工操作，故不在程序范围内。

录入书籍信息，以表格的形式让使用者填入所要入库的书籍的信息，包括书号、书名、作者，这是经过分类之后将书籍录入书籍链表中必不可少的操作，相当于对某一个新节点赋初值的操作。

录入信息之后(包括后面的为这本书赋值)，需要分成两种情况考虑，即当前储存中已经有该书籍的信息，这种时候只需要对这本书的库存量、存量做出对应的更改即可，而不需要申请一个新的节点，因为它们是同一本书籍。另一种情况，当前库存中没有这本书籍，这个时候，就需要有一个新节点来存储这本新书的信息，所有的信息，包括书籍信息、借阅者信息都需要赋初值，这样，一个存储有这本书籍信息的新节点就诞生了，待使用者确认之后这本书就可以插入到书籍链表中去了。

之后便是确认书籍信息，即显示录入的和系统操作之后的书籍信息，待确认。此时可以选择放弃本次操作，一般是因为录入信息有误，需要重新录入。

增加到书籍链表中，由于这本书如果之前在书籍链表中不存在，所以需要将它添加进书籍链表中。而它所处的位置其实并不重要，但是为了操作的方便，所有的新录入的书籍都选择插入在链表结尾的方式，至此，加上之前对于那些已经在链表中的书籍的改变信息的操作，对于n本书籍的采编入库功能就算是完成了。

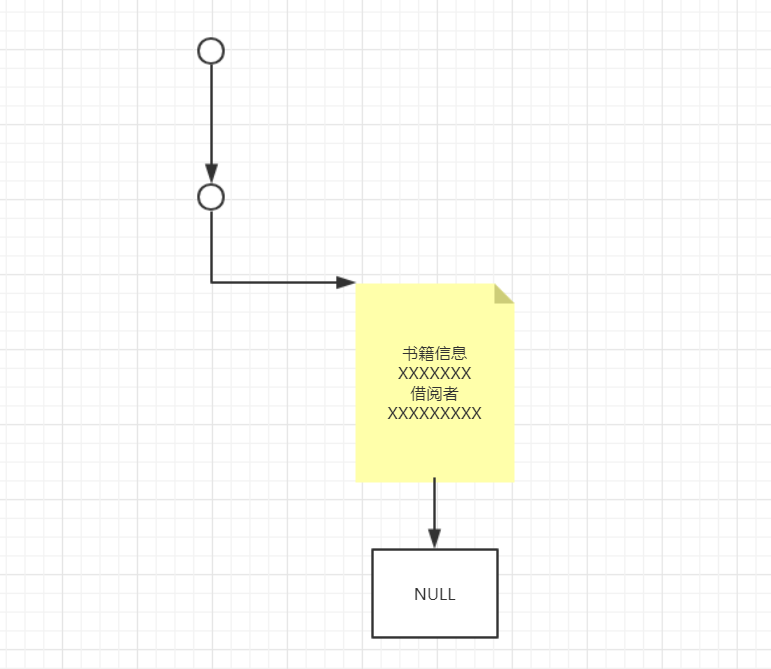


图 6 采编入库数据处理

(二)清除库存

清除库存就是对于已经在书籍链表中的书籍，如果这本书已经过时了或者需要更新换代

了，就需要从图书馆的许多书籍中找到这一本书籍，然后从链表中将这个节点的信息删除。从某种意义上来讲，这是采编入库的逆操作。需要经过输入书籍信息、寻找书籍、确认、删除的操作才可以最终完成。

输入书籍信息，是对一本书清除库存必不可少的任务，帮助我们寻找到这本书籍并最终完成删除的操作。需要输入的信息包括书号、书名、作者，而其中最为重要的一项则是书号，简单来说，书号就相当于学生的学生证号，通过这个学生证号可以方便地找到这位学生同时减少各种意外的发生。

得到了书籍信息之后，就需要确认书籍链表中是否真实存在这个书号所对应的书籍，如果书籍信息输入错误，根本找不到这本书的话，是无法完成清除库存这一操作的。简化来说，就是从书籍链表的第一个节点往下找，如果到最后一个节点都没有找到这样一本书，那么可以认为是输入了错误的书籍信息并报错。如果找到，就可以进行下一步的操作了。

确认所要删除的书籍信息，将系统根据你的书号找到的书籍信息显示在界面中，减少误删可能发生的概率，当你确认了书籍信息后，就可以执行删除操作了。

删除操作，从书籍链表中删除你根据你的输入找到的书籍，同时配合以图书馆对相关书籍的人为处理，就算是完成了书籍的清除库存操作。

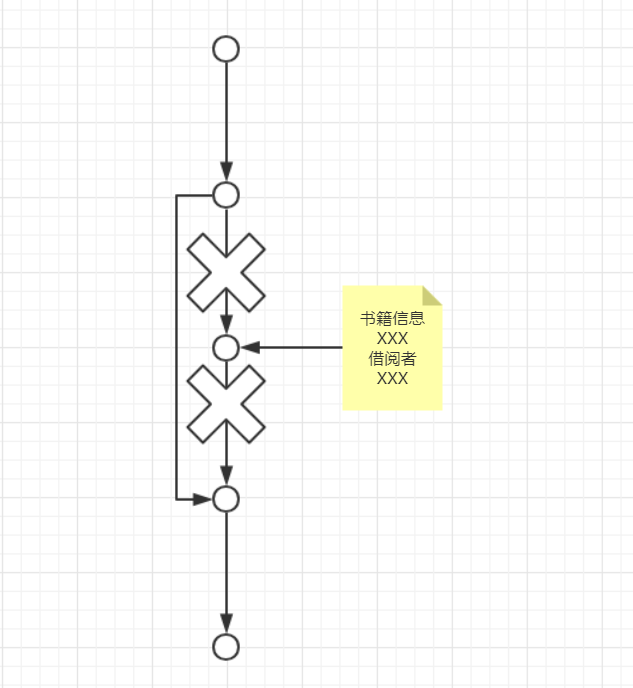


图 7 清除库存数据处理

(三)书籍信息

查找书籍信息就是根据我已经输入的信息帮我找到那些与我输入的信息相关的我不知道的信息。查找书籍信息的作用在这个系统中有列出某著者全部著作名和列出某书状态。

列出某著者全部著作名，可以分为获取著者名、输出某著者全部著作名。通过输入著者名，系统就知道了需要比较的信息。之后从头到尾查看整个书籍链表，进行比较，如果整个链表中都没有这位著者的著作，就显示图书馆内没有该著者的著作，而对于那些根据著者名找到的书籍名称，就按次序有条理地输出。对于提高这一操作的效率，由于在链表中查找，我想到的是如果只遍历一次链表可以提高效率，所以，如果找到一本这位著者的著作，就从遍历链表的操作中退出来，这样，无论书籍链表中有没有这位著者的著作，都只会从头到尾查找一次链表，对于有了第一本著作之后，就接着上次的接着遍历，直到输出这位著者的全部著作。这样，就算是完成了查看某著者全部著作名的操作。

列出某书状态，需要列出的信息有存储在书籍链表节点中的信息，还有如果这本书目前有在借的同学，也要列出最近到期的同学的信息，也即存储在借阅者链表节点中的信息。可以分为输入书籍信息、通过链表找到并输出相关的信息。输入了书籍信息后，系统就知道了比较的对象。如果输入的信息不存在于链表之中，则报错。不然就通过链表的节点输出所要的信息，链表的存在让这样的操作显得很方便。这时，需要进行一个判断，如果这本书有借阅者的话，就输出最近到期的借阅者的信息，由于借阅者是按照时间先后顺序链接起来的，所以第一个节点的信息就是所要找的借阅者信息，如果没有借阅者，就只是输出书籍链表中节点的信息。由于一本书籍的节点中包含了一个指向一个借阅者链表的指针，所以书籍就能够找到它的最早到期的借阅者。

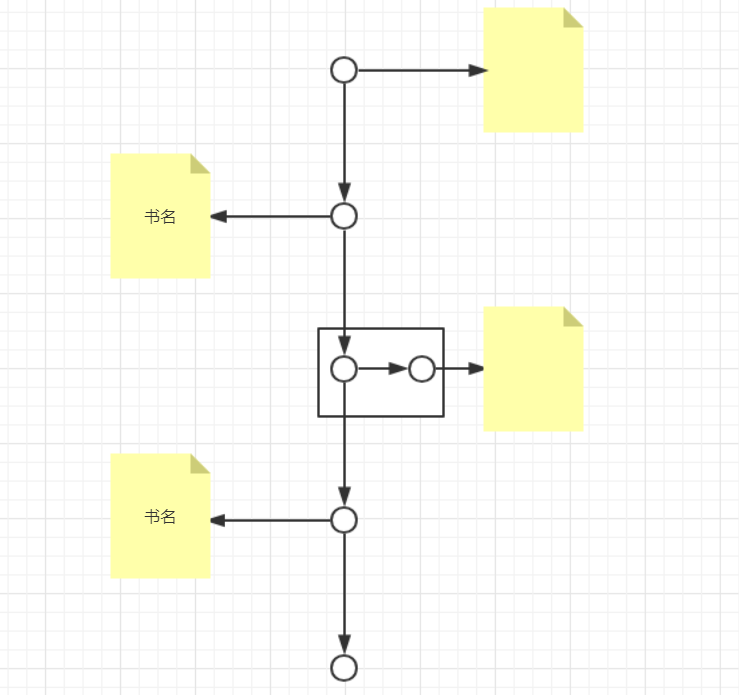


图 8 查看书籍信息数据处理

(四)借阅

借阅操作就是对于已经输入的书籍信息，找到链表中的该书籍信息，经过参数的修改后，可以给他们带离一段时间的操作。需要完成的事项有输入书籍信息、输入借阅者信息、确认、在书籍的借阅者链表中插入新信息。

输入书籍信息，是完成借阅一本书的必不可少的操作，在输入书籍信息之后，还需要确认输入的书籍信息在书籍链表中确实有这样的书籍可以提供借阅服务。

输入借阅者信息，是为了方便在借阅者链表中登记，间接地方便了还书服务。

确认需要借阅的信息，首先，这是建立在对于书籍的相关信息判断后发现允许借阅的前提上，相关信息就是现存量和可借量，如果可借量大于零，那么就是可以借阅这本书籍的，否则，即使图书馆中显示有这本书籍也无法完成借阅。其次需要完成还书日期的计算，由于选定的借书期限是3个月，所以要在界面上显示按照规定应该还书的日期，日期是通过时间的获取和累加计算出来的。这样，确认了借阅者信息无误之后，就可以选择确认信息。

在书籍的借阅者链表中插入新信息，将得到的借阅证号、借书日期和还书日期作为一个节点中的部分信息存储到借阅者链表中取出，和书籍的存储一样，都是存储在链表的末尾。这样，在归还的时候，因为有记录可以索引，就方便了归还服务。至此，算是完成了书籍的借阅。

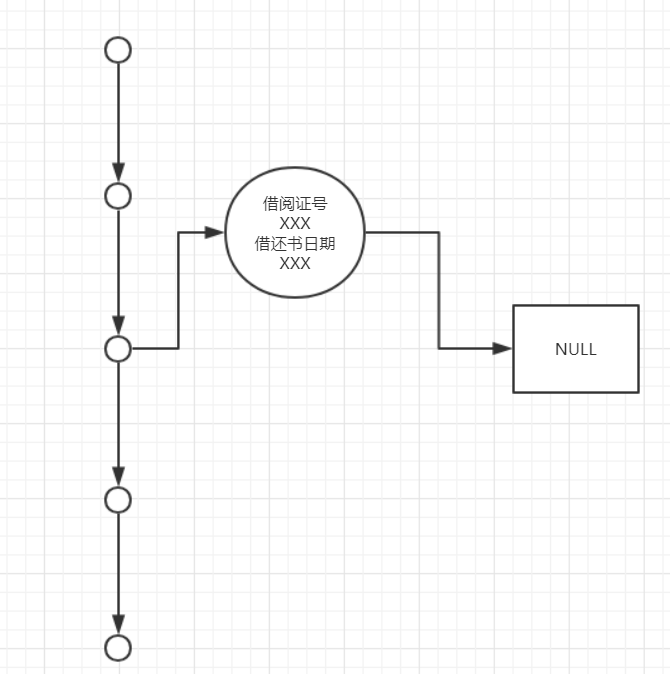


图 9 借阅操作数据处理

(五)归还

归还操作就是对于已经借阅出去的书籍收回图书馆中并注销相关信息的过程。从某种意义上来讲，归还操作是借阅操作的逆操作。归还操作包含输入书籍信息、输入借阅者信息、确认信息、从借阅者链表中删除信息。

输入书籍信息，可以确认是在哪一本书的借阅者链表之中寻找登记这位借阅者信息的节点。

输入借阅者信息，是为了找到对应的节点并完成删除操作。

确认信息，根据已经输入的信息，如果找到了信息，就会输出更详细的信息，包括了借书日期和还书日期等，如果确认这些信息准确无误，就接着执行从借阅者链表中删除信息的操作。

从借阅者链表中删除信息，可以类比于从书籍链表中删除某一本书籍的信息的操作，都是找到、执行链表操作、完成。至此，算是完成了书籍的归还操作。

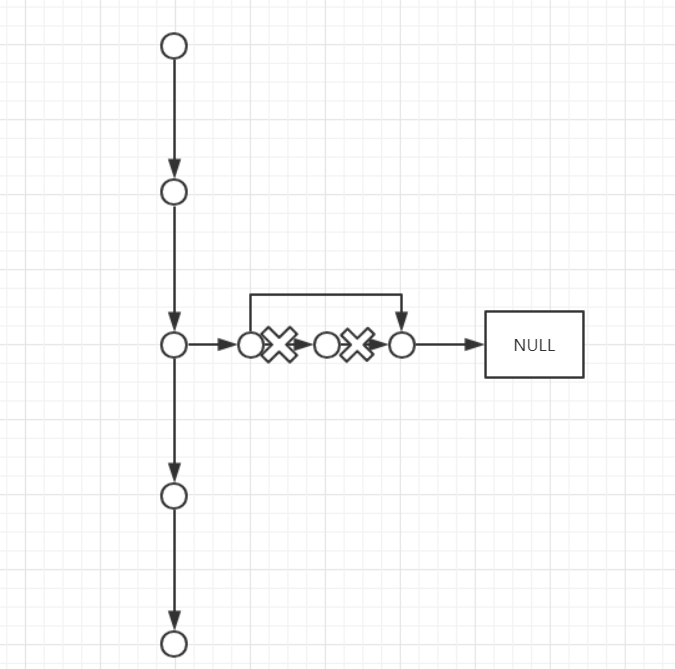


图 10 归还操作数据处理

(六)预约借书

预约借书功能就是要告诉想要借阅的人，在他想要借阅的时候会不会有空闲的书本留给他，并且在之后给他留下这样一本书。

预约借书需要考虑的有三种情况。

第一种情况，可借无书且在其间无人还书，这样，便预约失败。

第二种情况，可借有书，其间有无人还书就不重要了，当场就可以预约成功，并且修改书籍信息，让这本书可以保留下来直到真的被预约的人借走。

第三种情况，可借无书，但是其间有人还书，这样，可以先让可借的数量比原来减少一，在还书过后书本就被保留了下来。

这样，就算是完成了预约借书的事项。

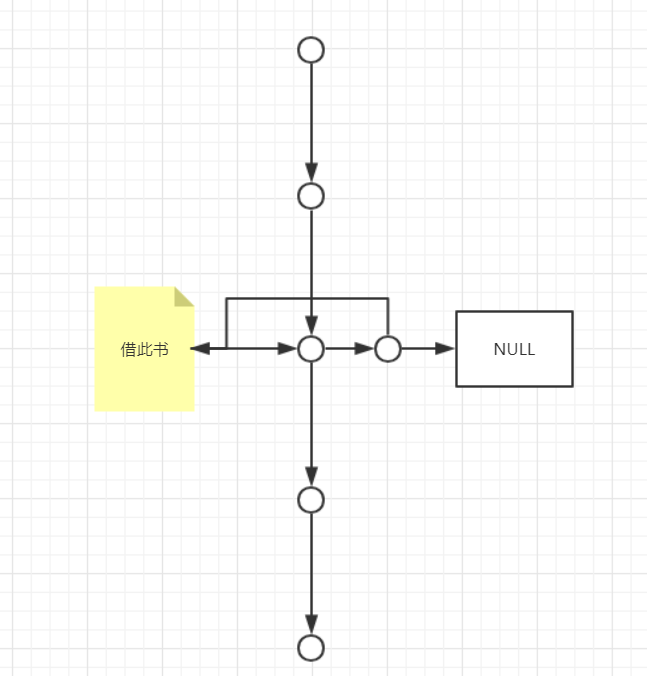


图 11 预约借书数据处理

四、数据结构与数据处理

(一)数据结构

如前所说，本系统是建立在链表的基础之上的，对应一本书，可以入库可以出库，对应的链表操作就是加入新节点和删除节点，同样的道理可以适用于借阅者身上。链表为书籍间的关联实现了可能，为借阅者之间的关联实现了可能，更为一本书籍和它的许多借阅者之间的关联实现了可能。

(二)数据处理

首先，有必要说明一下这个程序所用到的数据类型。对于书籍而言，书名、作者、书号是字符数组，可借量、现存量、库存总量、借阅者数量是整型，对于借阅者而言，借阅证号是无符号整型，借书日期和还书日期是整型。

下面是数据的处理的一些方面。

其一，针对输入的处理，因为在使用的过程中，每一个操作总是会有输入，这时便建立一个新的暂存书籍信息的节点或者是暂存借阅者信息的节点，然后这些信息的读入都会被存放到这些新建立的节点中，然后根据程序使用者后续的操作决定将这些节点插入到书籍链表中、借阅者链表中、在遍历链表的过程中与已知节点的信息进行比较或者是其他的操作。

其二，把输入的信息作为一个整体来看，他们会形成一个节点，这个节点可能会加入链表，可能用于删除链表中的已知节点。

其三，对于那些需要在需要判断节点是否存在的语句中，数值型的比较可以使用”==”,字符型的比较可以使用”strcmp”，这样，在找到了需要的节点之后就可以进行相关的操作了。

其四，是对于那些出错的数据而言的，由于对于错误的数据的修改，如果执行反操作的话，很比较麻烦，所以在每每有需要保存的内容的时候，都会有是否保存该内容的选项，如果选择n，当场放弃操作，而不用后续执行反操作。

五、程序亮点

在书籍节点中开辟了一个指向借阅者节点的指针，这样通过书籍就可以实现对借阅者的管理。

六、程序兼容性说明

对于一个合格的程序而言，实现程序的兼容性是不可避免的一个话题。

(一)输入输出

首先，就是对于输入的检查，能够做到识别非法输入，在向使用者报错之后，又能够自动转入下一次的菜单选项选择中，直到使用者选择退出程序运行。

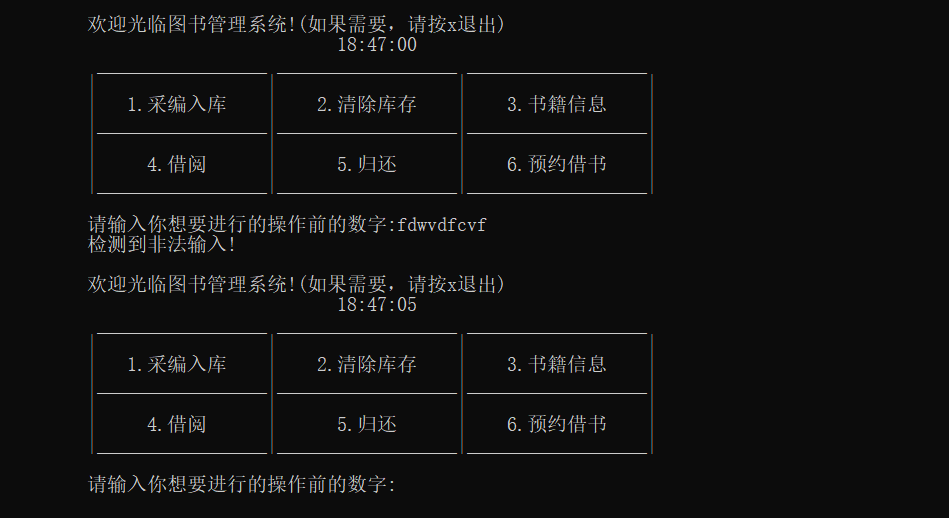


图 12 检测非法输入

(二)显示输入有误

如果出现对应的异样，能够向使用者显示出有异样的信息，并指出具体不能够运行程序是因为什么，帮助使用者找到正确的处理需要。

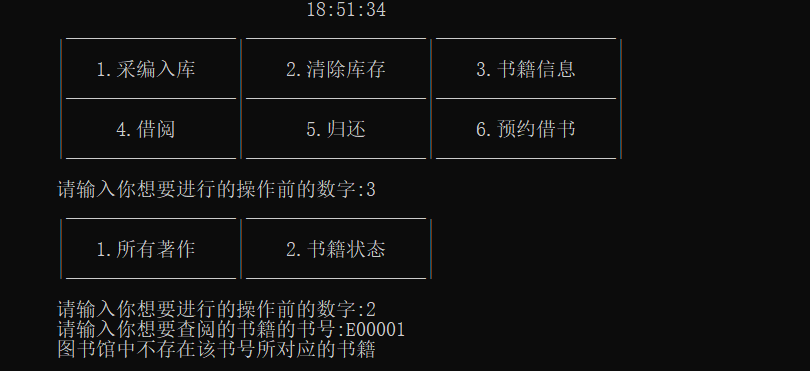


图 13 输入有误显示

(三)数据库功能

对于之前程序运行的结果可以在退出程序时自行保存，这样就可以不必保持系统始终处于运转中，也不怕数据丢失，需要的时候直接启动程序就好了。

七、总结与体会

在这次设计图书管理系统的过程中，我认识到，在真正开始编程之前，想好自己程序的构建大致思路，包括了数据结构的选择、选择显示的形式等等，是一件十分重要的事情，在开始之前想好，可能会省很大力气。而对于那些自己暂时没有办法解决的，需要有耐心，不能轻易放弃，而是要多思考，多尝试，问题的解决总是需要耗费精力的。

体会到了程序的兼容性对于一个想要做好的程序而言，是多么重要的一个参数，好的程序需要处理好许多可能出现失误的地方，同时，我把使用的方便程度也认为是程序兼容性的一部分，为使用者做尽量全面的考虑，是一个优秀编程者必不可少的做法。