

高级语言程序设计

课程设计报告

|  |  |  |
| --- | --- | --- |
| 课程设计成绩（总分20） | | 分数 |
| 程序 | 程序设计完整，功能齐全（8-10分） |  |
| 程序设计基本完整，功能基本完成（0-7分） |
| 报告 | 结构清晰完整，报告格式正确，语言通顺（8-10分） |  |
| 格式不正确，封面或字体有误，语言欠通顺（0-7分） |
|  | 总分 |  |

|  |  |
| --- | --- |
| 课程名称 | 高级语言程序设计 |
| 课程编号 | 08306135 |
| 报告题目 | 图书管理系统 |
| 序 号 |  |
| 姓 名 |  |
| 学 号 |  |
| 所在学院 |  |
| 专业名称 |  |

20 年 月

目 录

[1. 数据格式与文件描述 3](#_Toc514061021)

[1.1 数据文件格式 3](#_Toc514061022)

[1.2 功能描述 3](#_Toc514061023)

[1.3 要求 4](#_Toc514061024)

[2. 问题分析 5](#_Toc514061025)

[3. 总体设计 5](#_Toc514061026)

[3.1 功能模块设计 5](#_Toc514061027)

[3.2 系统界面设计 6](#_Toc514061028)

[3.3 数据结构设计 6](#_Toc514061029)

[3.4 函数设计 7](#_Toc514061030)

[4. 详细设计 8](#_Toc514061031)

[4.1 各个函数的调用关系图 8](#_Toc514061032)

[4.2 各个函数的功能描述 8](#_Toc514061033)

[5. 功能测试 12](#_Toc514061034)

[5.1 初始化界面 12](#_Toc514061035)

[5.2 查询图书信息 12](#_Toc514061036)

[5.3 删除图书信息 15](#_Toc514061037)

[5.4 修改图书信息 16](#_Toc514061038)

[5.5 插入图书信息 18](#_Toc514061039)

[5.6 排序图书 19](#_Toc514061040)

[5.7 借阅图书 21](#_Toc514061041)

[5.8 归还图书 21](#_Toc514061042)

[5.9 保存图书信息 22](#_Toc514061043)

[5.10 退出系统 23](#_Toc514061044)

[5.11 错误输入 23](#_Toc514061045)

[6. 总 结 24](#_Toc514061046)

图书管理系统

图书管理系统是单位进行图书管理的必备工具，它能够对海量的图书信息进行规范，使得管理工作系统化、自动化，提高图书管理工作的效率。图书信息同样需要使用结构体存放。

本实验涉及函数、结构体、链表、文件等方面的知识，学习利用链表处理数据的方法，熟练掌握文件操作，构建综合程序设计的思路及框架，提高综合设计软件系统的能力。

1. 数据格式与文件描述
   1. 数据文件格式

每条图书信息包括图书编号、书名、作者、价格、借阅人编号及借阅标志。多条图书信息存入一个txt文件作为图书馆数据。具体格式如图1-1所示。

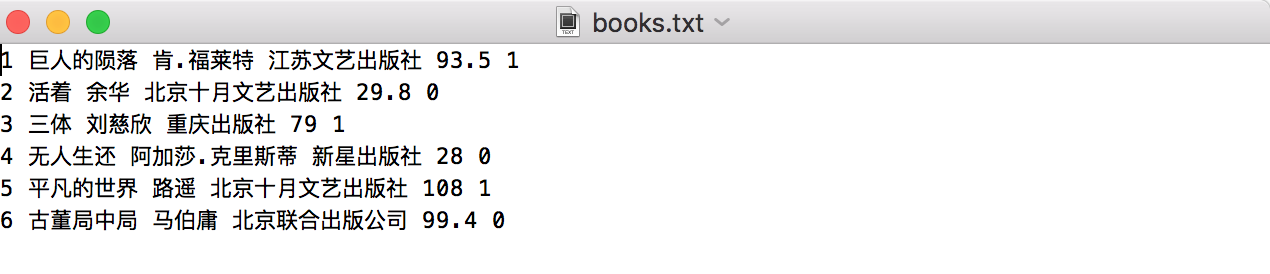


图1-1 图书信息

* 1. 功能描述

设计一个图书管理系统，利用单链表来处理图书信息，要求实现如下系统功能。

1. 初始化菜单

程序开始运行后显示菜单，要求菜单能完全展示本系统的功能，尽力做到设计美观。

(2) 查询图书信息

按图书编号、书名、作者进行部分查询。提示用户输入待查图书的编号、书名或者作者。此外还可以输出全部图书信息。

在单链表中逐个结点查找，若找到该图书，则输出图书信息；若未找到，则输出提示信息。

(3) 删除图书信息

删除指定图书的信息。提示用户输入待删除图书的编号，在单链表中逐个结点查找，若找到该图书，则删除图书信息并提示删除成功；若未找到，则输出提示信息。

(4) 修改图书信息

修改指定图书的信息。提示用户输入待修改图书的编号，在单链表中逐个结点查找. 若找到该图书，选择需要修改的部分，修改图书信息；若未找到，则输出提示信息。

(5) 插入图书信息

插入新的图书信息到指定位置。提示用户输入某个图书编号作为插入点。如果编号存在，提示编号已存在，重新选择编号，若编号不存在，则逐项输入新图书的信息，并设置借阅标志为“未借出”，将新的图书信息结点插入到单链表的末尾。

(6) 图书排序

按图书编号、价格进行排序。提示用输入选择的排序字段，按照用户所选择的字段进行排序，排序结果输出至屏幕上。

(7) 图书借阅

借阅指定图书。提示用户输入所借图书的编号，在单链表中逐个结点査找，若未找到该图书，输出提示信息。若找到该图书，就判断该书的借阅标志，若该书已借出，则输出“已借出”提示信息；若未借出，则修改借阅状态为“已借出”。输出所借图书的信息。

(8) 图书归还

归还指定图书。提示用户输入所归还图书的编号，在单链表中逐个结点查找，找到该图书后，若改书已经借出，则将其借阅标志设为“未借出”，输出提示信息。若该书未借出，输出提示信息。

(9) 保存图书信息

建立一个数据文件，将图书信息从单链表中写入到数据文件中进行保存。

结合以上内容编写源程序，并写出本实验的综合程序设计报告。

* 1. 要求

(1) 源程序编写要求

根据系统功能描述，采用模块化程序设计方法进行程序设计，要求程序结构清晰。上述各个功能模块要求分别用函数实现，在主函数中通过调用这些函数，完成系统功能的要求。代码书写规范，有简要的注释，给出数据和函数说明。

(2) 设计报告撰写要求

设计报告内容包括题目内容和要求、总体设计、详细设计、源代码、调试过程中的问题、总结等。

总体设计：对程序的整体设计思路进行描述，画出图书管理系统的总体功能模块图，说明系统使用的主要数据结构，给出包含上述功能的系统功能模块图。

详细设计：分析实现各函数功能的算法，描述函数的功能。

调试过程中的问题：记录程序编写和调试过程中遇到的各种问题，以及解决这些问题的途径和方法。

总结：回顾整个综合程序设计的过程，对学习到的设计方法和思路进行总结，写出个人体会。

1. 问题分析

图书信息的数据为文本格式，由多个数据项组成。在程序设计中，可以用结构体来存储，由于系统功能涉及图书信息插入和删除，使用链表结构存储信息更有利于插入和删除处理。因此，本实验将图书信息的存储在单链表的结点中，由数据域和指针域构成。数据部分包括：图书编号、书名、作者、价格、借阅人编号及借阅标志共六个成员；指针域存储其直接后继结点的地址。

图书信息的查找、删除、修改、插入、借阅、归还都需要先查找，找到目标结点后再进行相应操作。对图书信息的排序，可采用选择排序、冒泡排序、插入排序等，针对单链表结构的特点，图书管理系统可采用插入排序的方法实现按不同字段的排序。图书借阅时，找到目标结点后，需根据借阅标志判断图书是否借出，再完成后续操作。保存信息后，修改后的图书信息被写入数据文件。

1. 总体设计
   1. 功能模块设计

根据系统功能描述和问题分析，可将系统功能划分为若干模块，如图3-1所示。



图3-1 系统功能模块图

* 1. 系统界面设计

进入图书管理系统后，首先显示系统主菜单，菜单列出8个选项：查询图书信息、删除图书信息、修改图书信息、插入图书信息、图书排序、图书借阅、图书归还、保存图书信息和退出系统。选项编号为1～8，0为退出程序。

用户根据主菜单中显示的功能模块及其相应编号，选择相应编号，执行相应的功能。所选模块执行完毕后，返回系统主菜单。

* 1. 数据结构设计

实验内容涉及到多条记录的处理，每条记录含有不同类型的数据项，使用单链表来存储较为方便，链表结点设计如下：

typedef struct libraryBook{

//图书编号

int id;

//图书名称

string name;

//作者

string author;

//出版社

string press;

//价格

double price;

//是否借阅

bool flag;

//指向下一个book结点的指针

struct book \*next;

} book;

单链表结构中，每个节点的next作为指针域，用于存储其后续节点的地址，其他部分为数据域，存储结点中的数据。

* 1. 函数设计

函数设计的主要内容包括函数的名称，函数的功能与函数中参数类型的说明。

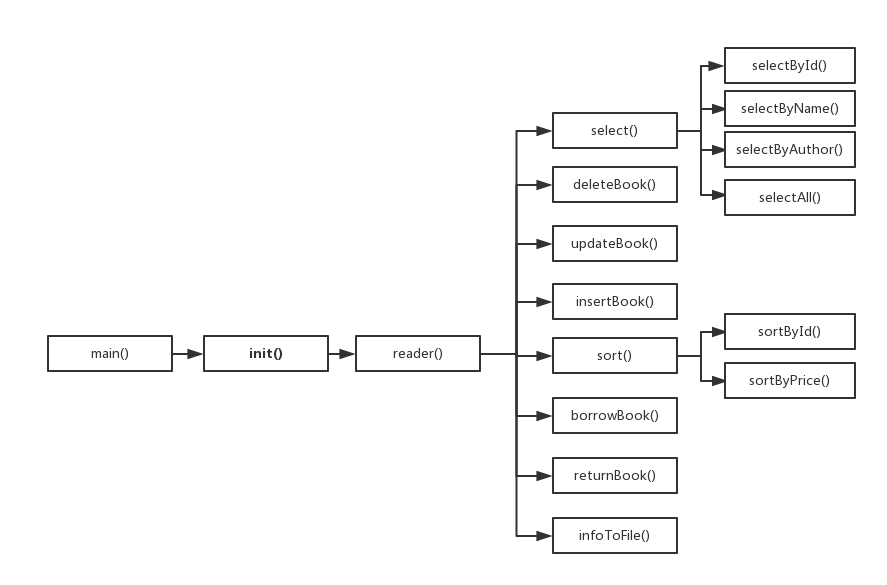
|  |  |  |
| --- | --- | --- |
| 函数名称 | 函数功能 | 参数说明 |
| void selectById(book\* head) | 根据图书的id号查询 | head表示图书链表的头指针 |
| void selectByName(book\* head) | 根据图书的名称查询 | head表示图书链表的头指针 |
| void selectByAuthor(book\* head) | 根据作者查询 | head表示图书链表的头指针 |
| void selectAll(book \*head) | 查询全部书籍 | head表示图书链表的头指针 |
| book\* deleteBook(book\* head) | 删除图书信息 | head表示图书链表的头指针 |
| book\* updateBook(book\* head) | 修改图书信息 | head表示图书链表的头指针 |
| void insertBook(book\* head) | 增加新的图书 | head表示图书链表的头指针 |
| void sortById(book\* head) | 按照编号排序 | head表示图书链表的头指针 |
| void sortByPrice(book\* head) | 按照价格排序 | head表示图书链表的头指针 |
| book\* borrowBook(book\* head) | 图书借阅 | head表示图书链表的头指针 |
| book\* returnBook(book\* head) | 图书归还 | head表示图书链表的头指针 |
| book\* infoToFile(book\* head) | 导出图书信息文件 | head表示图书链表的头指针 |
| void init() | 初始化界面 |  |
| book\* reader(); | 从文件中读取图书信息 |  |
| void printBooks(Book\* head) | 控制台打印输出全部图书信息 | head表示图书链表的头指针 |
| void printBook(Book\* p) | 控制台打印输出某一本图书信息 | p表示某一图书的指针 |

1. 详细设计

详细设计包括的源代码的框架及其功能描述。

* 1. 各个函数的调用关系图

如图4-1所示。



* 1. 各个函数的功能描述

4.2.1初始化系统界面

/\*\*

在控制台界面输出系统名称及其各个功能模块的名称

\*/

void init(){

}

4.2.2 具体功能的实现

/\*\*

查询函数，进入查询功能子菜单

输出选择排序方式

1-按照编号

2-按照名称

3-按照作者

\*/

void select(){

}

/\*\*

根据图书的id号查询

变量定义；

提示用户输入所查询图书的编号

如果存在，输出其图书信息

如果不存在，输出“没有这本书”

\*/

void selectById(book\* head){

}

/\*\*

根据图书的名称查询

变量定义；

提示用户输入所查询图书的名称

如果存在，输出其图书信息

如果不存在，输出“没有这本书”

\*/

void selectByName(book\* head){

}

/\*\*

根据作者查询

变量定义；

提示用户输入所查询图书的作者

如果存在，输出其此作者的所有图书信息

如果不存在，输出“没有这本书”

\*/

void selectByAuthor(book\* head){

}

/\*\*

查询全部书籍

如果存在，输出全部书籍

如果不存在，输出“图书馆里没有书”

\*/

void selectAll(book \*head){

}

/\*\*

删除图书信息

变量定义；

提示用户输入所要删除的图书的编号

如果存在，删除这本书，输出删除成功

如果不存在，输出“没有这本书”

\*/

book\* deleteBook(book\* head){

}

/\*\*

修改图书信息

变量定义；

提示用户输入所要修改的内容，1-书名，2-作者，3-出版社，4-价格，注意此处不能修改其借阅状态

如果存在，修改这本书，输出修改成功与修改后图书的信息

如果不存在，输出“没有这本书”

\*/

book\* updateBook(book\* head){

}

/\*\*

修改图书信息

变量定义；

逐条录入新增图书数据，注意借阅状态为0

如果不存在这本书，输出新增后图书的信息（当然也可以设置图书的数量）

如果存在，输出“已经有这本书了”

\*/

void insertBook(book\* head){

}

/\*\*

排序函数，进入排序功能子菜单

输出选择排序方式

1-按照编号

2-按照名称

3-按照价格

\*/

void sort(){

}

/\*\*

按照编号排序

变量定义；

调用printBooks()函数，输出排序后图书的信息

\*/

void sortById(book\* head){

}

/\*\*

按照价格排序，由低到高

变量定义；

调用printBooks()函数，输出排序后图书的信息

\*/

void sortByPrice(book\* head){

}

/\*\*

图书借阅

提示用户输入图书编号

如果存在且借阅状态为0，将借阅状态置为1

如果存在且借阅状态为1，输出已借出

如果不存在，输出“没有这本书”

\*/

book\* borrowBook(book\* head){

}

/\*\*

图书借阅

提示用户输入图书编号

如果存在且借阅状态为0，将借阅状态置为1

否则置为0

\*/

book\* returnBook(book\* head){

}

/\*\*

将图书信息生成文件并输出

\*/

book\* infoToFile(book\* head){

}

4.2.3 文件读写等工具函数

/\*\*

从文件中读取books

\*/

Book\* reader(){

}

/\*\*

输出books

\*/

void printBooks(Book\* head){

}

/\*\*

输出某一本图书

\*/

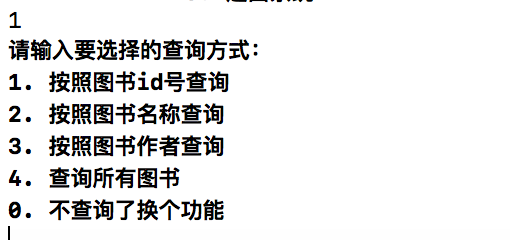
void printBook(Book\* p){

}

1. 功能测试
   1. 初始化界面

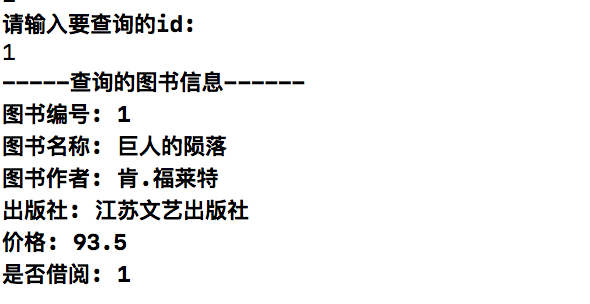


* 1. 查询图书信息

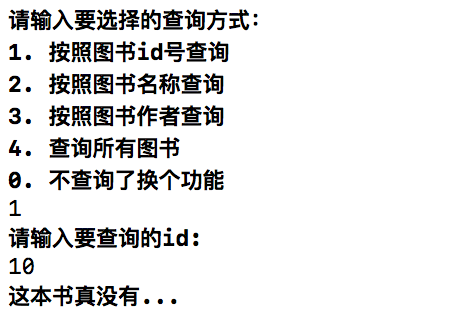


5.2.1 根据图书id查询

图书存在，输出本书信息

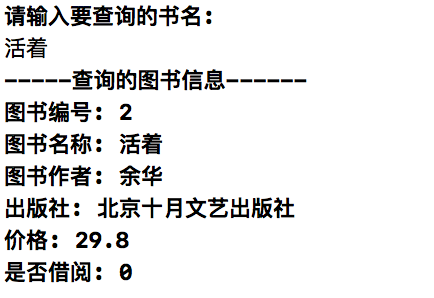


图书不存在，输出提示信息

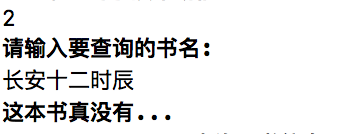


5.2.2 根据图书名称查询

图书存在，输出本书信息

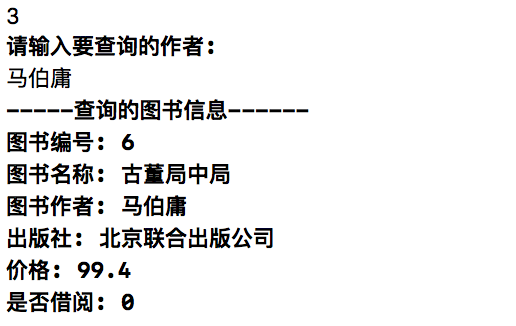


图书不存在，输出提示信息

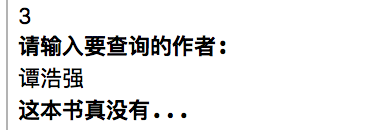


5.2.3 根据作者查询

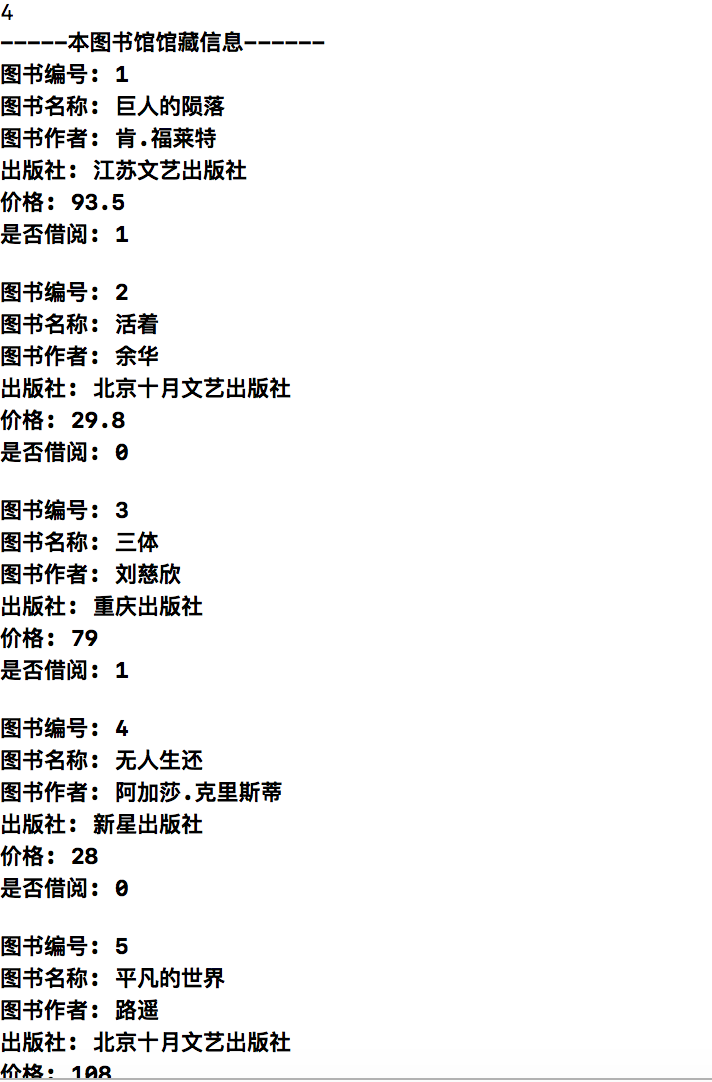
图书存在，输出本书信息



图书不存在，输出提示信息

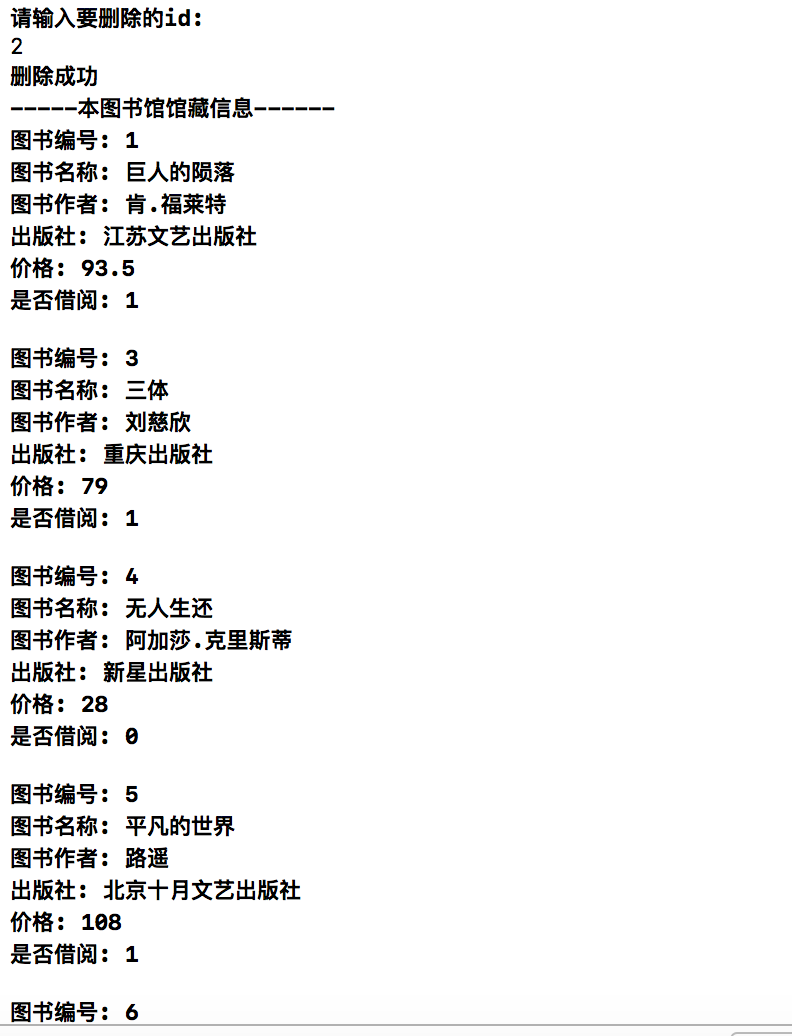


5.2.4 查询全部图书

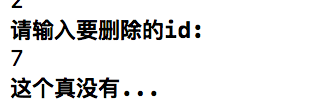


* 1. 删除图书信息

图书存在，删除信息，并输出全部图书。

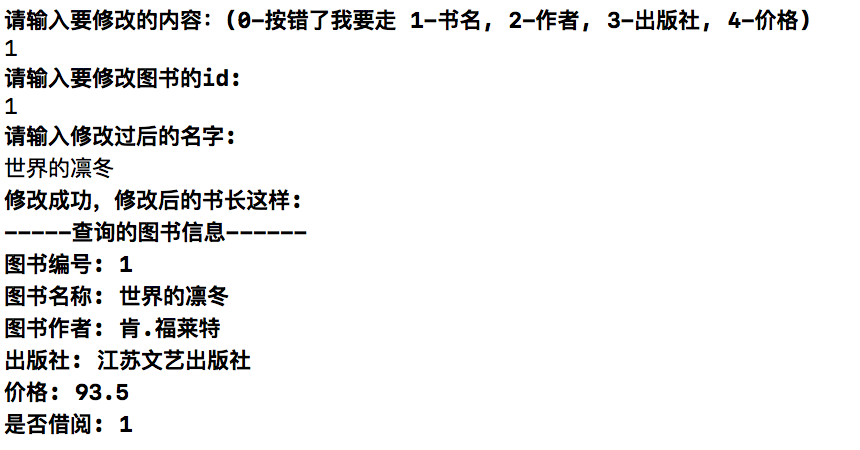


图书不存在，输出提示信息。

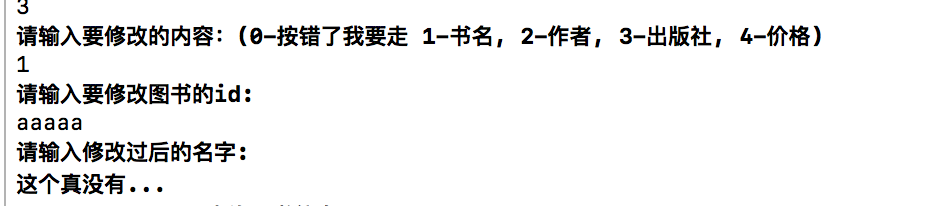


* 1. 修改图书信息

5.4.1 修改书名

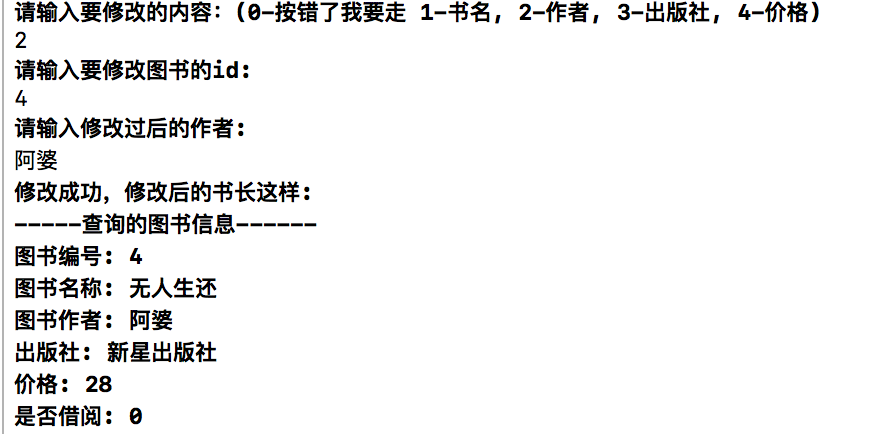


图书不存在，输出提示信息。



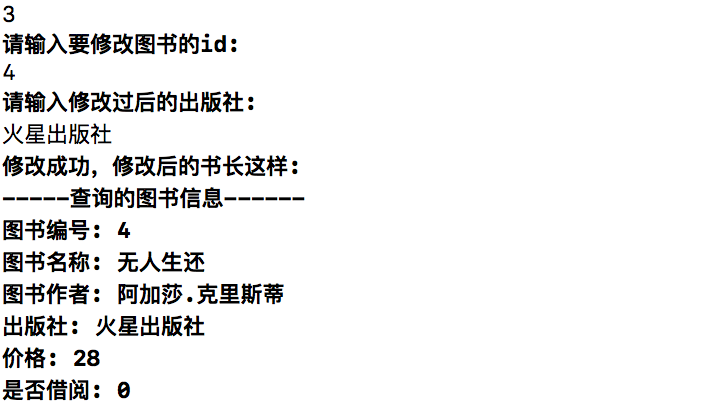
5.4.2 修改作者

图书存在，输出修改后的结果。



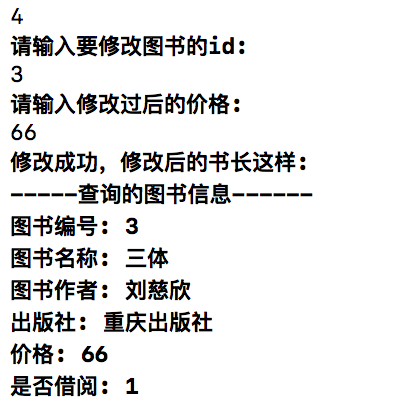
5.4.3 修改出版社

图书存在，输出修改后的结果。



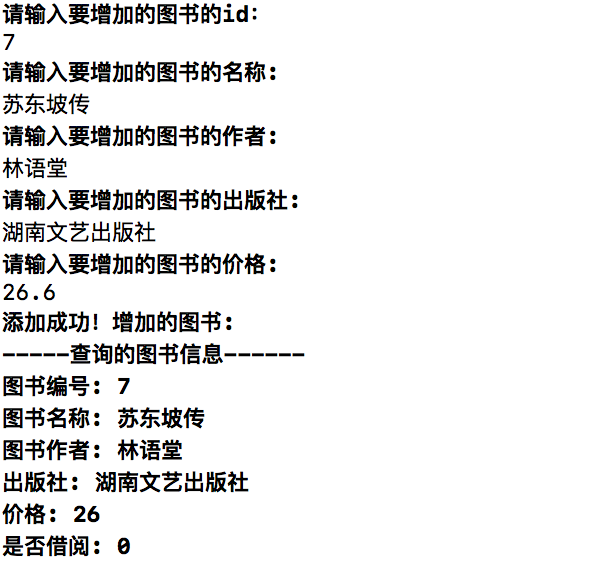
5.4.4 修改价格

图书存在，输出修改后的价格。

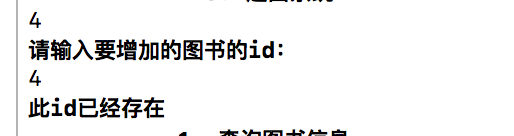


* 1. 插入图书信息

id不存在，增加图书并返回增加的图书信息。

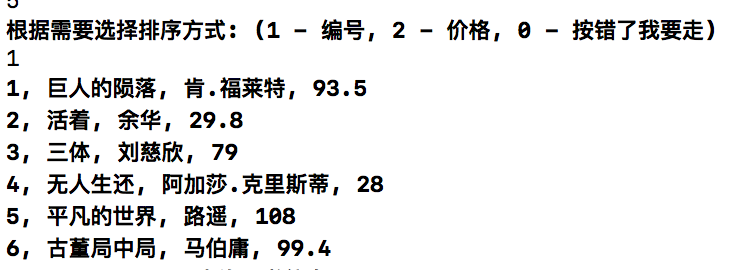
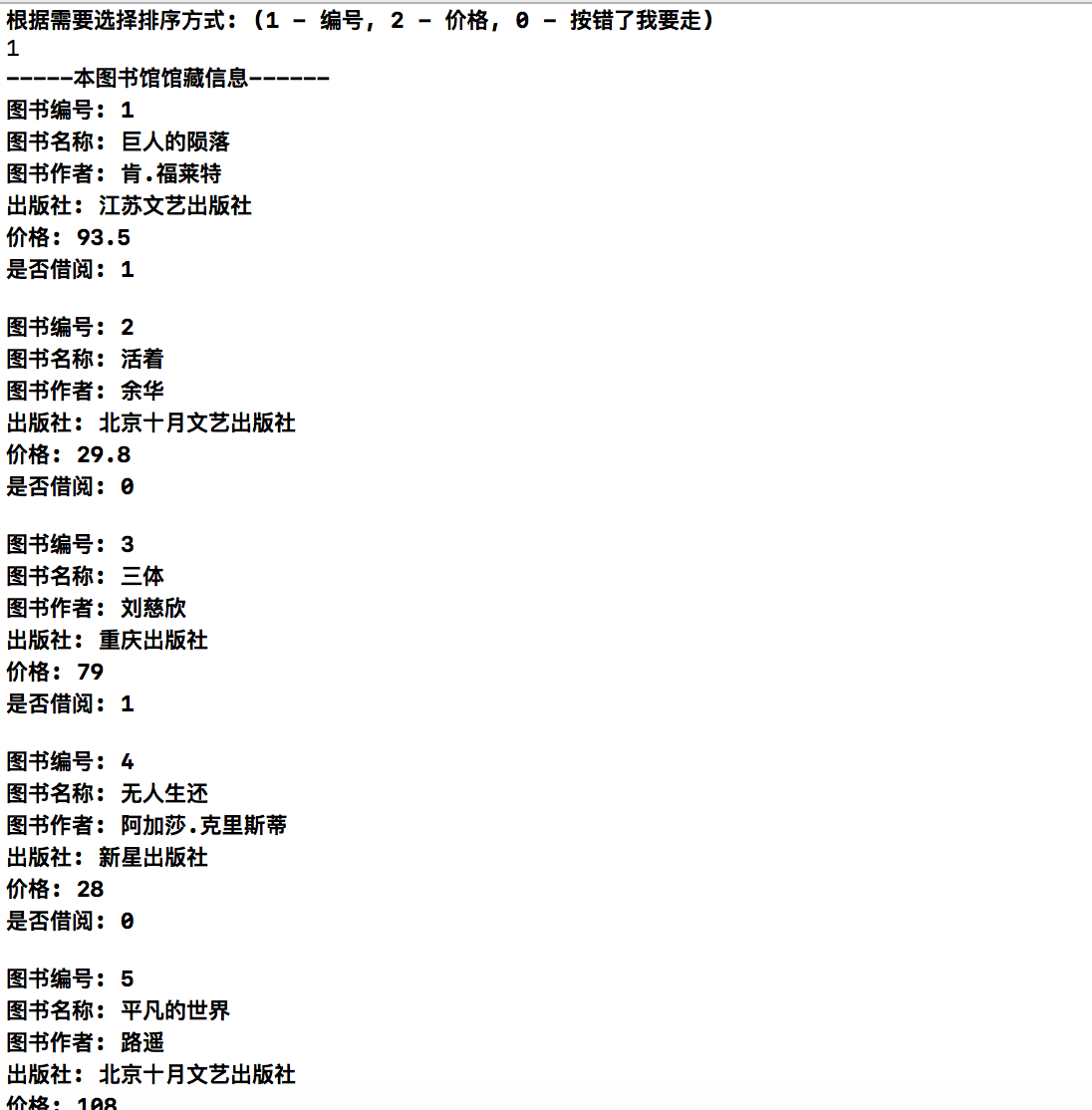


id存在，输出提示信息。

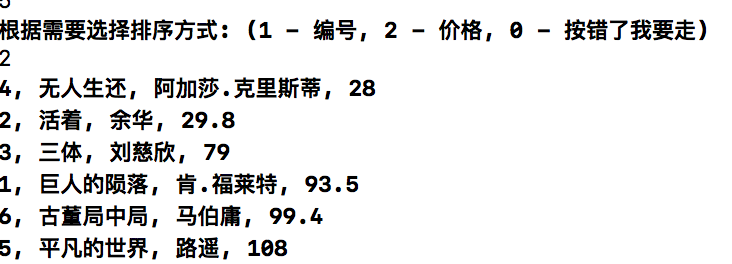


* 1. 排序图书

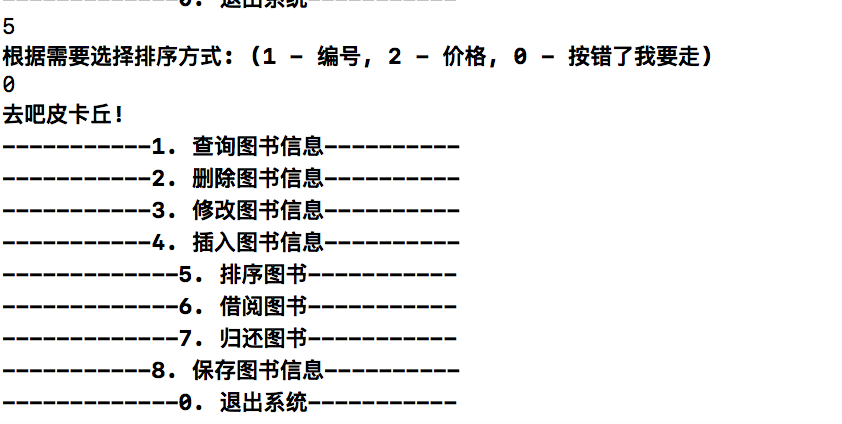
5.6.1 根据id排序

5.6.2 根据价格排序

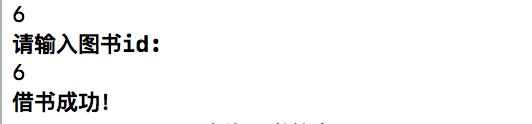


5.6.3 退出排序功能。

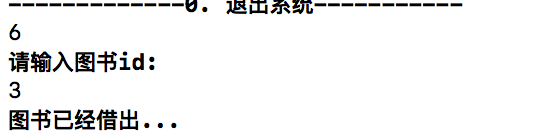


* 1. 借阅图书

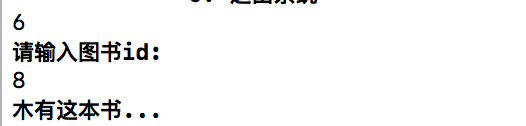
图书存在且并未借阅，输出借书成功。



图书存在且并已借阅，输出提示信息。

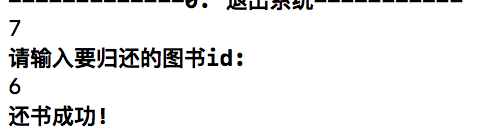


图书不存在，输出提示信息。

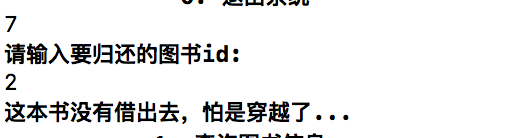


* 1. 归还图书

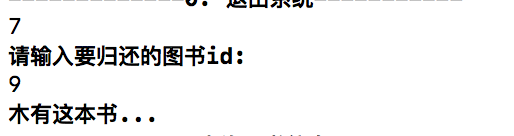
图书存在且借出，输出还书成功。



图书存在但并未借出，输出提示信息。

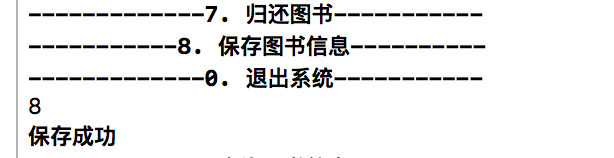


图书不存在，输出提示信息。



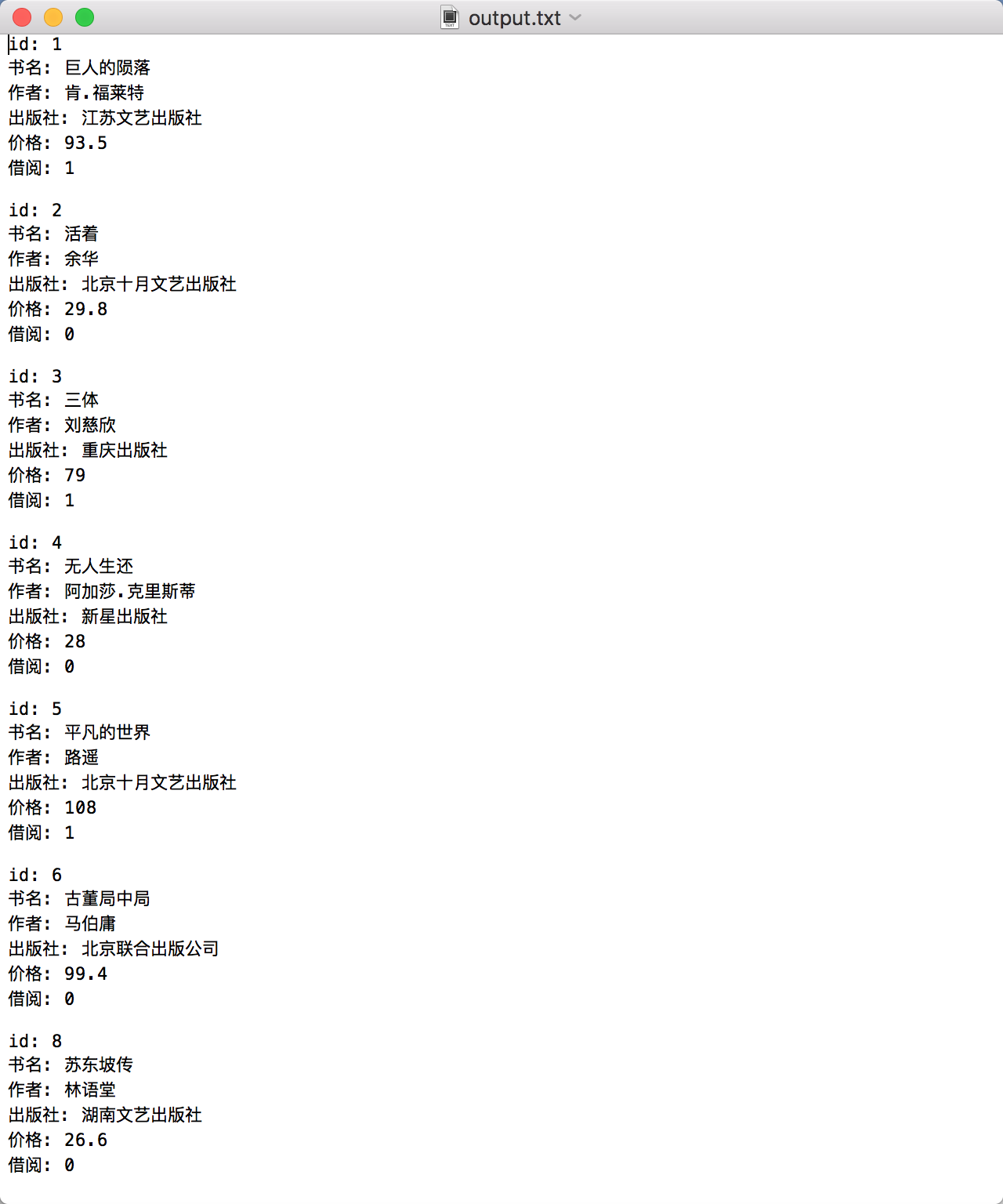
* 1. 保存图书信息

保存图书信息到桌面，并输出保存成功。

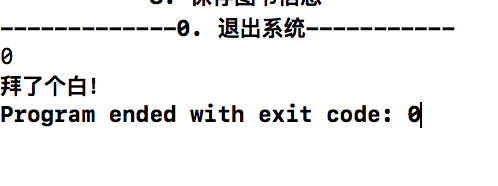


保存的txt文件。



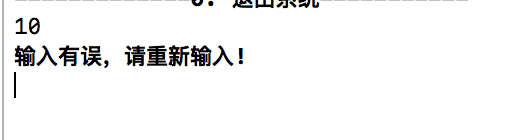


* 1. 退出系统



* 1. 错误输入

提示输入有误，需重新输入。



1. 总 结

主要写自己遇到了哪些问题，是如何解决的，有什么收获，现在的系统有什么缺陷，还可以从哪些角度优化本系统（功能补充、数据项补充、界面优化等）