

程序设计报告

（ 2018 / 2019 学年 第 一 学期）

题 目：**南邮图书馆管理系统**

|  |  |
| --- | --- |
| **专 业** | **自动化** |
| **组长 学号姓名** | **B17050322陈力** |
| **组员 学号姓名** | **B17050324 郝书睿** |
|  | **B17050323 邵侠鸿** |
| **指 导 教 师** | **薛景** |
| **指 导 单 位** | **计算机学院** |
| **日 期** | **2018/11/30** |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **成员分工** | | | | | | | | | |
| 组长（陈力） | | 编写增删修改查找图书代码，汇总各组员的代码，修改代码，组织组内工作，提出架构方式，提供总体编码方向。 | | | | | | | |
| 组员（郝书睿） | | 编写登录函数和\*掩盖密码函数，定义结构体，完善总的控制台操作界面，测试代码并写入报告。 | | | | | | | |
| 组员（邵侠鸿） | | 编写注册函数和控制台界面，完善登录函数，编写注销函数，录入具体图书数据，写除测试程序以外的实验报告。 | | | | | | | |
| **教师评价** | | | | | | | | | |
| 评价准则 | | | | 优秀 | 良好 | | 中等 | 合格 | 不合格 |
| 该课题组掌握程序设计开发的相关工程基础知识，并能够针对求解的工程问题，进行合理的分析与综合。 | | | |  |  | |  |  |  |
| 该课题组能够结合计算机软硬件资源，合理选用算法、数据结构、数据存储方式等技术手段，对求解的工程问题进行系统设计、建模及预测。 | | | |  |  | |  |  |  |
| 该课题组能够选择合适的程序设计语言与编程开发平台，对求解的工程问题进行系统模拟、仿真与预测。 | | | |  | √ | |  |  |  |
| 该课题组熟练掌握调试方法与工具，对程序开发过程中出现的问题进行分析、跟踪与调试，并在开发平台上对最终作品进行充分测试。 | | | |  |  | |  |  |  |
| 该课题组具备初步的软件开发能力，完成了课题要求的各项开发任务，功能实现全面。 | | | |  |  | |  |  |  |
| 该课题组具备初步的界面设计意识。用户使用界面布局合理，交互友好，操作简便。 | | | |  |  | |  |  |  |
| 该课题组正确、完整的回答了指导教师关于课题作品的提问，反映出其对课题内容，以及相关的工程基础知识具有较好的理解和掌握。 | | | |  |  | |  |  |  |
| 该课题组具备一定的表达能力与文字处理能力，能够熟练使用文字处理软件完成课题报告一份。报告完整记录了课题组的工作及总结，内容详实，格式规范。 | | | |  |  | |  |  |  |
| 该课题组具有较好的团队协作精神，能够努力沟通解决遇到的各种开发、工程问题。 | | | |  |  | |  |  |  |
| 该课题组具备一定自学能力与探索创新意识，能够充分利用各种网络与图书馆资源自学新知识与新技能。 | | | |  |  | |  |  |  |
| 该课题组在上机过程中遵守机房规章制度，出勤与平时表现规范。 | | | |  |  | |  |  |  |
| 最终作品和报告反映出该课题组具备较好的解决工程问题的能力。 | | | |  |  | |  |  |  |
| **本次程序设计能力达成评价** | | | | | | | | | |
| 学号姓名 | **B17050322 陈力** | | **B17050324 郝书睿** | | | **B17050323 邵侠鸿** | | | |
| 贡献度 |  | |  | | |  | | | |
| 最终成绩 |  | |  | | |  | | | |
| **备注：**  贡献度：高、较高、中等、较低、低  最终成绩：优秀、良好、中等、及格、不及格 | | | | | | | | | |
| **指导教师： 年 月 日** | | | | | | | | | |

**南邮图书馆管理系统**

**一、课题内容和要求**

南邮图书馆管理系统是一个具有排序、查找、计算、显示等功能的图书馆管理系统，用户能通过查找书名、索引号等分类查找图书并借阅或归还图书，管理员能通过系统增减图书信息。系统能实现对借阅与归还图书的记录，显示所有即将到期的图书并按借阅次数对所有图书进行排序。

1. **需求分析**

南邮图书馆管理系统的功能框架图如图1所示。

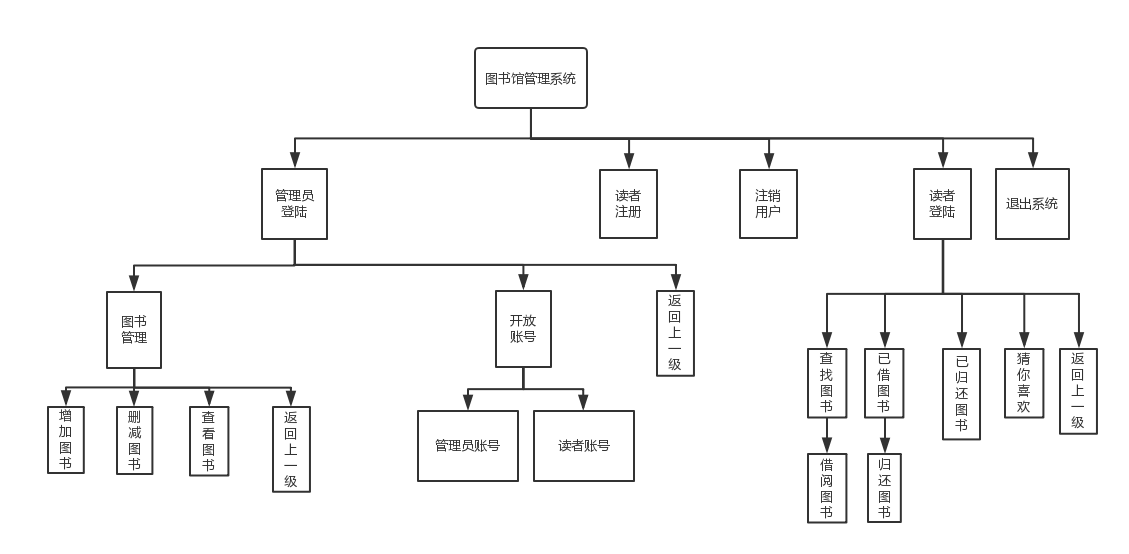


图1 功能框架图

（1）提供可操作的主菜单：输出菜单，用于显示若干个可选的功能选项。根据用户输入的选项来运行不同的功能，运行不同的函数。

（2）提供管理员副菜单：提供增加、删减、查看图书，开放账号等功能。根据用户输入的选项来运行不同的功能，运行不同的函数。

（3）提供读者副菜单：提供查找、借阅、归还图书等功能。根据用户输入的选项来运行不同的功能，运行不同的函数。

（4）在查看图书函数中实现图书已借阅次数排序输出，在读者查看已借图书函数中实现以归还日期排序。

**三、概要设计**

## 1 主要存储结构

主要使用csv文件存储，进行了模拟数据库操作。

实现了主关键字(primary\_key)、外键(foreigh\_key)的功能,及id的auto\_increment，字段的nullable、unique属性。

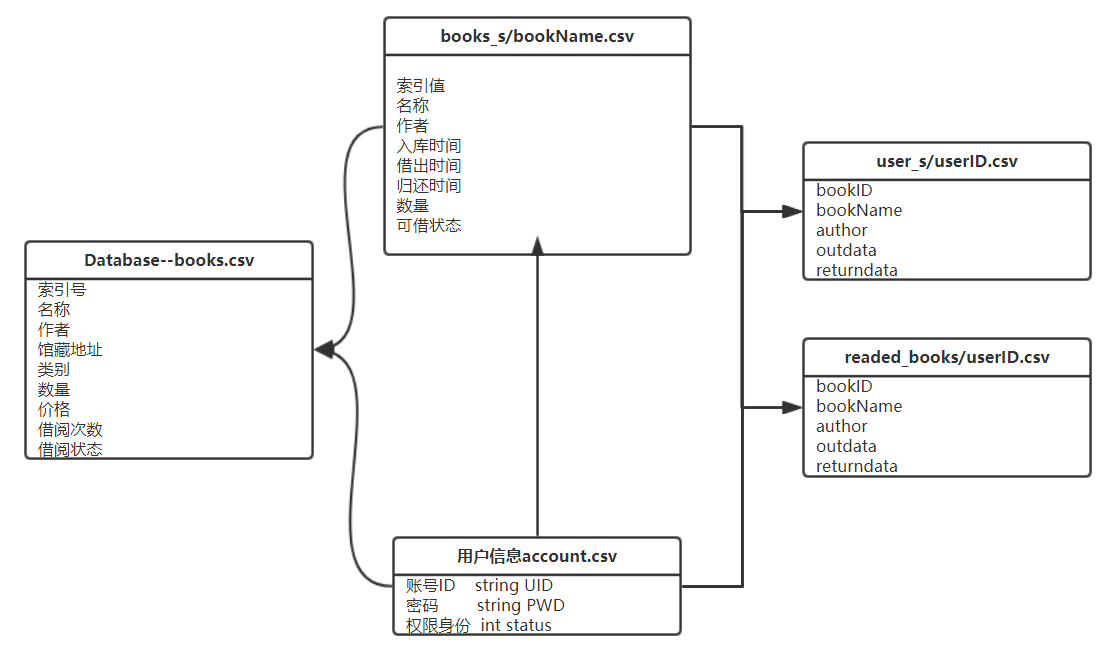


图2.数据存储关系

## 2 主要函数流程

2.1管理员功能：

（1）增书函数：在控制台选择增加图书后，在控制台按照顺序输入图书的信息，系统存储到生成的临时文件tmp.csv中，录入结束后tmp.csv替换掉原书本文件，并更名成原文件名。装载函数流程图如图2所示。

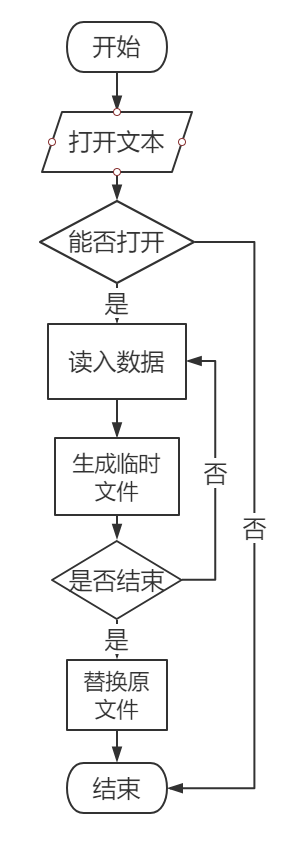


图3 增书函数

1. 删除函数：在控制台选择删除图书后，在控制台查找需删除的图书，系统存储到生成的临时文件tmp.csv中，录入结束后tmp.csv替换掉原书本文件，并更名成原文件名。删除函数流程图如图3所示。

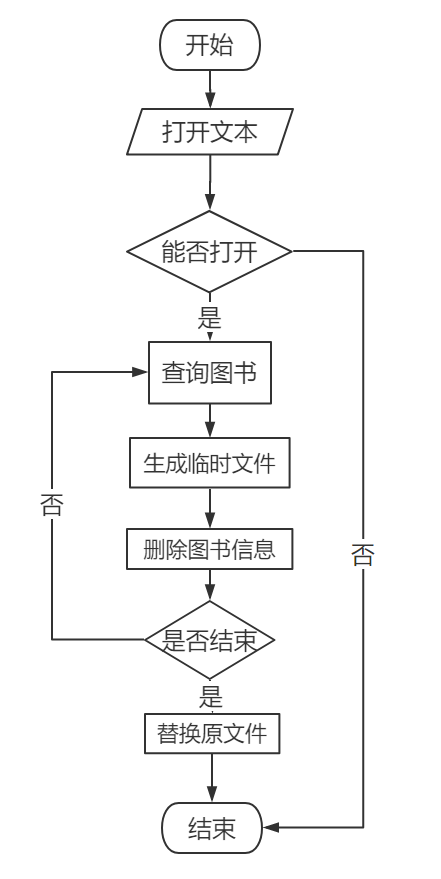


图4 删减函数

1. 查找函数：利用用户输入的书籍名称进行顺序查找并将找到的图书信息输出，查找函数流程图如图4所示。

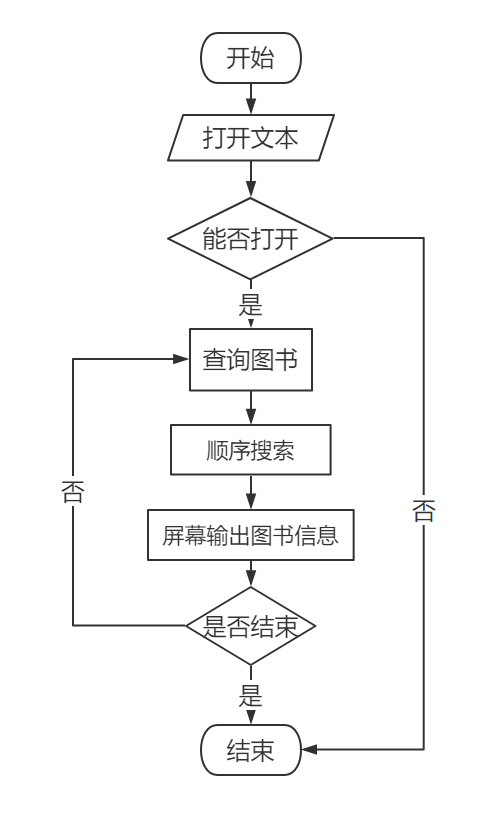


图5 查找函数

1. 排序函数（热门图书）：根据图书的借阅次数进行由大到小的排序展示。

排序函数流程图如图5所示。

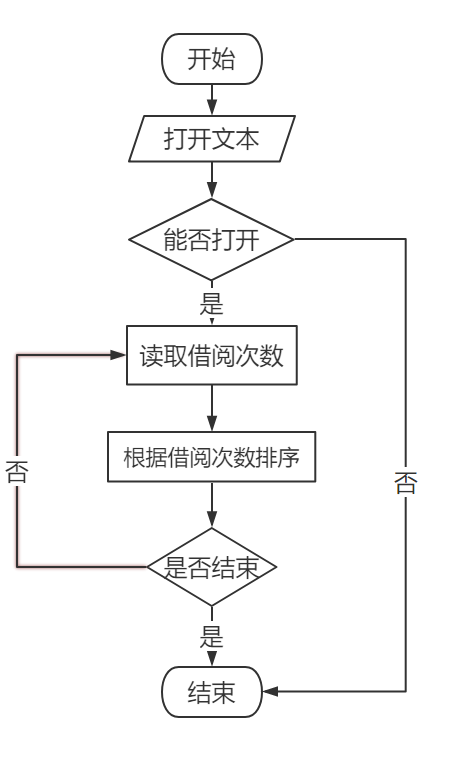


图6 排序函数

（5）推荐函数：通过分析读者最喜欢借的一类图书，给读者推荐相同类型的图书。

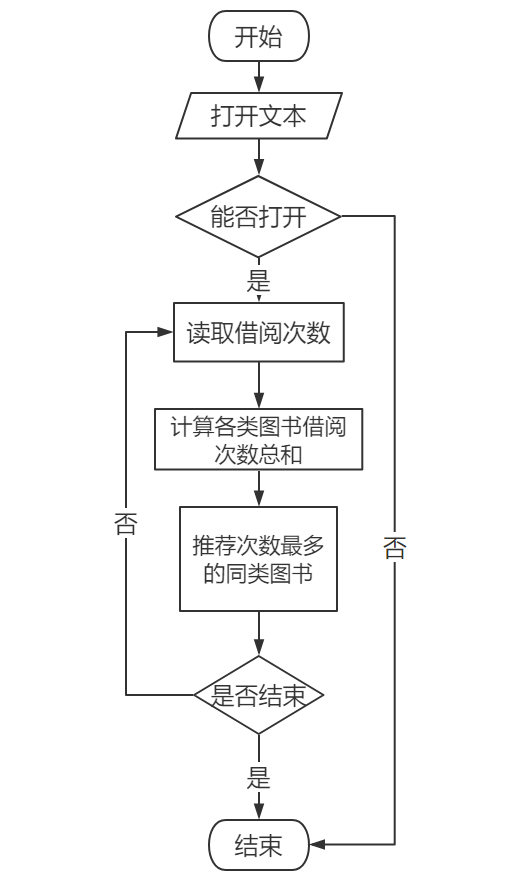


图7 推荐函数

**四、源程序代码**

## 1.头文件

### menu.h

1. /\*显示菜单\*/
2. **void** idMenu();
3. **void**menu();
4. **void**searchBookmenu();
5. **void**administratorMenu();
6. **void**studentMenu();
7. **int**registerStudent();
8. **void**AuthMenu();
9. **void**BOOKsMenu();
10. **void**READERsMenu();
11. /\*\*登录模块\*\*/
12. #include<string>
13. std::stringinputPassword();//掩盖密码
14. **void**logout();//注销登录
15. /\*\*登录模块\*\*/
16. /\*功能函数\*/
17. **int**showDueBooks();
18. **int**showBooksByorder();

### main.h

1. #define OK 1
2. #define FAIL 0
4. #define PASSER 0
5. #define AUTH 1
6. #define COMMONUSER 2
8. /\*\*\*\*\*\*\*\*\*\*PATH define\*\*\*\*\*\*\*\*\*\*\*\*\*/
9. /\*\*书种类.csv\*\*/
10. #define TPYESTMPFILE ".\\sources\\tmp.csv"
11. #define TYPESBOOK ".\\sources\\books.csv" //书总目录的位置
13. /\*\*books\_s下临时文件位置\*\*/
14. #define BOOKPATH ".\\sources\\books\_s\\"
15. #define TMPFILE ".\\sources\\books\_s\\tmp.csv"
17. /\*\*readed\_books下文件位置\*\*/
18. #define READEDBOOKPATH ".\\sources\\readed\_books\\"
20. /\*\*账号信息.csv\*\*/
21. #define ACCOUNTFILE ".\\sources\\account.csv"
22. #define USERPATH ".\\sources\\user\_s\\"
23. #define USERTMPFILE ".\\sources\\user\_s\\tmp.csv"

26. /\*\*PATH define\*/
27. #define RETURNBOOKCOL 5 // <=
28. #define BOOKCOL 8 // <
29. #define TYPESBOOKCOL 8 // <

### File.h

1. #include <string>
2. #include <map>
4. **using** **namespace** std;
5. #include "Classtype.h"
7. **int** confirmStatus(**const** string &UID,**const** string &PWD,CUser & user);
8. **int** readFromcsv(string filename);

11. **bool** addBookdetail(CTypesBook &tmp);
12. **void** update(string &uid,**int** changefield,string& changeValue); //总书
13. **void** updateDetail(string &bookname,**int** changefield,string& changeValue); //删除具体书
14. **void** updateDetail(string &bookname,map<**int**,string> &change) ; //具体书更新==>借阅

### Classtype.h

1. #include <string>
2. **using** **namespace** std;
3. #ifndef \_C\_CLASS\_H
4. #define \_C\_CLASS\_H
6. #include <vector>
8. /\*\*时间日期类\*\*/
9. **class** CData{
10. **public**:
11. **int** year;
12. **int** month;
13. **int** day;
14. **public**:
15. **void** formatTime(string ×tr);
16. string getReturntime(string &borrowtime);
17. **bool** valid(**int** y,**int** m,**int** d);
18. CData(string ×tr);
19. CData(){};
21. **inline** **bool** is\_leap(**int** y) {
22. **return** (y % 4 == 0 && y % 100 != 0) || (y % 400 == 0);
23. }
25. **int** operator-(string ×tr);
26. // string operator+(const CData& d);
27. };
29. /\*\*\*书种\*\*\*\*/
30. **class** CTypesBook{
31. **public**:
32. string BookID; //书ID
33. string BookName; //书名
34. string Author; //作者
35. string Address; //存储位置
36. string Category; //类别
37. **double** Price; //价格
38. **int** TotalNumber; // 总数
39. **int** readNumber; // 阅读次数
40. **int** state; // 借阅状态
41. **public**:
42. **void** update(string &uid,**int** changefield,string& changeValue);
43. };

46. /\*\*\*具体某种书的信息\*\*\*\*/
47. **class** CBook{
48. **public**:
49. string BookID; //根据BOOKID来确定打开的表
50. string BookName; //书名
51. string Author; //作者
52. string Category; //类别
53. string outDate; //借出时间
54. string returnData; //归还时间
55. //int readTimes; //借阅次数
56. //int readState; //可借状态
57. };


61. /\*\*\*\*用户权限\*\*\*\*\*/
62. **class** CUser{
63. **public**:
64. string UID;
65. **int** status;
66. **public**:
67. CUser():UID("0"),status(0){}
68. CUser(string id,**int** \_status):UID(id),status(\_status){}
69. };
71. **class** CcommonUser:**public** CUser{
72. **public**:
73. string UID;
74. **public**:
75. **void** borrowBook(); //显示所有图书的
76. /\*\*\*\*TODO:下一页功能===>存vector二维表,一次显示一点\*\*\*/
78. **void** createBookfile(CBook& book); //以用户ID创建文件
79. **void** returnBook(); //还书,显示自己借的图书
80. **void** lookBook(); //查看借阅了哪些书 ==>到期时间标注
82. **void** addReadedBook(vector<string> line);
83. **void** guessYoulike();
84. **void** readedBook();
86. **void** registeruser();
87. **void** topBooks();
88. };
90. /\*\*\*\*管理员权限\*\*\*\*\*/
91. **class** CAuth: **public** CcommonUser{
92. **public**:
93. string UID;
94. **public**:
95. **void** addBook();
96. **void** deleteBook();
97. **void** searchBook();
98. **void** modifyBook();
100. **void** registeruser();
101. };
103. #endif // \_C\_CLASS\_H

### boostFunction.h

## Cpp文件

1. #define CMD\_COLS 137
2. #define CMD\_LINES 35
4. #include <time.h>
5. #include <string>
6. #include <vector>
7. #include <cstdlib>
8. **using** **namespace** std;
10. **void** SetScreenGrid();
11. **void** SetSysCaption();
13. string getTime(); //获得系统时间
15. **void** renameFilename(string &oldfile,string &newfile);
17. **int** vecMAX(vector<string> &vec);
18. **int** vecSUM(vector<string> &vec);
20. vector< vector<string> > getTablecontent(string &path );
21. **inline** **int** cmp(std::vector< string > v1,std::vector<string> v2){
22. **return** atoi(v1.at(7).c\_str()) > atoi(v2.at(7).c\_str());
23. }
25. **inline** **bool** cmpPair(**const** pair<string, **int**>& lhs, **const** pair<string, **int**>& rhs) {
26. **return** lhs.second > rhs.second;
27. }

### main.cpp : 显示主程序,打印菜单等

1. #include <iostream>
2. #include <cstdio>
3. #include <cstdlib>
4. #include <conio.h> //掩盖密码
5. #include <cstring>
7. /\*自定义头文件\*/
8. #include "menu.h"
9. #include "file.h"
10. #include "main.h"
11. #include "boostFunction.h"
13. #include "Classtype.h"

16. **using** **namespace** std;
18. CUser user ; //声明一个全局的用户类实例
19. CTypesBook typesbook ;
20. CAuth auth ;/\*\*在认证后加\*\*/
21. CcommonUser commmonuser;
22. /\*
23. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
24. \* 系统环境：Windows10 操作系统 \*
25. \* 项目环境：Code Blocks 17.12 \*
26. \* 作者:陈力、郝书睿、邵侠鸿 \*
27. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
28. \*/
30. **int** main(){
31. /\*\*窗口设置\*\*/
32. SetScreenGrid();
33. SetSysCaption();
34. /\*\*窗口设置\*\*/
35. idMenu();
36. **return** 0;
37. }
39. **void** searchBookmenu(){
40. }
42. **int** showDueBooks(){
43. **return** 0;
44. }
46. **int** showBooksByorder(){
47. **return** 0;
48. }

51. /\*\*\*\*菜单模块\*\*\*\*/
53. **void** AuthMenu() {
54. system("cls");
55. printf("\n\n\n\n\n\n\n");
56. printf(" \t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");
57. printf(" \t | ①图书管理 |\n\n");
58. printf(" \t | ②开放账号 |\n\n");
59. printf(" \t | ③返回上一级 |\n\n");
60. printf(" \t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");
61. **int** i;
62. printf("请输入选项[1-3]:");
63. scanf("%d",&i);
64. **while**(i<1||i>3){
65. printf("请重新选择\n");
66. scanf("%d",&i);
67. }
68. **switch**(i){
69. **case** 1:
70. BOOKsMenu();**break**;
71. **case** 2:
72. auth.registeruser();**break**;
73. **case** 3:
74. idMenu();**break**;
75. }
76. }



81. **void** BOOKsMenu() //图书管理
82. {
83. system("cls");
84. printf("\n\n\n\n\n\n\n");
85. printf(" \t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");
86. printf(" \t | ①增加图书 |\n\n"); //未写
87. printf(" \t | ②删减图书 |\n\n");
88. printf(" \t | ③查看图书 |\n\n");
89. printf(" \t | ④返回上一级 |\n\n");
90. printf(" \t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");
91. **int** i;
92. **do**{
93. printf("请输入选项[1-4]:\n");
94. scanf("%d",&i);
95. **while**(getchar()!='\n') ;
96. }**while**(i<1||i>4);
97. **switch**(i){
98. **case** 1:
99. auth.addBook();
100. **break**;
101. **case** 2:
102. auth.deleteBook();
103. **break**;
104. **case** 3:
105. auth.searchBook();
106. **break**;
107. **case** 4:
108. AuthMenu();**break**;
109. }
110. }
111. **void** READERsMenu(){ //读者管理
112. system("cls");
113. printf("\n\n\n\n\n\n\n");
114. printf(" \t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");
115. printf(" \t | ①查找图书借阅 |\n\n");
116. printf(" \t | ②查看已借图书 |\n\n");
117. printf(" \t | ③查看TOP热门图书 |\n\n");
118. printf(" \t | ④猜你喜欢 |\n\n");
119. printf(" \t | ⑤查看已还图书 |\n\n");
120. printf(" \t | ⑥返回上一级 |\n\n");
121. printf(" \t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");
122. **int** i;
123. **do**{
124. printf("请输入选项[1-6]:\n");
125. scanf("%d",&i);
126. **while**(getchar()!='\n') ;
127. }**while**(i<1||i>6);
128. **switch**(i){
129. **case** 1:
130. commmonuser.borrowBook();
131. **break**;
132. **case** 2:
133. commmonuser.lookBook(); //查看后还书
134. **break**;
135. **case** 3: //查看还书功能
136. commmonuser.topBooks();
137. **break**;
138. **case** 4:
139. commmonuser.guessYoulike();
140. **break**;
141. **case** 5:
142. commmonuser.readedBook();
143. **break**;
144. **case** 6:
145. idMenu();
146. **break**;
147. }
148. }
150. /\*\*\*\*菜单模块\*\*\*\*/


154. /\*\*\*\*\*登录模块\*\*\*\*\*/
155. /\*输入账号\*/
156. **void** idMenu(){
157. **int** choice=0;
158. **do**{
159. system("cls");
160. printf(" \t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");
161. // printf("\n\n");
162. printf(" \t 南邮图书馆管理系统\n\n");
163. printf(" \t | ①读者登陆 |\n\n");
164. printf(" \t | ②读者注册 |\n\n");
165. printf(" \t | ③管理员登陆 |\n\n");
166. printf(" \t | ④注销当前用户 |\n\n");
167. printf(" \t | ⑤退出系统 |\n\n");
168. // printf("\n\n");
169. printf(" \t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");
170. // printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\n");
171. /\*\*考虑到退到主界面的问题,直接将printf全放循环里\*\*/
172. printf("请输入你的选项[1-5](Ctrl + c退出):\n");
173. scanf("%d",&choice);
174. **while**(getchar()!='\n') ; //避免有输入 非数值的情况 , 提高程序的鲁棒性和稳定性
175. **switch**(choice){
176. **case** 1:
177. studentMenu();
178. **break**;
179. **case** 2:
180. commmonuser.registeruser();
181. **break**;
182. **case** 3:
183. administratorMenu();
184. **break**;
185. **case** 4:
186. logout();
187. **break**;
188. **case** 5:
189. exit(0);
190. }
191. }**while**( choice != 1 || choice != 2 || choice != 3 || choice != 4|| choice != 5) ;
192. }

195. /\*管理员登录\*/
196. **void** administratorMenu(){
197. string UID;
198. string PWD;
199. **int** confirmed = **false**;
200. //判断是否认证成功
201. **if**( user.status == AUTH ){ //已认证过
202. system("cls");
203. AuthMenu();
204. **return**;
205. }
206. **do**{
207. cout <<"请输入管理员账号ID(输入Q退出):" <<endl;
208. cin >> UID;
209. // cout <<"请输入密码:" <<endl;
210. // cin >> PWD;
211. **if**(UID == "Q") **break**;
212. PWD = inputPassword();
213. confirmed = confirmStatus(UID,PWD,user);
214. **if**( !confirmed ) cout << "\n账号或密码错误,请尝试.账号ID输入'Q'退出" << endl;
215. **else**{ //数据库里有此账号数据,但不是管理员
216. **if**( user.status != AUTH ) {
217. confirmed = PASSER;
218. cout << "这个不是管理员账号请重新输入!" << endl;
219. }
220. }
221. // }while( !confirmed && UID != "Q");
222. }**while**( !confirmed); // UID输入Q时用return;重构
223. **if**(confirmed) {
224. user.status = AUTH;
225. auth.UID = UID;
226. system("cls");
227. AuthMenu();
228. }
229. }
231. /\*普通用户登录\*/
232. **void** studentMenu(){
233. string UID;
234. string PWD;
235. **int** confirmed = **false**;
236. //while(1){ //循环
237. **if**( user.status == COMMONUSER || user.status == AUTH){ //判断是否认证过
238. system("cls");
239. READERsMenu();
240. **return**;
241. }
242. **do**{
243. cout <<"请输入普通账号ID(或管理员账号):" <<endl;
244. cin >> UID;
245. **if**(UID == "Q") **return**;
246. // cout <<"请输入密码:" <<endl;
247. // cin >> PWD;
248. PWD = inputPassword();
250. confirmed = confirmStatus(UID,PWD,user);
251. **if**( !confirmed ) cout << "\n账号或密码错误,请尝试.账号ID输入'Q'退出\n";
252. // }while( !confirmed && UID != "Q");
253. }**while**( !confirmed); // UID输入Q时用return;重构
254. **if**(confirmed) {
255. //cout << "成功认证" << endl;
256. user.status = COMMONUSER;
257. commmonuser.UID = UID;
258. system("cls");
259. READERsMenu(); //进入普通用户界面
260. }
261. //}
262. }
264. **void** logout(){
265. user.status = 0;
266. cout << "注销成功" <<endl;
267. system("pause");
268. }

271. /\*\*\*掩盖密码\*\*\*/
272. string inputPassword(){
273. **int** i = 0;
274. **char** ch;
275. **char** passwd[15] = {0};
276. printf("请输入密码: ");
277. **while**((ch = getch())!='\r'){
278. **if**(i>10) //当密码超过10位
279. { fflush(stdin);
280. system("cls");
281. memset(passwd,0,**sizeof**(passwd));
282. i = 0;
283. printf("请输入不超过十位的密码: ");
284. **continue**;
285. }
286. **if**(ch == '\b'){ //当键入退格键时
287. **if**(i>0){
288. i--;
289. printf("\b \b");
290. passwd[i] = 0;
291. **continue**;
292. }**else**{
293. printf("\a"); //没有内容的时候
294. **continue**;
295. }
296. }**else**{
297. passwd[i] = ch;
298. printf("\*");
299. }
300. i++;
301. }
302. string PWD = passwd;
303. **return** PWD;
304. }
305. /\*\*\*掩盖密码\*\*\*/

308. /\*\*\*\*\*登录模块\*\*\*\*\*/

### file.cpp :对文件的操作,更新表的接口

1. #include <iostream>
2. #include <cstdlib>
3. #include <string>
4. #include <fstream>
5. #include <sstream>
6. #include <map>
8. #include "main.h"
9. #include "file.h"
11. #include "Classtype.h"
12. #include "boostFunction.h"
14. **using** **namespace** std;
16. /\*验证登录者身份信息\*/
17. **int** confirmStatus(**const** string &UID,**const** string &PWD,CUser & user){
18. //int confirmStatus(const string &UID,const string &PWD){
19. ifstream inFile;
20. string rUID;
21. string rPWD;
22. string rstatus;
23. // int confirmed = false;
24. //判断是否认证成功
25. inFile.open(ACCOUNTFILE, ios::in);
26. **if** (inFile.fail()){ //如果读取失败
27. cout << "Cannot open file" << endl;
28. **return** FAIL;
29. }
31. string lineStr;
32. **while** (getline(inFile, lineStr) ){ // 获取一行
33. stringstream ss(lineStr);
34. string str;
35. getline(ss, rUID, ','); // 按照逗号分隔的csv文件,将列取出
36. **if** ( rUID == UID){
37. getline(ss, rPWD, ','); // 取出密码列
38. **if** ( rPWD == PWD){ // 如果密码列相同,则匹配成功
39. // ===>使用 strcmp函数 , strcmp(rUID.c\_str(),UID.c\_str()) 好像并不简单 头文件cstring
40. getline(ss, rstatus, ',');
41. user.UID = rUID;
42. user.status = atoi(rstatus.c\_str());
43. inFile.close();
44. **return** OK;
45. }
46. }**else** **continue**; //如果账号ID不匹配则跳过该行
47. }
48. inFile.close();
49. **return** FAIL; //如果遍历后都没有则返回错误
50. }
52. /\*\*\*\*\*更新总目录\*\*\*\*\*/
53. /\*\*可拓展为根据ID更新\*\*/
54. **void** update(string &uid,**int** changefield,string& changeValue){
55. string filename = TYPESBOOK;
56. vector<vector<string> > row = getTablecontent(filename); //记录行数据
57. **for** (**int** i = 0; i < row.size(); ++i){
58. **if**(row.at(i).at(0) == uid){
59. **if**(changeValue == "delete"){ //删除该行
60. row.erase(row.begin()+i); **break**;
61. }
62. row.at(i).at(changefield) = changeValue;
63. **break**;
64. }
65. }
66. ofstream fout;
67. fout.open(TPYESTMPFILE,ios::out);
68. **for** (**int** i = 0; i < row.size() ; i++) {
69. **for**(**int** j = 0; j < TYPESBOOKCOL; j++) fout << row.at(i).at(j) << "," ; //最后一个","不妨碍
70. fout << endl;
71. }
72. fout.close();
73. string oldname = TPYESTMPFILE;
74. renameFilename(oldname,filename);
75. }
77. /\*\*可拓展为根据书名更新\*\*/
78. **void** updateDetail(string &bookname,map<**int**,string> &change){ //时间\借阅状态
79. string filename = BOOKPATH + bookname + ".csv";
80. vector<vector<string> > row = getTablecontent(filename); //记录行数据
82. **for** (**int** i = 0; i < row.size(); ++i){
83. **if**(row.at(i).at(1) == bookname){
84. **for**(map<**int**,string>::iterator iter = change.begin(); iter != change.end(); ++iter) //删除
85. row.at(i).at(iter->first) = iter->second;
86. **break**;
87. }
88. }
90. ofstream fout;
91. fout.open(TMPFILE,ios::out);
92. **for** (**int** i = 0; i < row.size() ; i++) {
93. **for**(**int** j = 0; j < TYPESBOOKCOL; j++) fout << row.at(i).at(j) << "," ; //最后一个","不妨碍
94. fout << endl;
95. }
96. fout.close();
97. string oldname = TMPFILE;
98. renameFilename(oldname,filename);
99. }
101. /\*\*\*删除图书\*\*\*/
102. **void** updateDetail(string &bookname,**int** changefield,string& changeValue){
103. string filename = BOOKPATH + bookname + ".csv";
104. vector<vector<string> > row = getTablecontent(filename); //记录行数据
105. **if**(row.size() == 1) {
106. remove(filename.c\_str());
107. **return** ;
108. }
109. **for** (**int** i = 0; i < row.size(); ++i){
110. **if**(row.at(i).at(0) == bookname){
111. **if**(changeValue == "delete"){
112. row.erase(row.begin()+i); **break**;
113. }
114. row.at(i).at(changefield) = changeValue;
115. **break**;
116. }
117. }
118. ofstream fout;
119. fout.open(TMPFILE,ios::out);
120. **for** (**int** i = 0; i < row.size() ; i++) {
121. **for**(**int** j = 0; j <= TYPESBOOKCOL; j++) fout << row.at(i).at(j) << "," ;
122. //最后一个","不妨碍
123. fout << endl;
124. }
125. fout.close();
126. string oldname = TMPFILE;
127. renameFilename(oldname,filename);
128. }
130. /\*\*\*具体书添加\*\*\*/
131. **bool** addBookdetail(CTypesBook &tmp){
133. fstream bookfile;
134. string path = BOOKPATH;
135. string filename = path + tmp.BookName + ".csv";
136. bookfile.open(filename.c\_str(),ios::app );
138. **if** (bookfile.fail()){ //如果读取失败
139. **return** **false**;
140. }
141. /\*\*是否添加表头\*\*/
142. vector< vector<string> > row = getTablecontent(filename);
143. **if**(row.empty()) bookfile << "索引值,名称,作者,入库时间,借出时间,归还时间,借阅次数,可借状态" << endl;
144. /\*\*是否添加表头\*\*/
145. **for**(**int** i=1;i <= tmp.TotalNumber ;i++){
146. stringstream ss;
147. ss << i;
148. //string bookid = tmp.BookID + "/" + to\_string(1); //Codeblock里C++标准非C++11无法使用to\_string
149. string bookid = tmp.BookID + "/" + ss.str();
150. string addtime = getTime();
151. string borrowTime ="#";
152. string returnTime ="#";
153. //用类完成会不会好点?
155. bookfile << bookid << ',' << tmp.BookName << ',' << tmp.Author << ','<< addtime << ',' <<
156. borrowTime << ',' << returnTime << ',' << "0" << ',' << "1" << endl;
157. //书的ID,书名,入库时间,借出时间,归还时间,节约次数,是否可借
159. }
160. bookfile.close();
161. **return** **true**;
162. }

### Classtype.cpp :类方法及主要函数

1. #include <iostream>
2. #include <fstream>
3. #include <string>
4. #include <sstream>
5. #include <cstdlib>
6. #include <algorithm>
7. #include <cstring>
9. #include "Classtype.h"
10. #include "main.h"
11. #include "boostFunction.h"
12. #include "file.h"
13. #include "menu.h"
14. **using** **namespace** std;
16. **extern** CcommonUser commmonuser;

19. /\*\*管理员CAuth成员函数\*\*/
20. /\*\*注册账号\*\*\*/
21. **void** CAuth::registeruser(){ //todo
22. **char** tep[20];
23. **char** user[20];
24. **char** password[20];
25. **char** yesorno='Y';
26. **int** i;
27. **FILE** \*fp;
28. system("cls");
29. printf("\t\t\t\t注册\n\n\n");
30. printf("\t\t\t请设置用户名：");
31. scanf("%s",user);
32. fp=fopen(ACCOUNTFILE,"r");
34. **do**{
35. printf("\n\t\t\t请设置密码：");
36. scanf("%s",password);
38. printf("\t\t\t请确认密码：");
39. scanf("%s",tep);
41. **while**(getchar() != '\n');
42. **do**{
43. printf("\t\t\t账号是否为管理员(YorN)：");
44. scanf("%c",&yesorno);
45. }**while**(yesorno != 'Y' && yesorno != 'N');

48. **if**(strcmp(tep,password)!=0)
49. printf("\n\t\t\t两次密码不一致请重新输入");
50. }**while**(strcmp(tep,password)!=0);
51. fp=fopen(ACCOUNTFILE,"a");
52. **if**(fp==NULL){
53. printf("\n\t注册失败\n");
54. exit(0);
55. }
56. **else**{
57. fputs(user,fp);
58. // char ch= ',';
59. fputc(',',fp);
60. fputs(password,fp);
61. fputc(',',fp);
62. **if**( yesorno=='Y' )fputc('1',fp);
63. **else** fputc('2',fp);
64. printf("\n\t注册成功");
65. }
66. fclose(fp);
67. system("pause");
68. }

71. **void** CAuth::addBook(){
72. fstream infile;
73. CTypesBook tmp;
74. infile.open(TYPESBOOK, ios::out| ios::in );
75. **if**(infile.fail()) {
76. cout << "添加失败" << endl;
77. **return** ;
78. }
80. vector<string> idcol; //找到最大的ID+1
81. vector<string> namecol; //不重复
82. string lineStr;
83. **while** (getline(infile, lineStr) ){ //获取行
84. stringstream ss(lineStr);
85. string bookid; //记录列数据,必须写在这个循环里
86. getline(ss, bookid, ',');
87. idcol.push\_back(bookid);
88. string bookname ;
89. getline(ss, bookname, ',');
90. namecol.push\_back(bookname);
91. }
92. **int** addid = vecMAX(idcol)+1;
93. stringstream inttostring;
94. inttostring << addid;
95. tmp.BookID = inttostring.str();
97. // cout << "请输入图书ID：" << endl;
98. // cin >> tmp.BookID ;
99. **int** first = 1;
100. **do**{
101. **if**(first == 1){
102. cout << "请输入图书名称：" << endl;
103. cin >> tmp.BookName ;
104. first = 0;
105. }**else**{
106. cout << "重复,请重新输入书名:" << endl;
107. cin >> tmp.BookName ;
108. }
109. }**while**(find(namecol.begin(),namecol.end(),tmp.BookName) != namecol.end());
110. //唯一书名不重复=>不能一次性增加==>单个增加
111. cout << "请输入书的作者：" << endl;
112. cin >> tmp.Author ;
113. cout << "请输入馆藏地址：" << endl;
114. cin >> tmp.Address ;
115. cout << "请输入类别：" << endl;
116. cin >> tmp.Category ;
117. **do**{
118. cout << "请输入书的数量：" << endl;
119. scanf("%d",&tmp.TotalNumber);
120. **while**(getchar()!='\n') ;
121. }**while**( tmp.TotalNumber == 0 ) ; //默认为0,非数字情况时不变
123. **do**{
124. cout << "请输入书的价格：" << endl;
125. scanf("%lf",&tmp.Price);
126. **while**(getchar() != '\n') ;
127. }**while**( tmp.Price == 0 ) ;
128. infile.clear();
129. infile << tmp.BookID << ',' << tmp.BookName << ',' <<
130. tmp.Author << ',' << tmp.Address << ','
131. << tmp.Category << ',' <<tmp.TotalNumber << ',' << tmp.Price << ',' << "0" << ',' << "1" << endl;
133. infile.close();
135. **if** (addBookdetail(tmp)) cout << "添加完毕" << endl;
136. **else** cout << "添加失败，请咨询后台" << endl;
137. system("pause");
138. }

141. /\*\*\* TODO: 将TypesBook里的可借阅状态和数量修改\*\*\*/
142. **void** CAuth::deleteBook(){
143. vector<vector<string> > row; //记录行数据
144. string PATH = BOOKPATH;
145. string searchname;
146. string Inputsearchname;
147. string id; //删除书id
148. **do**{ //多次输入
149. cout << "请输入书名,按Q退出查询:\n";
151. cin >> Inputsearchname;
152. searchname = PATH + Inputsearchname +".csv";
153. CTypesBook tmp;
154. ifstream infile;
156. infile.open(searchname.c\_str(),ios::in);
157. **if**(infile.fail()) {
158. **if**(Inputsearchname != "Q")
159. cout << "不存在该书,请重试" <<endl;
160. }
161. **else**{
162. string lineStr;
163. //printf("%10s\t%10s\t%10s\t%10s\t%10s\t%10s\t%10s\t\n",
164. // "索引值","名称作者","入库时间","借出时间","归还时间","数量","可借状态") ;
165. //表头
166. **while** (getline(infile, lineStr) ){ //获取行
167. stringstream ss(lineStr);
168. string str;
169. vector<string> col; //记录列数据,必须写在这个循环里
170. **while** (getline(ss, str, ',')) {
171. printf("%-8s\t",str.c\_str());
172. col.push\_back(str);
173. }
174. row.push\_back(col);
175. cout << endl;
176. }
178. infile.close();
179. **bool** flag = **false**;
180. **if**( !row.empty() ){
181. **while**( !flag ){
182. cout << "请输入要删除的书本ID:" << endl;
183. cin >> id;
184. **if**(id == "Q") **return**;
185. **for** (**int** i = 1; i < row.size(); i++){ /\*这边不能写int rowSize = row.size();否则会影响循环遍历\*/
186. **if** ( row.at(i).at(0) == id ){
187. row.erase(row.begin()+i);
188. flag = **true**;
189. }
190. }
191. }
192. }
193. ofstream out;
194. out.open(TMPFILE,ios::out);
195. **int** rowSize = row.size();
196. **for** (**int** i = 0; i < rowSize ; i++) {
197. **for**(**int** j = 0; j < BOOKCOL; j++) out << row.at(i).at(j) << "," ;
198. out << endl;
199. }
200. out.close();
201. string oldname = TMPFILE;
202. renameFilename( oldname, searchname);
203. cout << "删除成功!" << endl;
205. stringstream ss(id);
206. string bookid;
207. getline(ss,bookid,'/');
208. cout << row.size();
209. **if**( row.size() !=1 ){ //最后一本的情况
210. **int** booklef = row.size() - 1;
211. stringstream ss;
212. ss << booklef;
213. string num = ss.str();
214. update( bookid,5,num );
215. }**else** {
216. string command = "delete";
217. update( bookid,5,command);
218. }
219. }
220. }**while**(Inputsearchname != "Q");
221. system("pause");
222. }
224. /\*\*\* 查找指定书本,根据书名-->准确: \*\*\*/
225. /\*\*TODO:用contains...\*/
226. /\*\*\*TODO:通过ID查\*\*\*/
227. **void** CAuth::searchBook(){
228. string Inputsearchname;
229. **do**{ //多次输入
230. cout << "请输入书名,按Q退出查询:\n";
232. cin >> Inputsearchname;
233. string PATH = BOOKPATH;
234. string searchname = PATH + Inputsearchname +".csv";
235. CTypesBook tmp;
236. ifstream infile;
237. **if**(Inputsearchname == "Q") **return**;
239. infile.open(searchname.c\_str(),ios::in);
240. **if**(infile.fail()) {
241. cout << "不存在该文件,按任意键返回上级" ;
242. system("pause");
243. }
244. **else**{
245. string lineStr;
246. //printf("%10s\t%10s\t%10s\t%10s\t%10s\t%10s\t%10s\t\n","索引值","名称作者","入库时间","借出时间","归还时间","数量","可借状态") ;
247. //表头
248. **while** (getline(infile, lineStr) ){
249. stringstream ss(lineStr);
250. string str;
251. **while** (getline(ss, str, ',')) printf("%-8s\t",str.c\_str());
252. cout << endl;
253. }
254. system("pause");
255. **return**;
256. }
257. infile.close();
258. }**while**(Inputsearchname != "Q");
259. }


263. /\*\*\* TODO: \*\*\*/
264. //void CAuth::modifyBook(){
265. //vector<vector<string>> strArray; //二维vec存表数据
266. // CAuth::searchBook();
267. //
268. //}

271. /\*\*CcommonUser的成员函数实现\*\*/
272. // 借书-->具体和总书的借阅次数++
273. // 借书-->具体书的归还时间
274. // 借书-->总书里的可借阅状态
275. **void** CcommonUser::borrowBook(){
276. ifstream in;
277. string lineStr;
279. in.open(TYPESBOOK,ios::in);
280. **if**(in.fail()){
281. cout << "无法借阅" <<endl;
282. **return** ;
283. }
284. **while** (getline(in, lineStr) ){ //获取行
285. stringstream ss(lineStr);
286. string str;
287. **while** (getline(ss, str, ',')) {
288. printf("%-8s\t",str.c\_str());
289. }
290. cout << endl;
291. }
292. in.close();

295. /\*\*修改books\_s里面csv\*\*/
296. vector<vector<string> > row; //记录行数据
297. vector<string> canVec; //具体书里是否可借
298. vector<string> readtimesVec;
300. string Inputsearchname;
301. **do**{ //多次输入
302. cout << "请输入书名,按Q退出查询:\n";
304. cin >> Inputsearchname;
305. string PATH = BOOKPATH;
306. string searchname = PATH + Inputsearchname +".csv";
307. CTypesBook tmp;
308. ifstream infile;
310. infile.open(searchname.c\_str(),ios::in);
311. **if**(infile.fail()) {
312. **if**(Inputsearchname != "Q")
313. cout << "不存在该文件" <<endl;
314. }
315. **else**{
316. //printf("%10s\t%10s\t%10s\t%10s\t%10s\t%10s\t%10s\t\n",
317. // "索引值","名称作者","入库时间","借出时间","归还时间","数量","可借状态") ;
318. //表头
319. **while** (getline(infile, lineStr) ){ //获取行
320. stringstream ss(lineStr);
321. string str;
322. vector<string> col; //记录列数据,必须写在这个循环里
323. **while** (getline(ss, str, ',')) {
324. printf("%-8s\t",str.c\_str());
325. col.push\_back(str);
326. }
327. row.push\_back(col);
328. cout << endl;
329. }
330. string id;
332. **bool** flag = **false**;
333. **while**( !flag ){
334. cout << "请输入要借阅的书本ID(按Q退出):" << endl;
335. cin >> id;
336. **if**(id == "Q") **return**;
337. **for** (**int** i = 1; i < row.size(); i++){ /\*这边不能写int rowSize = row.size();否则会影响循环遍历\*/
338. canVec.push\_back(row.at(i).at(7)); //借阅状态
339. readtimesVec.push\_back(row.at(i).at(7)); //借阅次数
341. **if** ( row.at(i).at(0) == id ){
342. **if**( row.at(i).at(7) == "0")
343. cout << "当前书不可借,请重新选择" << endl;
344. **else**{ //可借时才能
345. row.at(i).at(4) = getTime();
346. CBook cbook; //声明个类好传参
347. cbook.BookID = row.at(i).at(0);
348. cbook.BookName = row.at(i).at(1);
349. cbook.Author = row.at(i).at(2);
350. // cbook.Category = row.at(i).at(3);
351. cbook.outDate = row.at(i).at(4);
352. //归还时间
353. CData borrowtime;
354. //cout << "borrowtime.getReturntime(row.at(i).at(5))" << borrowtime.getReturntime(row.at(i).at(4));
355. cbook.returnData = borrowtime.getReturntime(row.at(i).at(4));
356. row.at(i).at(5) = cbook.returnData;
357. /\*\*借阅次数\*\*/
358. **int** times = atoi(row.at(i).at(6).c\_str())+1;
359. stringstream ss;
360. ss << times;
361. row.at(i).at(6) = ss.str();
362. row.at(i).at(7) = "0"; //借阅状态
364. CcommonUser::createBookfile(cbook);
365. flag = **true**;
366. }
367. }
368. }
369. }
370. ofstream out;
371. out.open(TMPFILE,ios::out); //生成一个临时的文件来将新的数据写入
372. **int** rowSize = row.size();
373. **for** (**int** i = 0; i < rowSize ; i++) {
374. **for**(**int** j = 0; j < BOOKCOL; j++) out << row.at(i).at(j) << "," ;
375. out << endl;
376. }
377. out.close();
378. infile.close();
379. string oldname = TMPFILE;
380. renameFilename(oldname,searchname); //将临时文件替换为原来的.csv
381. cout << "借书成功!" << endl;

384. **char** num[20];
385. string readtime = itoa(vecSUM(readtimesVec),num,10 );
386. update(id,7,readtime);
388. **if** ( std::find(canVec.begin(), canVec.end(), "1") == canVec.end() ) //没找到可以借的
389. {
390. string no = "0";
391. update(id,8,no);
392. }
393. }
394. }**while**(Inputsearchname != "Q");
395. }

398. /\*\*用户在自己的下面创建借书记录:增加和删除===>TODO:借阅过的书,多加一个字段(Boolreturn)\*\*/
399. **void** CcommonUser::createBookfile(CBook &book){
400. fstream f;
401. string PATH = USERPATH;
402. string filename = PATH + **this**->UID + ".csv";
403. f.open(filename.c\_str(),ios::app);
404. **if**(f.fail()){
405. cout << "无法借阅" <<endl;
406. **return** ;
407. }
408. f << book.BookID << "," << book.BookName << "," << book.Author << ","
409. <<book.outDate << "," << book.returnData <<endl;
410. f.close();
411. }
413. **void** CcommonUser::lookBook(){
414. fstream f;
415. string PATH = USERPATH;
416. string filename = PATH + **this**->UID + ".csv";
417. string bookname; //更新具体书时需要
418. f.open(filename.c\_str(),ios::in);
419. **if**(f.fail()){
420. cout << "没有借阅信息,按任意键返回上级" <<endl;
421. system("pause");
422. READERsMenu();
423. **return** ;
424. }
425. f.close();
427. vector< vector<string> > row = getTablecontent(filename);
429. **if**( row.empty() ){ //为空情况
430. cout << "没有借过书,请找点书读读" <<endl;
431. system("pause");
432. **return** ;
433. }**else**{
434. string bookid;
435. **int** rowsize = row.size();
436. /\*\*\*\*\*三天紧急\*\*\*\*\*/
437. printf("%-8s\t%-8s\t%-8s\t%-8s\t%-8s\n", "索引值","书名","作者","借入时间","归还时间");
438. **for**(**int** i=0;i< rowsize;i++){
439. string shouldReturn = row.at(i).at(4);
440. CData shouldtime(shouldReturn);
441. string now = getTime();
442. **int** lefttime = shouldtime - now;
444. **if**( lefttime < 4 ) {
445. **if**( lefttime > 0) {
446. stringstream ss;
447. ss << lefttime;
448. row.at(i).at(4) = "还有" + ss.str() + "天,快!";
449. }
450. **else** **if**(lefttime ==0) row.at(i).at(4) = "请在当天归还,快!";
451. **else** {
452. **double** order = -0.1\*lefttime; //欠费账单
453. stringstream ss;
454. ss << order;
455. row.at(i).at(4) = "已经欠费" + ss.str() +"元";
456. }
457. }
458. }
459. //输出借阅书信息
460. **for** (**int** i = 0; i < rowsize ; ++i){
461. **int** colsize = row.at(i).size();
462. **for** (**int** j = 0; j < colsize ; ++j)
463. printf("%-8s\t", row.at(i).at(j).c\_str());
464. cout << endl;
465. }
467. /\*\*\*\*\*三天紧急\*\*\*\*\*/
468. **bool** flag = **false**;
469. **while**( !flag ){
470. cout << "请选择要归还的书ID(按Q退出):"<< endl;
471. cin >> bookid;
472. **if**(bookid =="Q") **return**;
473. **for** (**int** i = 0; i < row.size(); i++){ /\*这边不能写int rowSize = row.size();否则会影响循环遍历\*/
474. **if** ( row.at(i).at(0) == bookid ){
475. /\*将还书写入readed\_books文件夹下\*/
476. string returnPath = READEDBOOKPATH;
477. string returnfile = returnPath + **this**->UID + ".csv";
478. ofstream ff;
479. /\*\*归还时间是现在\*/
480. string returntime = getTime();
481. row.at(i).at(4)=returntime;
482. ff.open(returnfile.c\_str(),ios::app);
483. **int** colsize = row.at(i).size();
484. **for**(**int** j = 0; j < colsize; j++) ff << row.at(i).at(j) << ',';
485. ff << endl;
486. ff.close();
488. bookname = row.at(i).at(1);
490. row.erase(row.begin()+i);
491. flag = **true**;
492. **break**;
493. }
494. }
495. }
496. /\*\*写入用户借书临时文件\*\*/
497. ofstream out;
498. out.open(USERTMPFILE,ios::out);
499. **int** rowSize = row.size();
500. **for** (**int** i = 0; i < rowSize ; i++) {
501. **for**(**int** j = 0; j < RETURNBOOKCOL; j++)
502. out << row.at(i).at(j) << "," ;
503. //out << row.at(i).at(j) << "," ; //最后一个","不妨碍
504. out << endl;
505. }
506. out.close();
507. string oldname = USERTMPFILE;
508. renameFilename(oldname,filename);
509. /\*\*临时文件改名\*\*/
511. string canread = "1";
512. update(bookid,8,canread);
514. map<**int**,string> changefield;
515. changefield.insert(pair<**int**, string>(7, canread));
516. changefield.insert(pair<**int**, string>(6, "#"));
517. changefield.insert(pair<**int**, string>(5, "#"));
518. updateDetail(bookname,changefield);
520. cout << "还书成功!" <<endl;
521. }
522. cout << "按任意键返回上级" << endl;
523. //while(getchar()!='\n') ;
524. system("pause");

527. READERsMenu();
528. }
530. **void** CcommonUser::topBooks(){
531. vector< vector<string> > row;
532. string filename = TYPESBOOK;
533. row = getTablecontent(filename);
534. sort(row.begin()+1,row.end(),cmp);
535. **for** (**int** i = 0; i < row.size() ; ++i){
536. **int** colsize = row.at(i).size();
537. **for** (**int** j = 0; j < colsize ; ++j)
538. printf("%-8s\t", row.at(i).at(j).c\_str());
539. cout << endl;
540. }
541. system("pause");
542. }
544. **void** CcommonUser::guessYoulike(){
545. fstream f;
546. string PATH = READEDBOOKPATH;
547. string filename = PATH + **this**->UID + ".csv";
548. vector< vector<string> >row = getTablecontent(filename);
549. vector<string> bookname; //阅读过书名
550. map<string,**int**>classnum ; //阅读过书类别
551. **if**( row.empty() ){
552. printf("\t\t\t\t\t\t\t\t%s\n", "推荐热门图书");
553. **this**->topBooks();
554. **return** ;
555. }
556. **else**{
557. **int** rowsize = row.size();
558. **for**(**int** i=0; i < rowsize ; i++) bookname.push\_back(row.at(i).at(1));
559. string typebook =TYPESBOOK;
560. vector< vector<string> > typesbookrow = getTablecontent(typebook);
561. **int** typesbookrowsize = typesbookrow.size();
562. **int** booktotal = bookname.size();
564. printf("\t\t\t\t\t\t\t\t%s\n", "猜你喜欢");
565. **for**(**int** i=0;i < booktotal ;i++){
566. **for**(**int** j=0;j< typesbookrowsize ; j++) {
567. **if**( typesbookrow.at(j).at(1) == bookname.at(i) ) //书名匹配
568. **if**( classnum.count( typesbookrow.at(j).at(4)) == 0 ) classnum[typesbookrow.at(j).at(4)]=1;
569. **else** classnum[typesbookrow.at(j).at(4)]++;
570. // 类别=>typesbookrow.at(j).at(4) : 次数
571. **else** **continue**;
572. }
573. }
574. vector< pair<string, **int**> > v(classnum.begin(), classnum.end());
575. sort(v.begin(), v.end(),cmpPair);
576. **for**(**int** i=0;i< typesbookrowsize ; i++) {
577. **if**(typesbookrow.at(i).at(4) == v.at(0).first){
578. **int** colnum = typesbookrow.at(i).size();
579. **for**(**int** j=0;j<colnum;j++){
580. **if**( find(bookname.begin(),bookname.end(),typesbookrow.at(i).at(1) )==bookname.end() )
581. //书名不重复
582. printf("%-8s\t",typesbookrow.at(i).at(j).c\_str());
583. **else**;
584. }
585. printf("\n");
586. }**else** **continue**;
588. }
589. ; //次数最多的类别
590. }
591. system("pause");
592. }

595. **void** CcommonUser::readedBook(){
596. fstream f;
597. string PATH = READEDBOOKPATH;
598. string filename = PATH + **this**->UID + ".csv";
599. vector< vector<string> >row = getTablecontent(filename);
601. **if**( row.empty()) cout << "没有借阅记录" <<endl;
602. **else**{
603. **int** rowsize = row.size();
604. **for**(**int** i=0; i < rowsize ; i++){
605. **int** colsize = row.at(i).size();
606. **for**(**int** j=0;j<colsize;j++) printf("%-8s\t",row.at(i).at(j).c\_str());
607. cout << endl;
608. }
609. }
610. system("pause");
611. }


615. **void** CcommonUser::registeruser(){
616. **char** tep[20];
617. **char** user[20];
618. **char** password[20];
619. **FILE** \*fp;
620. system("cls");
621. printf("\t\t\t\t注册\n\n\n");
622. printf("\t\t\t请设置用户名：");
623. scanf("%s",user);
624. fp=fopen(ACCOUNTFILE,"r");
626. **do**{
627. printf("\n\t\t\t请设置密码：");
628. scanf("%s",password);
630. printf("\t\t\t请确认密码：");
631. scanf("%s",tep);
633. **if**(strcmp(tep,password)!=0)
634. printf("\n\t\t\t两次密码不一致请重新输入");
635. }**while**(strcmp(tep,password)!=0);
636. fp=fopen(ACCOUNTFILE,"a");
637. **if**(fp==NULL){
638. printf("\n\t注册失败\n");
639. exit(0);
640. }
641. **else**{
642. fputs(user,fp);
643. // char ch= ',';
644. fputc(',',fp);
645. fputs(password,fp);
646. fputc(',',fp);
647. fputc('2',fp);
648. printf("\n\t注册成功");
649. }
650. fclose(fp);
651. }
652. /\*\*CcommonUser的成员函数实现\*\*/

655. /\*\*CTypesBook的成员函数实现\*\*/




661. /\*\*CTypesBook的成员函数实现\*\*/
663. /\*\*CUserbooks的成员函数实现\*\*/




669. /\*\*CUserbooks的成员函数实现\*\*/
671. /\*\*CData\*\*/
672. CData::CData(string ×tr){
673. stringstream ss(timestr);
674. string str;
675. getline(ss, str, '/');
676. **this**->year = atoi(str.c\_str());
677. getline(ss, str, '/');
678. **this**->month = atoi(str.c\_str());
679. getline(ss, str, '/');
680. **this**->day = atoi(str.c\_str());
681. }
683. /\*\*计算时间差\*\*/
684. **int** CData::operator-(string ×tr){ //大的减小的,传入的小
685. stringstream ss(timestr);
686. string str;
687. **int** y,m,d;
689. **const** **int** m\_day[12 + 1] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
690. getline(ss, str, '/');
691. y = atoi(str.c\_str());
692. getline(ss, str, '/');
693. m = atoi(str.c\_str());
694. getline(ss, str, '/');
695. d = atoi(str.c\_str());
696. **int** leap = **this**->is\_leap(y);
698. **int** dely,delm,deld;
699. dely = **this**->year - y;
700. delm = **this**->month - m;
701. deld = **this**->day - d;
702. **int** gap=0;
703. **if**(leap){
704. gap += 366\*dely ;
705. **if**(m==2) gap += 29\*delm + deld;
706. **else** gap += m\_day[m]\*delm + deld;
707. }
708. **else** gap += 365\*dely + m\_day[m]\*delm + deld;
709. **return** gap;
710. }

713. **bool** CData::valid(**int** y, **int** m, **int** d) {
714. //不是润年情况
715. **const** **int** m\_day[12 + 1] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
716. **if** (m == 0 || d == 0) **return** **false**;
717. **if** (m > 12) **return** **false**;
718. //润年情况下,需要对2月份特殊处理
719. **if** ( CData::is\_leap(y) && m == 2 && d > 29) **return** **false**;
720. **if** ( CData::is\_leap(y) && m != 2 && d > m\_day[m]) **return** **false**;
721. //不是润年情况
722. **if** (!CData::is\_leap(y) && d > m\_day[m]) **return** **false**;
723. **return** **true**;
724. }
726. string CData::getReturntime(string &borrowtime){
727. **const** **int** m\_day[12 + 1] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
728. stringstream ss(borrowtime);
729. string str;
730. **int** year;
731. **int** month;
732. **int** day;
733. getline(ss, str, '/');
734. year = atoi(str.c\_str());
735. getline(ss, str, '/');
736. month = atoi(str.c\_str());
737. getline(ss, str, '/');
738. day = atoi(str.c\_str());
740. **if**( is\_leap(year) && month == 2 ){
741. day = day + 30 - 29 ;
742. month += 1;
743. }**else** **if**( is\_leap(year) && month != 2 ){
744. day = day + 30 - m\_day[month];
745. month += 1;
746. }**else**{ // !is\_leap(year)
747. day = day + 30 - m\_day[month];
748. month += 1;
749. }
750. **while**( !valid(year,month,day) ){
751. year ++;
752. month --;
753. }
755. std::stringstream timess;
756. timess<< year << "/" << month << "/" << day ;
757. **return** timess.str();
758. }

### boostFunction.cpp :主要功能文件

1. #include "boostFunction.h"
2. #include "main.h"
4. #include <fstream>
5. #include <cstdlib>
6. #include <cstdio>
7. #include <string>
8. #include <algorithm>
9. #include <numeric>
10. #include <iostream>
11. #include <sstream>
12. **using** **namespace** std;
13. /\*\*设置控制台屏幕大小\*\*/
14. **void** SetScreenGrid(){
15. **char** sysSetBuf[80];
16. sprintf(sysSetBuf, "mode con cols=%d lines=%d", CMD\_COLS, CMD\_LINES);
17. system(sysSetBuf);
18. }

21. /\*\*控制台标题\*\*/
22. **void** SetSysCaption(){
23. system("title 南邮图书馆系统");
24. }

27. /\*\*
28. 函数功能: 删除(改名)临时文件
29. 函数参数: 原文件名oldfile,新文件名newfile
30. 函数返回值:无
31. \*\*/
32. **void** renameFilename(string &oldfile,string &newfile){
33. remove(newfile.c\_str());
34. rename(oldfile.c\_str(), newfile.c\_str());
35. }

38. /\*\*
39. 函数功能:返回当前时间
40. 函数参数:无
41. 函数返回值:string currentTime
42. \*\*/
43. string getTime(){
44. **time\_t** timep;
45. time(&timep);
46. **char** tmp[64];
47. strftime(tmp, **sizeof**(tmp), "%Y/%m/%d",localtime(&timep) );
48. **return** tmp;
49. }

52. /\*\*计算vec中最大值\*\*/
53. **int** vecMAX(vector<string> & vec){
54. vector<**int**> v;
55. **for**(**int** i=0;i < vec.size();i++){
56. v.push\_back(atoi(vec.at(i).c\_str()));
57. }
58. **return** \*max\_element(v.begin(),v.end());
59. }
61. /\*\*计算vec中总和\*\*/
62. **int** vecSUM(vector<string> &vec){
63. vector<**int**> v;
64. **for**(**int** i=0;i < vec.size();i++){
65. v.push\_back(atoi(vec.at(i).c\_str()));
66. }
67. **return** accumulate(v.begin(),v.end(),1);
68. }
70. vector< vector<string> > getTablecontent(string &path ){
71. fstream f;
72. f.open(path.c\_str(),ios::in);
73. vector<vector<string> > row; //记录行数据
74. **if**(f.fail()){
75. cout << "信息不存在" <<endl;
76. **return** row;
77. }**else**{
79. /\*\*将表数据存入vec中\*\*/
80. string lineStr;
81. **while** (getline(f, lineStr) ){ //获取行
82. stringstream ss(lineStr);
83. string str;
84. vector<string> col; //记录列数据,必须写在这个循环里
85. **while** (getline(ss, str, ',')) col.push\_back(str);
86. row.push\_back(col);
87. }
88. **return** row;
89. } //没有返回值,直接为空
90. f.close();
91. }

**五、测试数据及其结果分析**

运行后，程序的主菜单界面如下，如图1-1所示：

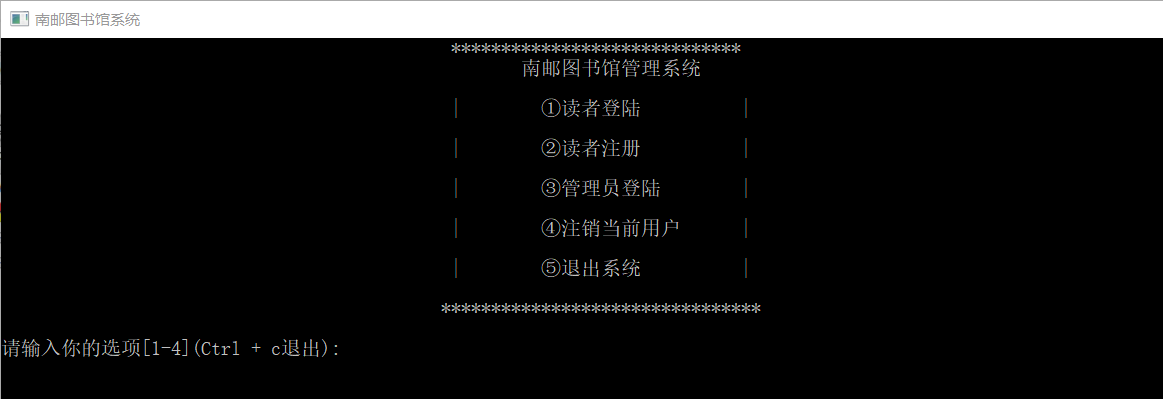


图1-1 主菜单界面

（1）读者登陆

输入：1

输入读者账号：1

输入密码：\*\*\*（123）

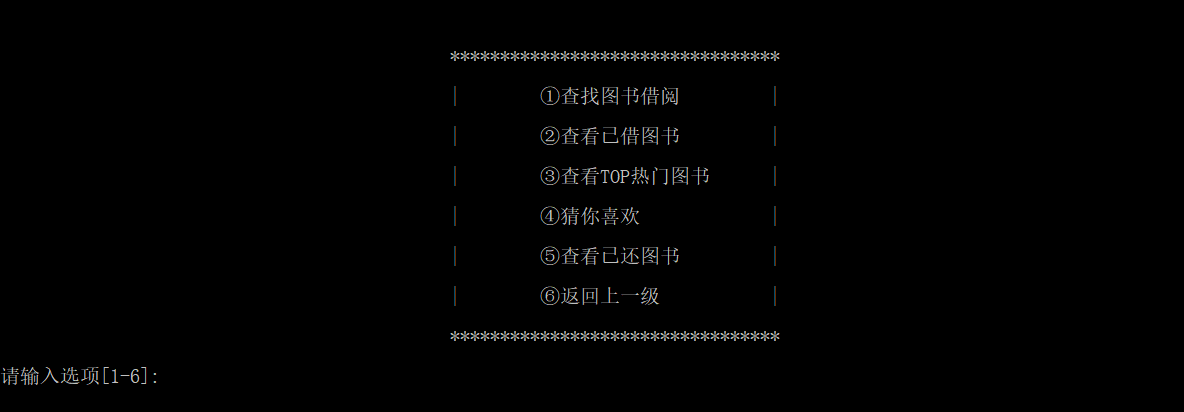
到达程序的读者操作界面，如图1-2所示：

图1-2读者操作界面

（2）读者借阅

输入：1

可看到图书馆当前所有图书，如图1-3所示

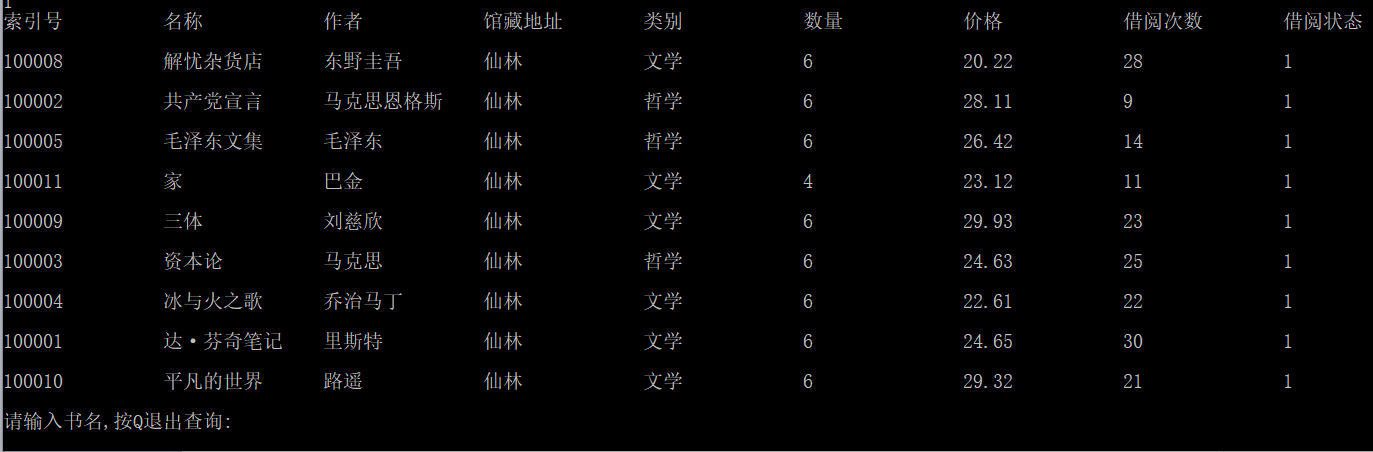


图1-3现存所有图书界面

根据书名查询图书并借阅：

例如借阅三体

输入：三体

可查看图书馆所有的三体图书如图1-4所示

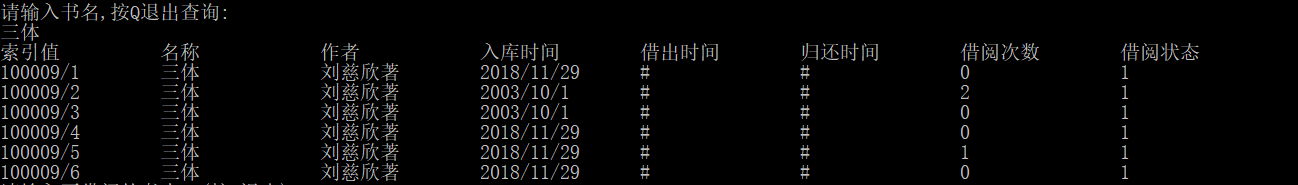


图1-4所有的三体图书

根据书的ID借阅具体某一本三体图书如图1-5所示

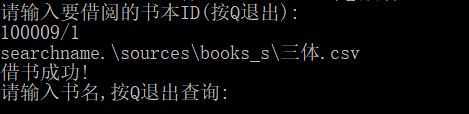


图1-5借阅成功

操作已完成，请继续操作

1. 读者查看已借图书并还书

例如归还之前所借的三体图书

在读者操作界面输入：2

可看到目前读者所借的所有图书，如图1-6所示

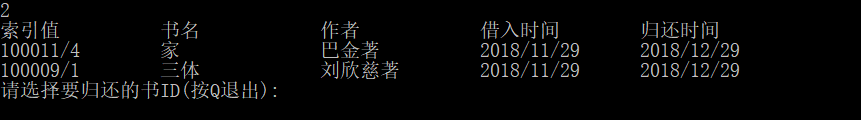


图1-6已借的所有图书

通过书的ID来归还图书，如图1-7所示

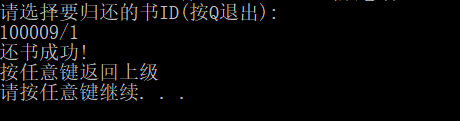


图1-7归还图书

此时该本三体归入库中，状态变为可借，如图1-8所示

操作已完成，请继续操作

1. 读者查看热门图书

在读者操作界面输入：3

可看到热门图书排行榜，如图1-9所示

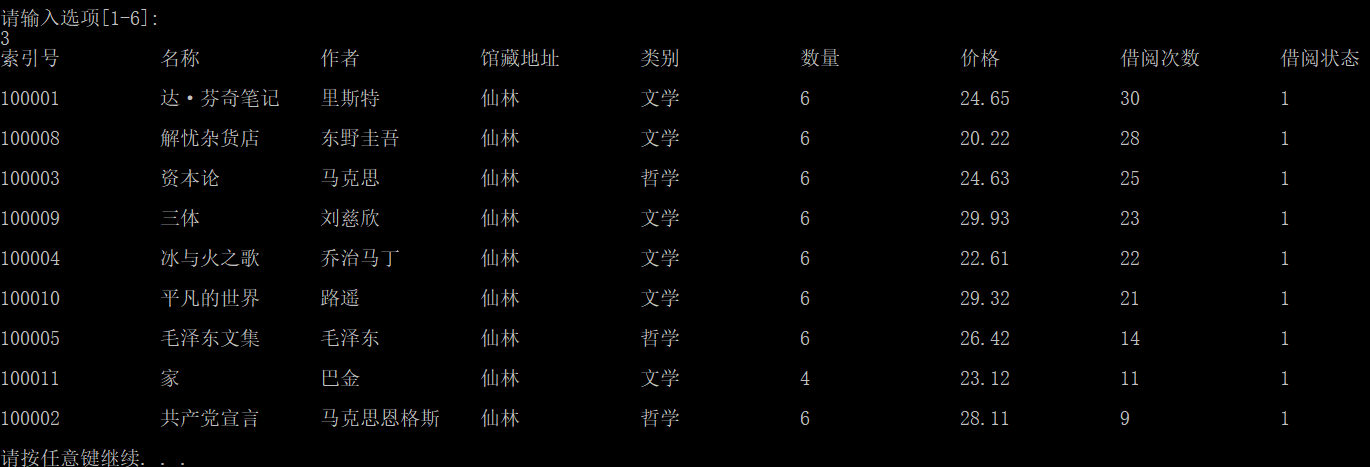


图1-9热门图书排行

操作已完成，请继续操作

1. 猜你喜欢系列

在读者操作界面输入：4

若该读者没有读过任何书籍，则显示信息不存在并推荐热门图书，如图1-10所示

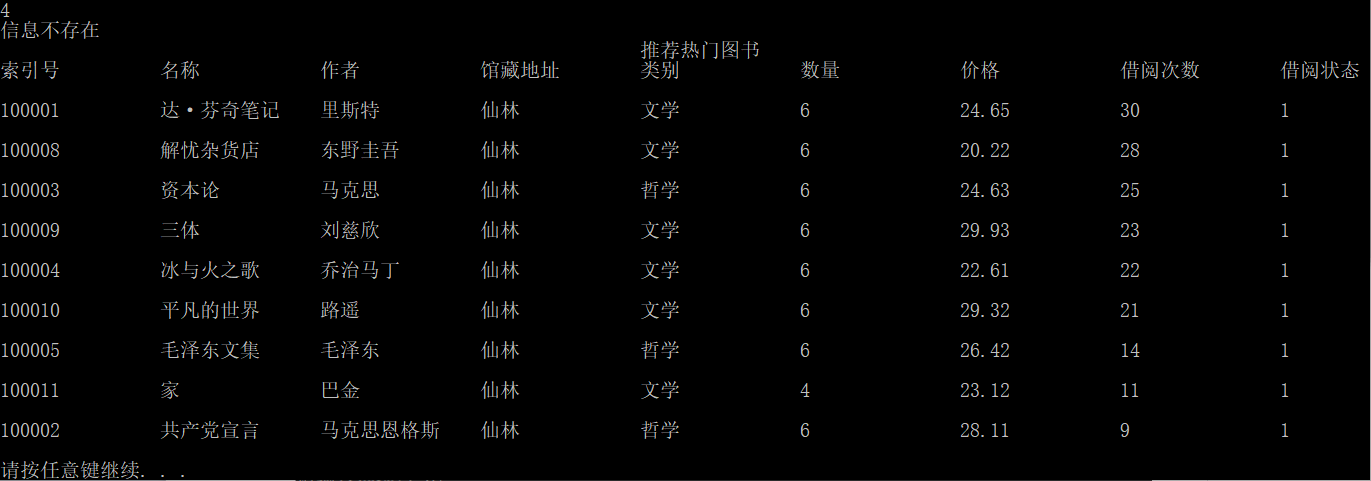


图1-10推荐热门图书（未借书读者）

若该读者借阅并归还书籍，则根据他读过的书来给他进行推荐，如图1-11所示

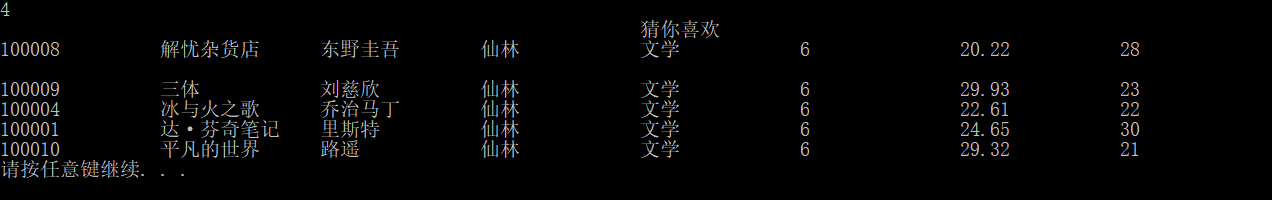


图1-11推荐图书（读过书读者）

操作已完成，请继续操作

(6)查看已还图书

在读者操作界面输入：5

即可看到之前所借的所有图书，如图1-12所示

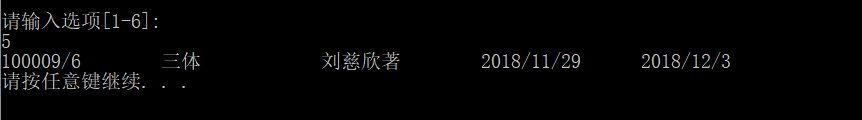


图1-12已还图书

操作已完成，请继续操作

（7）用户注册

例如注册一个名为HSR的账号

在主菜单界面输入：1

读者账号注册界面如图2-1所示

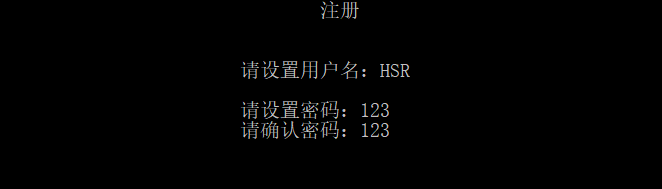


图2-1读者账号注册

此时存储账号的库中就出现了用户名为HSR的账号，如图2-2所示

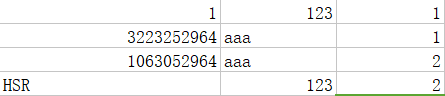


图2-2用户账号库

注：第一列表示用户账号的用户名

第二列表示用户账号的密码

第三列表示用户账号的等级（1表示管理员账号，2表示读者账号）

操作已完成，请继续操作

（8）管理员登陆

输入：3

输入管理员账号：1

输入密码：\*\*\*（123）

到达程序的管理员操作界面，如图3-1所示：

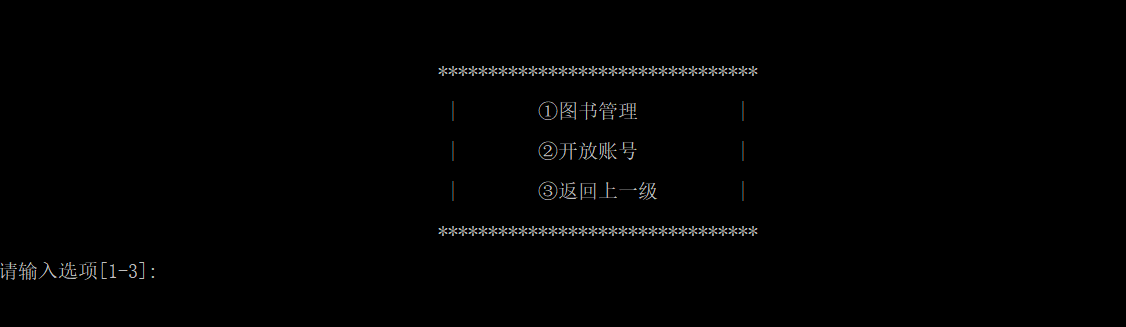


图3-1管理员操作界面

（9）图书管理（管理员）

在管理员操作界面输入：1

可看到管理员图书操作界面（可完成对图书的增加，删除，查看），如图3-2所示：

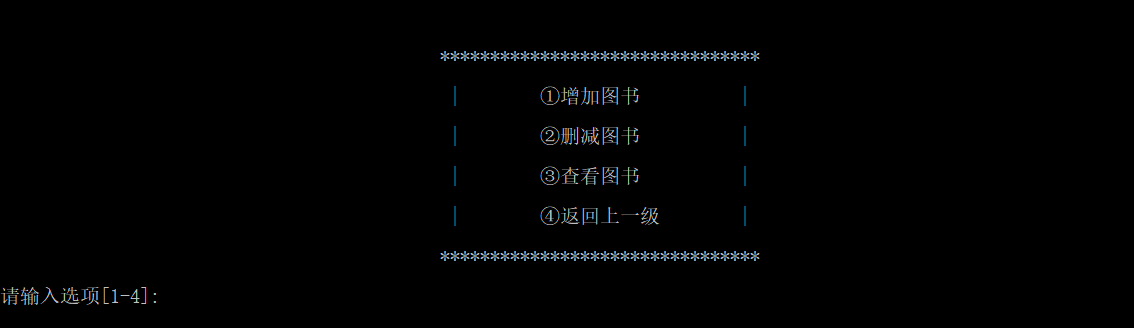


图3-2管理员图书操作界面

（10）增加图书

在管理员图书操作界面输入：1

可实现图书的增加

例如添加图书左耳，如图3-3所示

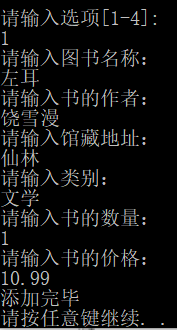


图3-3增加图书

此时，该书即被添加于保存所有图书的库内，如图3-4所示



图3-4增加过以后的书库

此时具体的书所在的库也添加了左耳图书，如图3-5所示



图3-5具体的书的书库

操作已完成，请继续操作

（11）删除图书

在管理员界面输入：2

可实现图书的删除

例如删除之前添加的图书左耳，如图3-6所示

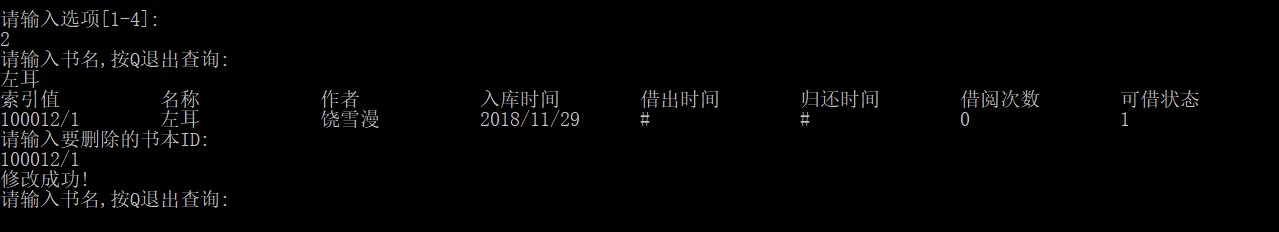


图3-6删除图书

操作已完成，请继续操作

（12）查看图书（管理员）

在管理员界面输入：3

可实现查看图书

例如查询三体，如图3-7所示

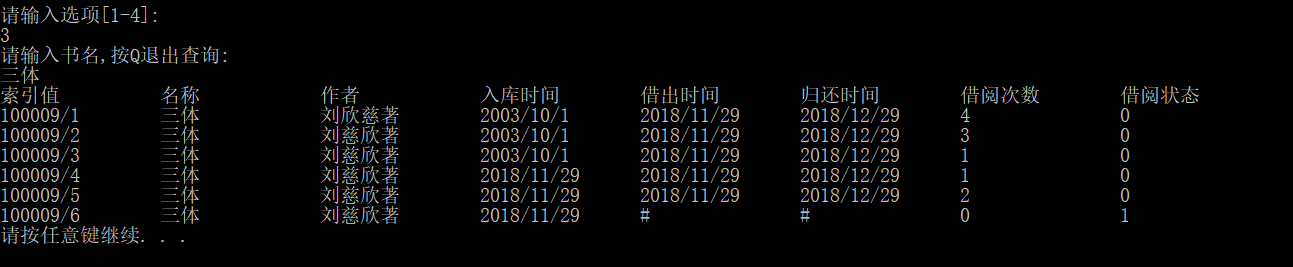


图3-7管理员查询图书

操作已完成，请继续操作

（13）开放账号注册（管理员操作）

在管理员操作界面输入：2

进行管理员账号或读者账号注册

例如注册一个名为HSR的管理员账号，如图3-8所示

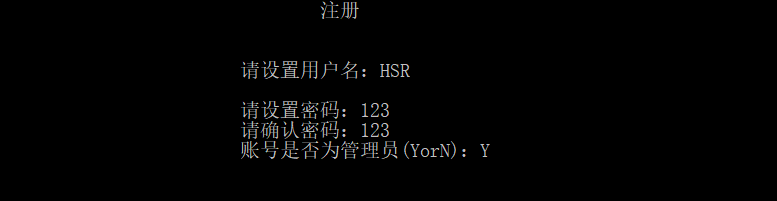


图3-8管理员注册管理员账号或读者账号

此时存储账号的库中就出现了用户名为HSR的管理员账号，如图3-9所示

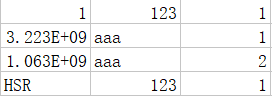


图3-9用户账号库

注：第一列表示用户账号的用户名

第二列表示用户账号的密码

第三列表示用户账号的等级（1表示管理员账号，2表示读者账号）

操作已完成，请继续操作

1. 注销当前用户

在主菜单界面输入：4

注销用户如图4-1所示

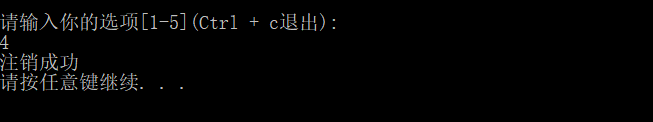


图4-1注销用户

操作已完成，请继续操作

1. 退出系统

在主菜单界面输入：5

退出系统如图5-1所示

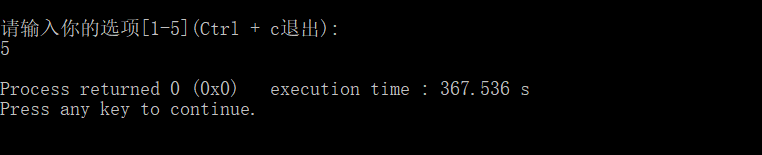


图5-1退出系统

操作已完成，请继续操作

结果分析如下。

1. 在主菜单界面时如果输入的内容不是1-5之间的数字，而是其他数字或字符（字母或符号），系统将继续执行等待下次正确的输入。
2. 在读者操作界面时如果输入的内容不是1-6之间的数字，而是其他数字或字符（字母或符号），系统将继续执行等待下次正确的输入。
3. 在管理员操作界面时如果输入的内容不是1-3之间的数字，而是其他数字或字符（字母或符号），系统将继续执行等待下次正确的输入。

（4）在图书管理（管理员）界面时如果输入的内容不是1-4之间的数字，而是其他数字或字符（字母或符号），系统将继续执行等待下次正确的输入。

1. 在需要输入所须查找的名称处如输入名称不在系统中，系统提示“不存在该书”

，需要再次输入库内所包含的图书。

**六、调试过程中的问题**

（1）模拟数据库，如何处理外键问题比较困难，后来主要通过文件名来对应链接关系。同时，数据库删除一条记录，要更新好几张表的信息，一开始都是直接在对应函数下又写了打开要更新表的操作，即冗余又容易出错，后来直接写了个接口来更新对应需要更新的表，就很方便了。

（2）C++和C在char\*和string的转换之间经常出错。此时IDE的优越性就体现出来了，有代码补全及显示功能就能减少在Bebug过程中的工作量。

（3）open的参数问题,ffstream.open参数filename为char \* ,string需要string.c\_str(),宏定义可以直接传入，打开模式ios::app在文件结束时必须加ffstream.close()，不然无法保存写入，ios::out可以,但时刻记住文本输入输出流的关闭是个好习惯。

（4）cpp标准头文件是在.h还是.cpp中include?普通的头文件重复包含是没多大影响的，但涉及类的定义的时候，重复包含是个大问题，通过条件编译#ifndef \_C\_CLASS\_H、

#define \_C\_CLASS\_H、#endif // \_C\_CLASS\_H 解决。

（5）操作二维表数据时,最好用宏定义在头文件定义每一列的含义和长度，使用宏定义可以避免操作错误的列。而且每一个表最好都有固定一系列的前缀，这个需要在架构的时候就想好定义。

1. **课程总结**

这次程序设计我们组设计的程序是南邮图书馆管理系统，这次设计课程充分考察和锻炼了我们对运用c语言和c++编写程序的能力。这次课程加深了我们对MicosoftVisualC++环境的了解，以及加深了对编程良好习惯作用的印象。特别是编程的良好习惯，可以大大提高效率，如注意相同内容的缩进和对齐，这样做可以使程序代码出错的情况下，可以快速并且便捷的查找到错误的行，利于修改。对代码的变量命名，代码内注释格式等可以有助于组内协作，大大提高小组协作编写代码的效率，有效地防止了函数作用不明，函数套用不畅等小组内易出现的问题。因为这次编码是小组合作，每个人的编码习惯不同，如果没有良好的编码习惯和统一函数、变量命名方式就极容易导致每个人代码合起来时练练报错。基础架构在编码过程中是非常重要的一环，只有建立了良好的基础架构，确定了之后编码的流程和方向才能按部就班地实现代码的功能和功能的总和，成为一个完整的可用的系统。一个合理的架构也可使我们在编码过程中不手忙脚乱，丢三落四。我们可以对着架构井井有条地进行编码和总和。真正地证明一个良好的开端是成功的一半。我们在这次编码中发现的另一个问题是，需要对代码及时的调试。及时的调试能发现代码中很多不易发现但重要的问题，每次完成一个编程任务，完成一段代码，都应当有目的的跟踪该程序的应用状况，随时总结，找到自己的不足。这次编码最重要的作用是教会了我们团队合作的重要性，一个人的能力是有限的，团队互帮互助，共同前进才是编码甚至以后生活中永恒的主题。这次程序设计使我们在c语言知识方面都得到了十足的进步，温故而知新，这对以后的学习提供了良好的帮助。总而言之，这是一次对我们以后学习生活都有重大意义的程序设计实验课。