

XI`AN TECHNOLOGICAL UNIVERSITY

课程设计报告

课程名称: C语言项目实践

专 业： 软件工程

班 级： 23060208

姓 名： 高明杰

学 号： 2023032369

指导教师： 黄姝娟

成 绩：

2024 年 7 月\_5\_日

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | C语言项目实践 | | | | | 指导教师 | | 黄姝娟 | |
| 课设题目4 | **药品销售管理系统** | | | | | | | |  |
| 学号 | 2023032369 | 姓名 | 高明杰 | 地点 | 工1-202 | | 完成时间 | | 2024/7/1  2024/7/5 |
| 设计任务 | 建立一个药品销售管理系统，利用**堆区创建数组**(**指针+动态内存)**实现，具有如下的五大功能模块，具体要求如下：  qq_pic_merged_1687695704408  药品销售管理系统 | | | | | | | | |
| 设计要求 | 1. 输入记录模块   逐个从文件中读取数据初始化数组结构。  药品包含属性: 药品名称，品牌，单价，库存量，生产日期（年月日格式），药品简介等。   1. 查询记录模块   分别按照**药品名称、年份(例如:2023年以前生产的药品)**进行查询，把满足条件的卡信息显示出来。   1. 更新记录模块   3.1 修改某一药品的属性信息  3.2 插入一种新药品信息；  3.3 购买药品，根据药品名称购买药品，若药品库存量为1，则根据药品名称删除信息; 若库存量大于1，则将该药品库存量减1。   1. 统计记录模块   分别按照库存量统计单价>=30元的所有药品信息，并输出统计结果。   1. 显示模块   显示所有药品的信息。   1. 输出记录模块   能够将所有药品信息保存到文件中，并能从文件中读取药品信息；  以上为最基本要求，可根据自己的理解在此基础上对各个模块增加和完善相应的功能，使得整个系统能更加接近实际的应用。 | | | | | | | | |
| 成果要求 | 1. 设计出完整流程图。  2．编写出全部模块对应的函数。  3．在集成开发工具中调试通过。  4．参加答辩。  5．写出完整、规范的课程设计报告，在指定的时间内提交指导教师。 | | | | | | | | |
| 设计报告  要求 | 1．封面：（格式附后）  2．课程设计任务书  3．课程设计报告：  ⑴ 系统总体方案  ⑵ 设计思路和主要步骤  ⑶ 各功能模块和流程图  ⑷ 设计代码  ⑸ 心得体会 | | | | | | | | |

**药品销售管理系统**

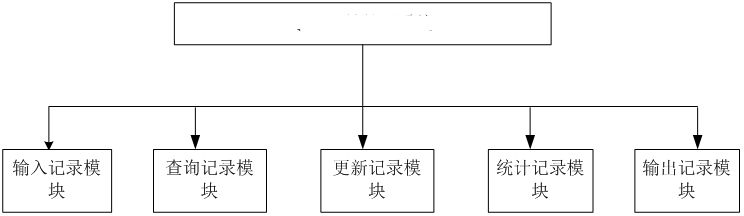
**一、**课程设计任务书

建立一个药品销售管理系统

二、课程设计报告

**2.1 系统总体方案**

利用堆区创建数组(指针+动态内存)实现，具有如下的五大功能模块，具体要求如下：



药品销售管理系统

**2.2设计思路和主要步骤**

2.2.1输入记录模块

逐个从文件中读取数据初始化数组结构。药品包含属性: 药品名称，品牌，单价，库存量，生产日期（年月日格式），药品简介等。

2.2.2查询记录模块

分别按照药品名称、年份(例如:2023年以前生产的药品)进行查询，把满足条件的卡信息显示出来。

2.2.3更新记录模块

2.3.1 修改某一药品的属性信息

2.3.2 插入一种新药品信息；

2.3.3 购买药品，根据药品名称购买药品，若药品库存量为1，则根据药品名称删除信息; 若库存量大于1，则将该药品库存量减1。

2.2.4统计记录模块

分别按照库存量统计单价>=30元的所有药品信息，并输出统计结果。

2.2.5显示模块

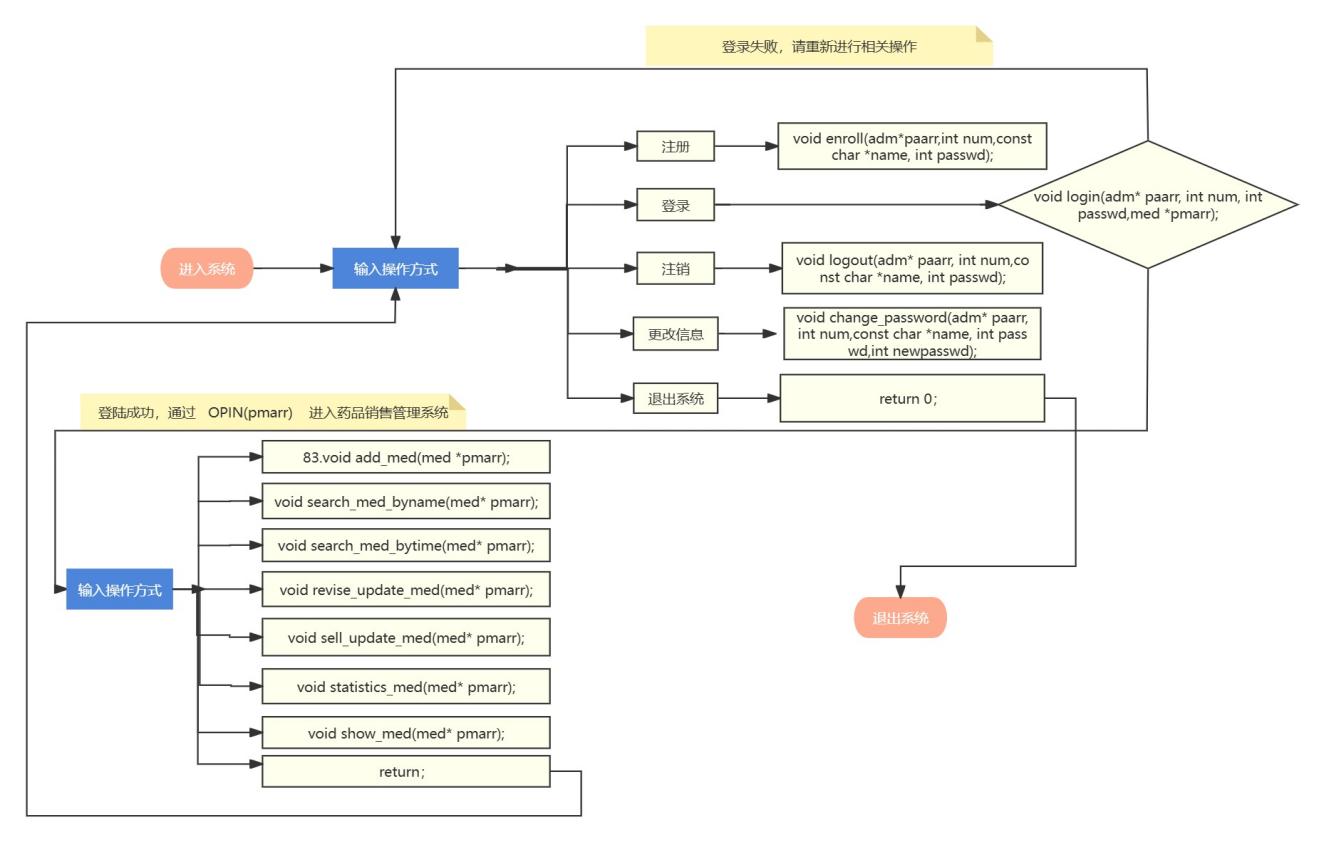
显示所有药品的信息。

2.2.6输出记录模块

能够将所有药品信息保存到文件中，并能从文件中读取药品信息。

**2.3各功能模块和流程图**

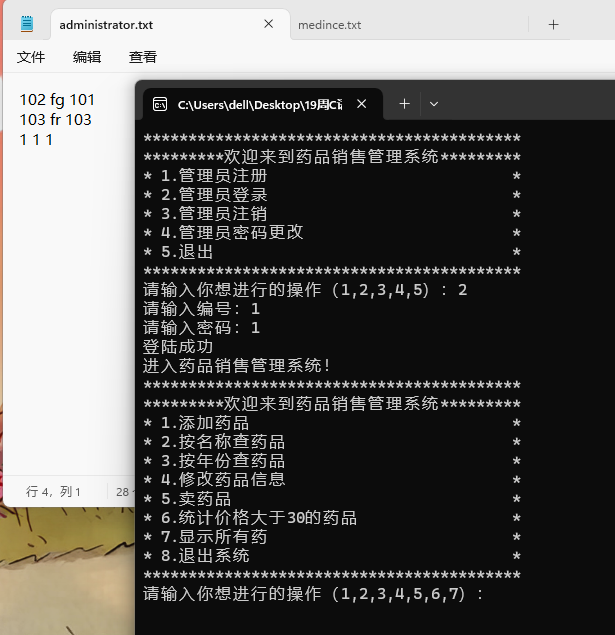
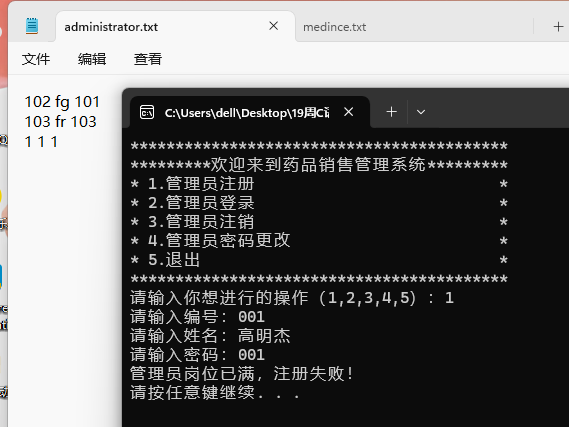
为了进一步提升药品销售管理系统的安全性与合规性，我精心设计并引入了一项关键的进入条件——管理员验证系统。这一创新举措旨在确保系统访问的严格控制，最大限度地保障系统内部数据的安全性与完整性，从而让药品销售管理流程更加可靠、高效。通过实施管理员验证系统，我们不仅能够有效防范未经授权的访问，还能为系统操作增设一道坚实的防线，全方位提升系统的安全防护能力。

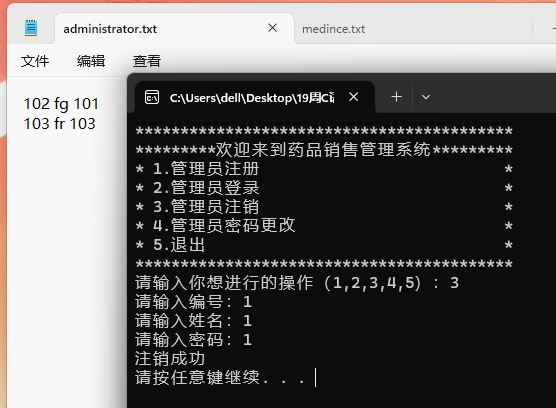


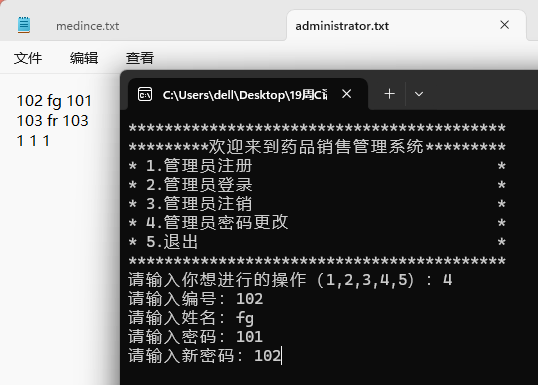
如上是通过ProcessOn制作的本药品管理系统的流程图。

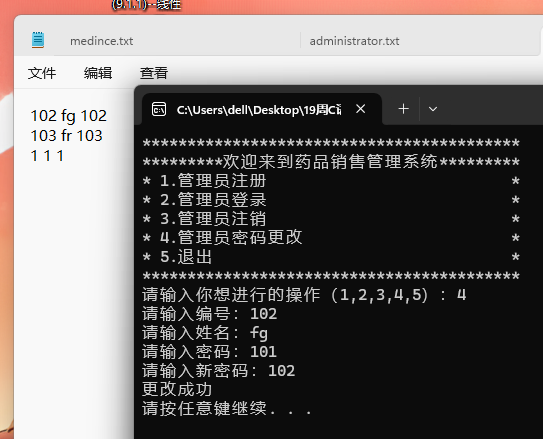
**2.4 代码编译放在本文末尾**

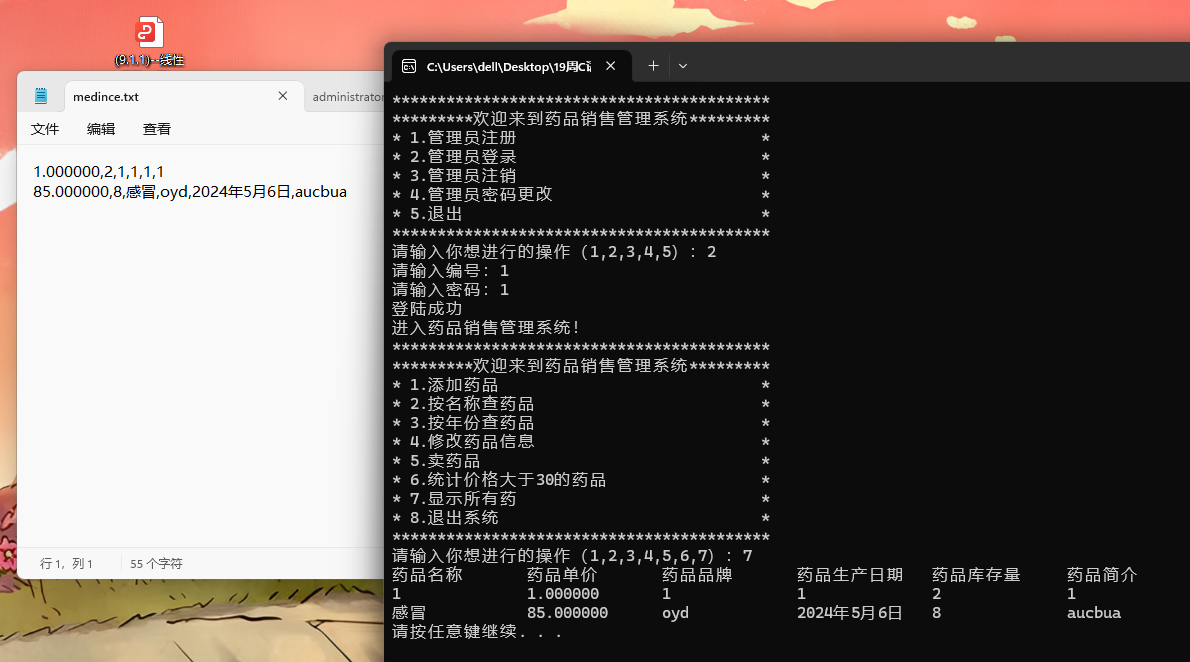
**2.5程序界面**











通过全面而详尽的测试与调试流程，我们可以确药品销售管理系统在实际运用中能够稳定流畅地运行，准确高效地处理用户输入的数据。同时，系统在面对各种异常情况时，也具备相应的应对策略，以保障运营的连贯性和安全性。在软件开发过程中，测试与调试是不可或缺的关键环节，它们协助开发者及时发现并解决潜在问题，进而显著提升系统的整体质量和可靠性。一个优秀的程序，必然能够经受住细致入微的推敲与考验。

**2.6总结**

药品销售管理系统是一款基于数据结构设计的信息管理系统，其目标是高效管理和操作药品的库存与销售信息。该系统巧妙地运用了C语言的数据结构以及动态内存管理功能，为用户提供了丰富的功能函数，支持药品信息的录入、储存、查询及删除等多项操作，从而大大提升了药品销售管理的便捷性和效率。

药品销售管理系统不仅充分实现了基础的信息管理功能，更进一步彰显了在实际软件开发过程中，如何精妙地运用数据结构和算法来构建一个稳定且高效的管理系统。借助这一系统，药品销售者能够更为便捷地管理药品库存与销售信息，同时，系统管理员也能轻松进行数据统计与分析，从而有力推动业务进展，显著提升运营效率。

**2.7 参考文献**

1. 谭浩强. *C程序设计*[M]. 北京：清华大学出版社，2005.
2. 严蔚敏，吴伟民. *数据结构（C语言版）*[M]. 北京：清华大学出版社，2011.
3. 张海藩，吕凤翥. *C语言程序设计教程（第4版）*[M]. 北京：人民邮电出版社，2010.
4. 王晓东. "医药信息系统设计与实现"[J]. *计算机知识与技术*，2012, 8(1): 42-44.
5. 李红. "药品销售管理系统的设计与实现"[D]. 河北大学，2015.
6. 李春葆. *数据库原理及应用*[M]. 北京：清华大学出版社，2010.
7. 赵仲民. "基于C/S模式的药品库存管理系统的设计与实现"[J]. *计算机技术与发展*，2011, 21(3): 164-167.
8. 张晓峰. "药品供应链管理信息系统的设计与开发"[J]. *计算机系统应用*，2009, 18(7): 112-115.
9. 陈立波. "药品销售管理软件的设计与实现"[D]. 浙江大学，2008.
10. 李晓明. "药品流通领域的供应链管理研究"[J]. *中国医药导刊*，2013, 5(2): 222-224.
11. 赵丽华. "药品销售管理系统需求分析与设计"[J]. *计算机知识与技术*，2010, 6(3): 304-306.
12. 王海燕. "药品销售管理信息系统的设计与实现"[D]. 电子科技大学，2014.

附录 代码

项目实践本文将代码分入5个文件中

1.Operation.h

1. #pragma once
2. #ifndef OPERATION\_H
3. #define ADMSIZE 3 *//目前设置管理员仅有三人*
4. #define MEDINCE 100   *//初始药*
5. *//设计管理员*
6. typedef struct administrator {
7. int a\_num;         *//编号*
8. char a\_name[20];  *//姓名*
9. int a\_passwd;  *//密码*
10. }administrator;
11. *//设计药品*
12. typedef struct medince {
13. float m\_price;            *//药品单价*
14. int m\_inventory;          *//药品库存量*
15. char m\_name[20];          *//药品名称*
16. char m\_brand[20];         *//药品品牌*
17. char m\_manufacture[20];   *//药品生产日期*
18. char m\_introduction[50];  *//药品简介*
19. }medince;
20. *//管理员数组*
21. typedef struct adm {
22. administrator a\_arr[ADMSIZE];
23. int a\_size;
24. }adm;
25. *//药品数组*
26. typedef struct med {
27. medince\* pmeds;      *//动态内存*
28. int m\_size;        *//有效药品数量*
29. int medcapacity;   *//药品数组容量*
30. }med;
31. *// File 文件操作*
32. *//药品文件读入*
33. void File\_read\_med(med\* pmarr, const char\* filepath);
34. *//药品文件写入*
35. void File\_write\_med(med \*pmarr, const char\* filepath);
36. *//管理员文件读入*
37. void File\_read\_adm(adm \*paarr, const char\* filepath);
38. *//管理员文件写入*
39. void File\_write\_adm(adm\* paarr, const char\* filepath);
40. *//Administrator 管理员操作*
41. *//初始化Administrator数组*
42. void AInit(adm \*paarr);
43. *//注册*
44. void enroll(adm\*paarr,int num,const char \*name, int passwd);
45. *//登录*
46. void login(adm\* paarr, int num, int passwd,med \*pmarr);
47. *//注销*
48. void logout(adm\* paarr, int num,const char \*name, int passwd);
49. *//更改密码*
50. void change\_password(adm\* paarr, int num,const char \*name, int passwd,int newpasswd);
51. *//Medince 药品操作*
52. *//初始化药品数组*
53. void MInit(med\* pmarr);
54. *//动态内存扩容*
55. void Grow(med\* pmarr);
56. *//添加药品*
57. void add\_med(med \*pmarr);
58. *//查询药品*
59. void search\_med\_byname(med\* pmarr);  *// 通过名称*
60. void search\_med\_bytime(med\* pmarr); *//通过时间*
61. *//更新药品*
62. void revise\_update\_med(med\* pmarr);  *// 修改药品信息*
63. void sell\_update\_med(med\* pmarr); *//卖药品*
64. *// 统计价格（例如统计大于30元药品）*
65. void statistics\_med(med\* pmarr);
66. *//显示药品信息*
67. void show\_med(med\* pmarr);
68. void show1();   *//登录前页面展示*
69. void OPIN(med \*pmarr);   *//药品管理系统进入*
70. void show2();   *//药品操作目录显示*
71. #endif OPERATION\_H
72. main.cpp
73. #define \_CRT\_SECURE\_NO\_WARNINGS
74. #include "Operation.h"
75. #include <stdio.h>
76. #include <stdlib.h>
77. #include <windows.h>
78. void show1() {
79. printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");
80. printf("\*\*\*\*\*\*\*\*\*欢迎来到药品销售管理系统\*\*\*\*\*\*\*\*\*\n");
81. printf("\* 1.管理员注册                           \*\n");
82. printf("\* 2.管理员登录                           \*\n");
83. printf("\* 3.管理员注销                           \*\n");
84. printf("\* 4.管理员密码更改                       \*\n");
85. printf("\* 5.退出                                 \*\n");
86. printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");
87. }   *//登录前页面展示*
88. static void Clear() {
89. Sleep(3000);
90. system("cls");
91. }
92. int main() {
93. int a = 0, b = 0;
94. adm paarr[1];
95. med pmarr[1];
96. *//初始化Administrator数组*
97. AInit(paarr);
98. *//初始化药品数组*
99. MInit(pmarr);
100. int num, passwd, newpasswd;
101. char name;
103. while (1) {
104. show1();   *//登录前页面展示*
105. printf("请输入你想进行的操作（1,2,3,4,5）：");
106. int n = 0;
107. scanf("%d", &n);
108. switch (n) {
109. case 1:
110. printf("请输入编号：");
111. scanf("%d", &num);
112. printf("请输入姓名：");
113. scanf("%s", &name);
114. printf("请输入密码：");
115. scanf("%d", &passwd);
116. enroll(paarr,num,&name,passwd );
117. Clear();
118. break;
119. case 2:
120. printf("请输入编号：");
121. scanf("%d", &num);
122. printf("请输入密码：");
123. scanf("%d", &passwd);
124. login(paarr,num, passwd,pmarr);
125. Clear();
126. break;
127. case 3:
128. printf("请输入编号：");
129. scanf("%d", &num);
130. printf("请输入姓名：");
131. scanf("%s", &name);
132. printf("请输入密码：");
133. scanf("%d", &passwd);
134. logout(paarr,num,&name, passwd);
135. Clear();
136. break;
137. case 4:
138. printf("请输入编号：");
139. scanf("%d", &num);
140. printf("请输入姓名：");
141. scanf("%s", &name);
142. printf("请输入密码：");
143. scanf("%d", &passwd);
144. printf("请输入新密码：");
145. scanf("%d", &newpasswd);
146. change\_password(paarr,num,&name, passwd,newpasswd);
147. Clear();
148. break;
149. case 5:
150. return 0;
151. default:
152. printf("请重新输入你想进行的操作！\n");
153. Clear();
154. *//show1();   //登录前页面展示*
155. }
156. }
157. return 0;
158. }
159. void show2() {
160. printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");
161. printf("\*\*\*\*\*\*\*\*\*欢迎来到药品销售管理系统\*\*\*\*\*\*\*\*\*\n");
162. printf("\* 1.添加药品                             \*\n");
163. printf("\* 2.按名称查药品                         \*\n");
164. printf("\* 3.按年份查药品                         \*\n");
165. printf("\* 4.修改药品信息                         \*\n");
166. printf("\* 5.卖药品                               \*\n");
167. printf("\* 6.统计价格大于30的药品                 \*\n");
168. printf("\* 7.显示所有药                           \*\n");
169. printf("\* 8.退出系统                             \*\n");
170. printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");
171. }*//药品操作目录显示*
172. void OPIN(med \*pmarr) {
173. while (1) {
174. show2();   *//药品操作目录显示*
175. printf("请输入你想进行的操作（1,2,3,4,5,6,7）：");
176. int n = 0;
177. scanf("%d", &n);
178. switch (n) {
179. case 1:
180. add\_med(pmarr);*//添加药品*
181. Clear();
182. break;
183. case 2:
184. search\_med\_byname( pmarr);*// 通过名称*
185. Clear();
186. break;
187. case 3:
188. search\_med\_bytime(pmarr);*//通过时间*
189. Clear();
190. break;
191. case 4:
192. revise\_update\_med(pmarr);  *// 修改药品信息*
193. Clear();
194. break;
195. case 5:
196. sell\_update\_med(pmarr); *//卖药品*
197. Clear();
198. break;
199. case 6:
200. statistics\_med(pmarr);
201. Clear();
202. break;
203. case 7:
204. show\_med(pmarr);
205. Clear();
206. break;
207. case 8:
208. return ;
209. Clear();
210. default:
211. printf("请重新输入你想进行的操作！\n");
212. Clear();
213. *//show2();   //登录前页面展示*
214. }
215. }
216. }
217. Administrator operation.cpp
218. #define \_CRT\_SECURE\_NO\_WARNINGS
219. #include "Operation.h"
220. #include <cassert>
221. #include <stdio.h>
222. #include <string.h>
223. *//初始化Administrator*
224. void AInit(adm\* paarr) {
225. assert(paarr != NULL);
226. paarr->a\_size = 0;
227. File\_read\_adm( paarr, "administrator.txt");
228. }
229. *//注册*
230. void enroll(adm\* paarr, int num, const char \*name, int passwd) {
231. if (paarr->a\_size == ADMSIZE) {
232. printf("管理员岗位已满，注册失败！\n");
233. return;
234. }
235. else {
236. paarr->a\_arr[paarr->a\_size ].a\_num = num;
237. strcpy(paarr->a\_arr[paarr->a\_size ].a\_name , name);
238. paarr->a\_arr[paarr->a\_size ].a\_passwd = passwd;
239. paarr->a\_size++;
240. printf("注册成功\n");
241. }
242. File\_write\_adm(paarr, "administrator.txt");
244. }
245. *//登录*
246. void login(adm\* paarr, int num, int passwd,med \*pmarr) {
247. int k = 0;
248. for (int i = 0; i < paarr->a\_size; i++) {
249. if ((paarr->a\_arr[i].a\_num == num)
250. && (paarr->a\_arr[i].a\_passwd == passwd)) {
251. printf("登陆成功\n");
252. printf("进入药品销售管理系统！\n");
253. OPIN(pmarr);
254. k = 1;
255. }
256. }if (k == 0) printf("登录失败\n");
258. }
259. *//注销*
260. void logout(adm\* paarr, int num, const char\* name, int passwd) {
261. int k = 0;
262. for (int i = 0; i < paarr->a\_size; i++) {
263. administrator\* pca =& paarr->a\_arr[i];
264. if ((paarr->a\_arr[i].a\_num == num) && !strcmp(paarr->a\_arr[i].a\_name, name)
265. && (paarr->a\_arr[i].a\_passwd == passwd)) {
266. int n = paarr->a\_size - i - 1;
268. memmove(pca, pca + 1, n \* sizeof(administrator));
269. *//memset(&(paarr->a\_arr[paarr->a\_size]), NULL, sizeof(administrator));*
271. printf("注销成功\n");
272. paarr->a\_size--;
273. k = 1;
274. *//return;*
276. }
277. }
278. if (k == 0) {
279. printf("注销失败，当前用户不存在或密码不正确\n");
280. }
281. File\_write\_adm(paarr,"administrator.txt");
282. }
283. *//更改密码*
284. void change\_password(adm\* paarr, int num, const char\* name, int passwd,int newpasswd) {
285. int k = 0;
286. for (int i = 0; i < paarr->a\_size; i++) {
287. *//administrator\* pcm = &paarr->a\_arr[i];*
288. if ((paarr->a\_arr[i].a\_num == num) && !strcmp(paarr->a\_arr[i].a\_name, name)
289. && (paarr->a\_arr->a\_passwd == passwd)) {
290. paarr->a\_arr[i].a\_passwd = newpasswd;
291. printf("更改成功\n");
292. k = 1;
293. *//return ;*
295. }
296. }
297. if (k == 0) {
298. printf("更改失败，当前用户不存在或密码不正确\n");
299. }
300. File\_write\_adm(paarr, "administrator.txt");
301. }
302. Medince operation.cpp
303. #define \_CRT\_SECURE\_NO\_WARNINGS
304. #include "Operation.h"
305. #include <cassert>
306. #include <malloc.h>
307. #include <stdio.h>
308. #include <string.h>
309. *//Medince 药品操作*
310. *//初始化药品数组*
311. void MInit(med\* pmarr) {
312. pmarr->pmeds = (medince\*)malloc(MEDINCE \* sizeof(medince));
313. assert(pmarr->pmeds != NULL);
314. if (pmarr->pmeds == NULL) {
315. return;
316. }
317. pmarr->medcapacity = MEDINCE;
318. pmarr->m\_size = 0;*//有效卡数量0*
319. *//printf("初始化数组成功\n");*
320. File\_read\_med(pmarr, "medince.txt");
321. }
322. *//动态内存扩容*
323. void Grow(med\* pmarr) {
324. medince\* p = (medince\*)realloc(pmarr->pmeds, 2 \* pmarr->medcapacity \* sizeof(medince));
325. assert(p != NULL);
326. if (p == NULL) {
327. free(pmarr->pmeds);
328. pmarr->pmeds = NULL;
329. return;
330. }
331. pmarr->pmeds = p;
332. pmarr->medcapacity \*= 2;
333. }
334. *//通过名称查询药品*
335. static medince\* searchmedbyname(med\* pmarr, const char name[]) {
336. for (int i = 0; i < pmarr->m\_size; i++) {
337. if (strcmp(pmarr->pmeds[i].m\_name, name)==0)
338. return &pmarr->pmeds[i];
339. }
340. return NULL;
341. }
342. *//添加药品*
343. void add\_med(med\* pmarr) {
344. if (pmarr->m\_size == pmarr->medcapacity) Grow(pmarr);
345. char name[20];
346. char manufacture[20];
347. *//如果将要输入的药品，名称和生产日期之前存在并相同，直接在库存量增加*
348. printf("请输入药品名称：");
349. scanf("%s", &name);
350. printf("请输入药品生产日期：");
351. scanf("%s", &manufacture);
352. if ((searchmedbyname(pmarr, name) != NULL) &&
353. strcmp(searchmedbyname(pmarr, name)->m\_manufacture , manufacture) ==0){
354. searchmedbyname(pmarr, name)->m\_inventory++;
355. printf("当前药品已存在，且生产日期相同，已在库存量增加。\n");
356. }
357. else {
358. strcpy(pmarr->pmeds[pmarr->m\_size].m\_name, name);
359. strcpy(pmarr->pmeds[pmarr->m\_size].m\_manufacture, manufacture);
360. float price;
361. printf("请输入药品单价：");
362. scanf("%f", &price);
363. pmarr->pmeds[pmarr->m\_size].m\_price = price;
364. int inventory;
365. printf("请输入药品库存量：");
366. scanf("%d", &inventory);
367. pmarr->pmeds[pmarr->m\_size].m\_inventory = inventory;
368. char brand[20];
369. printf("请输入药品品牌：");
370. */\*fgets(brand, sizeof(brand), stdin);*
371. size\_t len = strlen(brand);
372. if (len > 0 && brand[len - 1] == '\n') brand[len - 1] = '\0';\*/
373. scanf("%s", &brand);
374. strcpy(pmarr->pmeds[pmarr->m\_size].m\_brand, brand);
375. char introduction[50];
376. printf("请输入药品简介：");
377. scanf("%s", &introduction);
378. *//fgets(introduction, sizeof(introduction), stdin);*
379. *//size\_t len\_2 = strlen(introduction);*
380. *//if (len\_2 > 0 && introduction[len\_2 - 1] == '\n') introduction[len\_2 - 1] = '0';*
381. strcpy(pmarr->pmeds[pmarr->m\_size].m\_introduction, introduction);
382. pmarr->m\_size++;
383. printf("药品添加成功，请进下下一步操作\n");
384. }
385. File\_write\_med(pmarr, "medince.txt");
386. }
387. *//查询药品*
388. *// 通过名称*
389. void search\_med\_byname(med\* pmarr) {
390. char name[20];
391. printf("请输入药品名称：");
392. scanf("%s", &name);
393. if (searchmedbyname(pmarr, name) != NULL) {
394. *//打印查询到的药品信息*
395. printf("%-15s%-15s%-15s%-15s%-15s%-15s\n", "药品名称", "药品单价", "药品品牌", "药品生产日期", "药品库存量", "药品简介");
396. printf("%-15s%-15f%-15s%-15s%-15d%-15s\n", searchmedbyname(pmarr, name)->m\_name,
397. searchmedbyname(pmarr, name)->m\_price, searchmedbyname(pmarr, name)->m\_brand, searchmedbyname(pmarr, name)->m\_manufacture,
398. searchmedbyname(pmarr, name)->m\_inventory, searchmedbyname(pmarr, name)->m\_introduction);
399. }
400. else
401. printf("当前药品不存在，请重新进行相关操作！");
402. }
403. *//通过时间*
404. void search\_med\_bytime(med\* pmarr) {
405. char manufacture[20];
406. printf("请输入药品生产日期：");
407. scanf("%s", &manufacture);
408. int time = (manufacture[0]-'0') \* 1000 + (manufacture[1] - '0') \* 100 + (manufacture[2] - '0') \* 10 +( manufacture[3] - '0');
409. printf("将会查找%d年之前生产的药品\n", time);
410. printf("%-15s%-15s%-15s%-15s%-15s%-15s\n", "药品名称", "药品单价", "药品品牌", "药品生产日期", "药品库存量", "药品简介");
411. int k = 0;
412. for (int i = 0; i < pmarr->m\_size; i++) {
413. int time2 = (pmarr->pmeds[i].m\_manufacture[0] - '0') \* 1000 + (pmarr->pmeds[i].m\_manufacture[1] - '0') \* 100
414. + (pmarr->pmeds[i].m\_manufacture[2] - '0') \* 10 + (pmarr->pmeds[i].m\_manufacture[3] - '0');
415. if (time2 < time) {
416. k = 1;
418. printf("%-15s%-15f%-15s%-15s%-15d%-15s\n", pmarr->pmeds[i].m\_name, pmarr->pmeds[i].m\_price,
419. pmarr->pmeds[i].m\_brand, pmarr->pmeds[i].m\_manufacture, pmarr->pmeds[i].m\_inventory, pmarr->pmeds[i].m\_introduction);
420. }
421. }
422. if (k == 0) {
423. printf("抱歉未查询到相关药品\n");
424. }
425. }
426. *//更新药品*
427. *// 修改药品信息*
428. void revise\_update\_med(med\* pmarr) {
429. char name[20];
430. printf("请输入药品名称：");
431. scanf("%s", &name);
432. if (searchmedbyname(pmarr, name) != NULL) {
433. printf("%-15s%-15s%-15s%-15s%-15s%-15s\n", "药品名称", "药品单价", "药品品牌", "药品生产日期", "药品库存量", "药品简介");
434. printf("%-15s%-15f%-15s%-15s%-15d%-15s\n", searchmedbyname(pmarr, name)->m\_name,
435. searchmedbyname(pmarr, name)->m\_price, searchmedbyname(pmarr, name)->m\_brand, searchmedbyname(pmarr, name)->m\_manufacture,
436. searchmedbyname(pmarr, name)->m\_inventory, searchmedbyname(pmarr, name)->m\_introduction);
437. */\*printf("请输入药品名称：");*
438. scanf("%s", &searchmedbyname(pmarr, name)->m\_name);
439. printf("请输入药品生产日期：");
440. scanf("%s", &searchmedbyname(pmarr, name)->m\_manufacture);
441. printf("请输入药品单价：");
442. scanf("%f", &searchmedbyname(pmarr, name)->m\_price);
443. printf("请输入药品库存量：");
444. scanf("%d", &searchmedbyname(pmarr, name)->m\_inventory);
445. printf("请输入药品品牌：");
446. scanf("%s", &searchmedbyname(pmarr, name)->m\_brand);
447. printf("请输入药品简介：");
448. scanf("%s", &searchmedbyname(pmarr, name)->m\_introduction);\*/
449. char name[20];
450. char manufacture[20];
451. printf("请输入药品名称：");
452. scanf("%s", &name);
453. printf("请输入药品生产日期：");
454. scanf("%s", &manufacture);
455. strcpy(searchmedbyname(pmarr, name)->m\_name, name);
456. strcpy(searchmedbyname(pmarr, name)->m\_manufacture, manufacture);
457. float price;
458. printf("请输入药品单价：");
459. scanf("%f", &price);
460. searchmedbyname(pmarr, name)->m\_price = price;
461. int inventory;
462. printf("请输入药品库存量：");
463. scanf("%d", &inventory);
464. searchmedbyname(pmarr, name)->m\_inventory = inventory;
465. char brand[20];
466. printf("请输入药品品牌：");
467. *//fgets(brand, sizeof(brand), stdin);*
468. *//size\_t len = strlen(brand);*
469. *//if (len > 0 && brand[len - 1] == '\n') brand[len - 1] = '\0';*
470. scanf("%s", &brand);
471. strcpy(searchmedbyname(pmarr, name)->m\_brand, brand);
472. char introduction[50];
473. printf("请输入药品简介：");
474. scanf("%s", &introduction);
475. */\*fgets(introduction, sizeof(introduction), stdin);*
476. size\_t len\_2 = strlen(introduction);
477. if (len\_2 > 0 && introduction[len\_2 - 1] == '\n') introduction[len\_2 - 1] = '\0';\*/
478. strcpy(searchmedbyname(pmarr, name)->m\_introduction, introduction);
479. }
480. else
481. printf("当前药品不存在，请重新进行相关操作！");
482. File\_write\_med(pmarr, "medince.txt");
483. }
484. *//卖药品*
485. void sell\_update\_med(med\* pmarr) {
486. char name[20];
487. printf("请输入药品名称：");
488. scanf("%s", &name);
489. int k = 0;
490. for (int i = 0; i < pmarr->m\_size; i++) {
491. medince\* pcm = &pmarr->pmeds[i];
492. if (strcmp(pmarr->pmeds[i].m\_name, name) == 0) {
493. if (pmarr->pmeds[i].m\_inventory == 1) {
494. int n = pmarr->m\_size - i + 1;
495. memmove(pcm, pcm + 1, n \* sizeof(medince));
496. pmarr->m\_size--;
497. }
498. else pmarr->pmeds[i].m\_inventory--;
499. k = 1;
500. }
501. }
502. if(k==0)
503. printf("当前药品不存在，请重新进行相关操作！");
504. File\_write\_med(pmarr, "medince.txt");
505. }
506. *// 统计价格（例如统计大于30元药品）*
507. void statistics\_med(med\* pmarr) {
508. float price = 0;
509. printf("请输入价格：");
510. scanf("%f", &price);
511. printf("将会输出%f价格之上的药品\n");
512. printf("%-15s%-15s%-15s%-15s%-15s%-15s\n", "药品名称", "药品单价", "药品品牌", "药品生产日期", "药品库存量", "药品简介");
513. int k = 0;
514. for (int i = 0; i < pmarr->m\_size; i++) {
516. if (pmarr->pmeds[i].m\_price>price) {
517. printf("%-15s%-15f%-15s%-15s%-15d%-15s\n", pmarr->pmeds[i].m\_name, pmarr->pmeds[i].m\_price,
518. pmarr->pmeds[i].m\_brand, pmarr->pmeds[i].m\_manufacture, pmarr->pmeds[i].m\_inventory, pmarr->pmeds[i].m\_introduction);
519. k = 1;
520. }
521. }
522. if (k == 0) {
523. printf("未找到价格大于30的药品\n");
524. }
525. }
526. *//显示药品信息*
527. void show\_med(med\* pmarr) {
528. printf("%-15s%-15s%-15s%-15s%-15s%-15s\n", "药品名称", "药品单价", "药品品牌", "药品生产日期", "药品库存量", "药品简介");
529. for (int i = 0; i < pmarr->m\_size; i++) {
530. printf("%-15s%-15f%-15s%-15s%-15d%-15s\n", pmarr->pmeds[i].m\_name, pmarr->pmeds[i].m\_price,
531. pmarr->pmeds[i].m\_brand, pmarr->pmeds[i].m\_manufacture, pmarr->pmeds[i].m\_inventory, pmarr->pmeds[i].m\_introduction);
532. }
533. }
534. File operation.cpp
535. #define \_CRT\_SECURE\_NO\_WARNINGS
536. #include "Operation.h"
537. #include <cassert>
538. #include <stdio.h>
539. *// File 文件操作*
540. *//药品文件读入*
541. void File\_read\_med(med\* pmarr, const char\* filepath) {
542. assert(pmarr && filepath);
543. FILE\* fp = fopen(filepath, "r");
544. if (fp == NULL) {
545. fp = fopen(filepath, "w");
546. if (fp == NULL) {
547. printf("文件以写模式打开失败\n");
548. fclose(fp);
549. return;
550. }
551. }
552. int i = 0;
553. medince\* pc;
554. while (1) {
555. pc = &pmarr->pmeds[i];
556. if (fscanf(fp, "%f,%d,%[^,],%[^,],%[^,],%[^\n]\n", &pc->m\_price, &pc->m\_inventory,
557. &pc->m\_name, &pc->m\_brand, &pc->m\_manufacture, &pc->m\_introduction) == 6) {
558. i++;
559. pmarr->m\_size++;
560. }
561. else break;
562. if (pmarr->m\_size == pmarr->medcapacity) Grow(pmarr);
563. }
565. fclose(fp);
566. }
567. *//药品文件写入*
568. void File\_write\_med(med\* pmarr, const char\* filepath) {
569. assert(pmarr && filepath);
570. FILE\* fp = fopen(filepath, "w");
571. if (fp == NULL) {
572. printf("文件以写方式打开失败\n");
573. return;
574. }
575. for (int i = 0; i < pmarr->m\_size; i++) {
577. medince \*pc= &pmarr->pmeds[i];
578. fprintf(fp, "%f,%d,%s,%s,%s,%s\n", pc->m\_price, pc->m\_inventory,
579. pc->m\_name, pc->m\_brand, pc->m\_manufacture, pc->m\_introduction);
581. }
582. fclose(fp);
583. }
584. *//管理员文件读入*
585. void File\_read\_adm(adm\* paarr, const char\* filepath) {
586. assert(paarr && filepath);
587. FILE\* fp = fopen(filepath, "r");
588. if (fp == NULL) {
589. fp = fopen(filepath, "w");
590. if (fp == NULL) {
591. printf("文件以写模式打开失败\n");
592. fclose(fp);
593. return;
594. }
595. }
596. int i = 0;
597. administrator\* pc;
598. while (1) {   *//读取到文件末尾   feof(fp)!= EOF*
599. pc = &paarr->a\_arr[i];
600. if (fscanf(fp, "%d %s %d\n", &pc->a\_num, &pc->a\_name, &pc->a\_passwd) == 3) {
601. i++;
602. paarr->a\_size++;
603. }
604. else break;
605. }
606. fclose(fp);
607. }
608. *//管理员文件写入*
609. void File\_write\_adm(adm\* paarr, const char\* filepath){
610. assert(paarr && filepath);
611. FILE\* fp = fopen(filepath, "w");
612. if (fp == NULL) {
613. printf("文件以写方式打开失败\n");
614. return;
615. }
616. for (int i = 0; i < paarr->a\_size; i++) {
617. *//fwrite(&paarr->a\_arr[i], sizeof(administrator), 1, fp);*
618. fprintf(fp, "%d %s %d\n", paarr->a\_arr[i].a\_num, paarr->a\_arr[i].a\_name, paarr->a\_arr[i].a\_passwd);
620. }
621. fclose(fp);
622. }