

book.h

#pragma once

#define INITCAPACITY 10//图书的初始容量

#define EXPENDSIZE 2//扩容2倍

//图书管理

typedef struct Book {

int book\_id;

char bookname[20];

char author[10]; //作者

double price;

char time[30]; //上架图书时间

}Book;

typedef struct Library {

Book\* books;

int capacity; //存储书容量

int size; //有效书本个数

}Library;

void InitLibrary(Library\* lib);

bool Grow(Library\* lib); //扩容操作

//上架图书

bool InsertTail(Library\* lib); //arr[size++] = val;

bool Delete(Library\* lib);

//下架图书

bool DeleteByName(Library\* lib);

bool DeleteById(Library\* lib);

//查找图书 -- 书名 编号 价格区间 上架日期 (2023年) 作者

int Search(Library\* lib);

int SearchByName(Library\* lib);

int SearchById(Library\* lib);

bool SearchByPriceRange(Library\* lib);

int SearchByDate(Library\* lib);

int SearchByAuthor(Library\* lib);

//修改

int Change(Library\* lib);

int ChangeByName(Library\* lib);

int ChangeById(Library\* lib);

int ChangeByAuthor(Library\* lib);

void Show(Library\* lib);

===================================================

book.cpp

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include "book.h"

#include <cassert>

#include <corecrt\_malloc.h>

#include <graphics.h>

#include <time.h>

//图书管理

void GetTime(char\* arr) {

time\_t t;

time(&t);

struct tm\* ptm = localtime(&t);

sprintf(arr,"%d-%d-%d %d:%d:%d",ptm->tm\_year + 1900, ptm->tm\_mon + 1,ptm->tm\_mday,

ptm->tm\_hour, ptm->tm\_min, ptm->tm\_sec);

}

void InitLibrary(Library\* lib) {

assert(lib!=NULL);

lib->books = (Book\*)malloc(INITCAPACITY\*sizeof(Book));

assert(lib->books!=NULL);

lib->capacity = INITCAPACITY;

lib->size = 0;

}

bool Grow(Library\* lib) {//扩容操作

assert(lib != NULL);

int len = lib->capacity \* EXPENDSIZE;

Book\* tmp=(Book\*)realloc(lib->books,sizeof(Book)\*len);

if (NULL == tmp) {

return false;

}

lib->books = tmp;

lib->capacity = len;

return true;

}

//上架图书

bool InsertTail(Library\* lib) { //arr[size++] = val;

if (lib->size == lib->capacity && !Grow(lib)) {

MessageBox(NULL, "上架图书失败！！", "上架图书结果", MB\_OKCANCEL);

return false;

}

InputBox(lib->books[lib->size].bookname,50,"图书名","上架图书");

InputBox(lib->books[lib->size].author, 50, "作者名", "上架图书");

char buff[10] = { 0 };

bool flag;

do {//若编号重复，请重新录入编号

flag = false;

InputBox(buff, 10, "编号", "上架图书");

int id = atoi(buff);

lib->books[lib->size].book\_id = id;

for (int i = 0; i < lib->size; i++)

{

if(lib->books[i].book\_id==id){

MessageBox(NULL,"编号重复，请重新录入","上架图书",MB\_OKCANCEL);

flag = true;

break;

}

}

} while (flag);

InputBox(buff,10,"价格","上架图书");

lib->books[lib->size].price = atof(buff);

char time[50] = { 0 };

GetTime(time);

strcpy(lib->books[lib->size].time,time);

lib->size++;

MessageBox(NULL,"上架图书成功！！","上架图书结果",MB\_OKCANCEL);

return true;

}

//\*\*\*\*\*\*\*\*\*下架图书\*\*\*\*\*\*\*\*\*\*\*

bool Delete(Library\* lib) {

char buff[20] = { 0 };

InputBox(buff, 20, "下架书籍\n name or id", "下架图书");

if (strncmp(buff, "name", 4) == 0) {

return DeleteByName(lib);

}

if (strncmp(buff, "id", 2) == 0) {

return DeleteById(lib);

}

MessageBox(NULL,"该属性不存在\n","下架失败",MB\_OKCANCEL);

return false;

}

//考虑书名是否重复

bool DeleteByName(Library\* lib) {

char buff[20] = { 0 };

InputBox(buff, 20, "请输入下架图书书名", "下架图书");

for (int i = 0; i < lib->size; i++) {

if (!strcmp(lib->books[i].bookname, buff)) {

//找到待删除图书

for (int j = i + 1; j < lib->size; j++) {

lib->books[j - 1] = lib->books[j];

}

MessageBox(NULL, "下架图书成功！！", "下架图书结果", MB\_OK);

lib->size--;

return true;

}

MessageBox(NULL, "下架图书失败！！", "下架图书结果", MB\_OK);

return false;

}

}

bool DeleteById(Library\* lib) {

char buff[20] = { 0 };

InputBox(buff, 20, "请输入下架图书编号", "下架图书");

int id = atoi(buff);

for (int i = 0; i < lib->size; i++) {

if (lib->books[i].book\_id == id) {

//找到待删除图书

for (int j = i + 1; j < lib->size; j++) {

lib->books[j - 1] = lib->books[j];

}

lib->size--;//\*\*\*\*\*\*\*重点

MessageBox(NULL, "下架图书成功！！", "下架图书结果", MB\_OK);

return true;

}

}

MessageBox(NULL, "下架图书失败！！", "下架图书结果", MB\_OK);

return false;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*查找图书 -- 书名 编号 价格区间 上架日期 (2023年) 作者

int Search(Library\* lib) {

char buff[20] = { 0 };

bool flag;

do{

InputBox(buff, 20, "查找书籍\n 按照属性:\n书名\n作者\n编号\n年份\n价格区间\n", "查找图书");

flag = false;

if (strcmp(buff, "书名") == 0) {

return SearchByName(lib);

}

if (strcmp(buff, "作者") == 0) {

return SearchByAuthor(lib);

}

if (strcmp(buff, "编号") == 0) {

return SearchById(lib);

}

if (strcmp(buff, "年份") == 0) {

return SearchByDate(lib);

}

if (strcmp(buff, "价格区间") == 0) {

return SearchByPriceRange(lib);

}

else {

MessageBox(NULL, "该属性不存在\n请重新输入", "查找图书", MB\_OKCANCEL);

flag = true;

}

} while (flag);

return -1;

}

int SearchByName(Library\* lib) {

char buff[20] = { 0 };

InputBox(buff, 20, "请输入待查找图书书名", "查找图书");

for (int i = 0; i < lib->size; i++) {

if (!strcmp(lib->books[i].bookname,buff)) {

char res[128] = { 0 };//要打印查找到的书的信息

sprintf(res, "编号:%d 书名:%s 作者:%s 价格:%lf 上架时间:%s"

, lib->books[i].book\_id, lib->books[i].bookname, lib->books[i].author

, lib->books[i].price, lib->books[i].time);

strcat(res, "\n查找成功!\n");

MessageBox(NULL, res, "查找图书", MB\_OKCANCEL);

return i;

}

}

MessageBox(NULL, "查找失败", "查找图书", MB\_OKCANCEL);

return -1;

}

int SearchById(Library\* lib) {

char buff[20] = { 0 };

InputBox(buff, 20, "请输入待查找图书编号", "查找图书");

int id = atoi(buff);

for (int i = 0; i < lib->size; i++) {

if (lib->books[i].book\_id == id) {

char res[128] = { 0 };//要打印查找到的书的信息

sprintf(res,"编号:%d 书名:%s 作者:%s 价格:%lf 上架时间:%s"

,lib->books[i].book\_id, lib->books[i].bookname, lib->books[i].author

,lib->books[i].price, lib->books[i].time);

strcat(res, "\n查找成功!\n");

MessageBox(NULL, res, "查找图书", MB\_OKCANCEL);

return i;

}

}

MessageBox(NULL, "查找失败", "查找图书", MB\_OKCANCEL);

return -1;

}

bool SearchByPriceRange(Library\* lib) {

char buff[20] = { 0 };

InputBox(buff, 20, "请输入价格下限", "查找图书");

double lowprice = atof(buff);

InputBox(buff, 20, "请输入价格上限", "查找图书");

double highprice = atof(buff);

int index = -1;

char res[1024] = { 0 };//为什么要多创造这个数组，因为 在这个区间内有很多书，为了存多本书信息

char tmp[128] = { 0 };

for (int i = 0; i < lib->size; i++) {

if (lib->books[i].price>=lowprice&&lib->books[i].price<=highprice) {

sprintf(tmp, "编号:%d 书名:%s 作者:%s 价格:%lf 上架时间:%s\n"

, lib->books[i].book\_id, lib->books[i].bookname, lib->books[i].author

, lib->books[i].price, lib->books[i].time);

strcat(res,tmp);

}

}

if(res[0]=='\0'){

//!strcmp(res,'\0')

//strcmp( )函数传递的参数是指向字符串的指针，而不是单个的字符。

// 可以通过将该字符包装在一个字符串中，或者使用指针来传递。

MessageBox(NULL, "查找失败\n不存在该价格区间的书籍", "查找图书", MB\_OKCANCEL);

return index;

}

strcat(res, "查找成功！");

MessageBox(NULL, res, "查找图书", MB\_OKCANCEL);

return true;

}

int SearchByDate(Library\* lib) {

char buff[20] = { 0 };

InputBox(buff, 20, "请输入待查找年份\n格式:2023/2019", "查找图书");

char res[1024] = { 0 };

char tmp[128] = { 0 };//因为 有好多本书，所以都要存

bool flag = false;

for (int i = 0; i < lib->size; i++) {

char year[128] = { 0 };

strcpy(year,lib->books[i].time);

char\* y = strtok(year,"-");//分割函数

if (!strcmp(buff,y)) {

sprintf(tmp, "编号:%d 书名:%s 作者:%s 价格:%2.lf 上架时间:%s\n"

, lib->books[i].book\_id, lib->books[i].bookname, lib->books[i].author

, lib->books[i].price, lib->books[i].time);

strcat(res, tmp);

flag = true;

}

}

if(flag){

strcat(res, "查找成功！");

MessageBox(NULL, res, "查找图书", MB\_OKCANCEL);

}

else

{

MessageBox(NULL, "查找失败", "查找图书", MB\_OKCANCEL);

}

return flag;

}

int SearchByAuthor(Library\* lib) {

char buff[20] = { 0 };

InputBox(buff, 20, "请输入待查找图书作者", "查找图书");

char res[1024] = { 0 };

char tmp[128] = { 0 };

bool flag = false;

for (int i = 0; i < lib->size; i++) {

if (!strcmp(lib->books[i].author,buff)) {

sprintf(tmp, "编号:%d 书名:%s 作者:%s 价格:%lf 上架时间:%s"

, lib->books[i].book\_id, lib->books[i].bookname, lib->books[i].author

, lib->books[i].price, lib->books[i].time);

strcat(res,tmp);

flag = true;

}

}

if (!flag) {

MessageBox(NULL, "查找失败", "查找图书", MB\_OKCANCEL);

}

else {

strcat(res, "\n查找成功");

MessageBox(NULL, res, "查找图书", MB\_OKCANCEL);

}

return flag;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//修改 书名 编号 id 价格区间 上架日期

int Change(Library\* lib) {

char buff[20] = { 0 };

bool flag;

do {

InputBox(buff, 20, "修改书籍\n 按照属性:\n书名\n作者\n编号\n", "修改图书");

flag = false;

if (strcmp(buff, "书名") == 0) {

return ChangeByName(lib);

}

if (strcmp(buff, "作者") == 0) {

return ChangeByAuthor(lib);

}

if (strcmp(buff, "编号") == 0) {

return ChangeById(lib);

}

else {

MessageBox(NULL, "该属性不存在\n请重新输入", "修改图书", MB\_OKCANCEL);

flag = true;

}

} while (flag);

return -1;

}

int ChangeByName(Library\* lib) {

char Bold[20] = { 0 };

InputBox(Bold, 20, "请输入待修改图书书名", "修改图书");

char Bnew[20] = { 0 };

InputBox(Bnew, 20, "请输入修改后的图书书名", "修改图书");

int j = 0;

for (int i = 0; i < lib->size; i++) {

if (!strcmp(lib->books[i].bookname, Bold)) {

strcpy(lib->books[i].bookname, Bnew);

j = 1;

}

}

if (j) {

MessageBox(NULL, "修改成功", "修改图书", MB\_OKCANCEL);

}

else {

MessageBox(NULL, "修改失败", "查找图书", MB\_OKCANCEL);

}

return j;

}

int ChangeById(Library\* lib) {

char Bold[20] = { 0 };

InputBox(Bold, 20, "请输入待修改图书编号", "修改图书");

char Bnew[20] = { 0 };

InputBox(Bnew, 20, "请输入修改后的图书编号", "修改图书");

int j = 0;

for (int i = 0; i < lib->size; i++) {

if (lib->books[i].book\_id == atoi(Bold)) {

lib->books[i].book\_id = atoi(Bnew);

j = 1;

}

}

if (j) {

MessageBox(NULL, "修改成功", "修改图书", MB\_OKCANCEL);

}

else {

MessageBox(NULL, "修改失败", "查找图书", MB\_OKCANCEL);

}

return j;

}

int ChangeByAuthor(Library\* lib) {

char Bold[20] = { 0 };

InputBox(Bold, 20, "请输入待修改图书作者", "修改图书");

char Bnew[20] = { 0 };

InputBox(Bnew, 20, "请输入修改后的图书作者", "修改图书");

int j = 0;

for (int i = 0; i < lib->size; i++) {

if (!strcmp(lib->books[i].author, Bold)) {

strcpy(lib->books[i].author, Bnew);

j = 1;

}

}

if (j) {

MessageBox(NULL, "修改成功", "修改图书", MB\_OKCANCEL);

}

else {

MessageBox(NULL, "修改失败", "查找图书", MB\_OKCANCEL);

}

return j;

}

//展示

void Show(Library\* lib) {

char result[1024] = {0};

strcat(result,"编号\t书名\t作者\t价格\t上架时间\n");

char buff[128] = { 0 };//存一本书

for (int i = 0; i < lib->size;i++) {

sprintf(buff,"%d\t%s\t%s\t%2.lf\t%s\n",lib->books[i].book\_id,lib->books[i].bookname,lib->books[i].author,lib->books[i].price,lib->books[i].time);

strcat(result,buff);//把buff连接到result内，存到result内

memset(buff,0,128);

}

MessageBox(NULL,result,"书籍信息",MB\_OKCANCEL);

}

===================================================

book\_test.cpp

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <graphics.h>

#include "book.h"

#pragma comment(lib,"winmm.lib")

void Start(Library\* lib){

IMAGE img1,img2;

loadimage(&img1, "C:\\Users\\xxx\\Desktop\\封面.jpg", 640, 480, true);

loadimage(&img2, "C:\\Users\\xxx\\Desktop\\界面.jpg", 640, 480, true);

putimage(0, 0, &img1);

MOUSEMSG msg;

//播放音乐

while (1) {

msg = GetMouseMsg();

//播放音乐

if (msg.uMsg == WM\_LBUTTONDOWN && msg.x > 90 && msg.x < 134 && msg.y>186 && msg.y < 318) {

mciSendString("open C:\\Users\\xxx\\Desktop\\1.mp3 alias m", 0, 0, 0);

mciSendString("play m repeat", 0, 0, 0);//开始播放音乐，repeat重复播

break;

}

}

while(1){

msg = GetMouseMsg();

if (msg.uMsg == WM\_LBUTTONDOWN && msg.x > 330 && msg.x < 385 && msg.y>200 && msg.y < 250) {

char c[20] = { 0 };

InputBox(c,10, "用户名id:", "登录");

char d[20] = { 0 };

InputBox(d,10, "密码passwd", "登录");

if(!strcmp(c,"123")&&!strcmp(d,"123")) {

MessageBox(NULL,"登录成功!!!","登录",MB\_OKCANCEL);

//进入到下一个界面

putimage(0, 0, &img2);

break;

}

else {

MessageBox(NULL, "密码账号输入错误\n登录失败", "登录结果", MB\_OKCANCEL);

}

}

}

//printf("%d %d",msg.x,msg.y);

while (1) {

msg = GetMouseMsg();

if (msg.uMsg == WM\_LBUTTONDOWN && msg.x > 15 && msg.x < 150 && msg.y>83 && msg.y < 135) {

InsertTail(lib);// //点击上架图书

}

else if (msg.uMsg == WM\_LBUTTONDOWN && msg.x > 0 && msg.x < 100 && msg.y>0 && msg.y < 100) {

Show(lib);//显示全部图书

}

else if (msg.uMsg == WM\_LBUTTONDOWN && msg.x > 15 && msg.x < 150 && msg.y>210 && msg.y < 260) {

//printf("%d %d",msg.x,msg.y);

Delete(lib);//下架图书

}

else if (msg.uMsg == WM\_LBUTTONDOWN && msg.x > 500 && msg.x < 640 && msg.y>110 && msg.y < 166) {

Search(lib);//查找

}

else if (msg.uMsg == WM\_LBUTTONDOWN && msg.x > 500 && msg.x < 640 && msg.y>210 && msg.y < 260) {

Change(lib);//更改

}

else if (msg.uMsg == WM\_LBUTTONDOWN && msg.x > 270 && msg.x < 400 && msg.y>150 && msg.y < 200) {

MessageBox(NULL, "退出", "图书管理", MB\_OK);

break;

}

}

}

int main(){

Library library;

Book book;

InitLibrary(&library);

initgraph(640,480,SHOWCONSOLE);

//=========

Start(&library);

//===================

system("pause");

closegraph();

}