**南京航空航天大学金城学院**

**《C语言程序设计》课程设计报告**

**题目： 书籍信息管理系统的设计与实现**

**学号： 2020027316**

**姓名： 严奕杰**

**完成日期： 2021年2月20日**

**目 录**

**一、 需求分析**

**二、 程序运行平台**

**三、 系统主要功能**

**四、 系统各模块的功能说明**

**五、 程序结构（画流程图）**

**六、 数据结构说明**

**七、 函数功能说明（含特色函数）**

**八、 程序源代码**

**一、 需求分析**

设计与分析一个学生信息管理系统，能够对学生的学号、姓名和各科成绩进行统计、处理和更新，并且可以方便学校老师和领导对学生成绩进行整体分析。

该系统具有以下几个功能：

* 学生信息界面美观、简洁，采用菜单式显示功能选择；
* 能够从磁盘文件输入和输出数据；
* 能够对学生信息进行检索操作；
* 具有增、删、改信息的功能；
* 能够对学生的成绩从高到低排序；
* 使用结构体**链表**数据类型对学生信息进行描述和存储。

**二、 程序运行平台**

**Windows10**

**三、 系统主要功能**

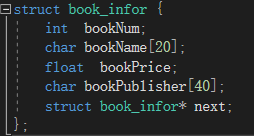
* 从磁盘文件输入和输出数据
* 对数据增删改查
* 对数据降序排序
* 用链表进行数据存储

**四、 系统各模块的功能说明**

1. 界面模块:与用户交互
2. 文件处理模块:读取,保存文件
3. 事件处理模块:应答用户选择
4. 数据处理模块:录入,查找,修改,显示存储数据

**五、 程序结构（画流程图）**

**六、 数据结构说明**



1. **七、 函数功能说明（含特色函数）**
2. ScreenDisplay:UI界面显示
3. Setup\_Infor:建立链表函数
4. Dle\_Infor:删除书籍信息
5. Mod\_Infor:修改书籍信息
6. Input\_Infor:录入书籍信息
7. Check\_Infor:查看书籍信息
8. Display\_Infor:显示所有书籍信息

**八、 程序源代码**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <malloc.h>

#include <stdlib.h>

#include <string.h>

#include "sourse.h"

/\*###############################################################

###################### 主程序 ##############################\*/

int main()

{

FILE\* fp;

char state; //用于存储一个字符（y or Y) ；

BOOK\_INFOR newstruct = { 0,"0",0.000,"0" }, \* head, \* Ftemp, \* Ftemp2; // Ftemp与Ftepm2在读写函数时使用。

int select = 0, num = 0; // 定义一个选择状态数select和一个书籍的学号num；

head = (BOOK\_INFOR\*)malloc(sizeof(BOOK\_INFOR)); // 先申请一个动态空间给head，避免野指针；

printf("\n\n\n\n\n\t\t\t\t\t欢迎进入书籍管理系统\n\n\n\n\t\t\t\t\t\t\t\tby 严奕杰\n");

printf("\n\n\n请问之前是否打开并保存过数据信息，如果是，则按 'Y' 键载入，否则按 'N' 键或其它字母新建数据！\n");

state = getchar();

getchar(); // 消除本次回车键；

/\*###############################################################

###################### 载入数据模块 #######################\*/

if (state == 'y' || state == 'Y')

{

fp = fopen("data.dat", "rb");

/\*###############################################################\*/

if (NULL == fp) // 判断文件是否打开成功

{

printf("文件不存在，打开文件失败\n");

printf("本程序将退出，按任意键结束");

getchar();

exit(-1);

}

/\*###############################################################\*/

Ftemp = head;

while (1 == fread(Ftemp, sizeof(BOOK\_INFOR), 1, fp)) // 判断文件是否读取完毕

{

Ftemp2 = Ftemp;

Ftemp->next = (BOOK\_INFOR\*)malloc(sizeof(BOOK\_INFOR));

Ftemp = Ftemp->next;

}

Ftemp->next = NULL; // 把导入的链表的最后一个位置的next标记为NULL;

free(Ftemp); // 释放掉没使用的空间；

fclose(fp);

printf("载入信息成功，请按G徤或其它键继续！\n");

getchar();

getchar(); // 消除本次的回车键；

}

/\*###############################################################\*/

else

{

free(head); // 释放掉文件开始部分申请的空间；

head = Setup\_Infor(); // 建立函数链表；

}

/\*################################################################

################### 进入整个循环 ########################

###############################################################\*/

LOOP:do

{

ScreenDisplay(); // 显示输出介面；

printf("\n\n\n\n请输入对应数字\n");

scanf("%d", &select);

getchar(); // 消除本次的回车键；

switch (select) {

case 1: // 录入书籍信息入口；

/\*###############################################################

#################### 录入书籍信息模块 #####################\*/

do

{

ScreenDisplay();

printf("请你输入书籍的编号：\n");

scanf("%d", &newstruct.bookNum);

getchar(); // 消除本次回车键；

printf("请你输入书籍的书名：\n");

scanf("%s", newstruct.bookName);

getchar(); // 消除本次的回车键；

printf("请你输入书籍的价格： \n");

scanf("%f", &newstruct.bookPrice);

getchar(); // 消除本次的回车键；

printf("请你输入书籍的出版商： \n");

scanf("%s", newstruct.bookPublisher);

getchar();

Input\_Infor(head, &newstruct);

printf("是否继续输入,Y or N ? \n");

state = getchar();

getchar(); // 消除本次的回车键；

} while (state == 'y' || state == 'Y');

break;

case 2: // 查看书籍信息入口；

/\*###############################################################

#################### 查看书籍信息模块 ####################\*/

do

{

ScreenDisplay();

printf("请你输入需要查看的书籍的编号 \n");

scanf("%d", &num);

getchar(); // 消除本次的回车键；

Check\_Infor(head, num);

printf("是否继续查看书籍信息, Y or N ? \n");

state = getchar();

getchar(); // 消除本次的回车键；

} while (state == 'y' || state == 'Y');

break;

case 3: // 删除书籍信息入口；

/\*###############################################################

#################### 删除书籍信息模块 ####################\*/

do

{

ScreenDisplay();

printf("请输入需要删除的书籍的编号：\n");

scanf("%d", &num);

getchar(); // 消除本次的回车键；

printf("确定要删除？ （Y or N）");

state = getchar();

getchar(); // 消除本次的回车键；

if (state == 'Y' || state == 'y')

head = Dle\_Infor(head, num);

printf("是否继续删除书籍信息, Y or N ? \n");

state = getchar();

getchar(); // 消除本次的回车键；

} while (state == 'y' || state == 'Y');

break;

case 4: // 修改书籍信息入口；

/\*###############################################################

#################### 修改书籍信息模块 ####################\*/

do

{

ScreenDisplay();

printf("请你输入需要修改的书籍的编号：\n");

scanf("%d", &newstruct.bookNum);

getchar(); // 消除本次的回车键；

printf("请你输入需要修改的书籍的书名：\n");

scanf("%s", newstruct.bookName);

getchar(); // 消除本次的回车键；

printf("请你输入需要修改的书籍的价格： \n");

scanf("%f", &newstruct.bookPrice);

getchar(); // 消除本次的回车键；

printf("请你输入需要修改的书籍的出版商： \n");

scanf("%s", newstruct.bookPublisher);

getchar(); // 消除本次的回车键；

head = Mod\_Infor(head, &newstruct);

printf("是否继续修改书籍信息, Y or N ? \n");

state = getchar();

getchar(); // 消除本次的回车键；

} while (state == 'y' || state == 'Y');

break;

case 5:

/\*###############################################################

#################### 显示所有书籍信息模块 ####################\*/

BubbleSort(head); //排序

do

{

Display\_Infor(head);

printf("是否需要再重复显示一次书籍信息, Y or N ? \n");

state = getchar();

getchar(); // 消除本次的回车键；

} while (state == 'y' || state == 'Y');

break;

case 6: // 退出书籍管理系统入口；

/\*###############################################################

#################### 退出书籍管理系统模块 ####################\*/

printf("是否保存以上操作， Y or N ? ");

state = getchar();

getchar(); // 消除本次的回车键；

if (state == 'y' || state == 'Y') // 加一个保存界面；

{

fp = fopen("data.dat", "wb");

/\*###############################################################\*/

if (NULL == fp) // 判断文件是否打开失败；

{

printf("Open files error\n");

exit(-1);

}

/\*###############################################################\*/

Ftemp = head;

while (Ftemp->next != NULL) // 写文件的判断条件；

{

fwrite((void\*)Ftemp, sizeof(BOOK\_INFOR), 1, fp);

Ftemp = Ftemp->next;

}

fwrite((void\*)Ftemp, sizeof(BOOK\_INFOR), 1, fp); // 把指针的最后一个写入的文件中；

fclose(fp);

printf("恭喜你，保存成功。\n");

}

printf("按任意键继续！\n");

getchar();

exit(0);

break;

/\*###############################################################

############ 当输入的不是有效的数字模块 ###################\*/

default:printf("你输入的有误，按任意键返回主界面\n ");

getchar();

}

} while (1);

/\*################################################################

################### 到此,整个循环结束 #####################

###############################################################\*/

return 0;

}

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdlib.h>

#include<stdio.h>

#include "sourse.h"

void BubbleSort(BOOK\_INFOR\* head)

{

BOOK\_INFOR\* cur = NULL;

BOOK\_INFOR\* teil = NULL;

cur = head->next;

while (cur != teil) {

while (cur->next != teil) {

if (cur->bookNum > cur->next->bookNum) {

float temp = cur->bookNum;

cur->bookNum = cur->next->bookNum;

cur->next->bookNum = temp;

}

cur = cur->next;

}

teil = cur;

cur = head->next;

}

}

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include "sourse.h"

void Check\_Infor(BOOK\_INFOR\* phead, int bookNum)

{

BOOK\_INFOR\* ptemp;

ptemp = phead->next;

while (ptemp->next != NULL && ptemp->bookNum != bookNum) //查找书籍信息所在链表中的位置；

ptemp = ptemp->next;

if (ptemp->bookNum == bookNum) //判断书籍信息所在链表中的位置；

{

printf("\t编号\t书名\t价格\t出版商\n");

printf("\t%d\t%s\t%.2f\t%s\n", ptemp->bookNum, ptemp->bookName, ptemp->bookPrice, ptemp->bookPublisher);

}

else

printf("无效的查找信息！\n");

}

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <malloc.h>

#include <stdlib.h>

#include "sourse.h"

BOOK\_INFOR\* Setup\_Infor(void)

{

BOOK\_INFOR\* phead, \* pnew, \* ptail;

phead = (BOOK\_INFOR\*)malloc(sizeof(BOOK\_INFOR)); //申请一个动态存储空间；

if (NULL == phead) //判断是否申请成功；

{

printf("申请一个动态存储空间失败\n");

exit(-1);

}

pnew = ptail = phead;

phead->next = NULL;

return phead;

}

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <string.h>

#include<stdlib.h>

void ScreenDisplay(void)

{

system("cls");

printf("\n\n\n\n\n");

printf("#############################################################################################\n");

printf("\t\t\t欢迎来到书籍信息管理系统\n");

printf("\t\t\t### 1 录入书籍信息\n");

printf("\t\t\t### 2 查看书籍信息\n");

printf("\t\t\t### 3 删除书籍信息\n");

printf("\t\t\t### 4 修改书籍信息\n");

printf("\t\t\t### 5 显示所有书籍信息\n");

printf("\t\t\t### 6 退出书籍管理系统\n");

printf("###############################################################################################\n");

}

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <malloc.h>

#include <stdlib.h>

#include <string.h>

#include "sourse.h"

BOOK\_INFOR\* Input\_Infor(BOOK\_INFOR\* phead, BOOK\_INFOR\* newstruct)

{

BOOK\_INFOR\* pnew, \* ptemp;

pnew = (BOOK\_INFOR\*)malloc(sizeof(BOOK\_INFOR)); //申请一个动态存储空间；

if (NULL == pnew) //判断是否申请成功；

{

printf("申请一个动态存储空间失败\n");

exit(-1);

}

if (NULL == phead->next) //判断是否为空链表，如果是，则直接放在链表开头

{

\*pnew = \*newstruct;

phead->next = pnew;

pnew->next = NULL;

}

else

{

ptemp = phead;

while (ptemp->next->next != NULL && newstruct->bookNum >= ptemp->next->bookNum) //判断书籍信息录入的位置

ptemp = ptemp->next;

if (newstruct->bookNum < ptemp->next->bookNum) //如果是在链表中找到了满足条件的位置，插入链表；

{

\*pnew = \*newstruct;

pnew->next = ptemp->next;

ptemp->next = pnew;

}

else // 这种情况下，直接放在链表为尾；

{

\*pnew = \*newstruct;

pnew->next = NULL;

ptemp->next->next = pnew;

}

}

return phead;

}

#include <stdio.h>

#include <malloc.h>

#include <stdlib.h>

#include <string.h>

#include "sourse.h"

BOOK\_INFOR\* Dle\_Infor(BOOK\_INFOR\* phead, int bookNum)

{

BOOK\_INFOR\* ptemp, \* pdel;

ptemp = phead;

while (ptemp->next->next != NULL && ptemp->next->bookNum != bookNum) //查找要删除的书籍信息所在链表中的位置

ptemp = ptemp->next;

if (ptemp->next->bookNum == bookNum) //判断是否是要删除的位置

{

pdel = ptemp->next;

ptemp->next = pdel->next;

free(pdel);

printf("删除成功\n");

}

else

{

printf("无效的查找内容！\n");

printf("删除失败！\n");

}

return phead;

}

#include <stdio.h>

#include<stdlib.h>

#include "sourse.h"

void Display\_Infor(BOOK\_INFOR\* phead)

{

system("cls");

BOOK\_INFOR\* ptemp;

if (phead->next == NULL) // 空链表；

printf("对不起，没有内容\n");

else

{

ptemp = phead->next;

printf("\t编号\t书名\t价格\t出版商\n");

while (ptemp->next != NULL)

{

printf("\t%d\t%s\t%.2f\t%s\n", ptemp->bookNum, ptemp->bookName, ptemp->bookPrice, ptemp->bookPublisher);

ptemp = ptemp->next;

}

printf("\t%d\t%s\t%.2f\t%s\n", ptemp->bookNum, ptemp->bookName, ptemp->bookPrice, ptemp->bookPublisher);

}

}

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <string.h>

#include "sourse.h"

BOOK\_INFOR\* Mod\_Infor(BOOK\_INFOR\* phead, BOOK\_INFOR\* modify) // 链表的 Head 地址与需要修改的结构体的地址；

{

BOOK\_INFOR\* ptemp;

ptemp = phead->next;

while (ptemp->next != NULL && ptemp->bookNum != modify->bookNum) // 查找要修改的书籍信息所在链表中的位置

ptemp = ptemp->next;

if (ptemp->bookNum == modify->bookNum) //判断是否是是要修改的位置

{

strcpy(ptemp->bookName, modify->bookName);

ptemp->bookPrice= modify->bookPrice;

strcpy(ptemp->bookPublisher, modify->bookPublisher);

printf("修改成功\n");

}

else // 这种情况是一直查到链表结尾仍没有的情况；

{

printf("没有你修改的书籍的信息的内容！\n");

printf("修改失败！\n");

}

return phead;

}

typedef struct book\_infor BOOK\_INFOR;

struct book\_infor {

int bookNum;

char bookName[20];

float bookPrice;

char bookPublisher[40];

struct book\_infor\* next;

};

void ScreenDisplay(void); // 界面函数；

BOOK\_INFOR\* Setup\_Infor(void); // 建立链表函数；

BOOK\_INFOR\* Dle\_Infor(BOOK\_INFOR\* head, int bookNum); // 删除书籍信息函数；

BOOK\_INFOR\* Mod\_Infor(BOOK\_INFOR\* head, BOOK\_INFOR\* modify); // 修改书籍信息函数；

BOOK\_INFOR\* Input\_Infor(BOOK\_INFOR\* phead, BOOK\_INFOR\* newstruct); // 录入书籍信息函数；

void Check\_Infor(BOOK\_INFOR\* phead, int bookNum); // 查看书籍信息函数；

void Display\_Infor(BOOK\_INFOR\* phead); // 显示所有信息函数；

void BubbleSort(BOOK\_INFOR\* head); // 自动排序