

航空航天大学金城学院

"C 语言程序设计"课程设计报告

图书信息管理系统

**:

**:

**:

**:

日期:

目录

一、需求分析:	3
二、程序的主要功能:	3
三、程序运行平台:	3
四、系统总框架图:	6
五、程序类的说明:	6
六、模块分析:	8
七、比拟有特色的函数:	9
八、存在的缺乏与编程体会.....	10
九、程序源代码:	10

一、需求分析

设计与分析一个图书信息管理系统，能够对图书的根据书名、价格或进展统计、处理和更新，并且可以方便学校教师和领导对图书进展整体分析。

二、程序的主要功能

- 图书信息界面美观、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；
- 能够对图书信息进展检索操作；
- 具有增、删、改信息的功能；
- 使用构造体链表数据类型对图书信息进展描述和存储。

三、程序运行平台

VC6.0

运行程序“library.e*e”，进入登录界面如下

**

***** 欢送光临 *****

***** 图书信息管理系统 *****

=====1-用户登录=====

=====0-退出系统=====

请输入您的选择:

一、输入1进展登录

图书信息管理系统

- || 1-添加图书
- 2-删除图书 ||
- || 3-图书列表
- 4-图书排序 ||
- || 5-查询图书
- 6-修改图书 ||
- || 7-录入数据
- 0-退出系统 ||

请输入所选择的序号:

选择 1 进入添加图书模块:

请输入您要添加的图书的

选择 2 进入删除图书信息模块:

请输入要删除的图书编号:

选择 3 进入显示图书列表模块

图书信息列表如下

=====

编号	图书名	作者名	类别	出版时间	价格
2	2	2	2	2	2.00
3	3	3	3	3	3.00
33	15	5	2	5	52 5.00
5	5	5	5	5	5.00

=====;

选择 4.进入修改个人信息模块

1-按**修改

请输入您的数字选择:

选择 5.进入图书排序模块

=====

=====

- | | | | |
|----|-----------|-----------|----|
| ** | 1-按图书编号排序 | 2-按出版时间排序 | ** |
| ** | 3-按图书价格排序 | 4-按图书名排序 | ** |
| ** | 5-按作者名排序 | 0-取消排序操作 | ** |

=====

=====

请输入您选择的编号:选择 6.进入修改图书模块

*请输入要修改的图书编号:

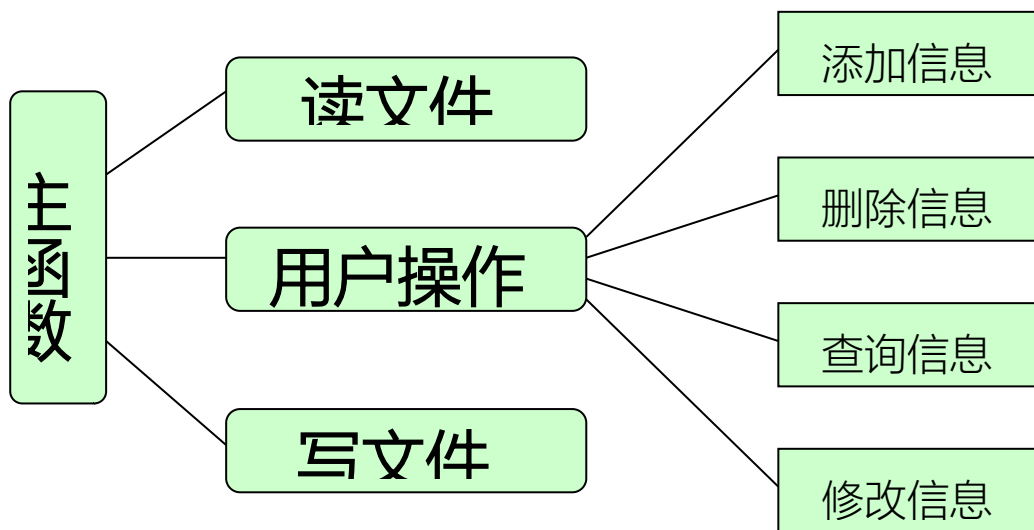
选择 7.进入录入信息模块

注意:输入图书编码为 0 时完毕!

请输入图书编号:

选择 0.退出系统

四、系统总框架图



五、程序类的说明

```
struct book{  
  
    int num;  
  
    char bname[50];  
  
    char wname[20];  
  
    char press[50];  
  
    char sort[50];  
  
    int time;
```

```
float price;

struct book *ne*t;//定义文件头指针

};
```

六、模块分析

1. 添加模块

系统将提示用户输入新添加的图书的信息，插入在链表中

2. 删除模块

首先由用户输入图书的编号，然后由删除模块判断 $p \rightarrow \text{num}$ 和用户输入的编号是否一样，一样则从链表中删除。

3. 图书列表模块

传递一个头指针给 `print_book` 函数，假设头指针不为空，则将整个链表的数据输出。

4. 图书排序模块

首先由用户判断由什么关键词进展排序，然后运行相应的运行模块。

5. 查询图书模块

系统将会把所有航班的信息显示出来。由用户输入图书的编号，判断是 $p \rightarrow \text{num}$ 和用户输入的编号是否一样，一样则输出。

6. 修改图书模块

按照编号检索，然后修改相应的容。

7. 录入模块

用 `while` 循环批量录入图书信息。

0. 退出系统

七、比拟有特色的函数

```
void num_pai*(u(struct book *head)
{
    struct book *a[1000],*p,*p1,*temp;

    int i,k,inde*,n=0;

    char b;

    p1=head;

    for(p=head;p;p=p->ne*t)

        n++;

    for(i=0;i<n;i++){

        a[i]=p1;

        p1=p1->ne*t;

    }

    for(k=0;k<n-1;k++){

        inde*=k;

        for(i=k+1;i<n;i++){

            if(a[i]->num<a[inde*]->num)

                inde*=i;

        }

        temp=a[inde*];

        a[inde*]=a[k];

        a[k]=temp;

    }
}
```

```

}

printf("排序成功!\n");

printf("是否显示排序结果"(y/n)\n");

scanf("%s",&b);

getchar();

switch(b){

case 'n':

    break;

case 'y':

    printf("

=====

=====\n");

    printf("    ** 编号   图书名   作者名   类别   出版时间   价格 **\n");

    for(i=0;i<n;i++){

        printf("    ** %d       %s       %s       %s       %s       %d

%.2f **\n",a[i]->num,a[i]->bname,a[i]->wname,a[i]->press,a[i]->sort,a[i]->time,a[i]->price);

    }

    printf("

=====

=====\n");

    break;

default:

```

```
        printf("您的输入有误!\n");  
  
        break;  
  
    }  
  
}
```

八、存在的缺乏与对策、编程体会。

排序的方式过于复杂，有待改良。

九、参考文献

百度文库

十、程序源代码

```
*include<stdio.h>  
  
*include<stdlib.h>  
  
*include<string.h>  
  
struct book{  
  
    int num;  
  
    char bname[50];  
  
    char wname[20];  
  
    char press[50];  
  
    char sort[50];  
  
    int time;  
  
    float price;
```

```

struct book *ne*t;//定义文件头指针

};

struct book *creatbook(); //创立链表

struct book *addbook(struct book *head); //添加图书

int yanzheng(struct book *head,int m); //验证新添加的图书编码是否已存在

void deletebook(struct book *head); //删除图书

void fprint(struct book *head); //将链表写入文件

struct book *load(); //从文件中读取信息并建成链表

void print_book(struct book *head); //将链表信息输出

void cha*un(struct book *head); //查询图书信息

void num_cha*un(struct book *head); //按图书编号查询图书

void wname_cha*un(struct book *head); //按作者名查询图书

void sort_cha*un(struct book *head); //按类别查询图书

void time_cha*un(struct book *head); //按出版时间查询图书

void bname_cha*un(struct book *head); //按图书名查询图书

void *iugai(struct book *head); //修改图书信息

void pai*u(struct book *head); //对图书进展排序

void num_pai*u(struct book *head); //按图书编号排序

void time_pai*u(struct book *head); //按图书出版时间排序

void price_pai*u(struct book *head); //按图书价格排序

void bname_pai*u(struct book *head); //按图书名排序

void wname_pai*u(struct book *head); //按作者名排序

```

```

int main()
{
    int choice,n,*y=1,c,c1=1234;

    char a,d,b[10],b1[10]="yjk";

    struct book *head=NULL;

    while(y){

        system("cls");

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

        printf("                ***** 欢送光临 *****\n\n");

        printf("                ***** 图书信息管理系统 *****\n\n\n");

        printf("\n\n");

        printf("                =====1-用户登录=====\\n");

        printf("                =====0-退出系统=====\\n");

        printf("                请输入您的选择:");

        scanf("%d",&n);

        printf("\\n");

        getchar();

        switch(n){

        case 0:

            y=0;

            break;

```

```

case 1:

printf("      请输入您的用户名:");

gets(b);

printf("\n");

printf("      请输入您的密码:");

scanf("%d",&c);

printf("\n");

if(strcmp(b,b1)!=0||c!=c1){

    printf("      验证失败,请重新输入!\n");

    scanf("%c",&d);

getchar();

    system("cls");

}

else{

    printf("      验证通过!请按 Enter 键进入!\n");

    scanf("%c",&d);

    getchar();

    *=1;

    while(*){

        system("cls");

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

        printf("      *图书信息管理系统*\n");

```

```

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

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

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

printf("                || 1-添加图书                2-删除图书

||\n\n");

printf("                || 3-图书列表                4-图书排序

||\n\n");

printf("                || 5-查询图书                6-修改图书

||\n\n");

printf("                || 7-录入数据                0-退出系统

||\n\n");

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

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

printf("请输入所选择的序号:");

scanf("%d",&choice);

getchar();

system("cls");

switch(choice)

{

case 0:

    *=0;break;

case 1:

```

```
head=load();

if(head==NULL){

    printf("文件为空,请先录入数据!\n");

    getchar();

    break;

}

else{

    head=addbook(head);

    printf("添加成功!\n");

    printf("是否将新信息保存到文件"(y/n)\n");

    scanf("%c",&a);

    getchar();

    switch(a){

    case 'n':

        break;

    case 'y':

        fprintf(head);

        printf("保存成功!\n");

        getchar();

        break;

    }

    break;

}
```

```
    }  
  
case 2:  
  
    head=load();  
  
    if(head==NULL){  
  
        printf("文件为空,请先录入数据!\n");  
  
        getchar();  
  
        break;  
  
    }  
  
    else{  
  
        deletebook(head);  
  
        getchar();  
  
        break;  
  
    }  
  
    break;  
  
case 3:  
  
    head=load();  
  
    if(head==NULL){  
  
        printf("文件为空,请先录入数据!\n");  
  
        getchar();  
  
        break;  
  
    }  
  
    else{
```

```
        print_book(head);

        getchar();

        break;
    }

case 4:

    head=load();

    if(head==NULL){

        printf("文件为空,请先录入数据!\n");

        getchar();

        break;
    }

    else{

        pai*u(head);

        getchar();
    }

    break;

case 5:

    head=load();

    if(head==NULL){

        printf("文件为空,请先录入数据!\n");

        getchar();

        break;
```

```
    }  
  
    else{  
  
        cha*un(head);  
  
        getchar();  
  
    }  
  
    break;  
  
case 6:  
  
    head=load();  
  
    if(head==NULL){  
  
        printf("文件为空,请先录入数据!\n");  
  
        getchar();  
  
        break;  
  
    }  
  
    else{  
  
        *iugai(head);  
  
        getchar();  
  
        break;  
  
    }  
  
    break;  
  
case 7:  
  
    printf("注意:输入图书编码为 0 时完毕!\n");  
  
    head=creatbook();
```

printf("是否将输入的信息保存到文件以覆盖文件中已存在的信息

"(y/n)\n");

getchar();

scanf("%c",&a);

getchar();

switch(a){

case 'n':

break;

case 'y':

fprint(head);

printf("保存成功!\n");

getchar();

break;

}

break;

default:

printf("您的输入有误,请重新输入!\n");

getchar();

break;

}

}

}

```
        break;

        default:

            printf("        您的输入有误! 请重新输入!\n");

            getchar();

            break;

        }

    }

}
```

//录入数据并形成链表

```
struct book *creatbook()

{

    struct book *head,*tail,*p;

    int num,time,n;

    char bname[50],wname[20],press[50],sort[50];

    float price;

    int size=sizeof(struct book);

    head=tail=NULL;

    printf("请输入图书编号:");

    scanf("%d",&num);

    printf("请输入图书名:");

    scanf("%s",bname);

    getchar();
```

```
printf("请输入作者名:");

scanf("%s",wname);

getchar();

printf("请输入:");

scanf("%s",press);

getchar();

printf("请输入类别:");

scanf("%s",sort);

getchar();

printf("请输入出版时间:");

scanf("%d",&time);

getchar();

printf("请输入价格:");

scanf("%f",&price);

getchar();

while(1){

    p=(struct book *)malloc(size);

    p->num=num;

    strcpy(p->bname,bname);

    strcpy(p->wname,wname);

    strcpy(p->press,press);

    strcpy(p->sort,sort);
```

```
p->time=time;

p->price=price;

p->next=NULL;

if(head==NULL)

    head=p;

else

    tail->next=p;

tail=p;

do{

printf("请输入图书编号:");

scanf("%d",&num);

n=yanzheng(head,num);

if(n==0)

    break;

else

    printf("您输入的编号已存在,请重新输入!\n");

}while(1);

if(num==0)

    break;

else

{

    printf("请输入图书名:");
```

```
scanf("%s",bname);

getchar();

printf("请输入作者名:");

scanf("%s",wname);

getchar();

printf("请输入:");

scanf("%s",press);

getchar();

printf("请输入类别:");

scanf("%s",sort);

getchar();

printf("请输入出版时间:");

scanf("%d",&time);

getchar();

printf("请输入价格:");

scanf("%f",&price);

getchar();

}

}

return head;

}
```

//插入结点, 并且插入后仍按一定顺序

```
struct book *addbook(struct book *head)
{
    struct book *ptr,*p1,*p2,*p;
    char bname[50],wname[20],press[50],sort[50];
    int size=sizeof(struct book);
    int num,time,n=1;
    float price;
    do{
        printf("请输入图书编号:");
        scanf("%d",&num);
        n=yanzheng(head,num);
        if(n==0)
            break;
        else
            printf("您输入的编号已存在,请重新输入!\n");
    }while(1);
    printf("请输入图书名:");
    scanf("%s",bname);
    getchar();
    printf("请输入作者名:");
    scanf("%s",wname);
    getchar();
```

```
printf("请输入:");  
  
scanf("%s",press);  
  
getchar();  
  
printf("请输入类别:");  
  
scanf("%s",sort);  
  
getchar();  
  
printf("请输入出版时间:");  
  
scanf("%d",&time);  
  
getchar();  
  
printf("请输入价格:");  
  
scanf("%f",&price);  
  
getchar();  
  
p=(struct book *)malloc(size);  
  
p->num=num;  
  
strcpy(p->bname,bname);  
  
strcpy(p->wname,wname);  
  
strcpy(p->press,press);  
  
strcpy(p->sort,sort);  
  
p->time=time;  
  
p->price=price;  
  
p2=head;  
  
ptr=p;
```

```

while((ptr->num>p2->num)&&(p2->ne*t!=NULL)){

    p1=p2;

    p2=p2->ne*t;

}

if(ptr->num<=p2->num){

    if(head==p2)

        head=ptr;

    else{

        p1->ne*t=ptr;

        p->ne*t=p2;

    }

}

else{

    p2->ne*t=ptr;

    p->ne*t=NULL;

}

return head;

}

```

//验证添加的图书编号是否已存在

```
int yanzheng(struct book *head,int m)
```

```
{
```

```
    struct book *p;
```

```

p=head;

while(p!=NULL){

    if(p->num==m)

        break;

    p=p->ne*t;

}

if(p==NULL)

    return 0;

else

    return 1;

}

//将新链表写入文件中

void fprint(struct book *head)

{

    FILE *fp;

    char ch='1';

    struct book *p1;

    if((fp=fopen("f1.t*t","w"))==NULL){

        printf("File open error!\n");

        e*it(0);

    }

    fputc(ch,fp);

```

```

for(p1=head;p1;p1=p1->ne*t){
    fprintf(fp,"%d      %s      %s      %s      %s      %d
%f\n",p1->num,p1->bname,p1->wname,p1->press,p1->sort,p1->time,p1->price);
}
fclose(fp);
}
//从文件中读取图书信息
struct book *load()
{
    FILE *fp;
    char ch;
    struct book *head,*tail,*p1;
    head=tail=NULL;
    if((fp=fopen("f1.t*t","r"))==NULL){
        printf("File open error!\n");
        e*it(0);
    }
    ch=fgetc(fp);
    if(ch=='1'){
        while(!feof(fp)){
            p1=(struct book *)malloc(sizeof(struct book));

```

```
fscanf(fp, "%d%s%s%s%s%d%f\n", &p1->num, p1->bname, p1->wname, p1->press, p1->sort, &p1->time, &p1->price);
```

```
    if(head==NULL)
```

```
        head=p1;
```

```
    else
```

```
        tail->ne*t=p1;
```

```
        tail=p1;
```

```
    }
```

```
    tail->ne*t=NULL;
```

```
    fclose(fp);
```

```
    return head;
```

```
 }
```

```
 else
```

```
     return NULL;
```

```
 }
```

```
//将整个链表的信息输出
```

```
void print_book(struct book *head)
```

```
{
```

```
    struct book *ptr;
```

```
    if(head==NULL){
```

```
        printf("\n 没有信息!\n");
```

```
        return;
```

```

}

printf("                图书信息列表如下\n");

printf("

=====\\n

");

printf("                编号   图书名   作者名   类别   出版时间   价格\n");

for(ptr=head;ptr;ptr=ptr->ne*t)

    printf("                %d       %s       %s       %s       %s       %d

%.2f\n",ptr->num,ptr->bname,ptr->wname,ptr->press,ptr->sort,ptr->time,ptr->price);

printf("

=====\\n

");

}

//删除图书信息

void deletebook(struct book *head)

{

    int a;

    char b,ch='1';

    struct book *p1,*p2;

    FILE *fp;

    printf("请输入要删除的图书编号:");

    scanf("%d",&a);

```

```

p1=head;

if(p1->num==a&&p1->ne*t==NULL){ //对于文件中只有一组数据

    printf("是否清空文件!(y/n)\n");

    getchar();

    scanf("%c",&b);

    getchar();

    switch(b){

    case 'n':

        break;

    case 'y':

        if((fp=fopen("f1.t*t","w"))==NULL){

            printf("File open error!\n");

            e*it(0);

        }

        fclose(fp);

        printf("文件已清空!\n");

    }

}

else{

    while(p1->num!=a&&p1->ne*t!=NULL){

        p2=p1;

        p1=p1->ne*t;

```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/588064014023006073>