# 图书管理系统设计报告

#### ——软件工程x班 xxx xxxxxxxx

**一、功能分析**

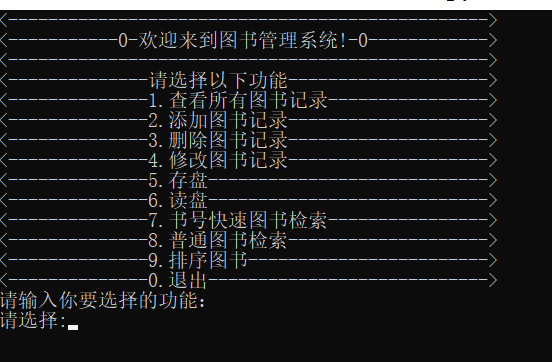
图书馆藏书丰富，书籍数目种类繁多，对于读者而言，要在大量的图书中查找检索到自己想要的书十分困难，而相对的，对于工作人员而言，要对大量的图书进行管理也是一件极其繁杂的事情。因而为了这类事情的相对便利，就需要图书管理系统进行辅助。

图书管理系统的主要功能有：1）添加、删除、修改图书记录 2）读盘、存盘 3)检索图书 4）排序图书。

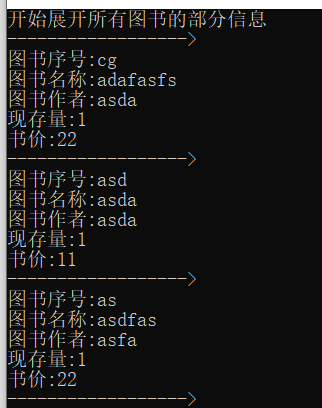
其中，就检索图书而言，可以对某项属性进行遍历查找，也可以通过书号建立索引（本次采用哈希表）来快速查找。就排序而言，可通过书名、图书编号、作者名、出版社对图书进行排序，而价格与数量的排序，对于图书馆而言相对无用。

**二、用户界面设计**

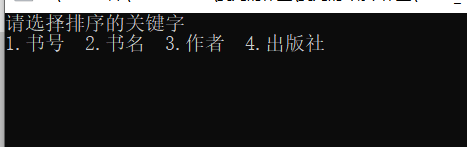
1、主界面

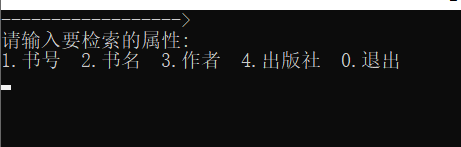


2、查看信息界面



3、关键字选择界面





**三、数据结构设计**

1、单链表，用于存储图书记录

typedef struct Book{

char book\_id[MAXSIZE];//书号

char book[MAXSIZE];//书名

char author[MAXSIZE];//作者

char publisher[MAXSIZE];//出版社

int money; //书价

char introduction[MAXSIZE];//简介

int num; //数量

struct Book \*next;

}LBook,\*LinkBook;

如下示例：

记录2

Head

记录1

……

记录n

NULL

1. 哈希表，用于建立书号快速索引

typedef LinkBook HashList[Max\_Num];//书号哈希表

如下示例：

getHash(记录2的书号)=key;

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 哈希表标号 | …… | …… | key | …… | …… | …… |
| 值 | …… | …… | 记录2的地址 | …… | …… | …… |

记录2

Head

记录1

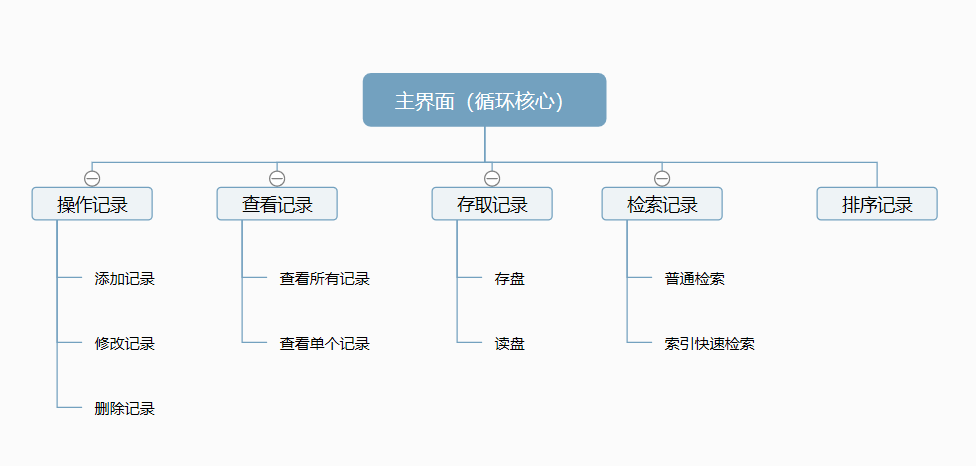
……

记录n

NULL

1. **模块结构设计**

1、程序功能结构图



tool()实现主界面循环调用子功能

2、操作记录模块

1）添加记录

//初始添加图书信息(前插法)，e是插入个数

Status AddBook(LinkBook &L,int e)

2）修改记录

//修改图书信息，p是待修改结点

Status AlterBook(LinkBook p)

3）删除记录

//删除图书信息操作,根据书号i删除

Status DelBook(LinkBook &L,char \*i)

3、查看记录

1）查看所有记录

//遍历图书信息并输出

void DisBook(LinkBook L)

1. 查看单个记录

//输出单个图书记录的全部信息

void printBook(LinkBook p)

4、存取记录

1）存盘

//存盘，首行为记录个数

Status Save()

2）读盘

//读盘，与存盘顺序对应

Status Save()

1. 检索记录
2. 普通检索

//普通检索，按关键字检索

Status Search()

1. 索引快速检索

//计算并返回书号book\_id的哈希值

//哈希构造方法：进制哈希，冲突方法：线性探测再散列

int getHash(char \*book\_id,HashList H)

//按图书序号查找图书信息，返回该图书的地址

LinkBook GetBook(LinkBook L,char \*n,HashList H){

1. 排序记录

//比较函数，ctr用于控制比较的关键字

int cmp(LinkBook a,LinkBook b,int ctr)

//图书排序，冒泡排序

void BubbleSort(int (\*cmp)(LinkBook,LinkBook,int))

1. **源代码**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <windows.h>

#define ERROR 0

#define OK 1

#define MAXSIZE 50//字符串容纳最大长度 50

#define Max\_Num 39877 //最大图书记录数

#define k 96            //k进制

typedef int Status;

typedef int ElemType;

int flag=0;

//图书信息

typedef struct Book{

char book\_id[MAXSIZE];//书号

char book[MAXSIZE];//书名

char author[MAXSIZE];//作者

char publisher[MAXSIZE];//出版社

int money;          //书价

char introduction[MAXSIZE];//简介

int num;                    //数量

struct Book \*next;

}LBook,\*LinkBook;

typedef LinkBook HashList[Max\_Num];//书号哈希表

LinkBook L;

HashList H;

int cmp(LinkBook a,LinkBook b,int ctr){

    switch(ctr){

        case 1:

            return strcmp(a->book\_id,b->book\_id);

        case 2:

            return strcmp(a->book,b->book);

        case 3:

            return strcmp(a->author,b->author);

        case 4:

            return strcmp(a->publisher,b->publisher);

    }

}

//初始化哈希表

void InitHash(HashList &H){

    for(int i=0;i<=Max\_Num;i++)

        H[i]=NULL;

    return;

}

//计算并返回书号book\_id的哈希值

int getHash(char \*book\_id,HashList H){

    int len =strlen(book\_id);

int val = 0;

    for(int i = 0; i < len ; i++){

        val =((val\*k)%Max\_Num + book\_id[i]-' ' + 1)%Max\_Num;

    }           //k进制数转换为十进制数 ,过程中进行取模

    int d=1; //偏移量

    while(H[val]!=NULL){//产生冲突

        if(strcmp(H[val]->book\_id,book\_id)==0){//相同字符串产生的冲突

            break;

        }

        else{   //不同字符串产生的冲突，进行线性探测再散列

            val=(val+d)%Max\_Num;

            d++;

        }

    }

return val;//返回哈希值

}

//输出单个图书记录的全部信息

void printBook(LinkBook p){

    if(p==NULL){

        printf("不存在该图书!\n");

        return;

    }

    printf("------------------>\n");

    printf("以下是该图书全部信息：\n");

    printf("图书序号:%s\n图书名称:%s\n图书作者:%s\n现存量:%d\n书价:%d\n",p->book\_id,p->book,p->author,p->num,p->money);

    printf("出版社:%s\n简介:%s\n",p->publisher,p->introduction);

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

return;

}

//初始化图书信息

Status InitBook(LinkBook &L){

L=(LinkBook)malloc(sizeof(LBook));

if(!L) return ERROR;

L->next=NULL;

L->num=0;

return OK;

}

//遍历图书信息

void DisBook(LinkBook L){

LBook \*p;

p=L->next;//指向第一个节点

if(p==NULL){

    printf("没有图书！\n");

    system("pause");

    return;

    }

printf("开始展开所有图书的部分信息\n");

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

while(p)//p!=NULL

{

printf("图书序号:%s\n图书名称:%s\n图书作者:%s\n现存量:%d\n书价:%d\n",p->book\_id,p->book,p->author,p->num,p->money);

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

p=p->next;

}

system("pause");

}

//删除图书信息操作,根据书号删除

Status DelBook(LinkBook &L,char \*i){

LBook \*p=L;

while(p->next)

{

if(strcmp(p->next->book\_id,i)==0){

Book \*s=p->next;

p->next=s->next;//p->next=p->next->next

free(s);

L->num--;//修正记录数

return OK;

}

p=p->next;

}

return ERROR;

}

//初始添加图书信息(前插法)--单插

Status AddBook(LinkBook &L,int e){

LBook \*p;

int i,j;

fflush(stdin);

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

p=(LinkBook)malloc(sizeof(LBook));

printf("请选择图书编号和数量:\n");

scanf("%s%d",p->book\_id,&j);

p->num=j;

printf("请插入图书的名字和作者名以及出版社名:\n");

scanf("%s%s%s",p->book,p->author,p->publisher);

printf("请插入图书的价格:\n");

scanf("%d",&j);

p->money=j;

printf("请插入图书的简介:\n");

scanf("%s",p->introduction);

p->next=L->next;

L->next=p;

L->num++;       //记录数

int val=getHash(p->book\_id,H);

    H[val]=p;

}

    return OK;

}

//按图书序号查找图书信息，返回该图书的地址

LinkBook GetBook(LinkBook L,char \*n,HashList H){

int val=getHash(n,H);

    if(H[val]==NULL || strcmp(H[val]->book\_id,n)==0)

        return H[val];

    else return NULL;

}

//普通检索

Status Search(){

fflush(stdin);

while(true){

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

        printf("请输入要检索的属性:\n");

        printf("1.书号 2.书名 3.作者 4.出版社 0.退出\n");

        int ctr;

        scanf("%d",&ctr);

        char s[MAXSIZE];

        LinkBook p=L->next;

        switch(ctr){

            case 1:

                scanf("%s",s);

                while(p){

                    if(strcmp(p->book\_id,s)==0)

                        printBook(p);

                    p=p->next;

                }

            break;

        case 2:

            scanf("%s",s);

                while(p){

                    if(strcmp(p->book,s)==0)

                        printBook(p);

                    p=p->next;

                }

            break;

        case 3:

            scanf("%s",s);

                while(p){

                    if(strcmp(p->author,s)==0)

                        printBook(p);

                    p=p->next;

                }

            break;

        case 4:

            scanf("%s",s);

                while(p){

                    if(strcmp(p->publisher,s)==0)

                        printBook(p);

                    p=p->next;

                }

            break;

        case 0:

            return OK;

        default:

            printf("无效指令，请重新输入!\n");

            break;

        }

        system("pause");

        system("cls");

    }

}

//图书排序

//冒泡排序的主要思想两两相性比较，每比较一次会把一轮最大或最小的数放在最后

void BubbleSort(int (\*cmp)(LinkBook,LinkBook,int)){

    printf("请选择排序的关键字\n");

    printf("1.书号 2.书名 3.作者 4.出版社\n");

    int ctr;

    scanf("%d",&ctr);

    LinkBook cur,tail;

    cur=L;

    tail=NULL;

    if(cur==NULL||cur->next==NULL){

        return;

    }

    while(cur!=tail){

        while(cur->next!=tail){

            if(cmp(cur, cur->next,ctr)>0){

                //修正哈希

                int p1=getHash(cur->book\_id,H),p2=getHash(cur->next->book\_id,H);

                LBook \*htmp=H[p1];

                H[p1]=H[p2];

                H[p2]=htmp;

                //cur->next与cur换值

                LBook tmp;

                strcpy(tmp.author,cur->author);

                strcpy(tmp.book,cur->book);

                strcpy(tmp.book\_id,cur->book\_id);

                strcpy(tmp.introduction,cur->introduction);

                tmp.money=cur->money;

                tmp.num=cur->num;

                strcpy(tmp.publisher,cur->publisher);

                strcpy(cur->author,cur->next->author);

                strcpy(cur->book,cur->next->book);

                strcpy(cur->book\_id,cur->next->book\_id);

                strcpy(cur->introduction,cur->next->introduction);

                cur->money=cur->next->money;

                cur->num=cur->next->num;

                strcpy(cur->publisher,cur->next->publisher);

                strcpy(cur->next->author,tmp.author);

                strcpy(cur->next->book,tmp.book);

                strcpy(cur->next->book\_id,tmp.book\_id);

                strcpy(cur->next->introduction,tmp.introduction);

                cur->next->money=tmp.money;

                cur->next->num=tmp.num;

                strcpy(cur->next->publisher,tmp.publisher);

            }

            cur=cur->next;

        }

        tail=cur;

        cur=L;

    }

}

//修改图书信息

Status AlterBook(LinkBook p){

    fflush(stdin);

if(p==NULL){

    printf("不存在该图书!\n");

    system("pause");

    return ERROR;

    }

    while(true){

        system("pause");

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

        printf("图书查到,请选择修改属性修改:\n");

        printf("1.书号 2.书名 3.作者 4.出版社 5.书价 6.数量 7.简介 0.退出\n");

        int ctr;

        scanf("%d",&ctr);

        switch(ctr){

            case 1:

                printf("图书序号为:%s\n请修改:",p->book\_id);

            scanf("%s",p->book\_id);

            break;

        case 2:

            printf("书名为:%s\n请修改:",p->book);

            scanf("%s",p->book);

            break;

        case 3:

            printf("作者:%s\n请修改:",p->author);

            scanf("%s",p->author);

            break;

        case 4:

            printf("出版社:%s\n请修改:",p->publisher);

            scanf("%s",p->publisher);

            break;

        case 5:

            printf("书价:%d\n请修改:",p->money);

            scanf("%d",p->money);

            break;

        case 6:

            printf("数量:%d\n请修改:",p->num);

            scanf("%d",p->num);

            break;

        case 7:

                printf("简介:%s\n请修改:",p->introduction);

            scanf("%s",p->introduction);

            break;

        case 0:

            return OK;

        default:

            printf("无效指令，请重新输入!\n");

            break;

        }

    }

}

//存盘

Status Save(){

    fflush(stdout);

    printf("存盘中……\n");

    FILE \*f = fopen("data.txt", "w");

    LinkBook p=L->next;

    fprintf(f,"%d\n",L->num);

    while(p){

        fprintf(f,"%s\n%s\n%s\n%s\n%s\n",p->book\_id,p->book,p->author,p->publisher,p->introduction);

        fprintf(f,"%d\n%d\n",p->num,p->money);

        p=p->next;

    }

    fclose(f);

    system("cls");

    printf("存盘完成！\n");

    system("pause");

    return OK;

}

//读盘

Status Read(){

    fflush(stdin);

    printf("读盘中……\n");

    FILE \*f = fopen("data.txt", "r");

    int tmp;

    fscanf(f,"%d",&tmp);

    LBook \*p;

    while(tmp--){

        int t1,t2;

        p=(LinkBook)malloc(sizeof(LBook));

        fscanf(f,"%s%s%s%s%s",p->book\_id,p->book,p->author,p->publisher,p->introduction);

        fscanf(f,"%d%d",&t1,&t2);

        p->num=t1,p->money=t2;

        p->next=L->next;

L->next=p;

L->num++;       //记录数

int val=getHash(p->book\_id,H);

    H[val]=p;

    }

    fclose(f);

    system("cls");

    printf("读盘完成！\n");

    system("pause");

    return OK;

}

void function(){//功能菜单

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

printf("<-----------0-欢迎来到图书管理系统!-0------------>\n");

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

printf("<--------------请选择以下功能-------------------->\n");

printf("<--------------1.查看所有图书记录---------------->\n");

printf("<--------------2.添加图书记录-------------------->\n");

printf("<--------------3.删除图书记录-------------------->\n");

printf("<--------------4.修改图书记录-------------------->\n");

printf("<--------------5.存盘---------------------------->\n");

printf("<--------------6.读盘---------------------------->\n");

printf("<--------------7.书号快速图书检索---------------->\n");

printf("<--------------8.普通图书检索-------------------->\n");

printf("<--------------9.排序图书------------------------>\n");

printf("<--------------0.退出---------------------------->\n");

}

void tool(){//功能实现

InitBook(L);

InitHash(H);

int i,choose;

char n[MAXSIZE];

ElemType e;

while(true){

    fflush(stdin);

    system("cls");

        function();

printf("请输入你要选择的功能：\n请选择:");

scanf("%d",&choose);

system("cls");

switch(choose){

    case 1:         //查看所有图书

DisBook(L);

break;

case 2:         //插入图书记录

printf("请插入多少种不同的图书:\n");

scanf("%d",&e);

AddBook(L,e);

break;

 case 3:             //删除图书

printf("请输入要删除的图书序号:\n");

scanf("%s",n);

if(DelBook(L,n)){

printf("删除成功!\n");

}

else{

printf("该图书不存在，请检查是否有这个图书号!\n");

}

system("pause");

break;

case 4:             //修改图书

printf("请输入要修改的图书号:\n");

scanf("%s",n);

AlterBook(GetBook(L,n,H));

break;

case 5:             //存盘

    if(!flag){

        printf("警告！当前未读盘，存盘将覆盖原数据！\n");

        printf("是否继续（y/n）:");

        char tmp;

        fflush(stdin);

                    scanf("%c",&tmp);

                    system("pause");

                    if(tmp!='y' && tmp!='Y') break;

                }

    Save();

                break;

case 6:             //读盘

                if(Read())

                    flag=1;

                break;

case 7:             //书号检索

    printf("请输入要检索的图书序号:\n");

scanf("%s",n);

printBook(GetBook(L,n,H));

system("pause");

                break;

case 8://普通检索

Search();

break;

case 9://图书排序

    BubbleSort(cmp);

    printf("排序成功！\n");

                system("pause");

break;

case 0://退出程序

printf("拜拜！\n");

exit(0);

default:

printf("无效输入，请重新选择！\n");

system("pause");

break;

}

}

}

int main()

{

    tool();

}