

实验四：小小图书馆

在社区中有一个小小图书馆，居民可以来阅读或借阅图书，现在要帮助图书管理员管理读者和图书，编写程序完成图书和读者信息的自动管理。

一、目标与要求

要求程序是一个简单的图书馆管理系统，主要完成的工作有：

- 借书
- 还书
- 图书管理
- 读者维护

其中，图书管理和读者维护功能中可以进行日常的数据添加、删除、查找、显示等功能，使日常图书借阅工作能正常进行。读者信息记录在 reader.dat 文件中，图书信息在 book.dat 文件中。编程基于字符界面的控制台应用程序来实现。源代码如下：

1. book.h(图书类)

```
#ifndef _book_h_
#define _book_h_
#include<iostream>
using namespace std;
class Book //图书类
{ private:
    int note; //删除标记（1：已删，0：未删）
    int number; //图书编号
    char name[10]; //书名
    int onshelf; //上架标志
public:
    Book() {}
    char *getname(); //获得书名
    int getnote(); //获得删除标记
    int getnumber(); //获得图书编号
    void setname(char na[]); //设置书名
    void delbook(); //删除图书
    void addbook(int n, char *na); //添加图书
    int borrowbook(); //借书操作
    void retbook(); //还书操作
    void list(); //输出图书
};
```

2. book.CPP(图书类实现)

```
#include<iomainip.h>
```

```

#include "stdafx.h"
#include "book.h"
char *Book::getname()
{ return name;}
int Book::getnote()
{ return note;}
int Book::getnumber()
{ return number;}
void Book::setname(char na[])
{strcpy(name, na);}
void Book::addbook(int n, char *na)
{   note=0;
    number=n;
    strcpy(name, na);
    onshelf=1;}
void Book::delbook()
{   note=1;}
int Book::borrowbook()
{return onshelf;}
void Book::retbook()
{   onshelf=1;}
void Book::list()
{cout<<setw(5)<<number<<setw(10)<<name<<setw(10)<<onshelf<<endl}#endif

```

3. reader.h(读者类)

```

#ifndef _reader_h_
#define _reader_h_
#include<iostream>
using namespace std;
const int Maxbor=5;                                //最大借阅量
class Reader                                         //读者类
{
private:
    int note;                                        //删除标记（1：已删，0：未删）
    int number;                                    //读者编号
    char name[10];                                  //读者姓名
    int borbook[Maxbor];                            //所借图书
public:
    Reader() {}
    char *getname();                                //获得姓名
    int getnote();                                  //获得删除标记
    int getnumber();                                //获得读者编号
    void setname(char na[]);                         //设置姓名
    void delbook();                                  //设置删除标记

```

```

        void addreader(int n, char *na);    //添加读者
        void borrowbook(int bookid);      //借书操作
        int retbook(int bookid);          //还书操作
        void list();                       //输出读者信息
};
#endif

```

4. reader.cpp(读者类实现)

```

#include<iomainip.h>
#include "stdafx.h"
#include "reader.h"
const int Maxbor=5;
#include <iostream.h> //最大借阅量
char *Reader::getname()
{ return name;}
int Reader::getnote()
{ return note;}
int Reader::getnumber()
{ return number;}
void Reader::setname(char na[])
{ strcpy(name, na);
}
void Reader::addreader(int n, char *na)
{
    note=0;
    number=n;
    strcpy(name, na);
    for(int i=0;i<Maxbar;i++)
        borbook[i]=0;}
void Reader::delbook()
{ note=1;}
void Reader::borrowbook(int bookid)
{ for(int i=0;i<Maxbar;i++)
    {
        if(borbook[i]==0) { borbook[i]=bookid; return;}
    }
}
int Reader::retbook(int bookid)
{
    for(int i=0;i<Maxbar;i++)
    {
        if(borbook[i]==bookid) { borbook[i]=0; return 1;}
    }
    return 0;
}

```

```

void Reader::list()
{
    cout<<setw(5)<<number<<setw(10)<<name<<"借书编号: [";
    for(int i=0;i<Maxbor;i++)
        if(borbook[i]!=0) cout<<borbook[i]<<" ";
    cout<<"]"<<endl;}

```

5. bdatabase.h(图书信息库类)

```

#ifndef _bdatadase_h_
#define _bdatabase_h_
#include "book.h"
#include<iostream>
using namespace std;
const int Maxb=100;                //最大图书数量
class BDatabase                    //图书库类
{
private:
    int top;                        //图书记录指针
    Book book[Maxb];               //图书记录
public:
    BDatabase();                  //构造函数初始化: 读文件
    void clear();                 //全部删除
    int addbook(int n, char *na);  //添加图书
    Book *query(int bookid);      //查找图书
    void list();                  //输出图书信息
    void bookdata();              //图书库维护
    ~BDatabase();                 //析构函数: 写文件
};
#endif

```

6. bdatabase.cpp(图书信息库类实现)

```

#include "stdafx.h"
#include<fstream>
#include<iostream>
#include"bdatabase.h"
using namespace std;
BDatabase::BDatabase()
{
    Book s;
    top=-1;
    fstream file;
    file.open("book.dat", ios::in|ios::binary);
    while(1)
    {
        file.read((char*)&s, sizeof(s));
        if(!file) break;
    }
}

```

```

        top++;
        book[top]=s;
    }
    file.close();}
void BDatabase::clear()
{ top=-1;}
Book *BDatabase::query(int bookid)
{
    for(int i=0;i<=top;i++)
        if(book[i].getnumber()==bookid&&book[i].getnote()==0)
            return &book[i];
    return NULL;}
int BDatabase::addbook(int n, char *na)
{
    Book *p=query(n);
    if(p==NULL)
    {
        top++;
        book[top].addbook(n, na);
        return 1;
    }
    return 0;}
void BDatabase::list()
{ for(int i=0;i<=top;i++)
    book[i].list();
}
BDatabase::~BDatabase()
{
    fstream file;
    file.open("book.dat", ios::out|ios::binary);
    for(int i=0;i<=top;i++)
        if(book[i].getnote()==0)
            file.write((char*)&book[i], sizeof(book[i]));
    file.close();
}
void BDatabase::bookdata()
{
    int choice=1;
    char bookname[20];
    int bookid;
    Book *b;
    while(choice!=0)
    {
        cout<<"图书维护: "<<endl;

```

```

        cout<<"-----"<<endl;
cout<<"1:新增 2: 更改 3: 删除 4: 查找 5: 显示 6: 全删 0: 退出"<<endl;
cin>>choice;
cout<<"-----"<<endl;
    switch(choice)
    {
        case 1:cout<<"输入图书编号: ";
            cin>>bookid;
            cout<<"输入图书名: ";
            cin>>bookname;
            addbook(bookid,bookname); break;
        case 2:cout<<"输入图书编号: ";
            cin>>bookid;
            b=query(bookid);
            if(b==NULL)
            { cout<<"该图书不存在!"<<endl;
              break; }
            cout<<"输入新的图书名: ";
            cin>>bookname;
            b->setname(bookname);break;
        case 3:cout<<"输入图书编号: ";
            cin>>bookid;
            b=query(bookid);
            if(b==NULL)
            { cout<<"该图书不存在!"<<endl;
              break; }
        case 4:cout<<"输入图书编号: ";
            cin>>bookid;
            b=query(bookid);
            if(b==NULL)
            { cout<<"该图书不存在!"<<endl;
              break;}
            b->list(); break;
        case 5:list();break;
        case 6:break;
    } //end switch
} //end while
}

```

7. rdatabase.h(读者信息库类)

```

#ifndef _rdatadase_h_
#define _rdatabase_h_
#include "reader.h"
#include<iostream>
using namespace std;

```

```

const int Maxr=100;                                //最大读者数
class RDatabase                                     //读者库类
{
    private:
        int top;                                    //读者记录指针
        Reader read[Maxr];                          //读者记录
    public:
        RDatabase();                                //构造函数初始化：读文件
        void clear();                               //删除所有读者信息
int addreader(int n, char *na);                     //添加读者记录时，先确定他//是
                                                    //否已经存在

        Reader *query(int readerid);                //按编号查找
        void list();                                 //输出所有读者信息
        void readerdata();                           //读者库维护
        ~RDatabase();                               //析构函数：写文件
};
#endif

```

8. rdatabase.cpp(读者信息库类实现)

```

#include "stdafx.h"
#include<fstream>
#include<iostream>
#include"rdatabase.h"
using namespace std;
RDatabase::RDatabase()
{
    Reader s;
    top=-1;
    ifstream file;
    file.open("reader.dat", ios::in|ios::binary);

    while(1)
    { file.read((char*)&s, sizeof(s));
      if(!file) break;
      top++;
      read[top]=s;}
    file.close();
}

void RDatabase::clear()
{ top=-1;}

Reader *RDatabase::query(int readerid)
{
    for(int i=0; i<=top; i++)
        if(read[i].getnumber()==readerid&&read[i].getnote()==0)    return
&read[i];
}

```

```

        return NULL;}
int RDatabase::addreader(int n,char *na)
{
    Reader *p=query(n);
    if(p==NULL)
    {
        top++;
        read[top].addreader(n,na);
        return 1;
    }
    return 0;}
void RDatabase::list()
{ for(int i=0;i<=top;i++)
    read[i].list();
}
RDatabase::~RDatabase()
{ fstream file;
  file.open("reader.dat",ios::out|ios::binary);
  for(int i=0;i<=top;i++)
      if(read[i].getnote()==0)
          file.write((char*)&read[i],sizeof(read[i]));
  file.close();}
void RDatabase::readerdata()
{ int choice=1;
  char readername[20];
  int readerid;
  Reader *r;
  while(choice!=0)
  {
      cout<<"读者维护: "<<endl;
      cout<<"-----"<<endl;
      cout<<"1:新增 2: 更改 3: 删除 4: 查找 5: 显示 6: 全删 0: 退出"<<endl;
      cin>>choice;
      cout<<"-----"<<endl;
      switch(choice)
      {
          case 1:cout<<"输入读者编号: ";
                  cin>>readerid;
                  cout<<"输入读者姓名: ";
                  cin>>readername;
                  addreader(readerid,readername);break;
          case 2:cout<<"输入读者编号: ";
                  cin>>readerid;
                  r=query(readerid);

```



```

        if(r==NULL)
        { cout<<"该读者不存在！"<<endl;
          break; }
        cout<<"输入新的姓名：";
        cin>>readername;
        r->setname(readername); break;
    case 3:cout<<"输入读者编号：";
            cin>>readerid;
            r=query(readerid);
            if(r==NULL)
            { cout<<"该读者不存在！"<<endl;
              break;}
    case 4:cout<<"输入读者编号：";
            cin>>readerid;
            r=query(readerid);
            if(r==NULL)
            { cout<<"该读者不存在！"<<endl;
              break;}
            r->list();break;
    case 5:list();break;
    case 6:break;
    } //end switch
} //end while
}

```

9. library.cpp(主函数实现)

```

#include "stdafx.h"
#include"reader.h"
#include"rdatabase.h"
#include"book.h"
#include"bdatabase.h"
#include<iostream>
using namespace std;
int main(int argc, char* argv[])
{ int choice=1,bookid,readerid;
  RDatabase ReaderDB;
  BDatabase BookDB;
  Reader *r;
  Book *b;
  cout<<"欢迎进入“小小图书馆”管理软件"<<endl;
  cout<<"-----"<<endl;
  while(choice!=0)
  { cout<<"-----"<<endl;
    cout<<"1: 借书 2 还书 3: 图书维护 4: 读者维护 0: 离开";
    cin>>choice;

```

```

cout<<"-----"<<endl;
    switch(choice)
    {case 1:
        cout<<"输入借书读者编号";
        cin>>readerid;
        cout<<"输入图书编号";
        cin>>bookid;
        r=ReaderDB.query(readerid);
        if(r==NULL)
        {cout<<"该读者不存在，不能借书！"<<endl;
        break;}
        b=BookDB.query(bookid);
        if(b=NULL)
        {cout<<"该图书不存在，不能借书！"<<endl;
        break;}
        if(b->borrowbook()==0)
        {cout<<"该图书已借出，不能借书！"<<endl;
        break; }
        r->borrowbook(b->getnumber()); break;
    case 2:
        cout<<"输入还书读者编号";
        cin>>readerid;
        cout<<"输入图书编号";
        cin>>bookid;
        r=ReaderDB.query(readerid);
        if(r==NULL)
        {cout<<"该读者不存在，不能还书！"<<endl;
        break;}
        b=BookDB.query(bookid);
        if(b=NULL)
        {cout<<"该图书不存在，不能还书！"<<endl;
        break;}
        b->retbook();
        r->retbook(b->getnumber()); break;
    case 3:
        BookDB.bookdata();break;
    case 4:
        ReaderDB.readerdata();break;
    }//end switch
} //end while
return 0;
}

```

