

课程设计成绩	
平时成绩(40%)	
设计报告+演示成绩(60%)	
总评成绩(100%)	

面向对象 课程设计报告

题 目： 图书信息管理系统

学院（系）：

学 号：

学生姓名：

时间： 2022-12-29 至 2023-1-13

目 录

一、设计目的	1
二、需求分析	3
三、总体设计	5
四、详细设计	7
五、具体实现	10
六、设计总结	44

一、设计目的

1.1 系统介绍

图书信息管理系统是针对借阅用户-管理员-图书馆的三方面连通性系统，主要是服务于用户日常对各类书籍的查询借阅以及管理员对图书馆信息的编辑处理。

图书馆规模不断扩大，图书数量呈现大幅度的增长，对图书的各种信息也颁发了相应的规定文件，各种信息量也成倍增加，但一直以来人们使用传统人工的方式管理文件档案，这种管理方式存在着许多缺点，如：效率低、保密性差，另外时间一长，将产生大量的文件和数据，这对于查找、更新和维护都带来了不少的困难。因此，必须制定一套合理、有效，规范和实用的图书管理系统，对图书资料进行集中统一的管理。

我们对于用户群体仅提供查询功能和借阅功能。针对管理人员的不同分配不同的个体权限，根据权限分配图书入库系统、入库书籍的查找功能、入库书籍信息编辑功能、书籍借阅功能、书籍借阅日志导出。

1.2 系统目标

- 1) 将所有书籍信息入档，较于人工管理，检索起来更为迅速、便捷。
- 2) 通过一个程序省去大面积存放档案的空间，节约成本。
- 3) 避免人工查询、更新、维护需要的大量时间成本。
- 4) 避免传统人工管理导致的档案纸张泛黄、老化等问题，实现对于所保存档案的稳定浏览。
- 5) 避免传统人工管理存在保密性不佳的问题，本系统将通过密码管理，对用户身份进行核验。
- 6) 通过账户等级区分，对不同等级的账户开放不同功能，从而解决调阅档案时的步骤繁琐、耗时耗力。

1.3 系统面向用户群体

本图书信息管理系统主要面向图书馆及相关的商户企业（例如：XX 书店）等需要图书管理功能以及需要记录借阅信息对象用户，最终用户细化至小型图书管理单位。

本产品可以给予目标客户及组织对象更加方便快捷的图书信息借阅等功能，帮助用户了解存书情况，对于统计方与被统计方更甚于责任方等多方面和个性化的设计体验，可以从时间、空间、系统效率、便携与云共享上让使用用户、客户及子对象受益。

二、需求分析

2.1 用户需求分析

功能类别	子功能	编号
图书管理员功能	个人信息修改与登陆系统	1001
	增加新的书籍信息	2001
	对同一书籍的不同本进行编号	2002
	对书籍、书本信息的查找	1002
	对书籍、书本信息的编辑	2004
	对书籍、书本信息的删除	2005
	对书本的借阅/归还	1003
	通过 ISBN 号对书籍信息自动填充	1004
	通过书本序号对书本信息自动填充	1005
	查询书本借阅历史	2006
	注销账户	1006
	退出系统	1007
读者功能	个人信息修改与登陆系统	1001
	对书籍、书本信息的查找	1002
	对书本的借阅/归还	1003
	通过 ISBN 号对书籍信息自动填充	1004
	通过书本序号对书本信息自动填充	1005
	注销账号	1006
	退出系统	1007

标注：1-全体功能、2-管理功能。书籍指同 ISBN 的所有书、书本指具体的某一本书。

2.2 功能需求分析

对于管理员权限：

在登录时自动获取身份权限，进入对应的二级对话框。关于书籍的信息录入后，生成相应的书本编号，对每一本书的编号要具有唯一性，同时在书的入库过程中针对各项信息完成登记，自动同步出书籍的库存总量。根据书籍的 ISBN 号、书本的编号，对所有书本进行检索，对数据库进行调取，查询书本信息。根据 ISBN 号，编辑不同书本的固定信息（例如：书名、出版社、版次），在特殊情况下也

可以直接更改借阅信息。可以根据 ISBN 号删除同名的所有书，也可以根据书号删除该本书的所有信息。

对于读者权限：

通过 ISBN 号检索到该书的可借阅数量，同时完成借阅以及归还功能。

2.3 非功能需求

较为简洁、清楚的 UI 菜单。客户能直观地接收到数据的反馈以及自己想要的功能。对软硬件没有较高的要求，但系统相应操作响应时间不超过 2 秒，最好可以通过键盘和鼠标等输入设备进行控制，也可与其他系统进行连接，实现嵌入式设计。

本软件须被测试完全，以避免任何数据储存及运算可能发生的错误；对重要数据会自动进行备份；软件必须采用检查点、恢复、重启动机制；系统数据库维护模块无远程处理功能。

对于不同权限用户，设置不同权限，对重要数据加密存储，条件允许的情况下可以让系统定时自动保存，省去人工操作保存的繁杂程序。

对登录人员进行统计。统计每一个登录系统人员的登录次数，和最后一次登录 IP 地址，保存完成的用户操作日志。

三、总体设计

3.1 功能模块

	图书信息管理系统	读者借阅系统	报错系统
用户登录	√	√	√
书籍录入	√		√
书本编号	√		√
书籍查找	√		√
书籍编辑	√		√
书籍删除	√		√
书籍借阅	√	√	√
借阅历史	√		√

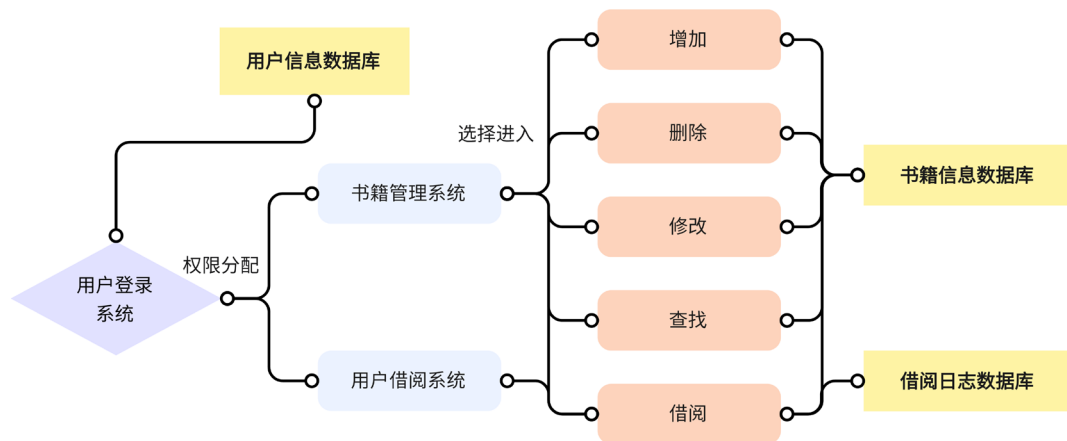
3.2 功能模块设计

- 1) **用户登录**: 输入用户名、密码, 进行登录, 将用户输入的信息与存储的链表信息进行匹配, 通过核验后将链表内的权限参数 (-1: 注销用户、0: 普通用户、1: 图书馆工作人员、2: 超级管理员) 取出, 进行分支判断。
- 2) **书籍录入**: 输入书名、ISBN、出版社、版次、录入数量、书籍分类、采购日期等信息, 完成存储功能。若是已录入过的书籍, 可通过 ISBN 自动调取相应的基础信息, 只需要更改录入数量和采购日期即可。根据每次录入的数量, 更新书籍的库存总量。
- 3) **书本编号**: 查询书籍数据库中某一 ISBN 对应的最大书本序号, 并在该序号后接续编号。
- 4) **书籍查找**: 输入 ISBN 后检索书籍信息, 输入 ISBN 和书本序号后检索书本信息并调取借阅状态。
- 5) **书籍编辑**: 输入 ISBN 和其他书籍信息可以编辑该 ISBN 对应的所有书本, 输入 ISBN、序号和其他书本信息后可以编辑该本书的信息。

- 6) **书籍删除：**输入 ISBN 后选择删除可以删除该 ISBN 对应的所有书本，输入 ISBN 和序号后可以删除该本书的信息，同步更新库存数量。
- 7) **书籍借阅：**输入 ISBN、序号确定借阅书本后，调取改书籍的借阅状态，展示可借阅数量，然后选择借阅/归还书本后，同步更新可借阅数量，并更新书籍的借阅状态。
- 8) **借阅历史：**每一次的借阅/归还行为，在借阅日志的链表中添加该书的新借阅状态，以便于查询某一本书的借阅历史，对书本情况拥有一个比较完整的了解。

四、详细设计

4.1 程序处理流程



4.2 模块详细设计

4.2.1 登录模块详细设计

预处理：

在 LibSystemDlg 对话框中创建用户名和密码的编辑框，并添加为可处理变量。

数据结构定义：

用户名、密码、权限参数

主要函数声明：

LoadUser() 加载 user.txt 进入链表。

CheckUser() 对用户名进行查找，然后进行密码的匹配。

OnInitDialog() 预处理函数。

OnBnClickedOk() 确认按键函数，调用 LoadUser()、CheckUser()，通过后调取权限参数（0：读者 1：管理员），进入对应的二级对话框。

全局变量声明：

CString: 用户名 m_csName, 密码 m_csPwd

4.2.2 增加模块详细设计

预处理:

在 CDlgAdd 对话框中创建书籍全部信息和入库信息的编辑框, 并添加为可处理变量。

数据结构定义:

书名、ISBN、出版社、库存总量、出版年月、分类、采购日期、同名序号、借阅状态

主要函数声明:

FindBNumByISBN(CString, CBook) 对已入库书籍, 查找其 ISBN 号对应的目前库存中最大地书本序号, 以便于新入库书本的后续编号。

AddBook() 增加新的书本信息, 结合循环功能实现批量录入。

UpdateBookInfo() 更新所有的书籍信息, 在本模块中用于更新所有书籍的库存总量。

Save() 将链表数据保存到本地。

SetDA() 将加载的链表数据首地址传入指针, 后续通过操作指针对链表数据进行操作。

OnInitDialog() 预处理函数, 完成书籍分类下拉框选项, spin 关联编辑框的数值规定, 以及链表数据载入。

OnBnClickedAbort() 根据 ISBN 信息填充其他书籍固定信息如书名、出版社、版次、书籍分类等等的按键函数。

OnBnClickedOk() 执行添加功能, 并将链表保存至本地的按键函数。

全局变量声明:

CString: 书名 m_csName, ISBN 号 m_csISBN, 出版社 m_csPublisher, 借阅状态 m_csBorrow。

int: 新增书本数量 m_nBNum, 出版年 m_nPYear, 出版月 m_nPMonth, 库存数量 m_nStock, 同名序号 m_nBNo。

CComboBox: 书籍分类下拉框 m_cbCatalog。

COleDataTime: 采购（入库）时间 m_dtBuy。

4.2.3 搜索与编辑模块详细设计

预处理:

在 CDlgFind 对话框中创建书籍 ISBN 信息和同名序号的编辑框用于查找, 以及其他信息的编辑框一并添加为可处理变量。

主要函数声明:

FindBookByNo() 通过ISBN和同名序号将链表指针指向那一行的数据。

GetPrevBook() 将指针挪到前一行的数据。

GetNextBook() 将指针挪到后一行的数据。

GetFirstBook() 将指针指向第一行数据, 用于该模块的初始界面展示。

FillWithBook() 将指针指向的那一行数据传进对话框中进行展示。

UpdateBook() 更新某一本书的信息。

UpdateBookInfo() 更新该ISBN对应的所有书籍信息。

OnInitDialog() 预处理函数, 完成书籍分类下拉框选项, spin 关联编辑框的数值规定, 以及链表数据载入。

OnBnClickedFind() 确认搜索按键函数。

OnBnClickedPrev() 查找上一本书信息按键函数。

OnBnClickedNext() 查找下一本书信息按键函数。

OnBnClickedUpdateinfo() 更新某一ISBN对应的所有书籍信息按键函数。

OnBnClickedUpdatebook() 更新某一本书信息按键函数。

全局变量声明:

CString: 书名 m_csName, ISBN 号 m_csISBN, 出版社 m_csPublisher, 书籍分类文本 m_csCatalog, 借阅状态 m_csBorrow。

int: 出版年 m_nPYear, 出版月 m_nPMonth, 库存数量 m_nStock, 同名序号 m_nBNo。

CComboBox: 书籍分类下拉框 m_cbCatalog。

COleDataTime: 采购（入库）时间 m_dtBuy。

4.2.4 删除模块详细设计

预处理:

在 CDlgDelete 对话框中创建书籍 ISBN 信息和同名序号的编辑框用于确定删除对象，并添加为可处理变量。

主要函数声明:

DeleBookByISBN() 通过 ISBN 对数据库进行查找，erase 每一条符合条件的数据。

DeleBookByNo() 通过 ISBN 和同名序号对数据库进行查找，erase 该书本数据。

OnInitDialog() 预处理函数，完成书籍分类下拉框选项，spin 关联编辑框的数值规定，以及链表数据载入。

OnBnClickedAbortNo() 根据 ISBN 和同名序号信息填充其他书本信息的按键函数。

OnBnClickedAbortIsbn() 根据 ISBN 填充其他书籍固定信息的按键函数。

OnBnClickedDeletebook() 根据 ISBN 和同名序号删除整条书本信息的按键函数。

OnBnClickedDeleteall() 根据 ISBN 删除所有对应书籍信息的按键函数。

全局变量声明:

CString: ISBN 号 m_csISBN。

int: 同名序号 m_nBNo。

4.2.5 借阅模块详细设计

预处理:

在 CDlgLend 对话框中创建书籍 ISBN 信息和同名序号的编辑框用于确定书本，创建借阅对象编辑框，并都添加为可处理变量。

数据结构定义:

ISBN、同名序号、借阅状态、操作日期

主要函数声明:

AddHistory() 增加新的借阅日志信息。

SaveHistory() 保存借阅日志。

LoadHistory() 载入借阅日志。

ReturnBook() 根据 ISBN 和同名序号，仅对书籍数据库更改借阅状态。

ReturnNum() 根据 ISBN 对书籍数据库检索，返回可借阅的该书籍数量。

OnInitDialog() 预处理函数，完成书籍分类下拉框选项，spin 关联编辑框的数值规定，链表数据载入。

OnBnClickedAbort() 根据 ISBN 和同名序号，填充信息校对的按键函数。

OnBnClickedLend() 将书籍借阅状态更新为借阅人的按键函数。

OnBnClickedReturn() 将书籍借阅状态更新为“未借出”状态的按键函数。

全局变量声明：

CString: ISBN 号 m_csISBN，借阅对象 m_csBorrow。

int: 同名序号 m_nBNo，可借阅数量 m_nReturn。

COleDateTime: 借阅状态操作时间 m_dtLTime。

五、具体实现

5.1 LibSystem 登录界面

user.txt 数据：用户名 密码 权限参数

界面截图：

代码：

```
void CLibSystemDlg::OnBnClickedOk()
{
    // TODO: 在此添加控件通知处理程序代码
    UpdateData(TRUE);
    CheckUser(m_csName);
    if (m_csName.GetBuffer() == m_it->m_sUName
        && m_csPwd.GetBuffer() == m_it->m_sPwd)
    {
        if (m_it->m_nAuthority == 1)
        {
            CDialogEx::OnCancel();
            CdlgIndex dlg;
            dlg.DoModal();
            CDialogEx::OnOK();
        }
        else if (m_it->m_nAuthority == 0)
```

```

        {
            CDialogEx::OnCancel();
            CDialogUserIndex dlg;
            dlg.DoModal();
            CDialogEx::OnOK();
        }
    }
else
{
    AfxMessageBox(_T("Anything wrong!"));
}
}

void CLibSystemDlg::LoadUser(CString csFullPath)
{
    m_listUser.clear();
    ifstream in(csFullPath, ios::in);
    if (in.is_open())
    {
        int num;
        in >> num;
        for (int i = 0; i < num; i++)
        {
            CUser user = CUser();
            user.Load(in);
            m_listUser.push_back(user);
        }
        in.close();
    }
}

void CLibSystemDlg::CheckUser(CString csUName)
{
    if (pList->size() == 0)
    {
        bFound = false;
        AfxMessageBox(_T("请保证user.txt与工程文件在同一路径下"));
    }
    for (list<CUser>::iterator it = pList->begin(); it != pList->end(); it++)
    {
        if (it->m_sUName == csUName.GetBuffer())
        {
            m_it = it;
            bFound = true;
        }
    }
}

```

```

        break;
    }
}
if (bFound == false)
{
    m_it = plist->begin();
}
}

```

5.2 管理员系统

界面截图：



代码：

```

void CDlgIndex::OnBnClickedButtonAdd()
{
    // TODO: 在此添加控件通知处理程序代码
    //CDlgAdd dlg;
    //dlg.SetDA(m_da);
    //dlg.DoModal();
    CDlgAdd* dlg = new CDlgAdd(this);
    CDataAdapter* cd = m_da;
    dlg->SetDA(cd);
    dlg->DoModal();
    delete dlg;
}

```



```

void CDlgIndex::OnBnClickedButtonFind()
{
    // TODO: 在此添加控件通知处理程序代码
    if (m_da->GetNumOfBook() == 0)
    {
        AfxMessageBox(_T("请先录入数据!"));
        return;
    }

    CDlgFind dlg;
    dlg.SetDA(m_da);
    dlg.DoModal();
}

```

```

void CDlgIndex::OnBnClickedButtonDelete()
{
    // TODO: 在此添加控件通知处理程序代码
    if (m_da->GetNumOfBook() == 0)
    {
        AfxMessageBox(_T("请先录入数据!"));
        return;
    }

    CDlgDelete dlg;
    dlg.SetDA(m_da);
    dlg.DoModal();
}

```

```

void CDlgIndex::OnBnClickedButtonInput()
{
    // TODO: 在此添加控件通知处理程序代码
    if (m_da->GetNumOfBook() == 0)
    {
        AfxMessageBox(_T("请先录入数据!"));
        return;
    }

    CDlgLend dlg;
    dlg.SetDA(m_da);
    dlg.DoModal();
}

```

```

BOOL CDlgIndex::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // TODO: 在此添加额外的初始化
    m_da = new CDataAdapter();
    m_da->Load();
    return TRUE; // return TRUE unless you set the focus to a control
    // 异常: OCX 属性页应返回 FALSE
}

void CDlgIndex::OnBnClickedCancel()
{
    // TODO: 在此添加控件通知处理程序代码
    CDialogEx::OnCancel();
    CLibSystemDlg dlg;
    dlg.DoModal();
}

```

5.2.1 CDlgAdd 增加界面

界面截图:

The screenshot shows a Windows-style dialog box titled "录入图书信息" (Enter Book Information). It contains the following fields and controls:

- 书名 (Book Title):** Text input field containing "离散数学" (Discrete Mathematics).
- ISBN:** Text input field containing "9787805130699". To its right is a button labeled "数据填充" (Data Fill).
- 出版社 (Publisher):** Text input field containing "上海科学技术文献出版社" (Shanghai Scientific, Technical, and Literary Publishing House).
- 采购数量 (Purchase Quantity):** Spin box set to "4".
- 出版年月 (Publication Date):** Two spin boxes for year and month, set to "2023" and "1" respectively.
- 索引类别 (Index Category):** Dropdown menu set to "自然应用科学" (Natural and Applied Sciences).
- 采购日期 (Purchase Date):** Date picker set to "2023/ 1/18".
- Buttons:** "确定" (OK) and "取消" (Cancel) buttons at the bottom.

数据: 书名、ISBN、出版社、库存量、版次、分类、入库日期、借阅状态

8												
离散数学	9787805130699	上海科学技术文献出版社	6	2023	1	自然应用科学	2023	1	16	1	未借出	
离散数学	9787805130699	上海科学技术文献出版社	6	2023	1	自然应用科学	2023	1	16	2	未借出	
abc 2	abc 2	2023	1	哲学	2023	1	14	2	abc			
abc 2	abc 2	2023	1	哲学	2023	1	14	5	未借出			
离散数学	9787805130699	上海科学技术文献出版社	6	2023	1	自然应用科学	2023	1	18	3	未借出	
离散数学	9787805130699	上海科学技术文献出版社	6	2023	1	自然应用科学	2023	1	18	4	未借出	
离散数学	9787805130699	上海科学技术文献出版社	6	2023	1	自然应用科学	2023	1	18	5	未借出	
离散数学	9787805130699	上海科学技术文献出版社	6	2023	1	自然应用科学	2023	1	18	6	未借出	

代码:

```

BOOL CDlgAdd::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // TODO: 在此添加额外的初始化
    m_nBNum = 1;
    m_nPYear = 2023;
    m_nPMonth = 1;
    m_csBorrow = _T("未借出");
    for (int i = 0; i < CataNum; i++)
        m_cbCatalog.AddString(g_BookCata[i]);
    m_cbCatalog.SetCurSel(0);
    UpdateData(FALSE);
    m_spinBNum.SetBuddy(GetDlgItem(IDC_EDIT_NUM));
    m_spinBNum.SetRange(1, 1000);
    m_spinYear.SetBuddy(GetDlgItem(IDC_EDIT_PYEAR));
    m_spinYear.SetRange(1950, 2050);
    m_spinMonth.SetBuddy(GetDlgItem(IDC_EDIT_PMONTH));
    m_spinMonth.SetRange(1, 12);
    GetDlgItem(IDC_EDIT_NAME)->SetFocus();
    return TRUE; // return TRUE unless you set the focus to a control
    // 异常: OCX 属性页应返回 FALSE
}

void CDlgAdd::SetDA(CDataAdapter* pDA)
{
    m_pDA = pDA;
}

void CDlgAdd::OnBnClickedOk()
{
    // TODO: 在此添加控件通知处理程序代码
    UpdateData(TRUE);
    CString csCatalog;

```

```

CBook book = CBook();

int nNextNo = 0;
m_pDA->FindBNumByISBN(m_csISBN.GetBuffer(), book);
if (book.m_nBNo > 0)
{
    nNextNo = m_pDA->m_it->m_nBNo;
    m_nStock = m_pDA->m_it->m_nStock + m_nBNum;
}
else
{
    m_nStock = m_nBNum;
}

csCatalog = g_BookCata[m_cbCatalog.GetCurSel()];
for (int i = nNextNo; i < m_nBNum + nNextNo; i++)
{
    m_nBNo = i + 1;
    m_pDA->AddBook(m_csName.GetBuffer(), m_csISBN.GetBuffer(),
m_csPublisher.GetBuffer(),
        m_nStock, m_nPYear, m_nPMonth, csCatalog.GetBuffer(),
        m_dtBuy.GetYear(), m_dtBuy.GetMonth(), m_dtBuy.GetDay(), m_nBNo,
m_csBorrow.GetBuffer());
}

m_pDA->UpdateBookInfo(m_csName.GetBuffer(), m_csISBN.GetBuffer(),
m_csPublisher.GetBuffer(),
        m_nStock, m_nPYear, m_nPMonth,
g_BookCata[m_cbCatalog.GetCurSel()].GetBuffer());
m_pDA->Save();

AfxMessageBox(_T("添加成功!"));
CDialogEx::OnOK();
}

void CDlgAdd::OnBnClickedAbort()
{
    // TODO: 在此添加控件通知处理程序代码
    CBook book = CBook();
    int index = 0;
    UpdateData(TRUE);
    string sISBN = m_csISBN.GetBuffer();

    if (m_pDA->FindBNumByISBN(sISBN, book))

```

```

{
    for (int i = 0; i < CataNum; i++)
        if (m_pDA->m_it->m_sCatalog == g_BookCata[i].GetBuffer())
            index = i;
    m_cbCatalog.SetCurSel(index);
    m_csName = m_pDA->m_it->m_sName.c_str();
    m_csPublisher = m_pDA->m_it->m_sPublisher.c_str();
    m_nPYear = m_pDA->m_it->m_nPYear;
    m_nPMonth = m_pDA->m_it->m_nPMonth;
    UpdatedData(FALSE);
}
else
{
    AfxMessageBox(_T("数据库中无该ISBN号，请先录入数据！"));
}
}
}

```

5.2.2 CDlgFind 搜索与编辑界面

界面截图：

数据：书名、ISBN、出版社、库存量、版次、分类、入库日期、借阅状态

8													
离散数学		9787805130699	上海科学技术文献出版社		6	2023	1	自然应用科学	2023	1	16	1	未借出
离散数学		9787805130699	上海科学技术文献出版社		6	2023	1	自然应用科学	2023	1	16	2	未借出
tets1	2	abc	2	2023	1	哲学	2023	1	14	2	test2		
test1	2	abc	2	2023	1	哲学	2023	1	14	5	未借出		
离散数学		9787805130699	上海科学技术文献出版社		6	2023	1	自然应用科学	2023	1	18	3	未借出
离散数学		9787805130699	上海科学技术文献出版社		6	2023	1	自然应用科学	2023	1	18	4	未借出
离散数学		9787805130699	上海科学技术文献出版社		6	2023	1	自然应用科学	2023	1	18	5	未借出
离散数学		9787805130699	上海科学技术文献出版社		6	2023	1	自然应用科学	2023	1	18	6	未借出

代码:

```

void CDlgFind::SetDA(CDataAdapter* pDA)
{
    m_pDA = pDA;
}

BOOL CDlgFind::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // TODO: 在此添加额外的初始化
    if (m_pDA && m_pDA->GetNumOfBook() > 0)
    {
        int index = 0;
        for (int i = 0; i < CataNum; i++)
        {
            m_cbCatalog.AddString(g_BookCata[i]);
            if (m_csCatalog == g_BookCata[i])
                index = i;
        }

        CBook book = CBook();
        m_pDA->GetFirstBook(book);
        m_pDA->FillWithBook(book, m_csName, m_csISBN, m_csPublisher,
            m_nStock, m_nPYear, m_nPMonth, m_csCatalog, m_dtBuy, m_nBNo,
            m_csBorrow);

        for (int i = 0; i < CataNum; i++)
            if (m_pDA->m_it->m_sCatalog == g_BookCata[i].GetBuffer())
                index = i;
        m_cbCatalog.SetCurSel(index);
    }
    UpdateData(FALSE);
    m_spinBNum.SetBuddy(GetDlgItem(IDC_EDIT_BNO));
    m_spinBNum.SetRange(1, 100);
    m_spinBNum.SetBuddy(GetDlgItem(IDC_EDIT_NUM));

```

```

        m_spinBNum.SetRange(1, 1000);
        m_spinYear.SetBuddy(GetDlgItem(IDC_EDIT_PYEAR));
        m_spinYear.SetRange(1950, 2050);
        m_spinMonth.SetBuddy(GetDlgItem(IDC_EDIT_PMONTH));
        m_spinMonth.SetRange(1, 12);
        GetDlgItem(IDC_EDIT_ISBN)->SetFocus();
        return TRUE; // return TRUE unless you set the focus to a control
        // 异常: OCX 属性页应返回 FALSE
    }

void CDlgFind::OnBnClickedFind()
{
    // TODO: 在此添加控件通知处理程序代码
    CBook book = CBook();
    int index = 0;
    UpdateData(TRUE);
    int nBNo = m_nBNo;
    string sISBN = m_csISBN.GetBuffer();

    m_pDA->FindBookByNo(sISBN, nBNo, book);
    m_pDA->FillWithBook(book, m_csName, m_csISBN, m_csPublisher,
        m_nStock, m_nPYear, m_nPMonth, m_csCatalog, m_dtBuy, m_nBNo, m_csBorrow);
    UpdateData(FALSE);

    for (int i = 0; i < CataNum; i++)
        if (m_pDA->m_it->m_sCatalog == g_BookCata[i].GetBuffer())
            index = i;
    m_cbCatalog.SetCurSel(index);
}

void CDlgFind::OnBnClickedPrev()
{
    // TODO: 在此添加控件通知处理程序代码
    CBook book = CBook();
    int index = 0;
    m_pDA->GetPrevBook(book);
    m_pDA->FillWithBook(book, m_csName, m_csISBN, m_csPublisher,
        m_nStock, m_nPYear, m_nPMonth, m_csCatalog, m_dtBuy, m_nBNo, m_csBorrow);
    UpdateData(FALSE);

    for (int i = 0; i < CataNum; i++)
        if (m_pDA->m_it->m_sCatalog == g_BookCata[i].GetBuffer())

```

```

        index = i;
        m_cbCatalog.SetCurSel(index);
    }

void CDlgFind::OnBnClickedNext()
{
    // TODO: 在此添加控件通知处理程序代码
    CBook book = CBook();
    int index = 0;
    m_pDA->GetNextBook(book);
    m_pDA->FillWithBook(book, m_csName, m_csISBN, m_csPublisher,
        m_nStock, m_nPYear, m_nPMonth, m_csCatalog, m_dtBuy, m_nBNo, m_csBorrow);
    UpdateData(FALSE);

    for (int i = 0; i < CataNum; i++)
        if (m_pDA->m_it->m_sCatalog == g_BookCata[i].GetBuffer())
            index = i;
    m_cbCatalog.SetCurSel(index);
}

void CDlgFind::OnBnClickedUpdateinfo()
{
    // TODO: 在此添加控件通知处理程序代码
    CBook book = CBook();
    int index = 0;
    UpdateData(TRUE);
    for (int i = 0; i < CataNum; i++)
    {
        if (m_cbCatalog.GetCurSel() == i)
        {
            index = i;
            break;
        }
    }
    m_cbCatalog.SetCurSel(index);

    m_pDA->UpdateBookInfo(m_csName.GetBuffer(), m_csISBN.GetBuffer(),
m_csPublisher.GetBuffer(),
        m_nStock, m_nPYear, m_nPMonth,
g_BookCata[m_cbCatalog.GetCurSel()].GetBuffer());
    AfxMessageBox(_T("书籍信息更改成功!"));
    m_pDA->Save();
}

```



```
}
```

```
void CDlgFind::OnBnClickedUpdatebook()
{
    // TODO: 在此添加控件通知处理程序代码
    CBook book = CBook();
    int index = 0;
    UpdateData(TRUE);

    m_pDA->UpdateBook(m_csISBN.GetBuffer(), m_dtBuy.GetYear(), m_dtBuy.GetMonth(),
        m_dtBuy.GetDay(), m_nBNo, m_csBorrow.GetBuffer());

    m_pDA->Save();
}
```

5.1.3 CDlgDelete 删除界面

界面截图：

删除图书信息

ISBN	9787805130699
书名	离散数学
出版社	上海科学技术文献出版社
出版年月	2023 1
索引类别	哲学
<input type="button" value="ISBN数据核验"/> <input type="button" value="删除该系列"/>	
书籍序号	1
采购日期	2023/ 1/12
借阅对象	未借出
库存数量	9
<input type="button" value="序号核验"/> <input type="button" value="删除该本"/> <input type="button" value="取消"/>	

数据：书名、ISBN、出版社、库存量、版次、分类、入库日期、借阅状态

5											
离散数学	9787805130699	上海科学技术文献出版社	5	2023	1	自然应用科学	2023	1	16	2	未借出
离散数学	9787805130699	上海科学技术文献出版社	5	2023	1	自然应用科学	2023	1	18	3	未借出
离散数学	9787805130699	上海科学技术文献出版社	5	2023	1	自然应用科学	2023	1	18	4	未借出
离散数学	9787805130699	上海科学技术文献出版社	5	2023	1	自然应用科学	2023	1	18	5	未借出
离散数学	9787805130699	上海科学技术文献出版社	5	2023	1	自然应用科学	2023	1	18	6	未借出

代码:

```
void CDlgDelete::SetDA(CDataAdapter* pDA)
{
    m_pDA = pDA;
}

BOOL CDlgDelete::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // TODO: 在此添加额外的初始化
    m_nPYear = 2023;
    m_nPMonth = 1;
    m_csBorrow = _T("未借出");
    for (int i = 0; i < CataNum; i++)
        m_cbCatalog.AddString(g_BookCata[i]);
    m_cbCatalog.SetCurSel(0);
    UpdateData(FALSE);
    m_spinBNum.SetBuddy(GetDlgItem(IDC_EDIT_BNO));
    m_spinBNum.SetRange(1, 100);
    m_spinBNum.SetBuddy(GetDlgItem(IDC_EDIT_NUM));
    m_spinBNum.SetRange(1, 1000);
    m_spinYear.SetBuddy(GetDlgItem(IDC_EDIT_PYEAR));
    m_spinYear.SetRange(1950, 2050);
    m_spinMonth.SetBuddy(GetDlgItem(IDC_EDIT_PMONTH));
    m_spinMonth.SetRange(1, 12);
    return TRUE; // return TRUE unless you set the focus to a control
    // 异常: OCX 属性页应返回 FALSE
}

void CDlgDelete::OnBnClickedAbortIsbn()
{
    // TODO: 在此添加控件通知处理程序代码
    CBook book = CBook();
    int index = 0;
    UpdateData(TRUE);
    string sISBN = m_csISBN.GetBuffer();
```

```

if (m_pDA->FindBNumByISBN(sISBN, book))
{
    for (int i = 0; i < CataNum; i++)
        if (m_pDA->m_it->m_sCatalog == g_BookCata[i].GetBuffer())
            index = i;
    m_cbCatalog.SetCurSel(index);
    m_csName = m_pDA->m_it->m_sName.c_str();
    m_csPublisher = m_pDA->m_it->m_sPublisher.c_str();
    m_nPYear = m_pDA->m_it->m_nPYear;
    m_nPMonth = m_pDA->m_it->m_nPMonth;
    UpdateData(FALSE);
}
else
{
    AfxMessageBox(_T("数据库中无该ISBN号, 请先录入数据!"));
}
}

void CDlgDelete::OnBnClickedAbortNo()
{
    // TODO: 在此添加控件通知处理程序代码
    CBook book = CBook();
    int index = 0;
    UpdateData(TRUE);
    int nBNo = m_nBNo;
    string sISBN = m_csISBN.GetBuffer();

    m_pDA->FindBookByNo(sISBN, nBNo, book);
    m_pDA->FillWithBook(book, m_csName, m_csISBN, m_csPublisher,
        m_nStock, m_nPYear, m_nPMonth, m_csCatalog, m_dtBuy, m_nBNo, m_csBorrow);
    UpdateData(FALSE);

    for (int i = 0; i < CataNum; i++)
        if (m_pDA->m_it->m_sCatalog == g_BookCata[i].GetBuffer())
            index = i;
    m_cbCatalog.SetCurSel(index);
}

void CDlgDelete::OnBnClickedDeletebook()
{
    // TODO: 在此添加控件通知处理程序代码
    UpdateData(TRUE);
    int nBNo = m_nBNo;
    CBook book = CBook();

```

```

        string sISBN = m_csISBN.GetBuffer();
        m_pDA->DeleBookByNo(sISBN, nBNo);
        m_pDA->UpdateBookInfo(m_csName.GetBuffer(), m_csISBN.GetBuffer(),
m_csPublisher.GetBuffer(),
            m_nStock - 1, m_nPYear, m_nPMonth,
g_BookCata[m_cbCatalog.GetCurSel()].GetBuffer());
        m_pDA->GetFirstBook(book);
        m_pDA->FillWithBook(book, m_csName, m_csISBN, m_csPublisher,
            m_nStock, m_nPYear, m_nPMonth, m_csCatalog, m_dtBuy, m_nBNo, m_csBorrow);
        UpdatedData(FALSE);
        m_pDA->Save();
    }

void CDlgDelete::OnBnClickedDeleteall()
{
    // TODO: 在此添加控件通知处理程序代码
    UpdatedData(TRUE);
    int nBNo = m_nBNo;
    string sISBN = m_csISBN.GetBuffer();
    CBook book = CBook();
    m_pDA->DeleBookByISBN(sISBN);
    m_pDA->GetFirstBook(book);
    m_pDA->FillWithBook(book, m_csName, m_csISBN, m_csPublisher,
        m_nStock, m_nPYear, m_nPMonth, m_csCatalog, m_dtBuy, m_nBNo, m_csBorrow);
    UpdatedData(FALSE);
    m_pDA->Save();
}

```

5.2.4 CDlgLend 借阅界面

界面截图：

数据:

data.txt

history.txt ISBN、同名序号、借阅状态、操作日期

```
1
9787805130699 2 wzj 2023 1 16
```

代码:

```
void CDlgLend::SetDA(CDataAdapter* pDA)
{
    m_pDA = pDA;
}

BOOL CDlgLend::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // TODO: 在此添加额外的初始化
    m_nPYear = 2023;
    m_nPMonth = 1;
    m_csBorrow = _T("");
    for (int i = 0; i < CataNum; i++)
        m_cbCatalog.AddString(g_BookCata[i]);
    m_cbCatalog.SetCurSel(0);
    UpdateData(FALSE);
    m_spinBNum.SetBuddy(GetDlgItem(IDC_EDIT_BNO));
    m_spinBNum.SetRange(1, 100);
    LoadHistory(_T("./history.txt"));
    m_dtLTime = COleDateTime::GetCurrentTime();
    pList = &m_listHistory;
    return TRUE; // return TRUE unless you set the focus to a control
    // 异常: OCX 属性页应返回 FALSE
}

void CDlgLend::OnBnClickedAbort()
{
    // TODO: 在此添加控件通知处理程序代码
```

```

CBook book = CBook();
int index = 0;
UpdatedData(TRUE);
int nBNo = m_nBNo;
string sISBN = m_csISBN.GetBuffer();

m_pDA->FindBookByNo(sISBN, nBNo, book);
m_pDA->FillWithBook(book, m_csName, m_csISBN, m_csPublisher,
    m_nStock, m_nPYear, m_nPMonth, m_csCatalog, m_dtBuy, m_nBNo, m_csBorrow);
m_nReturn = m_pDA->ReturnNum(sISBN);
UpdatedData(FALSE);

for (int i = 0; i < CataNum; i++)
    if (m_pDA->m_it->m_sCatalog == g_BookCata[i].GetBuffer())
        index = i;
m_cbCatalog.SetCurSel(index);
}

void CDlgLend::OnBnClickedLend()
{
    // TODO: 在此添加控件通知处理程序代码
    CBook book = CBook();
    int index = 0;
    UpdatedData(TRUE);

    m_pDA->ReturnBook(m_csISBN.GetBuffer(), m_nBNo, m_csBorrow.GetBuffer());
    m_pDA->Save();
    m_nReturn = m_pDA->ReturnNum(m_csISBN.GetBuffer());
    AddHistory(m_csISBN.GetBuffer(), m_nBNo, m_csBorrow.GetBuffer(),
        m_dtLTime.GetYear(), m_dtLTime.GetMonth(), m_dtLTime.GetDay());
    SaveHistory(_T("//history.txt"));
    UpdatedData(FALSE);
}

void CDlgLend::OnBnClickedReturn()
{
    // TODO: 在此添加控件通知处理程序代码
    CBook book = CBook();
    int index = 0;
    m_csBorrow = _T("未借出");

    m_pDA->ReturnBook(m_csISBN.GetBuffer(), m_nBNo, m_csBorrow.GetBuffer());

```

```

        m_pDA->Save();
        m_nReturn = m_pDA->ReturnNum(m_csISBN.GetBuffer());
        AddHistory(m_csISBN.GetBuffer(), m_nBNo, _T("归还"),
            m_dtLTime.GetYear(), m_dtLTime.GetMonth(), m_dtLTime.GetDay());
        SaveHistory(_T("../history.txt"));
        UpdateData(FALSE);
    }

void CDlgLend::AddHistory(string sISBN, int nNo, string sBorrow,
    int nLYear, int nLMonth, int nLDay)
{
    CHistory hist = CHistory();
    hist.SetHistory(sISBN, nNo, sBorrow, nLYear, nLMonth, nLDay);
    m_listHistory.push_back(hist);
}

void CDlgLend::LoadHistory(CString csFullPath)
{
    m_listHistory.clear();
    ifstream in(csFullPath, ios::in);
    if (in.is_open())
    {
        int num;
        in >> num;
        for (int i = 0; i < num; i++)
        {
            CHistory hist = CHistory();
            hist.Load(in);
            m_listHistory.push_back(hist);
        }
        in.close();
    }
}

void CDlgLend::SaveHistory(CString csFullPath)
{
    ofstream out(csFullPath, ios::out);
    if (out.is_open())
    {
        out << m_listHistory.size() << endl;
        for (list<CHistory>::iterator it = m_listHistory.begin(); it !=
m_listHistory.end(); it++)
            it->Save(out);
        out.close();
    }
}

```

```
}  
}
```

5.3 用户借阅系统

界面截图：



5.3.1 CDlgLend 借阅界面

同 5.2.4。

5.4 类

5.4.1 CUser 类

代码：

```
CUser::CUser()  
{  
  
}  
  
CUser::CUser(string sUName, string sPwd, int nAuthority)  
{  
    m_sUName = sUName;  
    m_sPwd = sPwd;  
    m_nAuthority = nAuthority;
```



```

}

CUser::~CUser()
{

}

void CUser::SetUser(string sUName, string sPwd, int nAuthority)
{
    m_sUName = sUName;
    m_sPwd = sPwd;
    m_nAuthority = nAuthority;
}

void CUser::Save(ofstream& out)
{
    out << m_sUName << _T("\t");
    out << m_sPwd << _T("\t");
    out << m_nAuthority << _T("\n");
}

void CUser::Load(istream& in)
{
    in >> m_sUName;
    in >> m_sPwd;
    in >> m_nAuthority;
}

```

5.4.2 CBook 类

代码:

```

CBook::CBook()
{

}

CBook::CBook(string sName, string sISBN, string sPublisher, int nStock,
    int nPYear, int nPMonth, string sCatalog,
    int nbYear, int nbMonth, int nbDay, int nbNo, string sBorrow)
{
    m_sName = sName;
    m_sISBN = sISBN;
    m_sPublisher = sPublisher;
    m_nStock = nStock;

```

```

        m_nPYear = nPYear;
        m_nPMonth = nPMonth;
        m_sCatalog = sCatalog;
        m_nbYear = nbYear;
        m_nbMonth = nbMonth;
        m_nbDay = nbDay;
        m_nBNo = nBNo;
        m_csBorrow = sBorrow;
    }

CBook::~CBook()
{

}

void CBook::SetBook(string sName, string sISBN, string sPublisher, int nStock,
    int nPYear, int nPMonth, string sCatalog,
    int nbYear, int nbMonth, int nbDay, int nBNo, string sBorrow)
{
    m_sName = sName;
    m_sISBN = sISBN;
    m_sPublisher = sPublisher;
    m_nStock = nStock;
    m_nPYear = nPYear;
    m_nPMonth = nPMonth;
    m_sCatalog = sCatalog;
    m_nbYear = nbYear;
    m_nbMonth = nbMonth;
    m_nbDay = nbDay;
    m_nBNo = nBNo;
    m_csBorrow = sBorrow;
}

void CBook::Save(ofstream& out)
{
    out << m_sName << _T("\t");
    out << m_sISBN << _T("\t");
    out << m_sPublisher << _T("\t");
    out << m_nStock << _T("\t");
    out << m_nPYear << _T("\t");
    out << m_nPMonth << _T("\t");
    out << m_sCatalog << _T("\t");
    out << m_nbYear << _T("\t");
    out << m_nbMonth << _T("\t");

```

```

        out << m_nbDay << _T("\t");
        out << m_nBNo << _T("\t");
        out << m_csBorrow << _T("\n");
    }

    void CBook::Load(istream& in)
    {
        in >> m_sName;
        in >> m_sISBN;
        in >> m_sPublisher;
        in >> m_nStock;
        in >> m_nPYear;
        in >> m_nPMonth;
        in >> m_sCatalog;
        in >> m_nbYear >> m_nbMonth >> m_nbDay;
        in >> m_nBNo;
        in >> m_csBorrow;
    }

```

5.4.3 CHistory 类

代码:

```

CHistory::CHistory()
{

}

CHistory::CHistory(string sISBN, int nNo, string sBorrow,
    int nLYear, int nLMonth, int nLDay)
{
    m_sISBN = sISBN;
    m_nNo = nNo;
    m_sBorrow = sBorrow;
    m_nLYear = nLYear;
    m_nLMonth = nLMonth;
    m_nLDay = nLDay;
}

CHistory::~CHistory()
{

}

void CHistory::SetHistory(string sISBN, int nNo, string sBorrow,

```

```

    int nLYear, int nLMonth, int nLDay)
{
    m_sISBN = sISBN;
    m_nNo = nNo;
    m_sBorrow = sBorrow;
    m_nLYear = nLYear;
    m_nLMonth = nLMonth;
    m_nLDay = nLDay;
}

void CHistory::Save(ofstream& out)
{
    out << m_sISBN << _T("\t");
    out << m_nNo << _T("\t");
    out << m_sBorrow << _T("\t");
    out << m_nLYear << _T("\t");
    out << m_nLMonth << _T("\t");
    out << m_nLDay << _T("\n");
}

void CHistory::Load(ifstream& in)
{
    in >> m_sISBN;
    in >> m_nNo;
    in >> m_sBorrow;
    in >> m_nLYear >> m_nLMonth >> m_nLDay;
}

```

5.4.4 CDataAdapter 类

代码:

```

CDataAdapter::CDataAdapter()
{
    m_pData = CData::CreateInstance();
}

CDataAdapter::~CDataAdapter()
{
    if (m_pData)
        CData::DeleteInstance();
    m_pData = NULL;
}

void CDataAdapter::AddBook(string sName, string sISBN, string sPublisher, int nStock,

```

```

        int nPYear, int nPMonth, string sCatalog,
        int nbYear, int nbMonth, int nbDay, int nBNo, string sBorrow)
    {
        m_pData->AddBook(sName, sISBN, sPublisher, nStock,
            nPYear, nPMonth, sCatalog, nbYear, nbMonth, nbDay, nBNo, sBorrow);
    }

void CDataAdapter::DeleBookByNo(string sISBN, int nBNo)
{
    m_pData->DeleBookByNo(sISBN, nBNo);
    AfxMessageBox(_T("书本删除成功!"));
}

void CDataAdapter::DeleBookByISBN(string sISBN)
{
    m_pData->DeleBookByISBN(sISBN);
    AfxMessageBox(_T("书籍删除成功!"));
}

void CDataAdapter::UpdateBookInfo(string sName, string sISBN, string sPublisher,
    int nStock, int nPYear, int nPMonth, string sCatalog)
{
    m_pData->UpdateBookInfo(sName, sISBN, sPublisher, nStock, nPYear, nPMonth,
sCatalog);
}

void CDataAdapter::UpdateBook(string sISBN, int nbYear, int nbMonth, int nbDay,
    int nBNo, string sBorrow)
{
    m_pData->UpdateBook(sISBN, nbYear, nbMonth, nbDay, nBNo, sBorrow);
    AfxMessageBox(_T("书本信息更改成功!"));
}

void CDataAdapter::ReturnBook(string sISBN, int nBNo, string sBorrow)
{
    m_pData->ReturnBook(sISBN, nBNo, sBorrow);
    AfxMessageBox(_T("借阅状态更新成功!"));
}

void CDataAdapter::Save()
{
    m_pData->Save(_T(".\\data.txt"));
}

```

```

void CDataAdapter::Load()
{
    m_pData->Load(_T(".\\data.txt"));
}

bool CDataAdapter::GetFirstBook(CBook& book)
{
    list<CBook>* pList = GetListOfBook();
    if (pList->size() == 0)
        return false;

    m_it = pList->begin();
    book = *m_it;
    return true;
}

bool CDataAdapter::GetPrevBook(CBook& book)
{
    list<CBook>* pList = GetListOfBook();
    bool bRes = false;

    if (pList->size() != 0)
    {
        if (m_it != pList->begin())
            m_it--;

        book = *m_it;
        bRes = true;
    }
    return bRes;
}

bool CDataAdapter::GetNextBook(CBook& book)
{
    list<CBook>* pList = GetListOfBook();
    bool bRes = false;
    if (pList->size() != 0)
    {
        m_it++;
        if (m_it == pList->end())
            m_it--;
        book = *m_it;
        bRes = true;
    }
}

```

```

        return bRes;
    }

bool CDataAdapter::GetLastBook(CBook& book)
{
    list<CBook>* pList = GetListOfBook();
    if (pList->size() == 0)
        return false;

    m_it = pList->end();
    m_it--;
    book = *m_it;
    return true;
}

bool CDataAdapter::FindBookByNo(string sISBN, int nBNo, CBook& book)
{
    list<CBook>* pList = GetListOfBook();
    if (pList->size() == 0)
        return false;

    bool bFound = false;
    for (list<CBook>::iterator it = pList->begin(); it != pList->end(); it++)
    {
        if (it->m_sISBN == sISBN && it->m_nBNo == nBNo)
        {
            m_it = it;
            book = *it;
            bFound = true;
            break;
        }
    }
    if (bFound == false)
    {
        AfxMessageBox(_T("看看你的书籍信息对不对呢"));
        m_it = pList->begin();
        book = *m_it;
    }
    return bFound;
}

bool CDataAdapter::FindBNumByISBN(string sISBN, CBook& book)
{
    list<CBook>* pList = GetListOfBook();

```

```

        if (pList->size() == 0)
            return false;

        bool bFound = false;
        int nNextNo = 1;
        for (list<CBook>::iterator it = pList->begin(); it != pList->end(); it++)
        {
            if (it->m_sISBN == sISBN)
            {
                if (it->m_nBNo >= nNextNo)
                {
                    nNextNo = it->m_nBNo;
                    m_it = it;
                    book = *it;
                    bFound = true;
                }
            }
        }
        return bFound;
    }

    void CDataAdapter::FillWithBook(CBook& book, CString& sName, CString& sISBN, CString&
sPublisher,
        int& nStock, int& nPYear, int& nPMonth, CString& sCatalog,
        COleDateTime& dtBuy, int& nBNo, CString& sBorrow)
    {
        sName = book.m_sName.c_str();
        sISBN = book.m_sISBN.c_str();
        sPublisher = book.m_sPublisher.c_str();
        nStock = book.m_nStock;
        nPYear = book.m_nPYear;
        nPMonth = book.m_nPMonth;
        dtBuy = COleDateTime(book.m_nbYear, book.m_nbMonth, book.m_nbDay, 0, 0, 0);
        nBNo = book.m_nBNo;
        sBorrow = book.m_csBorrow.c_str();
    }

    int CDataAdapter::ReturnNum(string sISBN)
    {
        list<CBook>* pList = GetListOfBook();
        if (pList->size() == 0)
            return 0;
        int nReturn = 0;
        for (list<CBook>::iterator it = pList->begin(); it != pList->end(); it++)

```



```

    {
        if (it->m_sISBN == sISBN && it->m_csBorrow == _T("未借出"))
        {
            nReturn++;
        }
    }
    return nReturn;
}

```

5.4.5 CData 类

代码:

```

int CData::m_staID = 0;
CData* CData::m_sInstance = NULL;
CData::CData()
{
    if (m_staID == 0)
        m_staID++;
    m_listBook = list<CBook>();
}

CData::~~CData()
{
}

CData* CData::CreateInstance()
{
    if (m_staID == 0)
        m_sInstance = new CData();
    return m_sInstance;
}

void CData::DeleteInstance()
{
    if (m_sInstance != NULL)
    {
        delete m_sInstance;
        m_staID = 0;
    }
    m_sInstance = NULL;
}

void CData::Save(CString csFullPath)

```

```

{
    ofstream out(csFullPath, ios::out);
    if (out.is_open())
    {
        out << m_listBook.size() << endl;
        for (list<CBook>::iterator it = m_listBook.begin(); it != m_listBook.end();
it++)
            it->Save(out);
        out.close();
    }
}

void CData::Load(CString csFullPath)
{
    m_listBook.clear();
    ifstream in(csFullPath, ios::in);
    if (in.is_open())
    {
        int num;
        in >> num;
        for (int i = 0; i < num; i++)
        {
            CBook book = CBook();
            book.Load(in);
            m_listBook.push_back(book);
        }
        in.close();
    }
}

void CData::AddBook(CBook& book)
{
    m_listBook.push_back(book);
}

void CData::AddBook(string sName, string sISBN, string sPublisher, int nStock,
int nPYear, int nPMonth, string sCatalog,
int nbYear, int nbMonth, int nbDay, int nbNo, string sBorrow)
{
    CBook Book = CBook();
    Book.SetBook(sName, sISBN, sPublisher, nStock, nPYear, nPMonth, sCatalog,
        nbYear, nbMonth, nbDay, nbNo, sBorrow);
    m_listBook.push_back(Book);
}

```

```

void CData::DeleBookByNo(string sISBN, int nBNo)
{
    bool bFound = false;
    for (list<CBook>::iterator it = m_listBook.begin(); it != m_listBook.end(); it++)
    {
        if (it->m_sISBN == sISBN && it->m_nBNo == nBNo)
        {
            m_listBook.erase(it);
            bFound = true;
            break;
        }
    }
    if (bFound == false)
    {
        AfxMessageBox(_T("看看你的书籍信息对不对呢"));
    }
}

```

```

void CData::DeleBookByISBN(string sISBN)
{
    bool bFound = false;
    for (list<CBook>::iterator it = m_listBook.begin(); it != m_listBook.end(); it++)
    {
        if (it->m_sISBN == sISBN)
        {
            m_listBook.erase(it);
            it = m_listBook.begin();
        }
    }
    list<CBook>::iterator it = m_listBook.begin();
    if (it->m_sISBN == sISBN)
    {
        m_listBook.erase(it);
        bFound = true;
    }
    if (bFound == false)
    {
        AfxMessageBox(_T("看看你的书籍信息对不对呢"));
    }
}

```

```

void CData::UpdateBookInfo(string sName, string sISBN, string sPublisher,

```

```

    int nStock, int nPYear, int nPMonth, string sCatalog)
{
    bool bFound = false;
    for (list<CBook>::iterator it = m_listBook.begin(); it != m_listBook.end(); it++)
    {
        if (it->m_sISBN == sISBN)
        {
            it->m_sName = sName;
            it->m_sPublisher = sPublisher;
            it->m_nPYear = nPYear;
            it->m_nPMonth = nPMonth;
            it->m_sCatalog = sCatalog;
            it->m_nStock = nStock;
            bFound = true;
        }
    }
    if (bFound == false)
    {
        AfxMessageBox(_T("看看你的书籍信息对不对呢"));
    }
}

void CData::UpdateBook(string sISBN, int nbYear, int nbMonth, int nbDay,
    int nBNo, string sBorrow)
{
    bool bFound = false;
    for (list<CBook>::iterator it = m_listBook.begin(); it != m_listBook.end(); it++)
    {
        if (it->m_sISBN == sISBN && it->m_nBNo == nBNo)
        {
            it->m_nbYear = nbYear;
            it->m_nbMonth = nbMonth;
            it->m_nbDay = nbDay;
            it->m_nBNo = nBNo;
            it->m_csBorrow = sBorrow;
            bFound = true;
        }
    }
    if (bFound == false)
    {
        AfxMessageBox(_T("看看你的书籍信息对不对呢"));
    }
}

```

```

void CData::ReturnBook(string sISBN, int nBNo, string sBorrow)
{
    bool bFound = false;
    for (list<CBook>::iterator it = m_listBook.begin(); it != m_listBook.end(); it++)
    {

        if (it->m_sISBN == sISBN && it->m_nBNo == nBNo)
        {
            it->m_nBNo = nBNo;
            it->m_csBorrow = sBorrow;
            bFound = true;
        }
    }
    if (bFound == false)
    {
        AfxMessageBox(_T("看看你的书籍信息对不对呢"));
    }
}

```

六、设计总结

在经历了两周的面向对象实验设计课程的学习过后，学到了很多 C++ 领域的相关知识，在课程中我学会了用 C++ 类的形式封装 Windows 的 API，即 MFC。虽然在大一的时候接触过面向对象程序设计，只学了一些很基础的程序语言和逻辑，但也对本次的课程打下了一定的基础。在本次程序设计中，无论是在框架设计中还是界面设计中都遇到了很多困难，但是经过组内的讨论，以及在网上寻找资料学习过后都能解决遇到的困难，我想这就是学习的乐趣。时间非常短暂，两周的课设一晃而过，但是依旧意犹未尽，非常感谢石老师与刘老师的悉心教导，希望在未来能有机会学到更多。

在调试过程中，我们学会了通过 Ctrl+单击函数进入函数体内部查看代码。在打断点，调试程序的过程中，对变量的处理也有了更深入的了解，当程序进入一些错误时，也学会从堆栈中查找究竟是哪一步的数据出现了问题，又是哪一步的数据被污染了。

在整个编写过程中，最先遇到的比较大的难题是如何给同一 ISBN 下的书本进行编号，在深刻理解了 AddBook 函数的概念后，在外面套了一层 for 循环实现批量的输入。在调试的过程中，也找到了调去数据库中最大序号的方法，对 for 循环进一步完善后，实现了接续编号的功能。至此为止，已经对链表、不同类直接的调用有了比较深刻的理解，进一步规划了后续的编码结构，开始分析哪些功能是我们能够实现的，哪些是现阶段难以解决的。其中最困难的一个问题是难以根据一个 ISBN 删除所有对应的书籍，erase 函数会让后一个节点变成现在的节点，我们选择了在 erase 后对下一个处理的节点进行一次重定义。

```
for (list<CBook>::iterator it = m_listBook.begin(); it != m_listBook.end(); it++)
{
    if (it->m_sISBN == sISBN)
    {
        m_listBook.erase(it);
        it = m_listBook.begin();
        bFound = true;
    }
}
```

在整个编码过程中，还有一些功能我们并没有时间或者能力来实现。在用户

信息存储方面，我们希望对用户密码进行加密；在书籍录入方面，我们希望新增一个书籍库链表，减少同一 ISBN 对应书本的重复信息过多，还希望可以自动生成一个新的书本号和条形码用于后期的信息处理，在资金允许的情况下，录入 RFID，为后续的图书馆自动扫描核对信息做好基础准备；在书籍信息更改层面，有些按键函数需要使用多次循环来实现不同的功能，增加时间复杂度，后续希望一次循环内进行多次判定，返回需要的参数；在管理员层面，希望可以加入管理员设置对话框，可以给不同的用户进行权限设置，对系统功能进行统一管理，同时更进对用户信息的增、删、改、查等功能；在用户层面，添加对账户的个人信息更改、查看；在数据层面，只能做到对单一数据的可视化查看，难以以表的形式呈现，希望可以有更好的可视化方案。

序号	学号	姓名	职责	主要工作	贡献度
1			组长	书籍数据维度增加，调用数据库对数据进行接续编号，简化数据录入流程，优化增、删、改、查的全部功能，新增用户数据链表、借阅日志链表，细节调试维护，报告撰写	65%
2			组员	报告撰写，实现参考课堂实例的基础功能，针对空指针产生数据错误的弹窗警告	35%