**Python大作业**

系统名称： 图书馆管理系统

姓 名：

班 级：

完成日期： 2019.12.31

**分工情况**

|  |  |  |
| --- | --- | --- |
| **学号** | **姓名** | **分工** |
|  |  | **100%** |

**成绩评定表**

|  |  |
| --- | --- |
| **姓名** | **成绩** |
| **张雪玉** |  |

**评语：**

**签名：**

**时间：**

目录

[一、 背景介绍 1](#_Toc29213825)

[二、 功能介绍 1](#_Toc29213826)

[三、 详细使用说明 1](#_Toc29213827)

[1. 启动软件 1](#_Toc29213828)

[2. 图书信息管理 2](#_Toc29213829)

[3. 图书借阅管理 4](#_Toc29213830)

[4. 非法操作处理 6](#_Toc29213831)

[四、 总结 7](#_Toc29213832)

[五、 源码 7](#_Toc29213833)

# 背景介绍

现代图书馆是大量信息资源的集合地，图书信息和用户信息等资料繁复，涉及很多的数据资料的存储及管理，传统的手工处理信息已经不适应社会的发展，不仅效率低下，而且安全性上也没有了保障。然而目前大多数图书馆都只是小区域性书馆，已经不能满足人们对大区域图书借阅的需求，为了能更好地服务读者，急需一套适合于的跨区域性图书管理软件来更加精准、高效的管理各种复杂的图书信息。

# 功能介绍

图书馆管理系统共分为两个主要功能：

1. 图书信息管理：
   1. 图书添加
   2. 图书删除
   3. 图书修改
   4. 图书查找
2. 图书借阅管理：
   1. 借阅图书
   2. 返还图书
   3. 查看借阅

# 详细使用说明

## 启动软件

在pycharm中点击运行打开软件，主界面如图3-1所示。



图3-1 系统主界面

## 图书信息管理

本系统主要包含图书信息录入，图书信息修改，查看详细信息，图书删除和搜索图书功能。首先介绍一下图书的信息录入功能，用户登入系统后，点击添加按钮，即可添加图书信息。用户需要输入图书的书名和作者信息，然后系统自动添加图书的编号和添加日期信息。

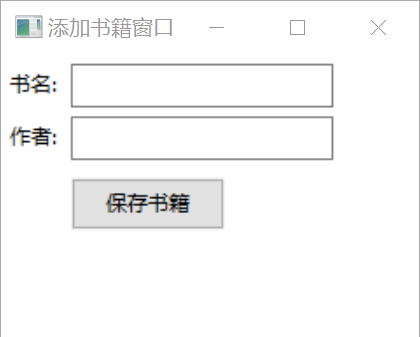


图3-2 添加书籍界面

接下来是图书信息的修改功能，首先选中某一本图书，然后鼠标单击修改按钮，即可对选中图书信息进行修改。



图3-3 图书信息修改界面

图书的删除功能与修改功能类似，首先选中某一本图书，然后点击删除按钮，该图书信息即可完成删除。



图3-4 图书信息删除

图书的查看功能是指查看图书的详细信息，系统主面板中由于空间限制，只能显示书号书名作者三个信息，如果想查看图书的详细信息例如添加时间、详细内容，则需要使用此功能进行查看。

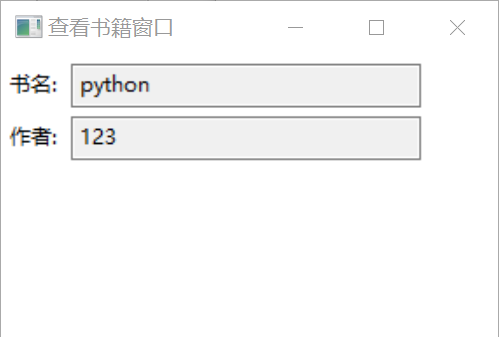


图3-5 图书详情查看

图书的搜索功能是指通过书名的全称或部分关键字对图书完成搜索的功能。具体用户点击搜索按钮，弹出搜索框，用户输入待搜索的图书信息关键字，系统即可在数据库中查询相应的图书。

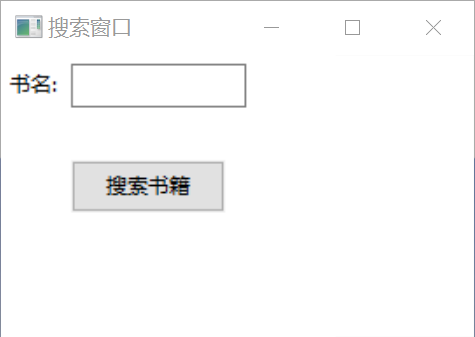


图3-6 图书搜索功能

## 3. 图书借阅管理

本系统提供对图书的借还信息录入维护的功能。

* 1. 借书信息查看

用户进入系统后，可以点击“借书信息”按钮，查看本系统当前已借阅图书的信息。系统会弹出“已借图书信息”对话框，对话框中显示借书条目信息，每行一条，每一条借书信息包含学生名、书名和借书日期。

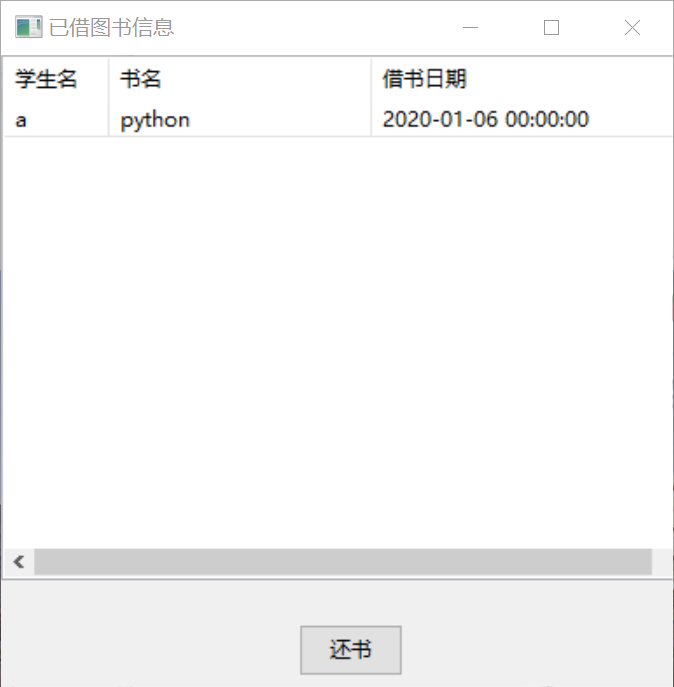


图3-7 借书信息查看

* 1. 图书借阅功能

用户进入系统后，可以首先选中需要借阅的书籍，然后点击“借书按钮”，完成对借书信息的录入。当用户点击按钮后，系统首先会弹出借书对话框，要求用户输入借书的学生姓名，接着系统会自动添加好借书日期，并向数据库中插入这条借书信息。



图3-8 图书借阅功能

* 1. 图书归还功能

用户可以通过还书功能完成对已借图书的归还操作。具体操作步骤为，首先用户点击“还书”按钮，然后系统会弹出还书对话框，接着用户输入待书的名称，图书归还完毕。

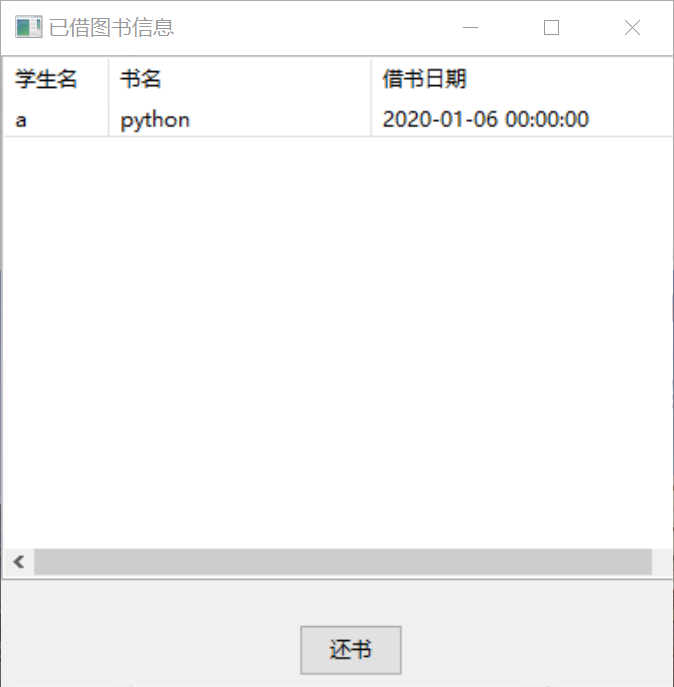


图3-9 图书归还界面1

除此之外，用户还可以在借书信息界面选中待还的图书条目。具体操作如下：点击借书信息按钮、选中待还的书籍条目、然后点击还书按钮，图书归还完毕。



图3-10 图书归还界面2

## 4. 非法操作处理

当系统判断用户触发了非法操作时，系统会报出相应的错误警告。例如当选中图书条目时，点击了搜索按钮，系统会提示“选中了条目！！！该操作非法”，如图3-11所示。



图3-11 非法信息提示

# 总结

随着社会进步和经济发展，人们的平均文化水平日益提升，对阅读的需求量越来越大。图书馆的图书存储量和借阅数量逐年上升。因此，如何管理好图书，对图书进行高效准确的整理、借还成为困扰图书馆管理人员的一大难题。图书馆管理系统能够帮助管理人员解决好这一问题，在系统中，图书的存储情况以及图书的借阅情况能够清晰的展示在系统界面上，为图书馆的管理提供了全新的管理方式，降低了馆员的劳动强度，简化了读者的借还书手续，提高了图书的借阅率。该系统功能完全，避免了之前人员统计错误而产生的问题，受到馆员和读者的一致好评。

# 源码

1. 增加功能

**class** AddFrame(wx.Frame):  
  
 *'''添加书籍弹出的小窗口'''* **def** \_\_init\_\_(self, parent, title):  
 *'''初始化该小窗口的布局'''* self.mainframe = parent  
 *#生成一个300\*300的框* wx.Frame.\_\_init\_\_(self, parent, title = title, size = (400, 250))  
  
 self.panel = wx.Panel(self, pos = (0, 0), size = (400, 250))  
 self.panel.SetBackgroundColour(**"#FFFFFF"**) *#背景为白色  
  
 #三个编辑框，分别用来编辑书名，作者，书籍相关信息* bookName\_tip = wx.StaticText(self.panel, label = **"书名:"**, pos = (5, 8), size = (35, 25))  
 bookName\_tip.SetBackgroundColour(**"#FFFFFF"**)  
 bookName\_text = wx.TextCtrl(self.panel, pos = (40, 5), size = (340, 25))  
 self.name = bookName\_text  
  
 author\_tip = wx.StaticText(self.panel, label = **"作者:"**, pos = (5, 38), size = (35, 25))  
 author\_tip.SetBackgroundColour(**"#FFFFFF"**)  
 author\_text = wx.TextCtrl(self.panel, pos = (40, 35), size = (340, 25))  
 self.author = author\_text  
  
 content\_tip = wx.StaticText(self.panel, label = **"内容:"**, pos = (5, 68), size = (340, 25))  
 content\_tip.SetBackgroundColour(**"#FFFFFF"**)  
 content\_text = wx.TextCtrl(self.panel, pos = (40, 65), size = (340, 100), style = wx.TE\_MULTILINE)  
 self.content = content\_text  
  
 save\_button = wx.Button(self.panel, label = **"保存书籍"**, pos = (160, 170))  
 self.Bind(wx.EVT\_BUTTON, self.saveBook, save\_button)  
  
 *#需要用到的数据库接口* self.dbhelper = DBHelper()  
  
  
 **def** saveBook(self, evt):  
 bookName = self.name.GetValue()  
 author = self.author.GetValue()  
 content = self.content.GetValue()  
 **if** bookName == **"" or** author == **"" or** content == **""**:  
 warn = wx.MessageDialog(self, message = **"不能为空"**, caption = **"错误警告"**, style = wx.YES\_DEFAULT | wx.ICON\_ERROR)  
 warn.ShowModal() *#提示错误* warn.Destroy()  
 **return** self.dbhelper = DBHelper()  
 datas = self.dbhelper.getAllBook()  
 datass = self.dbhelper.getAllLendBook()  
 **for** data **in** datas:  
 **if** data[1] == bookName:  
 warn = wx.MessageDialog(self, message=**"重复书名"**, caption=**"错误警告"**, style=wx.YES\_DEFAULT | wx.ICON\_ERROR)  
 warn.ShowModal() *# 提示错误* warn.Destroy()  
 **return  
 for** data **in** datass:  
 **if** data[1] == bookName:  
 warn = wx.MessageDialog(self, message=**"重复书名"**, caption=**"错误警告"**, style=wx.YES\_DEFAULT | wx.ICON\_ERROR)  
 warn.ShowModal() *# 提示错误* warn.Destroy()  
 **return** book = Book(bookName, author, content)  
 book\_id = self.dbhelper.insertBook(book)  
 Num = self.mainframe.list.GetItemCount()  
 **for** num **in** range(0, Num):  
 self.mainframe.list.DeleteItem(0)  
 self.mainframe.dbhelper = DBHelper()  
 datas = self.mainframe.dbhelper.getAllBook()  
 **for** data **in** datas:  
 index = self.mainframe.list.InsertItem(self.mainframe.list.GetItemCount(), str(data[0]))  
 self.mainframe.list.SetItem(index, 1, str(data[1]))  
 self.mainframe.list.SetItem(index, 2, str(data[2]))  
 self.Destroy()

1. 删除功能

**def** delBook(self, evt):  
 *'''删除书籍按钮，先选中,然后删除'''* selectId = self.list.GetFirstSelected()  
 **if** selectId == -1:  
 warn = wx.MessageDialog(self, message = **"未选中任何条目！！！"**, caption = **"错误警告"**, style = wx.YES\_DEFAULT | wx.ICON\_ERROR)  
 warn.ShowModal() *#提示错误* warn.Destroy()  
 **return  
 else**:  
 bookid = self.list.GetItem(selectId, 0).Text *#得到书本id* self.list.DeleteItem(selectId) *#先在listctrl中删除选中行* self.dbhelper.deleteBook(bookid)

1. 修改功能

**class** UpdateFrame(wx.Frame):  
 **def** \_\_init\_\_(self, parent, title, select\_id):  
 *'''初始化更新图书信息界面总布局'''* wx.Frame(parent, title = title, size = (400, 250))  
  
 *#用来调用父frame,便于更新* self.mainframe = parent  
 *#生成一个300\*300的框* wx.Frame.\_\_init\_\_(self, parent, title = title, size = (400, 250))  
  
 self.panel = wx.Panel(self, pos = (0, 0), size = (400, 250))  
 self.panel.SetBackgroundColour(**"#FFFFFF"**) *#背景为白色  
  
 #三个编辑框，分别用来编辑书名，作者，书籍相关信息* bookName\_tip = wx.StaticText(self.panel, label = **"书名:"**, pos = (5, 8), size = (35, 25))  
 bookName\_tip.SetBackgroundColour(**"#FFFFFF"**)  
 bookName\_text = wx.TextCtrl(self.panel, pos = (40, 5), size = (340, 25))  
 self.name = bookName\_text  
  
 author\_tip = wx.StaticText(self.panel, label = **"作者:"**, pos = (5, 38), size = (35, 25))  
 author\_tip.SetBackgroundColour(**"#FFFFFF"**)  
 author\_text = wx.TextCtrl(self.panel, pos = (40, 35), size = (340, 25))  
 self.author = author\_text  
  
 content\_tip = wx.StaticText(self.panel, label = **"内容:"**, pos = (5, 68), size = (340, 25))  
 content\_tip.SetBackgroundColour(**"#FFFFFF"**)  
 content\_text = wx.TextCtrl(self.panel, pos = (40, 65), size = (340, 100), style = wx.TE\_MULTILINE)  
 self.content = content\_text  
  
 save\_button = wx.Button(self.panel, label = **"保存修改"**, pos = (160, 170))  
 self.Bind(wx.EVT\_BUTTON, self.saveUpdate, save\_button)  
  
 *#选中的id和bookid* self.select\_id = select\_id  
 self.bookid = self.mainframe.list.GetItem(select\_id, 0).Text *#获取第select\_id行的第0列的值  
 #需要用到的数据库接口* self.dbhelper = DBHelper()  
 self.showAllText() *#展现所有的text原来取值* **def** showAllText(self):  
 *'''显示概述本原始信息'''* data = self.dbhelper.getBookById(self.bookid) *#通过id获取书本信息* self.name.SetValue(data[0]) *#设置值* self.author.SetValue(data[1])  
 self.content.SetValue(data[2])  
  
 **def** saveUpdate(self, evt):  
 *'''保存修改后的值'''* bookName = self.name.GetValue() *#获得修改后的值* author = self.author.GetValue()  
 content = self.content.GetValue()  
  
 print(**"书名:"**+bookName)  
 **if** bookName == **"" or** author == **"" or** content == **""**:  
 print(**"进来了"**)  
 warn = wx.MessageDialog(self, message = **"所有信息不能为空！！！"**, caption = **"错误警告"**, style = wx.YES\_DEFAULT | wx.ICON\_ERROR)  
 warn.ShowModal() *#提示错误* warn.Destroy()  
 **return  
 else**:  
 print(**"开始将修改后的数据保存到数据库中"**)  
 book = Book(bookName, author, content) *#将数据封装到book对象中* self.dbhelper.saveUpdate(self.bookid, book)  
 self.mainframe.list.SetItem(self.select\_id, 1, bookName)  
 self.mainframe.list.SetItem(self.select\_id, 2, author)  
 self.Destroy()

1. 查找功能

**class** Searchframe(wx.Frame):  
 *'''搜索书籍弹出的小窗口'''* **def** \_\_init\_\_(self, parent, title):  
 *'''初始化该小窗口的布局'''* self.mainframe = parent  
 wx.Frame.\_\_init\_\_(self, parent, title = title, size = (400, 250))  
  
 self.panel = wx.Panel(self, pos = (0, 0), size = (400, 250))  
 self.panel.SetBackgroundColour(**"#FFFFFF"**)  
 bookName\_tip = wx.StaticText(self.panel, label = **"书名:"**, pos = (5, 8), size = (35, 25))  
 bookName\_tip.SetBackgroundColour(**"#FFFFFF"**)  
 bookName\_text = wx.TextCtrl(self.panel, pos = (40, 5), size = (340, 25))  
 self.name = bookName\_text  
 save\_button = wx.Button(self.panel, label = **"搜索书籍"**, pos = (160, 170))  
 self.Bind(wx.EVT\_BUTTON, self.Search\_book, save\_button)  
  
 **def** Search\_book(self,txt):  
 bookName = self.name.GetValue()  
 *# 需要用到的数据库接口* self.dbhelper = DBHelper()  
  
 datass = self.dbhelper.getBookByName(bookName)  
 Num = self.mainframe.list.GetItemCount()  
 **for** num **in** range(0,Num):  
 self.mainframe.list.DeleteItem(0)  
 **if** datass == **None**:  
 warn = wx.MessageDialog(self, message=**"没有这本书"**, caption=**"错误警告"**)  
 warn.ShowModal() *# 提示错误* warn.Destroy()  
 **return** index = self.mainframe.list.InsertItem(self.mainframe.list.GetItemCount(), str(datass[0]))  
 self.mainframe.list.SetItem(index, 1, str(datass[1]))  
 self.mainframe.list.SetItem(index, 2, str(datass[2]))  
 self.Destroy()

1. 借书功能

**class** LendBook(wx.Frame):  
 **def** \_\_init\_\_(self, parent, title, select\_id,bookid):  
 *'''初始化更新图书信息界面总布局'''  
  
 #用来调用父frame,便于更新* self.bookid=bookid  
 self.mainframe = parent  
 *#生成一个300\*300的框* wx.Frame.\_\_init\_\_(self, parent, title = title, size = (200 , 130))  
  
 self.panel = wx.Panel(self, pos = (0, 0), size = (200, 130))  
 self.panel.SetBackgroundColour(**"#FFFFFF"**) *#背景为白色  
  
 #三个编辑框，分别用来编辑书名，作者，书籍相关信息* student\_tip = wx.StaticText(self.panel, label = **"学生名:"**, pos = (5, 20), size = (50, 25))  
 student\_tip.SetBackgroundColour(**"#FFFFFF"**)  
 student\_text = wx.TextCtrl(self.panel, pos = (60, 20), size = (100, 25))  
 self.studentname = student\_text  
  
 lend\_button = wx.Button(self.panel, label = **"确定"**, pos = (95, 60))  
 self.Bind(wx.EVT\_BUTTON, self.lend, lend\_button)  
  
 *#选中的id和bookid* self.select\_id = select\_id  
 *#需要用到的数据库接口* self.dbhelper = DBHelper()  
  
 **def** lend(self,evt):  
 *'''删除列表中的值，删除数据库的值，增加借阅图书数据库的值'''* **if** self.select\_id == -1:  
 warn = wx.MessageDialog(self, message=**"未选中任何条目！"**, caption=**"错误警告"**)  
 warn.ShowModal() *# 提示错误* warn.Destroy()  
 **return  
 else**:  
 stuname = self.studentname.GetValue()  
 alldata=self.dbhelper.getBookById(self.bookid)  
 bookname=alldata[0]  
 author=alldata[1]  
 content=alldata[2]  
 lendbook=Lendbook(stuname,bookname,author,content)  
 self.dbhelper.savelendbook(lendbook)  
 self.dbhelper.deleteBook(self.bookid)  
 self.Destroy()

1. 还书功能

**class** BackBook(wx.Frame):  
 **def** \_\_init\_\_(self, parent, title):  
 *'''初始化更新图书信息界面总布局'''  
  
 # wx.Frame(parent, title = title, size = (200, 125))  
  
 #用来调用父frame,便于更新* self.mainframe = parent  
 *#生成一个300\*300的框* wx.Frame.\_\_init\_\_(self, parent, title = title, size = (200 , 130))  
  
 self.panel = wx.Panel(self, pos = (0, 0), size = (200, 130))  
 self.panel.SetBackgroundColour(**"#FFFFFF"**) *#背景为白色* book\_tip = wx.StaticText(self.panel, label = **"书名:"**, pos = (5, 20), size = (50, 25))  
 book\_tip.SetBackgroundColour(**"#FFFFFF"**)  
 book\_text = wx.TextCtrl(self.panel, pos = (60, 20), size = (100, 25))  
 self.bookname = book\_text  
  
 lend\_button = wx.Button(self.panel, label = **"确定"**, pos = (95, 60))  
 self.Bind(wx.EVT\_BUTTON, self.back, lend\_button)  
 *# 需要用到的数据库接口* self.dbhelper = DBHelper()  
  
 **def** back(self,evt):  
 bkname=self.bookname.GetValue()  
 bookdata=self.dbhelper.getBook\_ByName(bkname)  
 book1 = Book(bookdata[0], bookdata[1], bookdata[2])  
 book\_id = self.dbhelper.insertBook(book1)  
 self.dbhelper.delLendMessgae(bkname)  
 self.mainframe.addToList(book\_id, book1)  
 self.Destroy()

1. 查阅借书功能

**class** Lend\_Message(wx.Frame):  
 **def** \_\_init\_\_(self,parent, title):  
 *'''初始化系统总体布局，包括各种控件'''* self.mainframe = parent  
  
 *#生成一个宽为400，高为400的frame框* wx.Frame.\_\_init\_\_(self, parent, title=title, size=(400, 400))  
  
 *#定一个网格布局* self.main\_layout = wx.BoxSizer(wx.VERTICAL)  
  
  
 *#生成一个列表* self.list = wx.ListCtrl(self, -1, size = (400,300), style = wx.LC\_REPORT | wx.LC\_HRULES | wx.LC\_VRULES) *#| wx.LC\_SINGLE\_SEL  
 #列表有散列，分别是书本ID,书名，添加日期* self.list.InsertColumn(0, **"学生名"**)  
 self.list.InsertColumn(1, **"书名"**)  
 self.list.InsertColumn(2, **"借书日期"**)  
 *#设置各列的宽度* self.list.SetColumnWidth(0, 60) *#设置每一列的宽度* self.list.SetColumnWidth(1, 150)  
 self.list.SetColumnWidth(2, 200)  
  
 *#添加一组按钮，实现增删改查,用一个panel来管理该组按钮的布局* self.panel = wx.Panel(self, pos = (0, 300), size = (400, 100))  
 back\_button = wx.Button(self.panel, label=**"还书"**, pos=(170, 25), size=(60, 30)) *# , size = (75, 30)* self.Bind(wx.EVT\_BUTTON, self.backbook, back\_button)  
  
  
 self.dbhelper = DBHelper()  
 datas = self.dbhelper.getAllLendBook()  
  
 **for** data **in** datas:  
 index = self.list.InsertItem(self.list.GetItemCount(), str(data[0]))  
 self.list.SetItem(index, 1, data[1])  
 self.list.SetItem(index, 2, str(data[2]))  
  
  
 **def** backbook(self,evt):  
 *'''还书，先把数据插入图书库，再从已借阅图书删除，删除显示信息'''* selectId = self.list.GetFirstSelected()  
 **if** selectId == -1:  
 warn = wx.MessageDialog(self, message=**"未选中任何条目！！！"**, caption=**"错误警告"**, style=wx.YES\_DEFAULT | wx.ICON\_ERROR)  
 warn.ShowModal() *# 提示错误* warn.Destroy()  
 **return  
 else**:  
 bookname = self.list.GetItem(selectId, 1).Text  
 bookdata=self.dbhelper.getdata(bookname)  
 book1=Book(bookdata[0][0],bookdata[0][1],bookdata[0][2])  
 book\_id=self.dbhelper.insertBook(book1)  
 self.dbhelper.delLendMessgae(bookname)  
 self.list.DeleteItem(selectId)  
 self.mainframe.addToList(book\_id, book1)