图书管理系统实验报告

马宇骁　PB19151769

更新：*December 12, 2021*

1　实验要求

* 概述：

实现一个简单的图书管理系统。图书基本信息包括：图书 ID，书名，ISBN，出版社，出版年月，作者，标签（可选做）。读者基本信息包括：读者 ID，读者姓名，读者类型，读者联系电话。

1.1　功能需求描述

（1）图书信息维护。批量导入书目信息、手工添加书目信息、删除书目信息、修改书目信息。图书购置批量入库（库存增加）、图书清理批量出库（因图书损坏，库存减少）。

（2）读者信息维护。批量导入读者信息、手工添加读者、删除读者、修改读者信息。

（3）图书信息查询。根据书名、作者、标签（可选做）、出版社和 ISBN 查询图书。显示库存数量，根据图书库存数量显示是否可借。

（4）读者信息查询。根据读者姓名或联系电话查询显示读者信息。查询显示读者在借图书信息。查询显示读者借阅历史。

（5）图书借出。

（6）图书归还。

（7）系统关闭。保存数据之后关闭系统。这些数据在系统启动的时候自动读入系统。

1.2　使用方法描述

（1）该系统由图书管理员操作，以上所有操作都不是由读者操作完成的。

（2）同一个 ID 的图书可能有多本，库存量随着入库、出库而改变，可借数量随着借出、归还而改变。库存量为可借数量与借出数量之和。

（3）读者有两个类型：会员和非会员。主要区别是借阅的数量上限。会员上限 8 本，非会员上限 4 本。当未还图书数量达到上限时，无法借阅。

（4）只有在图书库存数量为 0 时，才可以删除书目。

2 实现

2.1 代码部分

Python 程序显示如下：

```
import csv
import pandas as pd
import easygui as eg

class Book:
    def __init__(self, book_ID, book_name, ISBN, pub, date, author, number):
        self.book_name = book_name
        self.book_ID = book_ID
        self.ISBN = ISBN
        self.pub = pub
        self.date = date
        self.author = author
        self.number = number

    def __getitem__(self, item):
        if item in self.__dict__:
            return self.__dict__[item]

#status为1代表是会员
class Reader:
    def __init__(self, reader_ID, reader_name, tel, status):
        self.reader_name = reader_name
        self.reader_ID = reader_ID
        self.tel = tel
        self.status = status
        self.lend = {}
        self.lent = {}
    def __getitem__(self, item):
        if item in self.__dict__:
          return self.__dict__[item]

def insert_book():
    try:
        name = eg.enterbox(msg='请输入导入数据', title='图书管理系统管理员模
                                                    式', default='', strip=True,
                                                     image=None)
        c = pd.read_excel(name)
        for i in range(len(c)):
```

```
            m = []
            for k in c.loc[i]:
                m.append(k)
            books.append(Book(str(m[0]),str(m[1]),str(m[2]),str(m[3]),str(m[4]
                                                    ),str(m[5]),int(m[6])))
        eg.msgbox(msg="导入成功！",title="图书管理系统管理员模式",ok_button="
                                                    确定")
    except:
        eg.msgbox(msg="导入失败！",title="图书管理系统管理员模式",ok_button="
                                                    确定")

def add_book():
    m = eg.multenterbox(msg='增添图书', title='图书管理系统管理员模式', fields
                                                =['id','name','ISBN','pub','date'
                                                ,'author','number'], values=())
    books.append(Book(str(m[0]),str(m[1]),str(m[2]),str(m[3]),str(m[4]),str(m[
                                                5]),int(m[6])))
# （3）删除书目
def delete_book():
    id = eg.enterbox(msg='请输入删除的书id ', title='图书管理系统管理员模式',
                                                 default=' ', strip=True, image=
                                                None)

    for i in books:
        if i.book_ID == id:
            if i.number == 0:
                books.remove(i)
                break
    else:
        eg.msgbox(msg="操作拒绝",title="图书管理系统管理员模式",ok_button="确
                                                    定")

def modify_book():
    id = eg.enterbox(msg='待修改的书id ', title='图书管理系统管理员模式',
                                                default=' ', strip=True, image=
                                                None)
    id = str(id)
    for i in books:
        if i.book_ID == id:
            new = eg.multenterbox(msg=id, title='图书管理系统管理员模式修改图
                                                    书信息', fields=['name','
                                                    ISBN','pub','date','
                                                    author','number'], values
                                                    =())
```

```
            i.book_name = new[0] if  new[0] is not None else i.book_name
            i.ISBN = new[1] if new[1] is not None else i.ISBN
            i.pub = new[2] if new[2] is not None else i.pub
            i.date = new[3] if new[3] is not None else i.date
            i.author = new[4] if new[4] is not None else i.author
            i.number = new[5] if new[5] is not None else i.number
            i.number = int(i.number)
            break
    if i.book_ID != id:
        eg.msgbox(msg="不存在这本书",title="图书管理系统管理员模式",ok_button=
                                                    "确定")

# 输入书名以及要修改的属性
def add_num():
    n = eg.multenterbox(msg='增添已有图书', title='图书管理系统管理员模式',
                                                fields=['name','number'], values=
                                                ())
    num = int(n[1])
    name = n[0]
    for i in books:
        if i.book_name == name:
            i.number += num
            break
    if i.book_name != name:
        eg.msgbox(msg="不存在这本书",title="图书管理系统管理员模式",ok_button=
                                                    "确定")
def sub_num():
    n = eg.multenterbox(msg='减少已有图书', title='图书管理系统管理员模式',
                                                fields=['name','number'], values=
                                                ())
    num = int(n[1])
    name = n[0]
    for i in books:
        if i.book_name == name:
            if i.number == 0:
                eg.msgbox(msg="这本书没有了不能减少",title="图书管理系统管理员
                                                        模式",ok_button="确定
                                                        ")
                break
            else:
                if i.number-num < 0:
                    eg.msgbox(msg="这本书不能减少这么多",title="图书管理系统管
                                                            理员模式",
                                                            ok_button="确定")
```

```
                    break
                else:
                    i.number -= num
                    break
    if i.book_name != name:
        eg.msgbox(msg="不存在这本书",title="图书管理系统管理员模式",ok_button=
                                                    "确定")
# 即修改数量属性，且减少后数量不能小于0

# （6）分菜单
def manage_book():
    while True:
        choice = eg.buttonbox(msg='''
            图书管理系统:

            1.导入书目
            2.增加书目
            3.删除书目
            4.修改书目
            5.库存增加
            6.库存减少
            0.退出
            ''', title='管理员模式 ', choices=('1', '2', '3','4','5','6','0'),
                                                    image=None)

        if choice == '1':
            insert_book()
        elif choice == '2':
            add_book()
        elif choice == '3':
            delete_book()
        elif choice == '4':
            modify_book()
        elif choice == '5':
            add_num()
        elif choice == '6':
            sub_num()
        elif choice == '0':
            break

def insert_reader():
    try:
        name = eg.enterbox(msg=' 请输入导入数据 ', title='图书管理系统管理员模
                                                    式', default=' ', strip=True,
```

```
                                                    image=None)
            c = pd.read_excel(name)
            c = c.fillna(0)
            for i in range(len(c)):
                m = []
                for k in c.loc[i]:
                    m.append(k)
                readers.append(Reader(str(m[0]),str(m[1]),str(m[2]),int(m[3])))
                readers[i].lend = eval(c['lend'][i]) if c['lend'][i] != 0 else {}
                readers[i].lent = eval(c['lent'][i]) if c['lent'][i] != 0 else {}
            eg.msgbox(msg="导入成功！",title="图书管理系统管理员模式",ok_button="
                                                    确定")
        except:
            eg.msgbox(msg="导入失败！",title="图书管理系统管理员模式",ok_button="
                                                    确定")

    def add_reader():
        m = eg.multenterbox(msg='增添读者', title='图书管理系统管理员模式', fields
                                                    =['id','name','tel','status'],
                                                    values=())
        readers.append(Reader(str(m[0]),str(m[1]),str(m[2]),int(m[3])))


    def delete_reader():
        id = eg.enterbox(msg=' 请输入删除的读者id ', title='图书管理系统管理员模式
                                                    ', default=' ', strip=True, image
                                                    =None)
        for i in readers:
            if i.reader_ID == id:
                readers.remove(i)
                break
        if i.reader_ID != id:
            eg.msgbox(msg="不存在这个人",title="图书管理系统管理员模式",ok_button=
                                                    "确定")


    def modify_reader():
        id = eg.enterbox(msg=' 待修改的读者ID ', title='图书管理系统管理员模式',
                                                    default=' ', strip=True, image=
                                                    None)
        for i in readers:
            if i.reader_ID == id:
                new = eg.multenterbox(msg=id, title='图书管理系统管理员模式修改图
                                                    书信息', fields=['name','
```

```
                                                    tel','status'], values=()
                                                    )
            i.reader_name = new[0] if new[0] is not None else i.reader_name
            i.tel = new[1] if new[1] is not None else i.tel
            i.status = int(new[2]) if new[2] is not None else i.status
            break
    if i.reader_ID != id:
        eg.msgbox(msg="不存在这个人",title="图书管理系统管理员模式",ok_button=
                                                    "确定")



def manage_reader():
    while True:
        choice = eg.buttonbox(msg='''
            读者管理系统:

         1.导入读者
         2.增加读者
         3.删除读者
         4.修改读者信息
         0.退出
            ''', title='管理员模式 ', choices=('1', '2', '3','4','0'), image=
                                                    None)

        if choice == '1':
            insert_reader()
        elif choice == '2':
            add_reader()
        elif choice == '3':
            delete_reader()
        elif choice == '4':
            modify_reader()
        elif choice == '0':
            break


# 5.图书、读者信息查询
def search_book():
    while True:
        choice = eg.buttonbox(msg='''
            图书信息查询:

         1.书名
```

```
            2.作者
            3.出版社
            4.ISBN
            0.退出
                '''', title='管理员模式 ', choices=('1', '2', '3','4','0'), image=
                                                        None)

        if choice == '0':
            break
        key = eg.enterbox(msg=' 请输入查询的内容 ', title='图书管理系统管理员
                                                    模式', default=' ', strip=
                                                    True, image=None)
        temp = []
        tmp = []
        if choice == '1':
            for i in books:
                if i.book_name == key:
                    temp.append(i)
        elif choice == '2':
            for i in books:
                if i.author == key:
                    temp.append(i)
        elif choice == '3':
            for i in books:
                if i.pub == key:
                    temp.append(i)
        elif choice == '4':
            for i in books:
                if i.ISBN == key:
                    temp.append(i)

        for j in temp:
            for (k, v) in j.__dict__.items():
                t=[]
                t.append(k)
                t.append(v)
                tmp.append(t)
        eg.msgbox(msg=tmp,title="图书管理系统管理员模式",ok_button="确定")

def search_num():
    key = eg.enterbox(msg=' 请输入图书名字查询的库存 ', title='图书管理系统管
                                                理员模式', default=' ', strip=
                                                True, image=None)
    for i in books:
```

```
            if i.book_name == key:
                num = i.number
                break
        for i in readers:
            if i.lend[key]:
                num = num + i.lend[key]
        eg.msgbox(msg=num,title="图书管理系统管理员模式",ok_button="确定")


    def search_reader():
        while True:
            choice = eg.buttonbox(msg='''
                读者查询:

                1.读者名
                2.读者id
                3.联系方式
                0.退出
                ''', title='管理员模式 ', choices=('1', '2', '3','0'), image=None)

            if choice == '0':
                break
            key = eg.enterbox(msg=' 请输入查询的内容 ', title='图书管理系统管理员
                                                        模式', default=' ', strip=
                                                        True, image=None)
            temp = []
            tmp = []
            if choice == '1':
                for i in readers:
                    if i.reader_name == key:
                        temp.append(i)
            elif choice == '2':
                for i in readers:
                    if i.reader_ID == key:
                        temp.append(i)
            elif choice == '3':
                for i in readers:
                    if i.tel == key:
                        temp.append(i)
            for i in temp:
                for (k, v) in i.__dict__.items():
                    t=[]
                    t.append(k)
                    t.append(v)
```

```
                tmp.append(t)
        eg.msgbox(msg=tmp,title="图书管理系统管理员模式",ok_button="确定")



# 6、图书借还
#字典值相加
def returnSum(myDict):
    sum = 0
    for i in myDict:
        sum = sum + myDict[i]
    return sum


def lend_book():
    av = []
    for i in books:
        if i.number > 0:
            av.append(i.book_name)
    eg.msgbox(msg=av,title="可借的书",ok_button="确定")
    name = eg.multenterbox(msg='借书信息', title='图书管理系统管理员模式',
                                                fields=['book_name','reader_name'
                                                ], values=())
    name1 = name[0]
    name2 = name[1]
    for i in books:
        if i.book_name == name1:
            break
    if i.book_name != name1:
        eg.msgbox(msg='没这本书',title="图书管理系统管理员模式",ok_button="确
                                                    定")
        return 0
    for j in readers:
        if j.reader_name == name2:
            break
    if j.reader_name != name2:
        eg.msgbox(msg='没这个人',title="图书管理系统管理员模式",ok_button="确
                                                    定")
        return 0

    #jlend = 0
    if (j.lend):
        jlend = returnSum(j.lend)
    if (j.status == 1) and (jlend >= 8):
```

```
            eg.msgbox(msg='会员借书达到上限',title="图书管理系统管理员模式",
                                                    ok_button="确定")
        elif (j.status == 0) and (jlend >= 4):
            eg.msgbox(msg='借书达到上限',title="图书管理系统管理员模式",ok_button=
                                                    "确定")
        elif i.number == 0:
            eg.msgbox(msg='剩余不够',title="图书管理系统管理员模式",ok_button="确
                                                    定")
        else:
            if name1 in j.lend:
                j.lend[name1] += 1
            else:
                j.lend[name1] = 1
            if name1 in j.lent:
                j.lent[name1] += 1
            else:
                j.lent[name1] = 1
            i.number = i.number - 1
            eg.msgbox(msg='这本书借之后可借为:'+str(i.number),title="图书管理系统
                                                    管理员模式",ok_button="确定")


    def return_book():
        name = eg.multenterbox(msg='借书信息', title='图书管理系统管理员模式',
                                                fields=['book_name','reader_name'
                                                ], values=())
        name1 = name[0]
        name2 = name[1]
        for i in books:
            if i.book_name == name1:
                break
        if i.book_name != name1:
            eg.msgbox(msg='没这本书',title="图书管理系统管理员模式",ok_button="确
                                                    定")
            return 0
        for j in readers:
            if j.reader_name == name2:
                break
        if j.reader_name != name2:
            eg.msgbox(msg='没这个人',title="图书管理系统管理员模式",ok_button="确
                                                    定")
            return 0
        if name1 not in j.lend:
            eg.msgbox(msg='ta没借这本书',title="图书管理系统管理员模式",ok_button=
```

```
                                                    "确定")
            return 0
        j.lend[name1] -= 1
        i.number += 1
        eg.msgbox(msg='这本书还之后可借为:'+str(i.number),title="图书管理系统管理
                                                    员模式",ok_button="确定")

# 先判断读者是否能继续借书，若借书上限满、库存不够则不能继续借。

# 还书时判断读者是否持有这本书

# 其中为使类与字典转换，添加getitem

# 主函数以及菜单
books = []
readers = []
def main():
    B = pd.read_excel(r'C:\Users\Moriarty\Desktop\python\2021秋\大作业\origin.
                                                    xls')
    R = pd.read_excel(r'C:\Users\Moriarty\Desktop\python\2021秋\大作业\origin2
                                                    .xls')
    R = R.fillna(0)
    for i in range(len(B)):
        m = []
        for k in B.loc[i]:
            m.append(k)
        books.append(Book(str(m[0]),str(m[1]),str(m[2]),str(m[3]),str(m[4]),
                                                    str(m[5]),m[6]))
    for i in range(len(R)):
        m = []
        for k in R.loc[i]:
            m.append(k)
        readers.append(Reader(str(m[0]),str(m[1]),str(m[2]),int(m[3])))
        readers[i].lend = eval(R['lend'][i]) if R['lend'][i] != 0 else {}
        readers[i].lent = eval(R['lent'][i]) if R['lent'][i] != 0 else {}
    while True:
        a = eg.buttonbox(msg='''
            欢迎使用图书管理系统:

            1.图书信息维护
            2.读者信息维护
            3.图书信息查询
            4.读者信息查询
            5.图书借出
```

```
        6.图书归还
        7.库存
        0.退出
        ''', title='图书管理系统 ', choices=('1', '2', '3','4','5','6','7'
                                                ,'0'), image=None)
    if a == '1':
        manage_book()
    elif a == '2':
        manage_reader()
    elif a == '3':
        search_book()
    elif a == '4':
        search_reader()
    elif a == '5':
        lend_book()
    elif a == '6':
        return_book()
    elif a == '7':
        search_num()
    elif a == '0':
        for i in range(len(B)):
            B = B.drop([i])
        idd,nm,isbn,pub,date,au,nu = [],[],[],[],[],[],[]
        for m in books:
            idd.append(m.book_ID)
            nm.append(m.book_name)
            isbn.append(m.ISBN)
            pub.append(m.pub)
            date.append(m.date)
            au.append(m.author)
            nu.append(m.number)
        B['book_ID'] = idd
        B['book_name'] = nm
        B['ISBN'] = isbn
        B['pub'] = pub
        B['date'] = date
        B['author'] = au
        B['number'] = nu
        B.to_excel(r'C:\Users\Moriarty\Desktop\python\2021秋\大作业\origin
                                                .xls',index=None)

        for i in range(len(R)):
            R = R.drop([i])
        idd,nm,tel,st,ld,lt = [],[],[],[],[],[]
```

```
                for m in readers:
                    idd.append(m.reader_ID)
                    nm.append(m.reader_name)
                    tel.append(m.tel)
                    st.append(m.status)
                    ld.append(m.lend)
                    lt.append(m.lent)
                R['reader_ID'] = idd
                R['reader_name'] = nm
                R['tel'] = tel
                R['status'] = st
                R['lend'] = ld
                R['lent'] = lt
                R.to_excel(r'C:\Users\Moriarty\Desktop\python\2021秋\大作业\
                                                        origin2.xls',index=None)

                break

nm = '管理员登陆'
cd = '图书管理系统'
while True:
    id,mm = eg.multpasswordbox(nm,cd,(['用户名','密码']))
    if (id == 'tesla606') and (mm == '802366'):
        main()
        break
```

2.2 分析部分

所有功能通过菜单实现功能选择。

2.2.1 图形界面

为了方便，使用 easygui 包进行图形界面绘制，每次通过返回值判断函数读入或者下一个界面的跳转。

2.2.2 类

先创建两个类，定义为 class Book 和 class Reader。

其中，book 为图书包含名称，id，ISBN，出版社，日期，作者和数量总共 7 个属性；reader 包含姓名，id，电话，是否为会员，在借状态和历史借阅信息总共 6 个属性。两个类均定义__getitem__函数方便后续查询信息的时候使用。

2.2.3 书

* 批量插入：

通过 pandas 的读入 read_excel 将目标 xls 文件读入程序，并添加到类保存到 books 这个列表中。为了防止没有这个文件，使用 try 和 except 判断导入成功与否。

* 添加书：

在图形界面框输入图书信息，添加到 Book 类 append 进 books 列表中。

* 删除书：

先判断 books 列表的 Book 有没有这本书，若有再判断是不是库存为 0，若是就 remove；上述判断若有一个不是则操作拒绝。

* 修改书：

判断输入的 id 在不在 books 中：若在则输入修改的新信息（若相应修改框不输入则保持原信息）；若不在则弹出不存在这本书。

* 增加 & 减少库存：

判断是否有这本书，若有则增加/减少输入的数字（若减少的本书使得库存小于等于 0，则操作拒绝）；没有则输出不存在。

2.2.4 读者

* 批量插入：

通过 pandas 的读入 read_excel 将目标 xls 文件读入程序，过程几乎同 book 的读入。但考虑到 readers 的 lend 和 lent 的特殊性，做如下处理：对目标表格的所有空的内容填入 0

$$\Rightarrow$$

判断 lend 的位置的值是否为 0

$$\Rightarrow$$

是则赋值为空的 dict；否则使用 eval 将表格中的字符串转化为 dict 赋值。（lent 同 lend）

* 添加读者：

在图形界面框输入读者信息，类似添加书。

* 删除读者：

先判断 readers 列表的 Reader 有没有这个人，若有就 remove；没有就输出不存在。

* 修改读者：

判断输入的 id 在不在 readers 中：若在则输入修改的新信息（若相应修改框不输入则保持原信息）；若不在则弹出不存在这个人。

2.2.5 图书/读者信息查询

通过图书的书名/作者/出版社/ISBN 查询图书的所有信息，读者的名字/id/联系方式查询读者的所有信息。

通过判断输入的信息是否在图书/读者中存在，若存在则使用 for (k, v) in j.__dict__.items() 将相关信息 append 进 tmp 列表，然后输出列表。

2.2.6 库存查询

将类属性的 number 加上所有读者借阅（lend）中的该图书的数量即得到该图书的库存。

2.2.7 借出 & 归还书

借书功能时先输出可以借阅的书目。

输入需要借/还的书名和读者名，进行判断：这本书是否在书库信息中，读者名是否在读者信息中：若至少一项不满足则报出相关错误。都满足时：

* 对于借书：

先将 lend 字典所有借书的本数相加，若读者 status 为 1（会员），本书等于 8 则拒绝借书；status 为 0（非会员），本书等于 4 则拒绝借书。

若可以借书则再判断：若本数为 0 则输出可借阅本书不够；大于 0 则 lend 相应书的值加 1（没有则赋值该书为 1），lent 同。然后可借阅本书减 1，并输出现在可借阅本数。

* 对于还书：

判断这个人的 lend 中借没借过这本书：没有则输出没借过；借过则 lend 相应值减 1，number 加 1，lent 作为借阅历史不变。

2.2.8 主函数

创建 books 和 readers 两个列表，打开读取的文件地址（读者和书的信息分开存储）将相关信息读取加入类进列表。

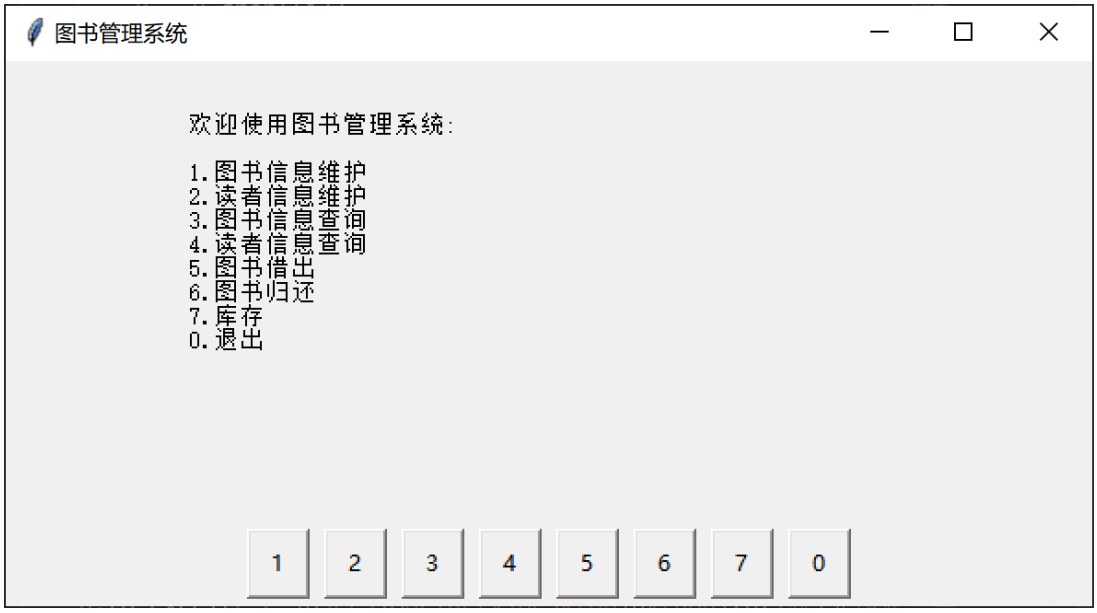
通过 ifelse 判断选择功能。

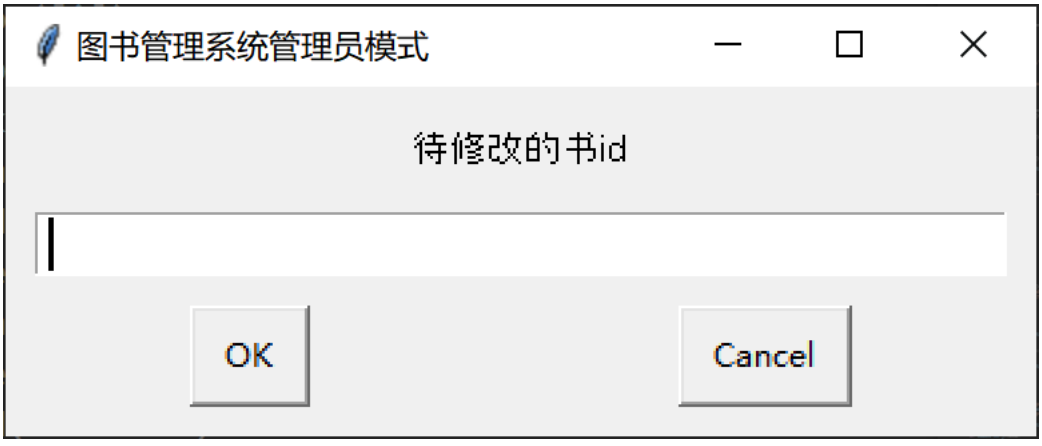
最后对于需要关闭系统时，将 books 和 readers 的内容写入两个文件，保存，方便下一次启动时自动读取上一次保存的信息。

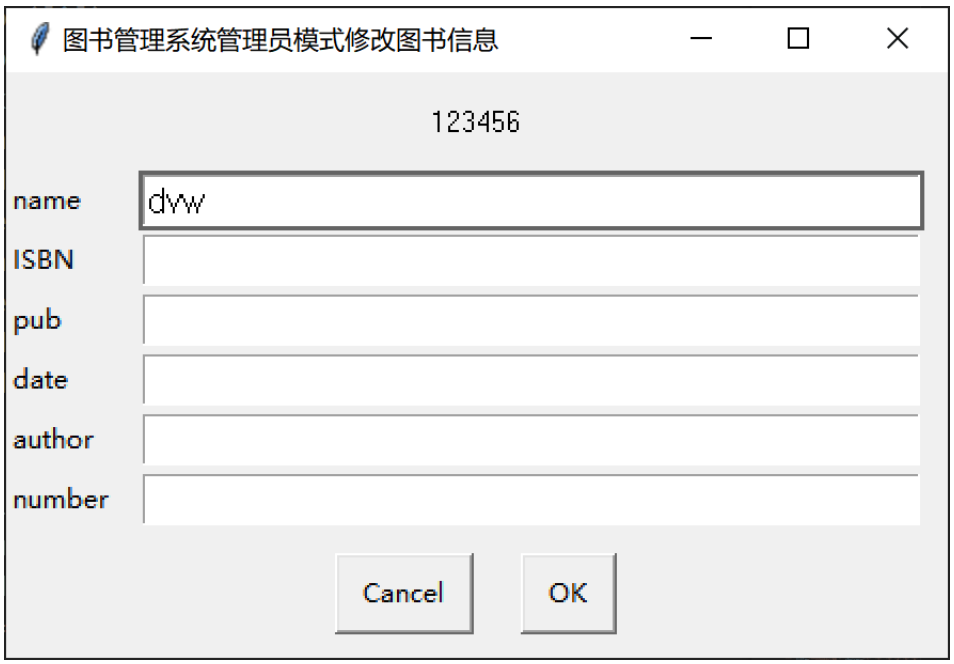
3 样例

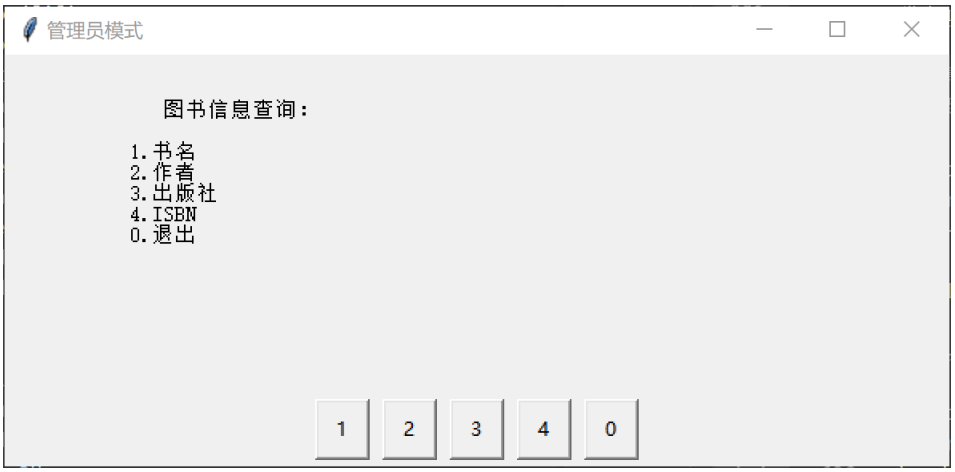
鉴于已经在助教上机检查中全部通过，故只选取一些界面截图展示：

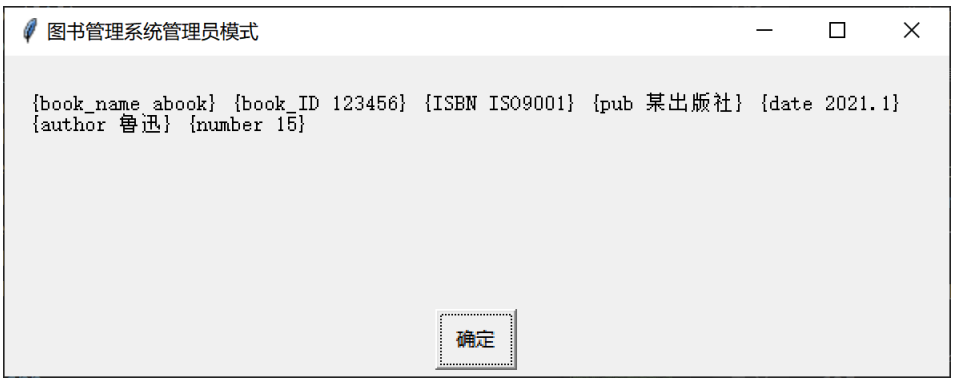

图书管理系统
管理员登陆
用户名
tesla606
密码
******
Cancel
OK


图书管理系统
欢迎使用图书管理系统:
1.图书信息维护
2.读者信息维护
3.图书信息查询
4.读者信息查询
5.图书借出
6.图书归还
7.库存
0.退出
1
2
3
4
5
6
7
0


图书管理系统管理员模式
待修改的书id
OK
Cancel


图书管理系统管理员模式修改图书信息
123456
name
dvw
ISBN
pub
date
author
number
Cancel
OK


管理员模式
图书信息查询：
1.书名
2.作者
3.出版社
4.ISBN
0.退出
1
2
3
4
0


图书管理系统管理员模式
{book_name abook} {book_ID 123456} {ISBN ISO9001} {pub 某出版社} {date 2021.1}
{author 鲁迅} {number 15}
确定