**文件1: library\_data\_entry\_excel.py**

#-\*- coding: utf8 -\*-

import urllib

import urllib.request

import xml.etree.ElementTree as ET

def get\_text(data, start, end):

if end - start<1 or start<100 or end<100 or data.find("数据库里没有这条请求记录.")>0 or data.find("<!-- filename: short-2-head -->")>0 or data[start:end] == " ":

text = "NA"

else:

text = data[start:end]

return text

def htmldecoding(text):

text = text.replace("&nbsp;"," ")

text = text.replace("&quot;",'"')

return text

def getinfo\_douban(isbn, number, position):

#豆瓣API

data = [int(isbn), "《NA》", "NA", "NA", "NA", "NA", "NA", "NA", int(number), "NA", "NA", "豆瓣但无信息", None, None, position]

#[isbn, "《"+ title + "》", author, publisher, pubdate, pages, price, subject, number, callnumber, summary, "来源"]

try:

r = urllib.request.urlopen("http://api.douban.com/book/subject/isbn/" + isbn)

f = open('book.xml','wb')

f.write(r.read())

f.close()

except Exception:

return data

try:

tree = ET.parse('book.xml')

root = tree.getroot()

except Exception:

return data

attributes = root.findall('{http://www.douban.com/xmlns/}attribute')

for attribute in attributes:

if attribute.attrib["name"] == "pages":

data[5] = attribute.text

if attribute.attrib["name"] == "author":

data[2] = attribute.text

if attribute.attrib["name"] == "price":

data[6] = attribute.text

if attribute.attrib["name"] == "publisher":

data[3] = attribute.text

if attribute.attrib["name"] == "pubdate":

data[4] = attribute.text

try:

data[1] = u"《" + root.find('./{http://www.w3.org/2005/Atom}title').text + u"》"

data[11] = "豆瓣"

data[10] = root.find('./{http://www.w3.org/2005/Atom}summary').text

except Exception:

return data

return data

def getinfo\_guotu1(isbn, number, position):

#中国国家图书馆API-key

apikey = urllib.request.urlopen("http://opac.nlc.cn/F/")

url = str(apikey.read())

apikey.close()

key\_start = url.find('<META HTTP-EQUIV="REFRESH" CONTENT="1200; URL=http://opac.nlc.cn:80/F/')+70

key\_end = key\_start + 56

key = url[key\_start:key\_end]

#中国国家图书馆API

website = urllib.request.urlopen("http://opac.nlc.cn/F/" + key + "?func=find-b&find\_code=ISB&request=" + isbn)

data = str(website.read(), encoding = "utf-8")

website.close()

#索书号

callnumber\_start = data.find('CALL-NO: ')+9

callnumber\_end = data[callnumber\_start:].find('DOC-NUMBER')+callnumber\_start-4

callnumber = htmldecoding(get\_text(data, callnumber\_start, callnumber\_end))

#标题

title\_start1 = data.find("题名与责任")+62

title\_start2 = data[title\_start1:].find(";'>")+title\_start1+3

title\_end = data[title\_start2:].find('[')+title\_start2-6

title = htmldecoding(get\_text(data, title\_start2, title\_end))

#内容简介

summary\_start = data.find("内容提要")+75

summary\_end = data[summary\_start:].find('</td>')+summary\_start-3

summary = htmldecoding(get\_text(data, summary\_start, summary\_end))

#页数

pages\_start = data.find('载体形态项')+76

pages\_end = data[pages\_start:].find('&nbsp')+pages\_start

pages = htmldecoding(get\_text(data, pages\_start, pages\_end))

#作者

author\_start1 = data.find("题名与责任")+62

author\_start2 = data[author\_start1:].find("&nbsp;/&nbsp;")+author\_start1+13

author\_end = data[author\_start2:].find('</A>')+author\_start2

if data[author\_start1:].find("&nbsp;/&nbsp;") == -1:

author\_start2 = data.find("AUTHOR: ")+8

author\_end = data[author\_start2:].find(' IMPRINT')+author\_start2-1

author = htmldecoding(get\_text(data, author\_start2, author\_end))

#价格

price\_start1 = data.find('ISBN: ')+6

price\_start2 = data[price\_start1:].find(' ')+price\_start1+1

price\_end = data[price\_start2:].find('TITLE:')+price\_start2-3

price = htmldecoding(get\_text(data, price\_start2, price\_end))

if price.find(" ")>0:

price = price[price.find(" ")+1:]

#出版社

publisher\_start1 = data.find('出版项')

publisher\_start2 = data[publisher\_start1:].find("&nbsp;:&nbsp;")+13+publisher\_start1

publisher\_end = data[publisher\_start2:].find(',&nbsp;')+publisher\_start2

publisher = htmldecoding(get\_text(data, publisher\_start2, publisher\_end))

#出版日期

pubdate\_start1 = data.find('出版项')

pubdate\_start2 = data[pubdate\_start1:].find(",&nbsp;")+7+pubdate\_start1

pubdate\_end = data[pubdate\_start2:].find('</A>')+pubdate\_start2

pubdate = htmldecoding(get\_text(data, pubdate\_start2, pubdate\_end))

if pubdate[-1] == ")":

pubdate = pubdate[:-1]

#主题

subject\_start1 = data.find("主题")

subject\_start2 = subject\_start1 + data[subject\_start1+6:].find("主题")

subject\_start3 = subject\_start2 + data[subject\_start2+16:].find("主题")

subject\_start4 = data[subject\_start3:].find(";'>") + subject\_start3 + 3

subject\_end = data[subject\_start4:].find("</A> ") + subject\_start4

subject = htmldecoding(get\_text(data, subject\_start4, subject\_end))

return [int(isbn), "《"+ title + "》", author, publisher, pubdate, pages, price, subject, int(number), callnumber, summary, "国家图书馆1", None, None, position]

def getinfo\_guotu2(isbn, number, position):

#中国国家图书馆API-key

apikey = urllib.request.urlopen("http://ucs.nlc.cn/F/")

url = str(apikey.read())

apikey.close()

key\_start = url.find('<META HTTP-EQUIV="REFRESH" CONTENT="1200; URL=http://ucs.nlc.cn:80/F/')+69

key\_end = key\_start + 57

key = url[key\_start:key\_end]

#中国国家图书馆API

website = urllib.request.urlopen("http://ucs.nlc.cn/F/" + key + "func=find-b&find\_code=ISB&request=" + isbn + "&local\_base=UCS01")

data = str(website.read(), encoding = "utf-8")

website.close()

#索书号

callnumber\_start = data.find('CALL-NO: ')+9

callnumber\_end = data[callnumber\_start:].find('DOC-NUMBER')+callnumber\_start-4

callnumber = htmldecoding(get\_text(data, callnumber\_start, callnumber\_end))

#标题

title\_start1 = data.find("题名与责任")+62

title\_start2 = data[title\_start1:].find(";'>")+title\_start1+3

title\_end = data[title\_start2:].find('[')+title\_start2-6

title = htmldecoding(get\_text(data, title\_start2, title\_end))

#内容简介

summary\_start = data.find("内容提要")+66

summary\_end = data[summary\_start:].find('</td>')+summary\_start-3

summary = htmldecoding(get\_text(data, summary\_start, summary\_end))

#页数

pages\_start = data.find('载体形态项')+67

pages\_end = data[pages\_start:].find('&nbsp')+pages\_start

pages = htmldecoding(get\_text(data, pages\_start, pages\_end))

#作者

author\_start1 = data.find("题名与责任")+62

author\_start2 = data[author\_start1:].find("&nbsp;/&nbsp;")+author\_start1+13

author\_end = data[author\_start2:].find('</A>')+author\_start2

if data[author\_start1:].find("&nbsp;/&nbsp;") == -1:

author\_start2 = data.find("AUTHOR: ")+8

author\_end = data[author\_start2:].find(' IMPRINT')+author\_start2-1

author = htmldecoding(get\_text(data, author\_start2, author\_end))

#价格

price\_start1 = data.find('ISBN: ')+6

price\_start2 = data[price\_start1:].find(' ')+price\_start1+1

price\_end = data[price\_start2:].find('TITLE:')+price\_start2-3

price = htmldecoding(get\_text(data, price\_start2, price\_end))

if price.find(" ")>0:

price = price[price.find(" ")+1:]

#出版社

publisher\_start1 = data.find('出版项')

publisher\_start2 = data[publisher\_start1:].find("&nbsp;:&nbsp;")+13+publisher\_start1

publisher\_end = data[publisher\_start2:].find(',&nbsp;')+publisher\_start2

publisher = htmldecoding(get\_text(data, publisher\_start2, publisher\_end))

#出版日期

pubdate\_start1 = data.find('出版项')

pubdate\_start2 = data[pubdate\_start1:].find(",&nbsp;")+7+pubdate\_start1

pubdate\_end = data[pubdate\_start2:].find('</A>')+pubdate\_start2

pubdate = htmldecoding(get\_text(data, pubdate\_start2, pubdate\_end))

if pubdate[-1] == ")":

pubdate = pubdate[:-1]

#主题

subject\_start1 = data.find("主题")

subject\_start2 = subject\_start1 + data[subject\_start1+6:].find("主题")

subject\_start3 = subject\_start2 + data[subject\_start2+16:].find("主题")

subject\_start4 = data[subject\_start3:].find(";'>") + subject\_start3 + 3

subject\_end = data[subject\_start4:].find("</A> ") + subject\_start4

subject = htmldecoding(get\_text(data, subject\_start4, subject\_end))

return [int(isbn), "《"+ title + "》", author, publisher, pubdate, pages, price, subject, int(number), callnumber, summary, "国家图书馆2", None, None, position]

def writer(isbn, number, position):

if len(isbn)<5 or number == "" or isbn[:1] == "91" or not isbn.isdigit():

bookinfo = [int(isbn), "WRONG ISBN", None, None, None, None, None, None, None, None, None, "WRONG ISBN", None, None, position]

print (isbn+": WRONG ISBN\n----------------------------")

else:

bookinfo = getinfo\_guotu1(isbn, number, position)

while len(bookinfo[1]) > 200:

bookinfo = getinfo\_guotu1(isbn, number, position)

if bookinfo[1] == "《NA》" and isbn[1] != "1":

bookinfo = getinfo\_guotu2(isbn, number, position)

if bookinfo[1] == "《NA》" and isbn[1] != "1":

bookinfo = getinfo\_douban(isbn, number, position)

print ("ISBN:", bookinfo[0])

print ("书籍名称:", bookinfo[1])

print ("作者:", bookinfo[2])

print ("出版社:", bookinfo[3])

print ("出版日期:", bookinfo[4])

print ("页数:", bookinfo[5])

print ("价格:", bookinfo[6])

print ("主题:", bookinfo[7])

print ("馆藏本数:", bookinfo[8])

print ("索书号:", bookinfo[9])

print ("书籍位置:", bookinfo[14])

print ("内容简介:", bookinfo[10])

print ("信息来源:", bookinfo[11])

print ("============================")

return bookinfo

**文件2: id\_generater.py**

# -\*- coding: utf-8 -\*-

from openpyxl import load\_workbook

reader = "读者信息.xlsx"

header1 = "==============================================================================="

try:

wr2 = load\_workbook(reader)

ws2 = wr2.worksheets[0]

except Exception:

input("请确认“图书馆信息.xlsx”和“读者信息.xlsx”文件保持关闭状态，并与该软件置于同一目录下！")

def reader\_id\_generater():

###自动生成借书号###

print (header1)

grade = {

"一": "1",

"二": "2",

"三": "3",

"四": "4",

"五": "5",

"六": "6",

"七": "7",

"八": "8",

"九": "9"

}

class\_num = {

"年级": "0",

"一": "1",

"二": "2",

"三": "3",

"四": "4",

"五": "5",

"六": "6",

"七": "7",

"八": "8",

"九": "9"

}

nrows = ws2.max\_row

classid\_dic = {}

for row in range(2,nrows+1):

if ws2.cell(row=row,column=4).value != "教师":

reader\_id = ""

classid = ws2.cell(row=row,column=4).value

if classid in classid\_dic.keys():

classid\_dic[classid] += 1

else:

classid\_dic[classid] = 1

count = classid\_dic[classid]

for g in grade:

if classid[0].find(g) != -1:

reader\_id += grade[g]

break

for n in class\_num:

if classid[1:].find(n) != -1:

reader\_id += class\_num[n]

break

if count < 10:

count\_str = "0" + str(count)

else:

count\_str = str(count)

reader\_id += count\_str

ws2.cell(row=row,column=1).value = int(reader\_id)

elif ws2.cell(row=row,column=4).value == "教师":

classid = ws2.cell(row=row,column=4).value

if classid in classid\_dic.keys():

classid\_dic[classid] += 1

else:

classid\_dic[classid] = 1

count = classid\_dic[classid]

ws2.cell(row=row,column=1).value = count

wr2.save(reader)

**文件3: library\_system.py**

# -\*- coding: utf-8 -\*-

from openpyxl import load\_workbook

from openpyxl.workbook import Workbook

import datetime

import json

import copy

from id\_generater import \*

from library\_data\_entry\_excel import \*

library = "图书馆信息.xlsx"

reader = "读者信息.xlsx"

data\_book = {}

data\_reader = {}

header1 = "==============================================================================="

header2 = "-------------------------------------------------------------------------------"

###从两个excel文档和一个json文档中载入初始数据###

try:

wr = load\_workbook(library)

wr2 = load\_workbook(reader)

ws = wr.worksheets[0]

ws2 = wr2.worksheets[0]

ws3 = wr.worksheets[1]

ws4 = wr2.worksheets[1]

except Exception:

input("请确认“图书馆信息.xlsx”和“读者信息.xlsx”文件保持关闭状态，并与该软件置于同一目录下！")

try:

with open('meta\_data.json', "r") as f:

json\_data = json.load(f)

supposed\_return\_days\_students = json\_data["student\_days"]

supposed\_return\_days\_teachers = json\_data["teacher\_days"]

except Exception:

input("请确认“meta\_data.json”文件保持关闭状态，并与该软件置于同一目录下！")

def read\_bookinfo():

###从excel文件读取图书信息###

nrows = ws.max\_row

for book in range(2, nrows+1):

if ws.cell(row=book,column=13).value == None:

borrowed\_times = 0

else:

borrowed\_times = ws.cell(row=book,column=13).value

if not str(ws.cell(row=book,column=1).value).isdigit():

isbn = "0"

else:

isbn = str(int(ws.cell(row=book,column=1).value))

if not str(ws.cell(row=book,column=9).value).isdigit():

number = 0

else:

number = int(ws.cell(row=book,column=9).value)

bookinfo = {

"row": book,

"ISBN": isbn,

"书籍名称": ws.cell(row=book,column=2).value,

"作者": ws.cell(row=book,column=3).value,

"出版社": ws.cell(row=book,column=4).value,

"出版日期": ws.cell(row=book,column=5).value,

"页数": ws.cell(row=book,column=6).value,

"价格": ws.cell(row=book,column=7).value,

"主题": ws.cell(row=book,column=8).value,

"馆藏本数": number,

"索书号": ws.cell(row=book,column=10).value,

"书籍位置": ws.cell(row=book,column=15).value,

"借阅次数": borrowed\_times,

"借阅记录": ws.cell(row=book,column=14).value,

"内容简介": ws.cell(row=book,column=11).value,

"信息来源": ws.cell(row=book,column=12).value

}

data\_book[isbn] = bookinfo

def read\_readerinfo():

###从excel文件读取读者信息###

nrows = ws2.max\_row

for r in range(2, nrows+1):

if ws2.cell(row=r,column=1).value == None:

continue

if ws2.cell(row=r,column=11).value == None:

count = 0

else:

count = ws2.cell(row=r,column=11).value

if ws2.cell(row=r,column=12).value == None:

borrowed\_times = 0

else:

borrowed\_times = ws2.cell(row=r,column=12).value

if not str(ws2.cell(row=r,column=1).value).isdigit():

reader\_id = 0

else:

reader\_id = str(int(ws2.cell(row=r,column=1).value))

readerinfo = {

"row": r,

"借书号": reader\_id,

"姓名": ws2.cell(row=r,column=2).value,

"性别": ws2.cell(row=r,column=3).value,

"单位": ws2.cell(row=r,column=4).value,

"所借书目": ws2.cell(row=r,column=5).value,

"借书日期": ws2.cell(row=r,column=6).value,

"应还日期": ws2.cell(row=r,column=7).value,

"还书日期": ws2.cell(row=r,column=8).value,

"借书记录": ws2.cell(row=r,column=9).value,

"借书权限": ws2.cell(row=r,column=10).value,

"过期次数": count,

"借阅次数": borrowed\_times

}

data\_reader[reader\_id] = readerinfo

def sort\_borrow\_log(d):

###生成借书记录###

log = {}

for reader\_id in d:

request = d[reader\_id]

if request["借阅次数"] == 0:

pass

else:

borrow\_log = request["借书记录"].split(",")

for book in borrow\_log:

if book.find("(") != -1:

log[datetime.datetime.strptime(book[book.find("(")+1:-1], "%Y-%m-%d %H:%M:%S")] = [request["单位"], request["姓名"], request["借书号"], "借书", book[:book.find("(")], data\_book[book[:book.find("(")]]["书籍名称"], data\_book[book[:book.find("(")]]["书籍位置"]]

elif book.find("{") != -1:

log[datetime.datetime.strptime(book[book.find("{")+1:-1], "%Y-%m-%d %H:%M:%S")] = [request["单位"], request["姓名"], request["借书号"], "丢书", book[:book.find("{")], data\_book[book[:book.find("{")]]["书籍名称"], data\_book[book[:book.find("{")]]["书籍位置"]]

elif book.find("[") != -1:

log[datetime.datetime.strptime(book[book.find("[")+1:-1], "%Y-%m-%d %H:%M:%S")] = [request["单位"], request["姓名"], request["借书号"], "还书", book[:book.find("[")], data\_book[book[:book.find("[")]]["书籍名称"], data\_book[book[:book.find("[")]]["书籍位置"]]

items=log.items()

backitems=[[v[0],v[1]] for v in items]

backitems.sort(reverse=True)

return log, [backitems[i][0] for i in range(0,len(backitems))]

def summary():

###统计馆藏本书、注册读者数等信息###

global summary1, summary2, summary3

summary1 = 0

summary2 = 0

summary3 = 0

for book in data\_book:

if data\_book[book]["馆藏本数"] == None:

pass

else:

summary1 += data\_book[book]["馆藏本数"]

summary2 += 1

for reader in data\_reader:

summary3 += 1

def info\_summary (data\_book):

###生成管理信息###

def sort\_value(d):

###排序模块###

items=d.items()

for v in items:

if type(v[0]) == int:

print (v[0],type(v[0]))

backitems=[[v[1]["借阅次数"],v[0]] for v in items]

backitems.sort(reverse=True)

return [backitems[i][1] for i in range(0,len(backitems))]

print (header1)

summary()

print ("图书馆现存图书【"+str(summary2)+"】种，共计图书【"+str(summary1)+"】册，注册读者【"+str(summary3)+"】人。")

print (header2)

print ("最受欢迎的10本图书：")

ranking = 1

for book in sort\_value(data\_book)[:10]:

print ("【"+str(ranking)+"】共借阅"+str(data\_book[book]["借阅次数"])+"次，书籍位置："+str(data\_book[book]["书籍位置"])+"，书籍名称："+str(data\_book[book]["书籍名称"]))

ranking += 1

print (header2)

print ("最勤奋的10位读者：")

ranking = 1

for r in sort\_value(data\_reader)[:10]:

print ("【"+str(ranking)+"】共借阅"+str(data\_reader[r]["借阅次数"])+"次，单位："+str(data\_reader[r]["单位"])+"，姓名："+str(data\_reader[r]["姓名"]))

ranking += 1

print (header2)

print ("过期未还书的读者：")

for reader in data\_reader:

if data\_reader[reader]["应还日期"] == None:

pass

elif data\_reader[reader]["应还日期"]<datetime.datetime.now():

print (data\_reader[reader]["借书号"], data\_reader[reader]["单位"], data\_reader[reader]["姓名"], data\_reader[reader]["所借书目"], data\_book[data\_reader[reader]["所借书目"]]["书籍名称"], "应还日期:", data\_reader[reader]["应还日期"].strftime('%Y-%m-%d'))

print (header2)

print ("今日借阅记录（查看详细借阅记录请打开“读者信息.xlsx”文件查询）：")

borrow\_log, sorted\_log = sort\_borrow\_log(data\_reader)

r = 2

for log in sorted\_log:

if log.date() == datetime.datetime.now().date():

print (log, borrow\_log[log][0], borrow\_log[log][1], borrow\_log[log][2], borrow\_log[log][3], borrow\_log[log][4], borrow\_log[log][5], borrow\_log[log][6])

ws4.cell(row=r,column=1).value = log

ws4.cell(row=r,column=2).value = borrow\_log[log][0]

ws4.cell(row=r,column=3).value = borrow\_log[log][1]

ws4.cell(row=r,column=4).value = borrow\_log[log][2]

ws4.cell(row=r,column=5).value = borrow\_log[log][3]

ws4.cell(row=r,column=6).value = borrow\_log[log][4]

ws4.cell(row=r,column=7).value = borrow\_log[log][5]

ws4.cell(row=r,column=8).value = borrow\_log[log][6]

r += 1

wr2.save("读者信息.xlsx")

print (header1)

input ("点击回车键确认退出。")

def input\_request (instruction):

###信息输入###

content = input(instruction)

while content == "":

content = input(instruction)

return content

def borrow\_book():

###借书模块###

print (header1)

readerid = input\_request("请输入读者【借书号】，退出借书请按【0】\n读者借书号:")

while readerid != "0":

try:

request\_reader = data\_reader[readerid]

print\_request\_reader(request\_reader, False)

if request\_reader["借书权限"] != "开通":

print (header2+"\n【该读者暂无借书权限！】")

input(header2+"\n点击回车键确认退出借书。")

return

if request\_reader["所借书目"] != None:

print (header2+"\n【请将已借书目归还后再借新书！】")

input(header2+"\n点击回车键确认退出借书。")

return

print (header2)

isbn = input\_request("请输入欲借书目【ISBN】，退出借书请按【0】\nISBN:")

if isbn == "0":

return

while isbn != "0":

try:

request\_book = data\_book[isbn]

print\_request\_book(request\_book, False)

print (header2)

confirm = input\_request("是否确认借书？确认请按【1】，取消请按【0】")

while confirm != "1" and confirm != "0":

confirm = input\_request("是否确认借书？确认请按【1】，取消请按【0】")

if confirm == "1":

#修改读者信息

ws2.cell(row=request\_reader["row"],column=5).value = isbn

#借书时间计算

now = datetime.datetime.now()

if request\_reader["单位"] == "教师":

return\_time = now + datetime.timedelta(days=int(supposed\_return\_days\_teachers))

else:

return\_time = now + datetime.timedelta(days=int(supposed\_return\_days\_students))

now\_str = now.strftime('%Y-%m-%d %H:%M:%S')

ws2.cell(row=request\_reader["row"],column=6).value = now

ws2.cell(row=request\_reader["row"],column=7).value = return\_time

if ws2.cell(row=request\_reader["row"],column=9).value == None:

ws2.cell(row=request\_reader["row"],column=9).value = isbn + "(" + now\_str + ")"

else:

ws2.cell(row=request\_reader["row"],column=9).value = request\_reader["借书记录"] + "," + isbn + "(" + now\_str + ")"

if ws2.cell(row=request\_reader["row"],column=12).value == None:

ws2.cell(row=request\_reader["row"],column=12).value = 1

else:

ws2.cell(row=request\_reader["row"],column=12).value += 1

wr2.save(reader)

#修改图书信息

if ws.cell(row=request\_book["row"],column=13).value == None:

ws.cell(row=request\_book["row"],column=13).value = 1

else:

ws.cell(row=request\_book["row"],column=13).value += 1

if ws.cell(row=request\_book["row"],column=14).value == None:

ws.cell(row=request\_book["row"],column=14).value = "(" + now\_str + ")" + request\_reader["借书号"] + request\_reader["单位"] + request\_reader["姓名"] + "[借书]"

else:

ws.cell(row=request\_book["row"],column=14).value += "\n " + "(" + now\_str + ")" + request\_reader["借书号"] + request\_reader["单位"] + request\_reader["姓名"] + "[借书]"

wr.save(library)

read\_bookinfo()

read\_readerinfo()

print ("\n【借书成功！】")

input(header2+"\n点击回车键确认退出借书。")

return

else:

return

except Exception:

print ("\n【ISBN不存在，请重新输入！】")

print (header2)

isbn = input\_request("请输入欲借书目【ISBN】，退出借书请按【0】\nISBN:")

if isbn == "0":

return

except Exception:

print ("\n【读者借书号不存在，请重新输入！】")

print (header2)

readerid = input\_request("请输入读者【借书号】，退出借书请按【0】\n读者借书号:")

def return\_book():

###还书模块###

print (header1)

readerid = input\_request("请输入读者【借书号】，退出还书请按【0】\n读者借书号:")

while readerid != "0":

try:

request\_reader = data\_reader[readerid]

print\_request\_reader(request\_reader, False)

if request\_reader["所借书目"] == None:

print (header2+"\n【该读者暂无借阅书目！】")

input(header2+"\n点击回车键确认退出。")

return

print (header2)

isbn = input\_request("请输入欲还书目【ISBN】，书目丢失请输入【9999】，退出还书请按【0】\nISBN/9999:")

while isbn != request\_reader["所借书目"]:

if isbn == "0":

return

if isbn == "9999":

borrowed\_book = request\_reader["所借书目"]

request\_book = data\_book[borrowed\_book]

now = datetime.datetime.now()

now\_str = now.strftime('%Y-%m-%d %H:%M:%S')

#修改读者信息

ws2.cell(row=request\_reader["row"],column=9).value = request\_reader["借书记录"] + "," + borrowed\_book + "{" + now\_str + "}"

ws2.cell(row=request\_reader["row"],column=10).value = "丢失"

ws2.cell(row=request\_reader["row"],column=5).value = None

ws2.cell(row=request\_reader["row"],column=6).value = None

ws2.cell(row=request\_reader["row"],column=7).value = None

ws2.cell(row=request\_reader["row"],column=8).value = now

wr2.save(reader)

#添加丢书读者信息

nrows = ws3.max\_row

ws3.cell(row=nrows+1,column=1).value = request\_reader["单位"]

ws3.cell(row=nrows+1,column=2).value = request\_reader["姓名"]

ws3.cell(row=nrows+1,column=3).value = request\_reader["性别"]

ws3.cell(row=nrows+1,column=4).value = request\_book["ISBN"]

ws3.cell(row=nrows+1,column=5).value = request\_book["书籍名称"]

ws3.cell(row=nrows+1,column=6).value = request\_book["价格"]

ws3.cell(row=nrows+1,column=7).value = now

#修改图书信息

ws.cell(row=request\_book["row"],column=9).value -= 1

ws.cell(row=request\_book["row"],column=14).value += "\n " + "(" + now\_str + ")" + request\_reader["借书号"] + request\_reader["单位"] + request\_reader["姓名"] + "[丢失]"

wr.save(library)

read\_bookinfo()

read\_readerinfo()

print (header2+"\n【读者借书权限已经关闭！】\n请赔偿所丢书目" + data\_book[request\_reader["所借书目"]]["书籍名称"] + "双倍价格！书籍价格为" + data\_book[request\_reader["所借书目"]]["价格"])

input(header2+"\n点击回车键确认退出。")

return

isbn = input\_request(header2+"\n【欲还书目与所借书目不符】，请输入欲还书目【ISBN】，书目丢失请输入【9999】，退出还书请按【0】\nISBN/9999:")

confirm = input\_request("是否确认还书？确认请按【1】，取消请按【0】")

while confirm != "1" and confirm != "0":

confirm = input\_request("是否确认还书？确认请按【1】，取消请按【0】")

if confirm == "1":

#还书时间计算

borrowed\_book = request\_reader["所借书目"]

request\_book = data\_book[borrowed\_book]

now = datetime.datetime.now()

now\_str = now.strftime('%Y-%m-%d %H:%M:%S')

supposed\_time = ws2.cell(row=request\_reader["row"],column=7).value

delta = supposed\_time - now

if delta.days < 0:

print ("【过期还书！过期次数加一！】")

overdue\_times = request\_reader["过期次数"] + 1

ws2.cell(row=request\_reader["row"],column=11).value = overdue\_times

if overdue\_times >= 2:

print ("【目前过期次数为" + str(overdue\_times) + "次，借书权限暂停！】")

ws2.cell(row=request\_reader["row"],column=10).value = "暂停"

else:

print ("【目前过期次数为" + str(overdue\_times) + "次，请按时还书！】")

ws2.cell(row=request\_reader["row"],column=9).value = request\_reader["借书记录"] + "," + isbn + "[" + now\_str + "]"

ws2.cell(row=request\_reader["row"],column=5).value = None

ws2.cell(row=request\_reader["row"],column=6).value = None

ws2.cell(row=request\_reader["row"],column=7).value = None

ws2.cell(row=request\_reader["row"],column=8).value = now

wr2.save(reader)

ws.cell(row=request\_book["row"],column=14).value += "\n " + "(" + now\_str + ")" + request\_reader["借书号"] + request\_reader["单位"] + request\_reader["姓名"] + "[还书]"

wr.save(library)

read\_bookinfo()

read\_readerinfo()

print ("\n【还书成功！】")

input(header2+"\n点击回车键确认退出。")

readerid = "0"

else:

readerid = "0"

except Exception:

print ("\n【读者借书号不存在，请重新输入！】")

print (header2)

readerid = input\_request("请输入读者【借书号】进行查询，退出还书请按【0】\n读者借书号:")

def print\_request\_book(request, all\_or\_not):

###打印书籍信息###

available\_number = request["馆藏本数"]

history = request["借阅记录"]

if history == None:

pass

else:

while history.find("[") != -1:

status = history[history.find("[")+1:history.find("[")+3]

if status == "借书":

available\_number -= 1

elif status == "还书":

available\_number += 1

history = history[history.find("[")+5:]

print ("书籍名称:", request["书籍名称"])

print ("作者:", request["作者"])

print ("出版社:", request["出版社"])

print ("价格:", request["价格"])

print ("馆藏本数:", request["馆藏本数"])

print ("可借本书:", available\_number)

print ("书籍位置:", "【"+str(request["书籍位置"])+"】")

if all\_or\_not == False:

if request["借阅次数"] == 0:

print ("借阅记录:","该书借阅次数:", request["借阅次数"])

else:

borrow\_log = request["借阅记录"]

log\_table = []

while borrow\_log.find("(") != -1:

log\_table.append(borrow\_log[borrow\_log.find("("):borrow\_log.find("]")+1])

borrow\_log = borrow\_log[borrow\_log.find("]")+1:]

print ("借阅记录:","该书借阅次数:", request["借阅次数"])

if len(log\_table) >= 5:

print (" 【早期记录已省略，仅显示最近5条记录，若要查询完整记录请进入管理员模式。】")

for log in log\_table[-5:]:

print (" ", log)

else:

for log in log\_table:

print (" ", log)

else:

print ("借阅记录:", request["借阅记录"])

def request\_book():

###返回检索书籍信息###

print (header1)

isbn = input\_request("请输入【ISBN】进行查询，退出请按【0】\nISBN:")

while isbn != "0":

try:

request = data\_book[isbn]

print\_request\_book(request, False)

isbn = input\_request(header2+"\n请输入【ISBN】进行查询，退出请按【0】\nISBN:")

except Exception:

print ("【ISBN或借阅记录中的读者借书号不存在，请重新输入！】")

isbn = input\_request(header2+"\n请输入【ISBN】进行查询，退出请按【0】\nISBN:")

def print\_request\_reader(request, all\_or\_not):

###打印读者信息###

print ("姓名:", request["姓名"])

print ("性别:", request["性别"])

print ("单位:", request["单位"])

if request["所借书目"] == None:

print ("所借书目:", "暂无所借书目")

else:

print ("所借书目:", request["所借书目"], data\_book[request["所借书目"]]["书籍名称"])

if request["借书日期"] == None:

print ("借书日期:")

print ("应还日期:")

else:

print ("借书日期:", request["借书日期"].strftime('%Y-%m-%d %H:%M:%S'))

print ("应还日期:", request["应还日期"].strftime('%Y-%m-%d %H:%M:%S'))

if request["还书日期"] == None:

print ("还书日期:")

else:

print ("还书日期:", request["还书日期"].strftime('%Y-%m-%d %H:%M:%S'))

if request["借阅次数"] == 0:

print ("借书记录:", "无借书记录")

else:

borrow\_log = request["借书记录"].split(",")

print ("借书记录:", "曾借阅图书" + str(request["借阅次数"]) + "本：")

if all\_or\_not == False:

log\_table = []

for book in borrow\_log:

if book.find("(") != -1:

log\_table.append(" "+book[book.find("(")+1:-1]+" 借书 "+data\_book[book[:book.find("(")]]["书籍名称"]+" 【"+str(data\_book[book[:book.find("(")]]["书籍位置"])+"】")

elif book.find("{") != -1:

log\_table.append(" "+book[book.find("{")+1:-1]+" 丢失 "+data\_book[book[:book.find("{")]]["书籍名称"]+" 【"+str(data\_book[book[:book.find("{")]]["书籍位置"])+"】")

elif book.find("[") != -1:

log\_table.append(" "+book[book.find("[")+1:-1]+" 还书 "+data\_book[book[:book.find("[")]]["书籍名称"]+" 【"+str(data\_book[book[:book.find("[")]]["书籍位置"])+"】")

if len(log\_table) >= 5:

print (" 【早期记录已省略，仅显示最近5条记录，若要查询完整记录请进入管理员模式。】")

for log in log\_table[-5:]:

print (log)

else:

for log in log\_table:

print (log)

else:

for book in borrow\_log:

if book.find("(") != -1:

print (" "+book[book.find("(")+1:-1], "借书", data\_book[book[:book.find("(")]]["书籍名称"], "【"+str(data\_book[book[:book.find("(")]]["书籍位置"])+"】")

elif book.find("{") != -1:

print (" "+book[book.find("{")+1:-1], "丢失", data\_book[book[:book.find("{")]]["书籍名称"], "【"+str(data\_book[book[:book.find("{")]]["书籍位置"])+"】")

else:

print (" "+book[book.find("[")+1:-1], "还书", data\_book[book[:book.find("[")]]["书籍名称"], "【"+str(data\_book[book[:book.find("[")]]["书籍位置"])+"】")

print ("借书权限:", request["借书权限"])

print ("过期次数:", request["过期次数"])

def request\_reader():

###返回检索读者信息###

print (header1)

readerid = input\_request("请输入读者【借书号】进行查询，退出请按【0】\n读者借书号:")

while readerid != "0":

try:

request = data\_reader[readerid]

print\_request\_reader(request, False)

readerid = input\_request(header2+"\n请输入读者【借书号】进行查询，退出请按【0】\n读者借书号:")

except Exception:

print ("【读者借书号或借阅记录中的书目不存在，请重新输入！】")

readerid = input\_request(header2+"\n请输入读者【借书号】进行查询，退出请按【0】\n读者借书号:")

def reader\_access\_revise():

###修改读者借书权限###

print (header1)

readerid = input\_request("请输入读者【借书号】，退出请按【0】\n读者借书号:")

while readerid != "0":

try:

request\_reader = data\_reader[readerid]

print\_request\_reader(request\_reader, False)

request = input\_request(header2+"\n开通读者借书权限请按【1】\n暂停读者借书权限请按【2】\n重置读者过期次数请按【3】\n退出请按【0】\n")

while request != "0":

if request == "1":

ws2.cell(row=request\_reader["row"],column=10).value = "开通"

wr2.save(reader)

read\_readerinfo()

print ("【" + request\_reader["单位"]+request\_reader["姓名"]+"借书权限开通成功！】")

request = input\_request(header2+"\n开通读者借书权限请按【1】\n暂停读者借书权限请按【2】\n重置读者过期次数请按【3】\n退出请按【0】\n")

elif request == "2":

ws2.cell(row=request\_reader["row"],column=10).value = "暂停"

wr2.save(reader)

read\_readerinfo()

print ("【" + request\_reader["单位"]+request\_reader["姓名"]+"借书权限暂停成功！】")

request = input\_request(header2+"\n开通读者借书权限请按【1】\n暂停读者借书权限请按【2】\n重置读者过期次数请按【3】\n退出请按【0】\n")

elif request == "3":

ws2.cell(row=request\_reader["row"],column=11).value = None

wr2.save(reader)

read\_readerinfo()

print ("【" + request\_reader["单位"]+request\_reader["姓名"]+"过期次数重置成功！】")

request = input\_request(header2+"\n开通读者借书权限请按【1】\n暂停读者借书权限请按【2】\n重置读者过期次数请按【3】\n退出请按【0】\n")

else:

print ("\n【错误代码，请重新输！】")

request = input\_request(header2+"\n开通读者借书权限请按【1】\n暂停读者借书权限请按【2】\n重置读者过期次数请按【3】\n退出请按【0】\n")

return

except Exception:

print ("\n【读者借书号不存在，请重新输入！】\n"+header2)

readerid = input\_request("请输入读者【借书号】，退出还按【0】\n读者借书号:")

def book\_info\_entry\_single():

###单本录入图书信息###

global wr,ws

print(header1+"\n请输入书籍ISBN、书籍本数、书籍位置，退出请输入【0】：")

isbn = input("ISBN：")

if isbn == "0":

return

number = input("本数：")

if number == "0":

return

if not number.isdigit():

number = "00"

position = input("位置：")

if position == "0":

return

while isbn != "0" and number != "0":

print ("----------------------------")

print ("【正在自动生成书籍信息，请稍后！】\n"+header2)

bookinfo = writer(isbn, number, position)

if str(bookinfo[0]) in data\_book.keys() :

print ("【书目已存在于数据库中】，信息如下：")

print ("书籍名称:", data\_book[str(bookinfo[0])]["书籍名称"])

print ("馆藏本数:", data\_book[str(bookinfo[0])]["馆藏本数"])

print ("书籍位置:", data\_book[str(bookinfo[0])]["书籍位置"])

print ("============================")

confirm = input\_request("【是否与现存书目信息合并】？\n现存书目将增加【"+str(bookinfo[8])+"】本，书籍位置依然为【"+str(data\_book[str(bookinfo[0])]["书籍位置"])+"】？\n确认请按【1】，取消请按【0】")

while confirm != "1" and confirm != "0":

confirm = input\_request("确认请按【1】，取消请按【0】")

if confirm == "1":

excel\_to\_json()

ws.cell(row=data\_book[str(bookinfo[0])]["row"],column=9).value += bookinfo[8]

wr.save(library)

wr = load\_workbook(library)

ws = wr.worksheets[0]

read\_bookinfo()

print ("【信息录入完成！】")

else:

excel\_to\_json()

existed = ws.max\_row

for i in range(15):

ws.cell(row=existed+1,column=i+1).value = bookinfo[i]

wr.save(library)

wr = load\_workbook(library)

ws = wr.worksheets[0]

read\_bookinfo()

print ("【信息录入完成！】")

print(header2+"\n请输入书籍ISBN、书籍本数、书籍位置，退出请输入【0】：")

isbn = input("ISBN：")

if isbn == "0":

return

number = input("本数：")

if number == "0":

return

if not number.isdigit():

number = "00"

position = input("位置：")

if position == "0":

return

def reader\_reset\_batch():

###重置读者信息###

nrows = ws2.max\_row

for row in range(2,nrows+1):

ws2.cell(row=row,column=5).value = None

ws2.cell(row=row,column=6).value = None

ws2.cell(row=row,column=7).value = None

ws2.cell(row=row,column=8).value = None

ws2.cell(row=row,column=9).value = None

ws2.cell(row=row,column=10).value = "开通"

ws2.cell(row=row,column=11).value = None

ws2.cell(row=row,column=12).value = None

wr2.save(reader)

def books\_reset\_batch():

###重置图书信息###

nrows = ws.max\_row

for row in range(2,nrows+1):

ws.cell(row=row,column=13).value = None

ws.cell(row=row,column=14).value = None

wr.save(library)

def admin():

###管理员模块###

global pw\_admin, wr2, ws2

instruction = "\n【管理员】请按指示进行相关操作：\n修改读者权限请按【1】\n单本录入图书请按【2】\n图书信息重置请按【3】\n读者信息重置请按【4】\n自动生成借书号请按【5】\n查看统计信息请按【6】\n设置还书期限请按【7】\n重置软件请按【8】\n查询书目完整信息请按【9】\n查询读者完整信息请按【10】\n恢复文件请按【11】\n退出请按【0】\n"

print (header1)

password = input("请输入【管理员密码】！退出请按【0】\n密码:")

if password == "0":

return

while password != pw\_admin:

password = input("【密码错误！】\n请输入密码！退出请按【0】\n密码:")

if password == "0":

return

content = input\_request(header1+instruction)

while content != "0":

if content == "1":

excel\_to\_json()

reader\_access\_revise()

content = input\_request(header1+instruction)

elif content == "2":

excel\_to\_json()

book\_info\_entry\_single()

content = input\_request(header1+instruction)

elif content == "3":

print (header1)

confirm = input\_request("该操作将清空所有图书借还书记录，确认请按【1】，取消请按【0】")

while confirm != "1" and confirm != "0":

confirm = input\_request("该操作将清空所有图书借还书记录，确认请按【1】，取消请按【0】")

if confirm == "1":

excel\_to\_json()

books\_reset\_batch()

read\_bookinfo()

print ("【操作完成，图书信息重置成功！】")

content = input\_request(header1+instruction)

elif content == "4":

print (header1)

confirm = input\_request("该操作将清空所有借还书记录，并开通所有读者借书权限，确认请按【1】，取消请按【0】")

while confirm != "1" and confirm != "0":

confirm = input\_request("该操作将清空所有借还书记录，并开通所有读者借书权限，确认请按【1】，取消请按【0】")

if confirm == "1":

excel\_to\_json()

reader\_reset\_batch()

read\_readerinfo()

print ("【操作完成，读者信息重置成功！】")

content = input\_request(header1+instruction)

elif content == "5":

excel\_to\_json()

reader\_id\_generater()

wr2 = load\_workbook(reader)

ws2 = wr2.worksheets[0]

read\_readerinfo()

print ("【操作完成，读者借书号成功生成！】")

content = input\_request(header1+instruction)

elif content == "6":

excel\_to\_json()

info\_summary(data\_book)

content = input\_request(header1+instruction)

elif content == "7":

print (header1)

student\_days = input("【学生】借书期限：")

teacher\_days = input("【教师】借书期限：")

confirm = input\_request("该操作将设置读者借书期限，但【不会】影响图书和读者信息。\n确认请按【1】，取消请按【0】")

while confirm != "1" and confirm != "0":

confirm = input\_request("该操作将设置读者借书期限，但【不会】影响图书和读者信息。\n确认请按【1】，取消请按【0】")

if confirm == "1":

with open('meta\_data.json', "r") as f1:

json\_data = json.load(f1)

json\_data["student\_days"] = student\_days

json\_data["teacher\_days"] = teacher\_days

with open('meta\_data.json', "w") as f2:

json.dump(json\_data,f2)

print ("读者借书期限设置成功！")

content = input\_request(header1+instruction)

elif content == "8":

print (header1)

confirm = input\_request("该操作将重置软件登录信息及密码，但【不会】影响图书和读者信息。\n确认请按【1】，取消请按【0】")

while confirm != "1" and confirm != "0":

confirm = input\_request("该操作将重置软件登录信息及密码，但【不会】影响图书和读者信息。\n确认请按【1】，取消请按【0】")

if confirm == "1":

print (header2)

with open('meta\_data.json', "r") as f1:

json\_data = json.load(f1)

json\_data["status"] = "0"

with open('meta\_data.json', "w") as f2:

json.dump(json\_data,f2)

welcome, pw, pw\_admin = initiallize()

return ("initiallized")

content = input\_request(header1+instruction)

elif content == "9":

print (header1)

isbn = input\_request("请输入【ISBN】进行查询，退出请按【0】\nISBN:")

while isbn != "0":

try:

request = data\_book[isbn]

print\_request\_book(request, True)

isbn = input\_request(header2+"\n请输入【ISBN】进行查询，退出请按【0】\nISBN:")

except Exception:

print ("【ISBN或借阅记录中的读者借书号不存在，请重新输入！】")

isbn = input\_request(header2+"\n请输入【ISBN】进行查询，退出请按【0】\nISBN:")

content = input\_request(header1+instruction)

elif content == "10":

print (header1)

readerid = input\_request("请输入读者【借书号】进行查询，退出请按【0】\n读者借书号:")

while readerid != "0":

try:

request = data\_reader[readerid]

print\_request\_reader(request, True)

readerid = input\_request(header2+"\n请输入读者【借书号】进行查询，退出请按【0】\n读者借书号:")

except Exception:

print ("【读者借书号或借阅记录中的书目不存在，请重新输入！】")

readerid = input\_request(header2+"\n请输入读者【借书号】进行查询，退出请按【0】\n读者借书号:")

content = input\_request(header1+instruction)

elif content == "11":

print (header1)

print ("正在恢复文件中，请稍后...")

json\_to\_excel()

print ("文件恢复完成，请返回根目录，并按文件名中的提示操作。")

input ("点击回车键确认退出。")

content = input\_request(header1+instruction)

else:

print("\n【错误代码，请重新输！】")

content = input\_request(header1+instruction)

def initiallize():

###初始化###

with open('meta\_data.json', "r") as f1:

json\_data = json.load(f1)

if json\_data["status"] == "0":

print ("【软件初始化】，请按提示输入相应内容！")

json\_data["status"] = "1"

json\_data["institution"] = input("【学校/机构名称】：")

json\_data["password"] = input("【登陆密码】：")

json\_data["administrator"] = input("【管理员密码】：")

with open('meta\_data.json', "w") as f2:

json.dump(json\_data,f2)

return (json\_data["institution"], json\_data["password"], json\_data["administrator"])

def excel\_to\_json():

###数据备份模块###

def excel\_to\_json\_reader(file\_name):

convert\_dict = copy.deepcopy(data\_reader)

for reader in convert\_dict:

for key in ["借书日期", "应还日期", "还书日期"]:

if convert\_dict[reader][key] != None:

convert\_dict[reader][key] = convert\_dict[reader][key].strftime('%Y-%m-%d %H:%M:%S')

with open(file\_name + '\_backup.json',"w") as f:

json.dump(convert\_dict,f)

def excel\_to\_json\_library(file\_name):

with open(file\_name + '\_backup.json',"w") as f:

json.dump(data\_book,f)

excel\_to\_json\_reader(reader)

excel\_to\_json\_library(library)

def json\_to\_excel():

###数据还原模块###

def json\_to\_excel\_library(file\_name):

wb = Workbook()

ws0 = wb.worksheets[0]

ws0.title = u"书籍信息"

ws1 = wb.create\_sheet()

ws1.title = u'书籍丢失信息'

title = ["ISBN", "书籍名称", "作者", "出版社", "出版日期", "页数", "价格", "主题", "馆藏本数", "索书号", "内容简介", "信息来源", "借阅次数", "借阅记录", "书籍位置"]

title2 = ["单位", "姓名", "性别", "ISBN", "书名", "价格", "登记时间"]

for i in range(len(title)):

ws0.cell(row=1,column=i+1).value = title[i]

for i in range(len(title2)):

ws1.cell(row=1,column=i+1).value = title2[i]

with open(file\_name, "r") as f:

json\_data = json.load(f)

for book in json\_data:

book\_single = [json\_data[book][key] for key in title]

existed = ws0.max\_row

for i in range(len(title)):

ws0.cell(row=existed+1,column=i+1).value = book\_single[i]

wb.save(file\_name[:file\_name.find(".")]+"（恢复文件，请将括号连带此提示删除并替换损坏文件即可使用）.xlsx")

def json\_to\_excel\_reader(file\_name):

wb = Workbook()

ws0 = wb.worksheets[0]

ws0.title = u"读者信息"

ws1 = wb.create\_sheet()

ws1.title = u'借阅记录'

title = ["借书号", "姓名", "性别", "单位", "所借书目", "借书日期", "应还日期", "还书日期", "借书记录", "借书权限", "过期次数", "借阅次数"]

title2 = ["时间", "单位", "姓名", "借书号", "动作", "ISBN", "书名", "书籍位置"]

for i in range(len(title)):

ws0.cell(row=1,column=i+1).value = title[i]

for i in range(len(title2)):

ws1.cell(row=1,column=i+1).value = title2[i]

with open(file\_name, "r") as f:

json\_data = json.load(f)

for reader in json\_data:

reader\_single = []

for key in title:

if key in ["借书日期", "应还日期", "还书日期"]:

if json\_data[reader][key] != None:

d = datetime.datetime.strptime(json\_data[reader][key], '%Y-%m-%d %H:%M:%S')

else:

d = None

reader\_single.append(d)

else:

reader\_single.append(json\_data[reader][key])

existed = ws0.max\_row

for i in range(len(title)):

ws0.cell(row=existed+1,column=i+1).value = reader\_single[i]

wb.save(file\_name[:file\_name.find(".")]+"（恢复文件，请将括号连带此提示删除并替换损坏文件即可使用）.xlsx")

json\_to\_excel\_library('图书馆信息.xlsx\_backup.json')

json\_to\_excel\_reader('读者信息.xlsx\_backup.json')

def main():

###主界面###

instruction = "\n请按指示进行相关操作：\n借书请按【1】\n还书请按【2】\n查询书目信息请按【3】\n查询读者信息请按【4】\n管理各类信息请按【5】\n帮助请按【6】\n退出请按【0】\n"

print ("请确认“图书馆信息.xlsx”和“读者信息.xlsx”文件保持关闭状态，并与该软件置于同一目录下！")

excel\_to\_json()

print (header1)

print ("欢迎进入"+ welcome + "图书馆管理系统！")

print (header1)

password = input("请输入密码！退出请按【0】\n密码:")

if password == "0":

return

while password != pw:

password = input("\n【密码错误！】\n请输入密码！退出请按0\n密码:")

if password == "0":

return

print (header1)

print ("【密码正确，欢迎使用！】")

print (header2)

summary()

print ("图书馆现存图书【"+str(summary2)+"】种，共计图书【"+str(summary1)+"】册，注册读者【"+str(summary3)+"】人。")

print ("学生借书期限【"+ supposed\_return\_days\_students + "】天，教师借书期限【" + supposed\_return\_days\_teachers + "】天。")

content = input\_request(header1+instruction)

while content != "0":

if content == "1":

excel\_to\_json()

borrow\_book()

content = input\_request(header1+instruction)

elif content == "2":

excel\_to\_json()

return\_book()

content = input\_request(header1+instruction)

elif content == "3":

request\_book()

content = input\_request(header1+instruction)

elif content == "4":

request\_reader()

content = input\_request(header1+instruction)

elif content == "5":

if admin() == "initiallized":

input("软件重置成功，请点击回车键后重新打开软件！")

return

content = input\_request(header1+instruction)

elif content == "6":

print (header2 + "\n感谢使用此套图书管理系统！\n软件作者：宋嘉勋\n联系方式：E-mail:jiaxun.song@outlook.com | Phone:15201410329 | Wechat:360911702")

content = input\_request(header1+instruction)

else:

print ("\n【错误代码，请重新输！】")

content = input\_request(header1+instruction)

welcome, pw, pw\_admin = initiallize()

read\_bookinfo()

read\_readerinfo()

main()