import pickle

F=open("Number of Books.txt","r")

class Book5:

L=[] #List of Book names

N=int(F.read())

F.close()

bookvar={}

def \_\_init\_\_(self,name,genre,summary,author,qty,price,no=N+1):

self.number=no

self.name=name

self.genre=genre

self.summary=summary

self.author=author

self.Qty=qty

self.price=price

self.status=""

Book5.N+=1

F=open("Number of Books.txt","w")

F.write(str(Book5.N))

F.close()

if self.Qty==0:

self.Status="Not available"

else:

self.Status="Available"

def Buy(self):

if self.Status=="Available":

print "The price is",self.price

while True:

ans=raw\_input("Do you want to buy it? (Y/N)")

if ans in ['Y','y','Yes','yes']:

self.Qty-=1

StoreBooks()

if self.Qty==0 or self.Qty<0:

self.Status="Not available"

return 'success'

elif ans in ['N','n','No','no']:

return "\*"

else:

print "Please enter (Yes/No) only"

continue

else:

print "Sorry, the book is currently not available"

return "\*"

def Issue(self):

if self.Status=="Available":

duration=raw\_input("Enter the duration: (minutes)")

if duration in ['cancel','Cancel']:

return 'cancel'

self.Qty-=1

StoreBooks()

print "Book",self.name,"Issued successfully!"

if self.Qty==0 or self.Qty<0:

self.Status="Not available"

return "success"

else:

print "Sorry, the book is currently not available"

def Display(self):

print "\n\n"

print "Name:",self.name

print "Author:",self.author

print "Genre:",self.genre

print "Description:",self.summary

print "\nStatus:",self.status

print "Quantity available:",self.Qty

print "Price:",self.price

def Return(self):

self.Qty+=1

print "Book returned successfully!"

def Setstatus():

for i in range(1,Book5.N+1):

if Book5.bookvar[i].Qty>0:

Book5.bookvar[i].status="Available"

else:

Book5.bookvar[i].status="Not Available"

def Displaylist(): #Displays list of books(catalogue)

for i in range(1,len(Book5.L)+1):

print i,":",Book5.L[i-1]

def DonateBook():

name=raw\_input("Enter the Book's name:")

name=name.title()

qty=raw\_input("Enter quantity of books:")

qty=qty.title()

if name=='Cancel' or qty=='Cancel':

return 'cancel'

qty=int(qty)

l=''

for i in range(1,len(Book5.L)+1):

if name==Book5.bookvar[i].name:

h=raw\_input("Confirm? (Yes/No)")

l="P"

if h in ['Y','y','Yes','yes']:

Book5.bookvar[i].Qty+=qty

StoreBooks()

print "\n\nThank you for your donation!"

break

elif name==Book5.L[i-1]:

break

if l!='P':

genre=raw\_input("Enter its Genre:")

author=raw\_input("Enter its Author:")

qty=int(raw\_input("Enter the quantity of books:"))

price=int(raw\_input("Enter the price of the book:"))

summary=raw\_input("Enter the description of the book:")

B1=Book5(name,genre,summary,author,qty,price,Book5.N+1)

StoreBook(B1)

print "\n\nThank you for you donation!"

Book5.bookvar[Book5.N+1]=B1

Book5.L+=[B1.name]

#Change: Book\_no in Library\_main after tuple

def StoreBook(B):

F=open("B\_ooks.dat","ab")

pickle.dump(B,F)

F.close()

def StoreBooks():

F=open("B\_ooks.dat","wb")

for i in range(1,Book5.N+1):

pickle.dump(Book5.bookvar[i],F)

F.close()

def GetBooks():

F=open("B\_ooks.dat","rb")

try:

D={}

for i in range(1,Book5.N+1):

D[1]=pickle.load(F)

Book5.bookvar[i]=D[1]

F.close()

return Book5.bookvar

except EOFError:

F.close()

return Book5.bookvar

if Book5.N>0:

Book\_no=GetBooks() #the first statement of the program

Book5.bookvar=dict(Book\_no)

Setstatus()

try:

for i in range(1,Book5.N+1):

Book5.L+=[Book\_no[i].name]

except KeyError:

pass

#Library Card module

""" Creates a New Library Card, Retrieves existing Cards and provides/sends them to the main module.

Classes:

LibCard

functions: display()

Issue()

Buy()

Returnbook()

Functions:

Editcard (To t=edit the details of a card (name/comment))

Storecard (To store a newly generated Card)

ChangeStoredcard (To store the information changed in a card)

Getcard (To retrieve cards from data file)

CNumber (To generate a unique card number)

Newcard (To create a new card(LibCard) object)

"""

#---------------------------------------#

import pickle

import datetime

def CNumber(p):

n=p+1

while True:

yield n

n+=1

class LibCard:

with open("Number of Cards.txt","r") as F: #Filechangelocation

N=int(F.read())

def \_\_init\_\_(self,Cno,name,D\_o\_J,Ino,Bno,Comment):

self.Number=Cno

self.Name=name

self.D\_o\_J=D\_o\_J

self.issues=Ino

self.buys=Bno

self.Comments=Comment

self.issued=[]

self.bought=[]

LibCard.N+=1

F=open("Number of Cards.txt","w+") #Filechangelocation

F.write(str(LibCard.N))

F.close()

self.d=0 #Number of displays

def display(self):

print """

+-----------------------------+

| The Reader's Havens""",(30-27)\*' ',"""|

| 2/7, Potsdam, Berlin""",(30-27)\*' ',"""|

| |

|Card Number/ID:""",self.Number,(30-(len(str(self.Number))+19))\*' ',"""|

|Name:""",self.Name,(30-(len(self.Name)+9))\*' ',"""|

|Date of Joining:""",self.D\_o\_J,(30-(len(self.D\_o\_J)+20))\*' ',"""|

|Number of Books issued:""",self.issues,(30-(len(str(self.issues))+27))\*' ',"""|

|Comments:""",self.Comments,(30-(len(self.Comments)+13))\*' ',"""|

+-----------------------------+\n"""

if self.d>0:

print "Previous activities\n"

print "Books issued:\n"

for i in range(0,len(self.issued)):

print '\t',self.issued[i][0],self.issued[i][1],self.issued[i][2],self.issued[i][3],'('+self.issued[i][4]+')'

print "\nBooks bought:\n"

for i in range(0,len(self.bought)):

print '\t',self.bought[i][0],self.bought[i][1],self.bought[i][2],self.bought[i][3]

self.d=1

def Issue(self,name,bn):

self.issues+=1

self.issued.append((str(self.issues)+'.',name,' ['+str(bn)+']',datetime.date.today().strftime("%d-%m-20%y"),"pending"))

ChangeStoredcard(Card)

def Buy(self,name,bn):

self.buys+=1

self.bought.append((str(self.buys)+'.',name,' ['+str(bn)+']',datetime.date.today().strftime("%d-%m-20%y")))

ChangeStoredcard(Card)

def Returnbook(self,name,bn):

for i in range(0,len(self.issued)):

if self.issued[i][1]==name:

if self.issued[i][4]=='pending':

self.issued[i]=(self.issued[i][0],self.issued[i][1],self.issued[i][2],self.issued[i][3],'returned')

return 'success'

else:

print "The book has been already returned"

else:

print 'The given book has not been issued by you yet'

def Newcard():

card\_no=next(NewNo)

Name=raw\_input("Enter your Name:")

D\_o\_J=datetime.date.today().strftime("%d-%m-20%y")

booksissued=0

booksbought=0

Remarks=raw\_input("Enter remarks/comments(optional) (max size 18):")

Card[LibCard.N]=LibCard(card\_no,Name,D\_o\_J,booksissued,booksbought,Remarks)

print "Your Library Card has been created successfully"

print "\nPlease note your card number"

Card[LibCard.N].display()

Storecard(Card[LibCard.N])

def Storecard(card):

F=open("Cards.dat","a+b") #Filechangelocation

pickle.dump(card,F)

F.close()

def Getcard():

Card={}

for i in range(1,LibCard.N+1):

try:

x=pickle.load(F)

Card[i]=x

except EOFError:

F.close()

break

return Card

def Editcard(Card):

while True:

n=int(raw\_input("Enter your card number:"))

if n in range(1,LibCard.N+1):

print "\nSelect you query:"

print "1. Edit your Name"

print "2. Edit Comment"

x=int(raw\_input("Your input: "))

if x==1:

newname=raw\_input("Enter field Name:")

Card[n].Name=newname

print "\n"

print "Your updated card info:"

Card[n].display()

ChangeStoredcard(Card)

return Card #Use this value (in the variable Card)

elif x==2:

newcomment=raw\_input("Enter field Comment:")

Card[n].Comments=newcomment

print "Your updated card info:"

Card[n].display()

ChangeStoredcard(Card)

return Card

else:

print "Given Card Number",n,"is invalid"

continue

def ChangeStoredcard(Card):

with open("Cards.dat","wb") as F:

for i in range(1,LibCard.N+1):

pickle.dump(Card[i],F)

NewNo=CNumber(0)

Card={} #Card : Dictionary with all the card objects

if LibCard.N>0:

F=open("Cards.dat","r+b") #Filechangelocation

Card=Getcard()

F.close()

#The Library.py module

#The program starts and this menu appears

import sys

import LibraryCard

import book3

import pickle

I=0

#-----------------The Main Menu-----------------#

def Menu():

print """\n\n

\t\t\t\tWELCOME TO THE READER's HAVENS

\t\t\t #------------------------------#"""

print "\t\tMENU"

print "\t\tSelect your Option/Choice"

print "\t\t1. View Book Catalogue/Database","\t\t 9. \_\_Search\_\_"

print "\t\t2. Issue a Book","\t\t\t\t ----------"

print "\t\t3. Buy a Book"

print "\t\t4. Return a Book"

print "\n\t\t5. Create your Library Card (New to the library?)"

print "\t\t6. View Member details"

print "\t\t7. Edit Member Details"

print "\n\t\t8. Donate a Book"

print "\n\t\t10. About Us\t\t\t\t\t 11. Exit"

print "\t "

print "\n\n\t\t(Enter 'cancel' to go back to the main menu after selecting an option)"

response=int(raw\_input("\n\n\t\tYour Response: "))

print "\n\n"

return response

#------------------End of Menu------------------#

def BacktoMenu(Message):

while True:

r4=raw\_input(Message)

r4=r4.title()

if r4 in ['Y','Yes']:

return True

elif r4 in ['N','No']:

return False

else:

print "\nPlease enter (Yes/No) only"

continue

Card=LibraryCard.Card #Card : Dictionary with all the card objects

I+=len(Card)

while True:

response=Menu()

#------------------Exit-------------------#

if response==11:

sys.exit() #Exit

#----------------About Us-----------------#

elif response==10:

print """About Us:

This is a Mini-Library (investigatory) project by Tanishq, Shivanshu and Tanmay

of class XII-D of DLF Public School"""

if BacktoMenu("Go back to the menu (Y/N)?"):

continue

else:

sys.exit()

#---------------New Member---------------#

elif response==5:

LibraryCard.Newcard()

I+=1

if BacktoMenu("Do you want to go back to Main Menu?"):

continue

else:

sys.exit()

#---------------View Member Details---------------#

elif response==6:

while True:

n=int(raw\_input("Enter your card number:"))

try:

if str(Card[n]):

pass

except KeyError:

print "Invalid Card number"

if BacktoMenu("Enter again? "):

continue

else:

break

print "\nYour card details:"

Card[n].display()

if BacktoMenu("\nGo back to the main menu (Y/N)?"):

break

else:

sys.exit()

#---------------Edit Member details---------------#

elif response==7: #Edit Card

while True:

Card=LibraryCard.Editcard(Card)

LibraryCard.Card=dict(Card)

if BacktoMenu("Do you want to change something else?"):

continue

else:

if BacktoMenu("Do you want to go back to the menu?"):

break

else:

sys.exit()

#----------------View Books in the Library----------------#

elif response==1:

book3.Displaylist()

while True:

if BacktoMenu("\nView details of any book? (Y/N)"):

n=int(raw\_input("Enter book number:"))

book3.Book5.bookvar[n].Display()

print "\n\n"

if BacktoMenu("View the Book list again?"):

book3.Displaylist()

continue

else:

break

else:

break

if BacktoMenu("Go back to the main menu (Y/N)?"):

continue

else:

sys.exit()

#---------------Issue a Book---------------#

elif response==2:

n=raw\_input("Enter you Card/Member number:")

n=n.title()

while True:

bn=int(raw\_input("Enter the serial number of the Book to be issued (To go to Main Menu and see Book List, Enter 9999)"))

if bn==9999 or n=='Cancel':

break

elif bn in range(0,book3.Book5.N+1):

n=int(n)

p=book3.Book\_no[bn].Issue()

if p=="success":

Card[n].Issue(name=book3.Book5.L[bn-1],bn=bn)

Card[n].issues+=1

p="\*"

break

else:

print "Book not found"

if BacktoMenu("Enter Book serial number again?"):

continue

else:

if BacktoMenu("Go back to Main Menu?"):

bn="Menu"

break

else:

sys.exit()

if bn==9999 or n=='Cancel':

continue

elif p=="\*":

if BacktoMenu("Go back to Main Menu?"):

continue

else:

sys.exit()

#---------------Buy Book---------------#

elif response==3:

n=raw\_input("Enter you Card/Member number:")

n=n.title()

if n=='Cancel':

continue

n=int(n)

while True:

bn=int(raw\_input("Enter the Book serial number you want to buy: (To go to Main Menu and see Book List, Enter 9999)"))

if bn==9999:

break

elif bn in range(0,book3.Book5.N+1):

p=book3.Book5.bookvar[bn].Buy()

if p=='success':

Card[n].Buy(name=book3.Book5.L[bn-1],bn=bn)

print "Book",book3.Book5.bookvar[bn].name,"bought successfully!"

p="\*"

break

else:

print "Book not found"

if BacktoMenu("Enter Book serial number again?"):

continue

else:

if BacktoMenu("Go back to Main Menu?"):

p=9999

break

else:

sys.exit()

if bn==9999:

continue

elif p=="\*":

if BacktoMenu("Go back to Main Menu?"):

continue

else:

sys.exit()

#---------------Return Book---------------#

elif response==4:

n=raw\_input("Enter your Card number/ID:")

n=n.title()

if n=='Cancel':

continue

bn=raw\_input("Enter the book number/ID:")

bn=bn.title()

if bn=='Cancel':

continue

n,bn=int(n),int(bn)

q=Card[n].Returnbook(name=book3.Book5.bookvar[bn].name,bn=bn)

if q=='success':

book3.Book5.bookvar[bn].Return()

if BacktoMenu("Return to the main menu? (Y/N)"):

continue

else:

sys.exit()

#---------------Donate Book---------------#

elif response==8:

r=book3.DonateBook()

if r=='cancel':

continue

book3.Book\_no=book3.Book5.bookvar

#-----------------Search-----------------#

elif response==9:

while True:

bname=raw\_input("Enter the name of the book:")

bname=bname.title()

if bname=='cancel':

break

else:

for i in range(0,len(book3.Book5.bookvar)):

if bname==book3.Book5.L[i]:

book3.Book5.bookvar[i+1].Display()

break

else:

print "Book with that name not found."

if BacktoMenu("Search again for a book?"):

continue

else:

if BacktoMenu("Go back to the main menu?"):

break

else:

sys.exit()

#-----------------END-----------------#