课程名称：Python语言与编程

实验名称：堆栈与队列的实现

1. 实验目的

在python语言中实现栈、队列的功能。

1. 实验内容

1.堆栈。一个堆栈(Stack)是一种具有后进先出(last-in-first-out，LIFO)特性的数据结构。实现：

1）pushstack() 向堆栈中压入一个数据项

2）popstack() 从堆栈中移出一个数据项

3）isempty() 如果堆栈是空的，返回布尔值1,否则返回0

4) peekstack() 取出堆栈顶部的数据项，但并不移除它

2.队列。一个队列(queue)是一种具有先进先出(first-in-first-out，FIFO)特性的数据结构。数据从前端被移除，从后端被加入。实现：

1)enqueue() 在队列的尾部加入一个新的数据

2)dequeue()在队列的头部取出一个数据，返回它并且把它从队列中删除

3)putfile()将队列中数据写入文件中

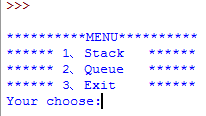
4)getfile()将文件中数据加入队列中

1. 实验环境

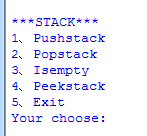
Python2.7.9

1. 实验结果

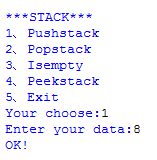
菜单（MENU）：



1、栈(STACK)：

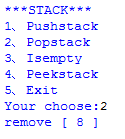
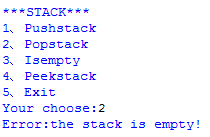


（1）、入栈(pushstack)：



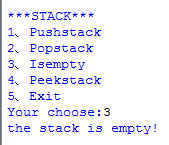
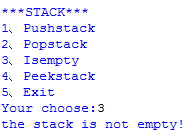
（2）、出栈(popstack)：

①成功： ②失败（栈空）：

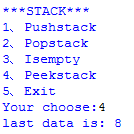
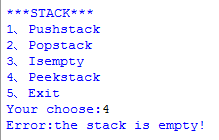
（3）、判断是否为空(isempty)：

①栈空 ②栈不空：

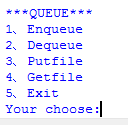
 

（4）、获取顶层数据(peekstack)：

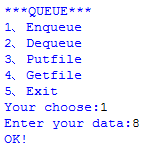
①成功： ②失败（栈空）：

2、队列(QUEUE)：

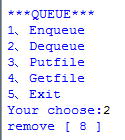
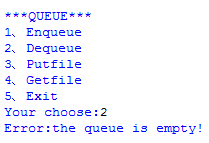


（1）、入队(enqueue)：



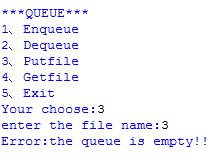
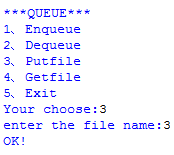
（2）、出队(dequeue)：

①成功： ②失败（队列为空）：

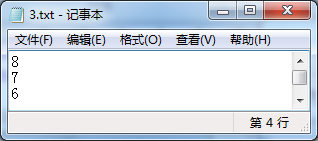
 

（3）、写入文件(putfile)：

①成功： ②失败（队列空）：

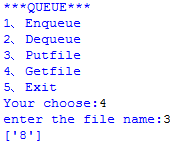
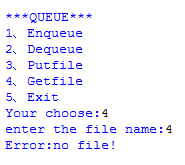
 

③文件存储情况（一行一个数据）：



（4）、读取文件(getfile)：

①成功： ②失败（文件名不存在）：

1. 程序：

import os

class stack():

def \_\_init\_\_(self):

self.stack=[]

def pushstack(self,data):

self.stack.append(data)

def popstack(self):

if len(self.stack)==0:

return None

else:

return self.stack.pop(-1)

def isempty(self):

if len(self.stack)==0:

return True

else:

return False

def peekstack(self):

if len(self.stack)==0:

return None

else:

return self.stack[-1]

class queue():

def \_\_init\_\_(self):

self.queue=[]

def enqueue(self,data):

self.queue.append(data)

def dequeue(self):

if len(self.queue)==0:

return None

else:

return self.queue.pop(0)

def putfile(self,filename):

if len(self.queue)==0:

return False

else:

file\_object=open(filename+'.txt','w')

for i in self.queue:

file\_object.write(i)

file\_object.write('\n')

file\_object.close()

return True

def getfile(self,filename):

if os.path.exists(filename+'.txt'):

file\_object = open(filename+'.txt')

for line in file\_object:

self.enqueue(line.strip('\n'))

file\_object.close()

return True

else:

return False

stack=stack()

queue=queue()

while True:

print '\n\*\*\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*\*\*'

print '\*\*\*\*\*\* 1、Stack \*\*\*\*\*\*'

print '\*\*\*\*\*\* 2、Queue \*\*\*\*\*\*'

print '\*\*\*\*\*\* 3、Exit \*\*\*\*\*\*'

menu=input("Your choose:")

if menu==1:

while True:

print '\n\*\*\*STACK\*\*\*'

print '1、Pushstack'

print '2、Popstack'

print '3、Isempty'

print '4、Peekstack'

print '5、Exit'

choose=input("Your choose:")

if choose==1:

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

stack.pushstack(data)

print 'OK!'

elif choose==2:

char=stack.popstack()

if char:

print 'remove [',char,']'

else:

print 'Error:the stack is empty!'

elif choose==3:

if stack.isempty():

print 'the stack is empty!'

else:

print 'the stack is not empty!'

elif choose==4:

char=stack.peekstack()

if char:

print 'last data is:',char

else:

print 'Error:the stack is empty!'

else:

break

elif menu==2:

while True:

print '\n\*\*\*QUEUE\*\*\*'

print '1、Enqueue'

print '2、Dequeue'

print '3、Putfile'

print '4、Getfile'

print '5、Exit'

choose=input("Your choose:")

if choose==1:

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

queue.enqueue(data)

print 'OK!'

elif choose==2:

char=queue.dequeue()

if char:

print 'remove [',char,']'

else:

print 'Error:the queue is empty!'

elif choose==3:

filename=raw\_input("enter the file name:")

isok=queue.putfile(filename)

if isok:

print 'OK!'

else:

print 'Error:the queue is empty!!'

elif choose==4:

filename=raw\_input("enter the file name:")

isok=queue.getfile(filename)

if isok:

print queue.queue

else:

print 'Error:no file!'

else:

break

else:

break