**[Python实现简单的记账本功能](https://www.cnblogs.com/xkops/p/6257026.html)**

目标：

　　1.使用序列化cPickle

　　2.账户中钱要大于花费的钱，否则提示请存钱

　　2.编写函数，实现存钱，花钱，查询及退出功能

1.序列化

　　pickle是python实现序列化的模块，次模块存在使用C语言编写模块，用法相同，但执行效率更高，所以优先使用C模块编写的序列化模块cPickle。

2.编写函数，实现存钱，花钱，查询及退出功能

代码如下：

[root@localhost python]# cat new\_account.py

#!/usr/bin/env python

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

import os,time

import cPickle as p

def save\_money(wallet, record, amount, comment):

date = time.strftime("%Y-%m-%d")

with open(wallet) as fobj:

balance = p.load(fobj) + amount

with open(wallet, 'wb') as fobj:

p.dump(balance, fobj)

with open(record, 'a') as fobj:

fobj.write(

"%-12s%-8s%-8s%-10s%-20s\n" % (

date, 'N/A', amount, balance, comment

)

)

def cost\_money(wallet, record, amount, comment):

date = time.strftime("%Y-%m-%d")

with open(wallet) as fobj:

balance = p.load(fobj) - amount

if balance < 0:

print "余额不足，请先存钱或进行其他操作！"

else:

with open(wallet, 'wb') as fobj:

p.dump(balance, fobj)

with open(record, 'a') as fobj:

fobj.write(

"%-12s%-8s%-8s%-10s%-20s\n" % (

date, amount, 'N/A', balance, comment

)

)

def query\_money(wallet, record):

print "%-12s%-8s%-8s%-10s%-20s" % (

'date', 'cost', 'save', 'balance', 'comment'

)

with open(record) as fobj:

for line in fobj:

print line,

with open(wallet) as fobj:

print "New Balance:\n%s" % p.load(fobj)

def show\_menu():

w\_file = 'wallet.data'

r\_file = 'record.txt'

cmds = {

'0': save\_money,

'1': cost\_money,

'2': query\_money

}

prompt = """(0) save money

(1) spend money

(2) query detail

(3) quit

Please input your choice(0/1/2/3): """

if not os.path.isfile(w\_file):

with open(w\_file, 'w') as fobj:

p.dump(0, fobj)

if not os.path.isfile(r\_file):

os.mknod(r\_file)

while True:

args = (w\_file, r\_file)

choice = raw\_input(prompt).strip()[0]

if choice not in '0123':

print "Invalid input, Try again."

continue

if choice in '01':

amount = int(raw\_input("Amount: "))

comment = raw\_input("Comment: ")

args = (w\_file, r\_file, amount, comment)

if choice == '3':

break

cmds[choice](\*args)

if \_\_name\_\_ == '\_\_main\_\_':

print show\_menu()

•运行代码，测试效果

[root@localhost python]# python new\_account.py

(0) save money

(1) spend money

(2) query detail

(3) quit

Please input your choice(0/1/2/3): 2

date cost save balance comment

New Balance:

0

(0) save money

(1) spend money

(2) query detail

(3) quit

Please input your choice(0/1/2/3): 1

Amount: 100

Comment: cost 100

余额不足，请先存钱或进行其他操作！

(0) save money

(1) spend money

(2) query detail

(3) quit

Please input your choice(0/1/2/3): 0

Amount: 100

Comment: save 100

(0) save money

(1) spend money

(2) query detail

(3) quit

Please input your choice(0/1/2/3): 2

date cost save balance comment

2017-01-06 N/A 100 100 save 100

New Balance:

100

(0) save money

(1) spend money

(2) query detail

(3) quit

Please input your choice(0/1/2/3): 1

Amount: 101

Comment: cost 101

余额不足，请先存钱或进行其他操作！

(0) save money

(1) spend money

(2) query detail

(3) quit

Please input your choice(0/1/2/3): 1

Amount: 100

Comment: cost 100

(0) save money

(1) spend money

(2) query detail

(3) quit

Please input your choice(0/1/2/3): 2

date cost save balance comment

2017-01-06 N/A 100 100 save 100

2017-01-06 100 N/A 0 cost 100

New Balance:

0

(0) save money

(1) spend money

(2) query detail

(3) quit

Please input your choice(0/1/2/3):

\*附录

1.如下是自己初次编写的代码，函数不具备通用性功能。

#!/usr/bin/env python

#coding:utf8

import os,sys

import time

'''

1.运行该脚本会生成一个balance.txt文件，并设置初始账户余额：￥10000

2.运行该脚本会生成一个account.txt文件，并记录账户消费信息详情。

'''

def save():

date = time.strftime("%Y-%m-%d")

cost = 0

while 1:

try:

save = int(raw\_input("请输入存款金额: ").strip())

except ValueError:

print "\033[31m请输入数值类型，重新输入！\033[0m"

continue

except (KeyboardInterrupt,EOFError):

sys.exit("\n\033[31m程序退出\033[0m")

if save <= 0:

print "\033[31m请输入一个大于0的存款金额：\033[0m"

continue

while 1:

try:

comment = str(raw\_input("请输入存款信息: "))

except (KeyboardInterrupt,EOFError):

sys.exit("\n\033[31m程序退出\033[0m")

if not comment:

continue

break

break

balance = rekcon\_balance(save,cost)

a.write('%-12s%-12s%-12s%-12s%-12s\n' %(date, cost, save, balance, comment))

a.flush()

with open('balance.txt', 'w') as b:

balance = str(balance)

b.write(balance)

def cost():

save = 0

date = time.strftime("%Y-%m-%d")

while 1:

try:

cost = int(raw\_input("请输入消费金额: ").strip())

except ValueError:

print "\033[31m请输入数值类型，重新输入!!!\033[0m"

continue

except (KeyboardInterrupt,EOFError):

sys.exit("\n\033[31m程序退出\033[0m")

if cost <= 0:

print "\033[31m请输入一个大于0的消费金额：\033[0m"

continue

break

balance = rekcon\_balance(save,cost)

while balance == -1:

print "\033[31m余额不足，请充值或进行其他操作!!!\033[0m"

break

else:

while 1:

try:

comment = str(raw\_input("请输入消费信息: "))

except (KeyboardInterrupt,EOFError):

sys.exit("\n\033[31m程序退出\033[0m")

if not comment:

continue

break

a.write('%-12s%-12s%-12s%-12s%-12s\n' %(date, cost, save, balance, comment))

with open('balance.txt', 'w') as b:

balance = str(balance)

b.write(balance)

a.flush()

def rekcon\_balance(save,cost):

try:

with open('balance.txt', 'r') as b:

balance = b.readline()

balance = int(balance)

except IOError:

balance = 10000

balance += save

if cost > balance:

balance = -1

return balance

balance -= cost

# with open('balance.txt', 'w') as f:

# balance = str(balance)

# f.write(balance)

return balance

def balance():

try:

with open('balance.txt', 'r') as b:

balance = b.readline()

except IOError,e:

balance = 10000

print "\033[31m初始账户余额:\033[0m￥%s" % balance

else:

print "\033[31m当前账户余额:\033[0m￥%s" % balance

def view():

print '账户金额详细信息'.center(78,'\*')

print "%-12s%-12s%-12s%-12s%-12s\n" %('Date', 'Cost', 'Save', 'Balance', 'Comment'),

with open('account.txt','r') as b:

for line in b.readlines():

print line,

print '\*'.center(70,'\*')

def show\_menu():

cmds = {

'0': save, '1': cost, '2': balance, '3': view, '4': quit

}

prompt = """\033[32m-----------------------------

(0): save money

(1): cost money

(2): balance

(3): view detail

(4): quit

-----------------------------\033[0m

Please Input Your Choice: """

while 1:

try:

choice = raw\_input(prompt).strip()[0]

except (KeyboardInterrupt,EOFError):

sys.exit("\n\033[31m程序退出\033[0m")

except IndexError:

print "\033[31m无效输入，请重新输入!!!\033[0m"

continue

if choice not in '01234':

print "\033[31m无效输入，请重新输入!!!\033[0m"

continue

if choice == 4:

break

cmds[choice]()

if \_\_name\_\_ == '\_\_main\_\_':

a = open('account.txt','a')

print show\_menu()

a.close()