## 各组组长，把银行的各个模块分给组员，组员负责设计银行的各个功能的用例，最终生成界面版的测试报告，组长打包发给我

import random  
*#银行库*bank = {} *# username : {password,money......}*bank\_name = **"中国工商银行昌平支行"**bank\_choice = {**"1"**:**"开户"**,**"2"**:**"存钱"**,**"3"**:**"取钱"**,**"4"**:**"转账"**,**"5"**:**"查询"**,**"6"**:**"Bye"**} *# 银行业务选项  
# 开户成功的信息模板*myinfo=**'''**\033**[0;32;40m  
------------账户信息------------  
账号：{account}  
姓名：{username}  
密码：{password}  
地址：  
 国家：{country}  
 省份：{province}  
 街道：{street}  
 门牌号：{door}  
账户余额：{money}  
注册银行名：{bank\_name}  
-------------------------------**\033**[0m  
'''***# 欢迎模板*welcome = **'''  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* 中国工商银行账户管理系统 \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* 选项 \*  
'''**welcome\_item = **'''\* {0}.{1} \*'''**def print\_welcome():  
 print(welcome,end=**""**)  
 keys = bank\_choice.keys()  
 for i in keys:  
 print(welcome\_item.format(i,bank\_choice[i]))  
 print(**"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**)  
  
*# 输入帮助方法：chose是打印选项*def inputHelp(chose,datatype=**"str"**):  
 while True:  
 print(**"请输入"**,chose,**":"**)  
 i = input(**">>>:"**)  
 if len(i) == 0:  
 print(**"该项不能为空！请重新输入！"**)  
 continue  
 if datatype != **"str"**:  
 return int(i)  
 else:  
 return i  
  
*# 判断是否存在该银行选项*def isExists(chose,data):  
 if chose in data:  
 return True  
 return False  
  
  
*# 获取随机码*def getRandom():  
 li = **"0123456789qwertyuiopasdfghjklzxcvbnmZXCVBNMASDFGHJKLQWERTYUIOP"** string = **""** for i in range(8):  
 string = string + li[int(random.random()\* len(li))]  
 return string  
  
*# 通过账号获取账户信息*def findByAccount(account):  
 for i in bank.keys():  
 if bank[i][**"account"**] == account:  
 return i  
 return None  
  
  
  
  
  
  
*# 银行的开户方法*def bank\_addUser(username,password,country,province,street,door,money):  
 *# for i in range(100):  
 # bank["张三" + str(i)] = {}* if len(bank) >= 100:  
 return 3  
 elif username in bank:  
 return 2  
 else: *# 正常开户：存储到银行* bank[username] = {  
 **"account"**:getRandom(),  
 **"password"**:password,  
 **"country"**:country,  
 **"province"**:province,  
 **"street"**:street,  
 **"door"**:door,  
 **"money"**:money,  
 **"bank\_name"**:bank\_name  
 }  
 return 1  
  
*# 银行的存钱方法*def bank\_saveMoney(ac,money):  
 for i in bank.keys():  
 if bank[i][**"account"**] == ac:  
 print(bank[i][**"money"**])  
 bank[i][**"money"**] += money  
  
 return True  
 return False  
  
*# 银行的查询功能*def bank\_selectUser(account,password):  
  
 uname = findByAccount(account)  
  
 if uname != None and len(uname) != 0:  
 if password == bank[uname][**"password"**]:  
 user = bank[uname]  
 print(myinfo.format(account=user[**"account"**],  
 username=uname,  
 password=user[**"password"**],  
 country=user[**"country"**],  
 province=user[**"province"**],  
 street=user[**"street"**],  
 door=user[**"door"**],  
 money=user[**"money"**],  
 bank\_name=user[**"bank\_name"**]  
 ))  
 else:  
 print(**"用户密码错误！"**)  
 else:  
 print(**"该用户不存在！"**)  
  
*# 银行的取钱功能*def bank\_takeMoney(account,password,money):  
 uname = findByAccount(account)  
 if uname != None:  
 if bank[uname][**"password"**] == password:  
 if bank[uname][**"money"**] < money:  
 return 3  
 else:  
 bank[uname][**"money"**] -= money  
 return 0  
 else:  
 return 2  
 else:  
 return 0  
  
*# 银行的转账功能*def bank\_transformMoney(outputaccount,inputaccount,outputpassword,outputmoney):  
 status = bank\_takeMoney(outputaccount,outputpassword,outputmoney)  
 if status == 1:  
 return status  
 elif status == 2:  
 return status  
 elif status == 3:  
 return status  
  
 if inputaccount != None and findByAccount(inputaccount) != None:  
 bank\_saveMoney(inputaccount,outputmoney)  
 return 0  
 else:  
 return 1

import unittest  
from day14.bank.bankdemo import bank\_takeMoney  
from day14.bank.bankdemo import findByAccount  
from day14.bank.bankdemo import bank\_saveMoney  
  
  
  
class TestBank1(unittest.TestCase):  
 def testBank\_transformMoney(self):  
 outputaccount=165146145  
 inputaccount=9842894892  
 outputpassword=**"123456"** outputmoney=**"654321"** status = bank\_takeMoney(outputaccount, outputpassword, outputmoney)  
 if status == 1:  
 return status  
 elif status == 2:  
 return status  
 elif status == 3:  
 return status  
  
 if inputaccount != None and findByAccount(inputaccount) != None:  
 bank\_saveMoney(inputaccount, outputmoney)  
 return 0  
 else:  
 return 1

mport unittest  
from day14.bank.debit import TestBank1  
from HTMLTestRunner import HTMLTestRunner  
  
*#创建测试集*suite = unittest.TestSuite()  
suite.addTest(TestBank1(**"testBank\_transformMoney"**))  
  
f = open(**"银行转账.html"**,**"w+"**,encoding=**"utf-8"**)  
htmlrunner = HTMLTestRunner.HTMLTestRunner(  
 stream=f, *# 将生成的报告写入到f文件里* title=**"银行转账的测试报告"**, *# 报告的标题* description=**"这是一个银行转账的测试"**, *# 报告的描述* verbosity=1,  
)  
htmlrunner.run(suite)