**อธิบายการทำงาน**

การเขียนโค้ดโปรแกรมสร้างคลาส BankAccount ที่ encapsulate ข้อมูลและคุณสมบัติต่างๆรวมไปถึงเมธอดที่เกี่ยวข้องกับการจัดการบัญชีธนาคาร

**ขั้นตอนการเขียนโค้ดโปรแกรม**

1. สร้าง Class BankAccount ซึ่งมี attribute ที่ประกอบไปด้วย (account\_number, balance, name) โดยกำหนดให้ทุกพารามิเตอร์เป็นแบบ private ด้วย double underscores

2. กำหนดตัวแปรและสร้างฟังก์ชัน ประกอบด้วย

1) ฟังก์ชันเก็บเลขบัญชี,เงินและชื่อผู้ใช้บัญชี

2) ฟังก์ชันในการฝากเงิน โดยมี 2 เงื่อนไข

- กรณีค่าเงินมากกว่า 0 ให้บวกจำนวนเงินนั้นเข้าบัญชีและแสดงผล

- กรณีค่าเงินที่น้อยกว่า 0 หรือ ข้อมูลอื่นให้แสดงข้อความให้ใส่จำนวนเงินที่ถูกต้อง

3) ฟังก์ชั่นในการถอนเงิน โดยมี 3 เงื่อนไข

- กรณีค่าค่าเงินมากกว่า 0 ให้ลบจำนวนเงินนั้นออกบัญชีและแสดงผล

- กรณีค่าเงินมากกว่าเงินในบัญชีให้แสดงข้อความาเงินในบัญชีไม่พอกับเงินที่จะถอน

- กรณีค่าเงินน้อยกว่า 0 หรือ ข้อมูลอื่นให้แสดงข้อความให้ใส่จำนวนเงินที่ถูกต้อง

4) ฟังก์ชันส่วนดอกเบี้ยเงินฝาก กำหนดให้อัตราดอกเบี้ยร้อยละ 0.02

5) ฟังก์ชันส่วนค่าธรรมเนียม กำหนดค่าธรรมเนียมไว้ที่ 20 บาท โดยมี 2 เงื่อนไข

- เมื่อทำการถอนเงินมากกว่าในบัญชี จะหักเงินค่าธรรมเนียม

- เมื่อทำการถอนเงินน้อยกว่าหรือเท่ากับเงินในบัญชี จะหักเงินค่าธรรมเนียม

6) สร้างฟังก์ชันแสดงข้อมูลในตัวแปร ประกอบด้วย เลขบัญชี,เงินในบัญชีและชื่อผู้ใช้บัญชี

3.สร้างส่วนของการแสดงผล

โค้ดโปรแกรม

class BankAccount: #สร้างคลาสบัญชีธนาคาร

    def \_\_init\_\_(self, account\_number, balance, name): #สร้างฟังก์ชันและพารามิเตอร์

        self.\_\_account\_number = account\_number #การกำหนดค่าให้กับตัวแปรใช้ double underscores เพื่อระบุให้เป็น private

        self.\_\_balance = balance

        self.\_\_name = name

    def get\_Accout\_number(self):  #สร้างฟังก์ชันเก็บเลขบัญชี

        return self.\_\_account\_number

    def \_\_get\_balance(self): #สร้างฟังก์ชันเก็บเงินในบัญชี

        return self.\_\_balance

    def get\_name(self): #สร้างฟังก์ชันเก็บชื่อเจ้าของบัญชี

        return self.\_\_name

    def deposite(self, amount): #สร้างฟังก์ชันในส่วนของการฝากเงิน

        if amount > 0: #ถ้าค่าที่เข้ามามากกว่า 0

            self.\_\_balance += amount # ให้บวกค่าค่าเงินในบัญชีและค่าที่เข้ามารวมกัน

            print('ฝากเงินเข้าบัญชีจำนวน:', amount)

            print('ยอดเงินคงเหลือ:', self.\_\_balance)

        else:

            print('กรุณาใส่จำนวนเงินที่ถูกต้อง')

    def withdraw(self, amount): #สร้างฟังก์ชันในส่วนของการถอนเงิน

        if 0 < amount <= self.\_\_balance: #ถ้าค่าที่ต้องการถอนมากกว่า 0

            self.\_\_balance -= amount #จะลบค่าที่ต้องการถอนออกจากเงินในบัญชี

            print('ถอนเงินจากบัญชีจำนวน:', amount)

            print('ยอดเงินคงเหลือ:', self.\_\_balance)

        elif amount > self.\_\_balance: #กรณีค่าที่ต้องการถอนมากกว่าเงินในบัญชี

            print('ยอดเงินไม่เพียงพอ')

        else:

            print('กรุณาใส่จำนวนเงินที่ถูกต้อง')

    def add\_interest(self, interest\_rate=0.02): #สร้างฟังก์ชันในส่วนของดอกเบี้ยเงินฝาก ดอกเบี้ยร้อยละ 0.03

        self.interest\_rate = interest\_rate

        interest = self.\_\_balance \* self.interest\_rate #คูณค่าเงินในบัญชีรวมเข้ากับเปอร์เซ็นต์ดอกเบี้ย

        self.\_\_balance += interest #รวมค่าเงินในบัญชีกับค่าดอกเบี้ยที่เพิ่ม

        print('เพิ่มดอกเบี้ยจากบัญชีออมทรัพย์:', interest)

        print('ยอดเงินคงเหลือ:', self.\_\_balance)

    def checking(self, amount, vat=20): #สร้างฟังก์ชันในส่วนของการจ่ายเช็ค ค่าธรรมเนียม 30

        self.vat = vat

        print('ต้องการจ่ายเช็คจำนวน:', amount) #แสดงค่าเช็คที่ต้องการจ่าย

        if amount > self.\_\_balance: #กรณีค่าที่ต้องการนั้นมากกว่าเงินในบัญชี

            print('ยอดเงินไม่เพียงพอ')

            self.\_\_balance -= self.vat #หักค่าธรรมเนียมออกจากเงินในบัญชี

            print('หักค่าธรรมเนียม:', self.vat)

            print('ยอดเงินคงเหลือ:', self.\_\_balance)

        else:

            self.\_\_balance -= self.vat #หักค่าธรรมเนียมออกจากเงินในบัญชี

            print('หักค่าธรรมเนียม:', self.vat)

            self.withdraw(amount)

    def show(self): #สร้างฟังก์ชันในการแสดงข้อมูลเลขบัญชี ชื่อ จำนวนเงิน

        print('เลขบัญชี: '+ str(self.get\_Accout\_number()), '| ชื่อบัญชี: ' +str(self.get\_name()))

        print('ยอดเงินในบัญชี: '+str(self.\_\_get\_balance()))

#ส่วนของการแสดงผล

Acc1=BankAccount(630910323,5000,'Tawan Chaiyamart') #กำหนดค่าลงในพารามิเตอร์ของคลาสบัญชีธนาคาร

Acc1.show() #เรียกใช้ฟังก์ชันในการแสดงข้อมูลเลขบัญชี ชื่อ จำนวนเงิน

print(' ')

Acc1.deposite(5000) #เรียกใช้ฟังก์ชันในส่วนของการฝากเงิน

print(' ')

Acc1.withdraw(3000) #เรียกใช้ฟังก์ชันในส่วนของการถอนเงิน

print(' ')

Acc1.add\_interest() #เรียกใช้ฟังก์ชันในส่วนของดอกเบี้ยเงินฝาก

print(' ')

print('-----------------------------------')

Acc2=BankAccount(630910316,8000,'Chokthawee Fakanong')#กำหนดค่าลงในพารามิเตอร์ของคลาสบัญชีธนาคาร

Acc2.show() #เรียกใช้ฟังก์ชันในการแสดงข้อมูลเลขบัญชี ชื่อ จำนวนเงิน

print(' ')

Acc2.checking(10000) #เรียกใช้ฟังก์ชันในส่วนของการจ่ายเช็ค

print(' ')

Acc2.checking(2000) #เรียกใช้ฟังก์ชันในส่วนของการจ่ายเช็คแต่เปลี่ยนค่าในพารามิเตอร์

print(' ')

print('-----------------------------------')

Acc1.show()#เรียกใช้ฟังก์ชันในการแสดงข้อมูลเลขบัญชี ชื่อ จำนวนเงิน

print(' ')

Acc2.show()#เรียกใช้ฟังก์ชันในการแสดงข้อมูลเลขบัญชี ชื่อ จำนวนเงิน

print(' ')

ผลการรัน

