

## อธิบายการทำงาน

การเขียนโค้ดโปรแกรมสร้างคลาส BankAccount ที่ encapsulate ข้อมูลและคุณสมบัติต่างๆรวมถึงเมธอดที่เกี่ยวข้องกับการจัดการบัญชีธนาคาร

### 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) ฟังก์ชันในการถอนเงิน โดยมี 3 เงื่อนไข

- กรณีค่าเงินมากกว่า 0 ให้ลบจำนวนเงินนั้นออกบัญชีและแสดงผล

- กรณีค่าเงินมากกว่าเงินในบัญชีให้แสดงข้อความว่าเงินในบัญชีไม่พอกับเงินที่จะถอน

- 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_Account_number(self): #สร้างฟังก์ชันเก็บเลขบัญชี
        return self.__account_number

    def __get_balance(self): #สร้างฟังก์ชันเก็บเงินในบัญชี
        return self.__balance

    def get_name(self): #สร้างฟังก์ชันเก็บชื่อเจ้าของบัญชี
        return self.__name

    def deposit(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): #สร้างฟังก์ชันในส่วนของการดอกเบี้ยเงินฝาก 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('ขอเงินไม่เพียงพอ')
            self.__balance -= self.vat #หักค่าธรรมเนียมออกจากเงินในบัญชี
            print('หักค่าธรรมเนียม:', self.vat)
            print('ขอเงินคงเหลือ:', 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_Account_number()), '| ชื่อบัญชี: '
+str(self.get_name()))
        print('ขอเงินในบัญชี: '+str(self.__get_balance()))

#ส่วนของการแสดงผล

Acc1=BankAccount(630910323,5000,'Tawan Chaiyamart') #กำหนดค่าลงในพารามิเตอร์ของคลาสบัญชีธนาคาร
Acc1.show() #เรียกใช้ฟังก์ชันในการแสดงข้อมูลเลขบัญชี ชื่อ จำนวนเงิน
print(' ')
Acc1.deposit(5000) #เรียกใช้ฟังก์ชันในส่วนของการฝากเงิน
print(' ')
Acc1.withdraw(3000) #เรียกใช้ฟังก์ชันในส่วนของการถอนเงิน
print(' ')
Acc1.add_interest() #เรียกใช้ฟังก์ชันในส่วนของการดอกเบี้ยเงินฝาก
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

เลขบัญชี: 630910323 | ชื่อบัญชี: Tawan Chaiyamart  
ยอดเงินในบัญชี: 5000

ฝากเงิน เข้าบัญชีจำนวน: 5000  
ยอดเงินคงเหลือ: 10000

ถอนเงินจากบัญชีจำนวน: 3000  
ยอดเงินคงเหลือ: 7000

เพิ่มดอกเบี้ยจากบัญชีออมทรัพย์: 140.0  
ยอดเงินคงเหลือ: 7140.0

-----  
เลขบัญชี: 630910316 | ชื่อบัญชี: Chokthawee Fakanong  
ยอดเงินในบัญชี: 8000

ต้องการจ่ายเช็คจำนวน: 10000  
ยอดเงินไม่เพียงพอ  
หักค่าธรรมเนียม: 20  
ยอดเงินคงเหลือ: 7980

ต้องการจ่ายเช็คจำนวน: 2000  
หักค่าธรรมเนียม: 20  
ถอนเงินจากบัญชีจำนวน: 2000  
ยอดเงินคงเหลือ: 5960

-----  
เลขบัญชี: 630910323 | ชื่อบัญชี: Tawan Chaiyamart  
ยอดเงินในบัญชี: 7140.0

เลขบัญชี: 630910316 | ชื่อบัญชี: Chokthawee Fakanong  
ยอดเงินในบัญชี: 5960