

เอกสารอธิบายการทำงานของโปรแกรม

การเขียนโปรแกรมนี้นี้เป็นการสร้างคลาส BankAccount ที่ encapsulate ข้อมูล และคุณสมบัติต่างๆ รวมไปถึงเมธอดที่เกี่ยวข้องกับการจัดการบัญชีธนาคาร

โดยเริ่มจากสร้าง Class BankAccount ซึ่งมี attribute ประกอบไปด้วย account_number เลขที่บัญชี, balance ยอดเงินในบัญชี, name ชื่อเจ้าของบัญชี โดยกำหนดให้ทุกพารามิเตอร์เป็นแบบ private ด้วย double underscores หลังจากนั้นทำการกำหนดค่าให้กับตัวแปร แล้วเริ่มสร้างฟังก์ชันคือ

1. ฟังก์ชันเก็บเลขบัญชี, เก็บเงินในบัญชี, เก็บชื่อเจ้าของบัญชี

2. ฟังก์ชันในส่วนของการฝากเงินโดยแบ่งเป็น กรณีที่หนึ่งถ้าค่าที่เข้ามามากกว่า 0 ให้บวกค่าเงินในบัญชีและค่าที่เข้ามารวมกันแล้วแสดงผล และกรณีที่สอง ถ้าค่าที่เข้ามาเป็นอื่นจะแสดงข้อความให้ใส่จำนวนเงินที่ต้องการ

3. ฟังก์ชันในส่วนของการถอนเงิน ถ้าค่าที่เข้ามามากกว่า 0 จะลบค่าที่ต้องการถอนออกจากเงินในบัญชีแล้วแสดงผล แต่ถ้าค่าที่ต้องการถอนมากกว่าเงินในบัญชีระบบจะแจ้งเตือนว่าเงินในบัญชีไม่พอกับเงินที่จะถอน หรือถ้าค่าที่เข้ามาเป็นอื่นระบบจะแจ้งเตือนว่าให้ใส่ค่าจำนวนเงิน

4. ฟังก์ชันในส่วนของการดอกเบี้ยเงินฝาก โดยกำหนดให้ดอกเบี้ยร้อยละ 0.03 เริ่มต้นจากกำหนดค่าให้กับตัวแปร สร้างตัวแปรเก็บค่าที่คูณค่าเงินในบัญชีรวมเข้ากับเปอร์เซ็นต์ดอกเบี้ย แล้วรวมค่าเงินในบัญชีกับค่าดอกเบี้ยที่เพิ่ม หลังจากนั้นแสดงผล

5. สร้างฟังก์ชันในส่วนของการจ่ายเช็ค ซึ่งมีค่าธรรมเนียมเท่ากับ 30 บาท เริ่มโดยการกำหนดค่าให้กับตัวแปร ถ้าค่าที่ต้องการนั้นมากกว่าเงินในบัญชีระบบจะแจ้งเตือนว่าเงินในบัญชีไม่พอกับเงินที่ต้องการแต่จะค่าธรรมเนียมหัก 30 บาทจากยอดเงินในบัญชี แต่ถ้าหากค่าที่ต้องการนั้นน้อยกว่าหรือเท่ากับเงินในบัญชี จะทำการหักค่าทำเนียม 30 บาทและเรียกใช้ฟังก์ชันของการถอนเงิน

6. และสร้างฟังก์ชันที่ทำการแสดงข้อมูลในตัวแปรซึ่งประกอบไปด้วย เลขบัญชี ชื่อเจ้าของบัญชี และยอดเงินในบัญชี

หลังจากสร้างฟังก์ชันทั้งหมดแล้วจะเป็นในส่วนของการแสดงผล

1. กำหนดข้อมูลชุดแรกลงในพารามิเตอร์ของคลาสบัญชีธนาคาร

2. เรียกใช้ฟังก์ชันในการแสดงข้อมูลเลขบัญชี ชื่อ จำนวนเงิน

- 3.เรียกใช้ฟังก์ชันในส่วนของการฝากเงิน
- 4.เรียกใช้ฟังก์ชันในส่วนของการถอนเงิน
- 5.เรียกใช้ฟังก์ชันในส่วนของการดอกเบี้ยเงินฝาก
- 6.กำหนดข้อมูลชุดที่สองลงในพารามิเตอร์ของคลาสบัญชีธนาคาร
- 7.เรียกใช้ฟังก์ชันในการแสดงข้อมูลเลขบัญชี ชื่อ จำนวนเงิน
- 8.เรียกใช้ฟังก์ชันในส่วนของการจ่ายเช็ค
- 9.เรียกใช้ฟังก์ชันในส่วนของการจ่ายเช็คแต่เปลี่ยนค่าในพารามิเตอร์
- 10.หลังจากนั้นเรียกใช้งานฟังก์ชัน show() เพื่อแสดงข้อมูลสุดท้ายของทั้งสองชุด

CODE บัญชีธนาคาร

```
class BankAccount: #สร้างคลาสบัญชีธนาคาร และสร้าง attribute ที่ประกอบไปด้วย

    def __init__(self, account_number, balance, name): #สร้างฟังก์ชันและพารามิเตอร์

        self.__account_number = account_number #การกำหนดค่าให้กับตัวแปรใช้ double
underscores เพื่อระบุให้เป็น private

        self.__balance = balance #การกำหนดค่าให้กับตัวแปร

        self.__name = name #การกำหนดค่าให้กับตัวแปร

    def get_Account_number(self): #สร้างฟังก์ชันเก็บเลขบัญชี

        return self.__account_number # สร้างเมธอดเพื่อให้เข้าถึงคุณสมบัติ

    def __get_balance(self): #สร้างฟังก์ชันเก็บเงินในบัญชี

        return self.__balance #สร้างเมธอดเพื่อให้เข้าถึงคุณสมบัติ

    def get_name(self): #สร้างฟังก์ชันเก็บชื่อเจ้าของบัญชี

        return self.__name #สร้างเมธอดเพื่อให้เข้าถึงคุณสมบัติ
```

```
def deposit(self, amount): #สร้างฟังก์ชันในส่วนของการฝากเงิน

    if amount > 0: #ถ้าค่าที่เข้ามามากกว่า 0

        self.__balance += amount # ให้นำบวกค่าเงินในบัญชีและค่าที่เข้ามารวมกัน

        print('ฝากเงินเข้าบัญชีจำนวน:', amount) #แสดงค่าที่เข้ามา

        print('ยอดเงินคงเหลือ:', self.__balance) #แสดงค่าที่รวมกันแล้ว

    else: #ถ้าค่าที่เข้ามาเป็นอื่น

        print('กรุณาใส่จำนวนเงินที่ถูกต้อง') #ระบบจะแจ้งเตือนให้ใส่ค่าจำนวนเงิน

def withdraw(self, amount): #สร้างฟังก์ชันในส่วนของการถอนเงิน

    if 0 < amount <= self.__balance: #ถ้าค่าที่ต้องการถอนมากกว่า 0

        self.__balance -= amount #จะลบค่าที่ต้องการถอนออกจากเงินในบัญชี

        print('ถอนเงินจากบัญชีจำนวน:', amount) #แสดงค่าที่ต้องการถอน

        print('ยอดเงินคงเหลือ:', self.__balance) #แสดงเงินหลังจากการถอน

    elif amount > self.__balance: #ถ้าค่าที่ต้องการถอนมากกว่าเงินในบัญชี

        print('ยอดเงินไม่เพียงพอ') #ระบบจะแจ้งเตือนว่าเงินในบัญชีไม่พอกับเงินที่จะถอน

    else: #ถ้าค่าที่เข้ามาเป็นอื่น

        print('กรุณาใส่จำนวนเงินที่ถูกต้อง') #ระบบจะแจ้งเตือนให้ใส่ค่าจำนวนเงิน

def add_interest(self, interest_rate=0.03): #สร้างฟังก์ชันในส่วนของการดอกเบี้ยเงินฝาก ดอกเบี้ยร้อยละ 0.03

    self.interest_rate = interest_rate #กำหนดค่าให้กับตัวแปร
```

```
interest = self.__balance * self.interest_rate #คูณค่าเงินในบัญชีรวมเข้ากับเปอร์เซ็นต์ดอกเบี้ย

self.__balance += interest #รวมค่าเงินในบัญชีกับค่าดอกเบี้ยที่เพิ่ม

print('เพิ่มดอกเบี้ยจากบัญชีออมทรัพย์:', interest) #แสดงค่าดอกเบี้ยที่เพิ่ม

print('ยอดเงินคงเหลือ:', self.__balance) #แสดงค่าที่รวมกับดอกเบี้ยแล้ว


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

    self.vat = vat #กำหนดค่าให้กับตัวแปร

    print('ต้องการจ่ายเช็คจำนวน:', amount) #แสดงค่าเช็คที่ต้องการจ่าย

    if amount > self.__balance: #ถ้าค่าที่ต้องการนั้นมากกว่าเงินในบัญชี

        print('ยอดเงินไม่เพียงพอ') #ระบบจะแจ้งเตือนว่าเงินในบัญชีไม่พอกับเงินที่ต้องการ

        self.__balance -= self.vat #จะหักค่าธรรมเนียมออกจากเงินในบัญชี

        print('หักค่าธรรมเนียม:', self.vat) #แสดงค่าธรรมเนียมที่ตั้งไว้

        print('ยอดเงินคงเหลือ:', self.__balance) #แสดงค่าที่หักค่าธรรมเนียมแล้ว

    else: #ถ้าเป็นอื่น

        self.__balance -= self.vat #จะหักค่าธรรมเนียมออกจากเงินในบัญชี

        print('หักค่าธรรมเนียม:', self.vat) #แสดงค่าธรรมเนียมที่ตั้งไว้

        self.withdraw(amount) #เรียกใช้attributeของการถอนเงิน


def show(self): #สร้างฟังก์ชันในการแสดงข้อมูลเลขบัญชี ชื่อ จำนวนเงิน

    print('เลขบัญชี: ' + str(self.get_Account_number()), '| ชื่อบัญชี: ' + str(self.get_name())) #แสดงเลขบัญชีและชื่อเจ้าของบัญชี

    print('ยอดเงินในบัญชี: ' + str(self.__get_balance())) #แสดงยอดเงินในบัญชี
```

#ส่วนของการแสดงผล

Acc1=BankAccount(10203040,10000,'Song Yong') #กำหนดค่าลงในพารามิเตอร์ของคลาสบัญชีธนาคาร

Acc1.show() #เรียกใช้ฟังก์ชันในการแสดงข้อมูลเลขบัญชี ชื่อ จำนวนเงิน

print(' ') #แสดงข้อความว่างในที่นี้ใช้เป็นบรรทัดว่าง

Acc1.deposit(5000) #เรียกใช้ฟังก์ชันในส่วนของการฝากเงิน

print(' ') #แสดงข้อความว่างในที่นี้ใช้เป็นบรรทัดว่าง

Acc1.withdraw(3000) #เรียกใช้ฟังก์ชันในส่วนของการถอนเงิน

print(' ') #แสดงข้อความว่างในที่นี้ใช้เป็นบรรทัดว่าง

Acc1.add_interest() #เรียกใช้ฟังก์ชันในส่วนของการดอกเบี้ยเงินฝาก

print(' ') #แสดงข้อความว่างในที่นี้ใช้เป็นบรรทัดว่าง

print('-----') #แสดง-ในที่นี้ใช้เป็นเส้นแบ่งข้อมูล

Acc2=BankAccount(11223344,10000,'Yun Yue')#กำหนดค่าลงในพารามิเตอร์ของคลาสบัญชีธนาคาร

Acc2.show() #เรียกใช้ฟังก์ชันในการแสดงข้อมูลเลขบัญชี ชื่อ จำนวนเงิน

print(' ') #แสดงข้อความว่างในที่นี้ใช้เป็นบรรทัดว่าง

Acc2.checking(20000) #เรียกใช้ฟังก์ชันในส่วนของการจ่ายเช็ค

print(' ') #แสดงข้อความว่างในที่นี้ใช้เป็นบรรทัดว่าง

Acc2.checking(2000) #เรียกใช้ฟังก์ชันในส่วนของการจ่ายเช็คแต่เปลี่ยนค่าในพารามิเตอร์

print(' ') #แสดงข้อความว่างในที่นี้ใช้เป็นบรรทัดว่าง

print('-----') #แสดง-ในที่นี้ใช้เป็นเส้นแบ่งข้อมูล

Acc1.show()#เรียกใช้ฟังก์ชันในการแสดงข้อมูลเลขบัญชี ชื่อ จำนวนเงิน

print(' ') #แสดงข้อความว่างในที่นี้ใช้เป็นบรรทัดว่าง

```
Acc2.show()#เรียกใช้ฟังก์ชันในการแสดงข้อมูลเลขบัญชี ชื่อ จำนวนเงิน  
  
print(' ')#แสดงข้อความว่างในที่นี้ใช้เป็นบรรทัดว่าง
```

ผลการรัน

```
เลขบัญชี: 10203040 | ชื่อบัญชี: Song Yong  
ยอดเงินในบัญชี: 10000  
  
ฝากเงิน เข้าบัญชีจำนวน: 5000  
ยอดเงินคงเหลือ: 15000  
  
ถอนเงินจากบัญชีจำนวน: 3000  
ยอดเงินคงเหลือ: 12000  
  
เพิ่มดอกเบี้ยจากบัญชีออมทรัพย์: 360.0  
ยอดเงินคงเหลือ: 12360.0  
  
-----  
เลขบัญชี: 11223344 | ชื่อบัญชี: Yun Yue  
ยอดเงินในบัญชี: 10000  
  
ต้องการจ่ายเช็คจำนวน: 20000  
ยอดเงินไม่เพียงพอ  
หักค่าธรรมเนียม: 30  
ยอดเงินคงเหลือ: 9970  
  
ต้องการจ่ายเช็คจำนวน: 2000  
หักค่าธรรมเนียม: 30  
ถอนเงินจากบัญชีจำนวน: 2000  
ยอดเงินคงเหลือ: 7940  
  
-----  
เลขบัญชี: 10203040 | ชื่อบัญชี: Song Yong  
ยอดเงินในบัญชี: 12360.0  
  
เลขบัญชี: 11223344 | ชื่อบัญชี: Yun Yue  
ยอดเงินในบัญชี: 7940
```