

อธิบายการทำงานของโปรแกรม

โปรแกรมนี้อธิบายการทำงานของบัญชีธนาคาร โดยใช้คลาส BankAccount ดังนี้

1. การสร้างคลาส BankAccount

Python

```
class BankAccount:
```

```
    def __init__(self, account_number, balance, name, account_type,  
interest_rate=0.0):
```

```
        self.__account_number = account_number
```

```
        self.__balance = balance
```

```
        self.__name = name
```

```
        self.__account_type = account_type
```

```
        self.__interest_rate = interest_rate
```

คลาส BankAccount มีตัวแปรสมาชิก (attributes) ดังนี้

- account_number: หมายเลขบัญชี
- balance: ยอดเงินคงเหลือ
- name: ชื่อเจ้าของบัญชี
- account_type: ประเภทบัญชี ("SAVINGS" หรือ "CHECKING")
- interest_rate: อัตราดอกเบี้ย

คลาสนี้มีฟังก์ชัน (methods) ดังนี้

- deposit: ฝากเงิน
- withdraw: ถอนเงิน
- get_balance: ดูกยอดเงินคงเหลือ
- get_account_number: ดูหมายเลขบัญชี
- get_name: ดูชื่อเจ้าของบัญชี
- get_account_type: ดูประเภทบัญชี
- calculate_interest: คำนวณดอกเบี้ย
- transfer_money: โอนเงินไปยังบัญชีอื่น

2. การสร้างวัตถุ (object) ของคลาส BankAccount

Python

```
account1 = BankAccount(630910596_1, 10000, "YOSSAKORN HEMHORMWONG",  
"SAVINGS", 0.015)
```

```
account2 = BankAccount(630910596_2, 5000, "YOSSAKORN HEMHORMWONG",  
"CHECKING", 0.0)
```

โค้ดนี้สร้างวัตถุ (object) ของคลาส BankAccount สองตัว ตัวแปร account1 เก็บข้อมูลของบัญชีแรก และตัวแปร account2 เก็บข้อมูลของบัญชีที่สอง

3. การฝากเงิน ถอนเงิน ดูยอดเงินคงเหลือ ดูข้อมูลบัญชี คำนวณดอกเบี้ย โอนเงิน

Python

```
account1.deposit(5000)
```

```
account2.deposit(2000)
```

```
account1.withdraw(2000)
```

```
account2.withdraw(1000)
```

```
print(f"ยอดเงินคงเหลือในบัญชี {account1.get_account_number()} :  
{account1.get_balance()}")
```

```
print(f"ยอดเงินคงเหลือในบัญชี {account2.get_account_number()} :  
{account2.get_balance()}")
```

```
print(f"ดอกเบี้ยของบัญชี {account1.get_account_number()} :  
{account1.calculate_interest()}")
```

```
print(f"ดอกเบี้ยของบัญชี {account2.get_account_number()} :  
{account2.calculate_interest()}")
```

```
account1.transfer_money(account2, 1000)
```

```
print(f"ยอดเงินคงเหลือในบัญชี {account1.get_account_number()} :  
{account1.get_balance()}")  
print(f"ยอดเงินคงเหลือในบัญชี {account2.get_account_number()} :  
{account2.get_balance()}")
```

โค้ดนี้

- ฝากเงิน 5,000 บาท บัญชี account1
- ฝากเงิน 2,000 บาท บัญชี account2
- ถอนเงิน 2,000 บาท บัญชี account1
- ถอนเงิน 1,000 บาท บัญชี account2
- พิมพ์ยอดเงินคงเหลือของบัญชี account1 และ account2
- พิมพ์ดอกเบี้ยของบัญชี account1 และ account2
- โอนเงิน 1,000 บาท จากบัญชี account1 ไปยังบัญชี account2
- พิมพ์ยอดเงินคงเหลือของบัญชี account1 และ account2 อีกครั้ง

ผลลัพธ์

ยอดเงินคงเหลือในบัญชี 630910596_1 : 13000

ยอดเงินคงเหลือในบัญชี 630910596_2 : 6000

ดอกเบี้ยของบัญชี 630910596_1 : 195

ดอกเบี้ยของบัญชี 630910596_2 : 0

ยอดเงินคงเหลือในบัญชี 630910596_1 : 12000

ยอดเงินคงเหลือในบัญชี 630910596_2 : 7000

5. อธิบายผลลัพธ์

- ยอดเงินคงเหลือของบัญชี account1 หลังจากฝากเงิน 5,000 บาท คือ $10,000 + 5,000 = 15,000$ บาท
- ยอดเงินคงเหลือของบัญชี account2 หลังจากฝากเงิน 2,000 บาท คือ $5,000 + 2,000 = 7,000$ บาท
- ยอดเงินคงเหลือของบัญชี account1 หลังจากถอนเงิน 2,000 บาท คือ $15,000 - 2,000 = 13,000$ บาท

- ยอดเงินคงเหลือของบัญชี account2 หลังจากถอนเงิน 1,000 บาท คือ $7,000 - 1,000 = 6,000$ บาท
- ดอกเบี้ยของบัญชี account1 คำนวณจาก $13,000 * 0.015 = 195$ บาท
- ดอกเบี้ยของบัญชี account2 คำนวณจาก $6,000 * 0.0 = 0$ บาท
- ยอดเงินคงเหลือของบัญชี account1 หลังจากโอนเงิน 1,000 บาท ไปยังบัญชี account2 คือ $13,000 - 1,000 = 12,000$ บาท
- ยอดเงินคงเหลือของบัญชี account2 หลังจากรับเงินโอน 1,000 บาท จากบัญชี account1 คือ $6,000 + 1,000 = 7,000$ บาท