

프로그래밍2 튜터링

(6회차)

조세연

목표

- class 실습

class 예제1

1. 기차(Train) 클래스를 생성하시오.

2.기차(Train) 클래스의 인스턴스를 생성하시오.

3.기차(Train) 클래스에 기차 종류, 탑승 요금, 출발지, 도착지를 입력받는 생성자를 추가하시오.

4. Train 클래스에 기차를 움직이는 move() 라는 함수를 추가하시오.

-> 출발지와 도착지를 출력하고, 출발지를 도착지로 변경

```
탑승 요금 : 25000  
서울역 -> 용산역  
출발지 : 용산역  
기차 종류 : ktx 출발지 : 용산역 도착지 : 용산역 탑승 요금 : 25000
```

5. Train 클래스에 탑승 요금을 출력하는 getOn() 함수를 추가하시오.

6. Train 클래스에 출발지를 출력하는 getOff() 함수를 추가하시오.

7. Train 클래스에 기차의 정보를 출력하는 print() 함수를 추가하시오.

-> 기차 종류, 출발지, 도착지, 탑승 요금을 출력

예제1 - 문제풀이

```
class Train:
```

```
    def __init__(self, type, fare, start, end):
```

```
        self.type = type
```

```
        self.amount = fare
```

```
        self.start = start
```

```
        self.end = end
```

```
    def move(self):
```

```
        print(self.start, '->', self.end)
```

```
        self.start = self.end
```

```
    def getOn(self):
```

```
        print("탑승 요금: ", self.amount)
```

```
    def getOff(self):
```

```
        print("출발지: ", self.start)
```

```
    def print(self):
```

```
        print("기차 종류: ", self.type, "출발지: ", self.start, "도착지: ", self.end, "탑승 요금: ", self.amount)
```

```
train = Train("ktx", 25000, "서울역", "용산역")
```

```
train.getOn()
```

```
train.move()
```

```
train.getOff()
```

```
train.print()
```

class 예제2

1. 돈(Money) 클래스를 정의하시오.

2. 돈(Money) 클래스의 인스턴스를 생성하고, 본인의 이름 변수(ex.seyeon)로 바인딩하시오.

3. 돈(Money) 클래스에 money라는 클래스 변수를 생성한 뒤, 10000으로 초기화하는 생성자를 추가하시오.

-> 반드시 10,000원이 아니어도 됨!! 본인이 원하는 초기값으로 셋팅.

4. 돈(Money) 클래스에 물건을 구매하는 buy_something() 함수를 추가하시오.

-> 물건을 구매시, money가 차감되어야 함.

5. 돈(Money) 클래스에 돈을 적립할 수 있는 save_money() 함수를 추가하시오.

6. 돈(Money) 클래스에 현재 보유 현금을 출력하는 print_my_money() 함수를 추가하시오.

7. (2)에서 생성한 인스턴스를 통해 (4), (5)에서 만든 함수를 각각 사용해보고, (6) 함수를 통해 현재 보유 현금을 출력하시오.

```
튜터링 /example/6차시 /Exam1.py  
구매 완료, 남은 금액은 10000 원 입니다.  
현재 보유 현금은 5000 원 입니다.
```

예제2 - 문제풀이

```
class Money:
    def __init__(self):
        self.money = 10000

    def buy_something(self, price):
        if price > self.money:
            print('보유 금액이 부족합니다.')
        else:
            print('구매 완료, 남은 금액은 ', self.money, '원 입니다.')
            self.money -= price

    def save_money(self, money):
        self.money += money

    def print_my_moeny(self):
        print('현재 보유 현금은 ', self.money, '원 입니다.')

seyeon = Money()
seyeon.buy_something(5000)
seyeon.print_my_moeny()
```

class 예제3

- 1. 계좌(Account)라는 클래스를 생성하시오.
- 2. 계좌(Account) 클래스의 인스턴스를 생성하시오.
- 3. 계좌(Account) 클래스에 아래와 같은 생성자를 추가하시오.
 - 1. 은행 이름, 예금주, 계좌번호, 잔액을 초기화 하시오.
 - 2. 생성자에서는 예금주와 초기 잔액만 입력 받으시오.
 - 3. 은행 이름은 "신한은행"으로 하시오.
 - 4. 계좌 번호는 3자리-2자리-6자리로 랜덤하게 생성되도록 하시오. (random 함수 사용, zfill 함수 사용)
- 4. 계좌(Account) 클래스에 입금을 위한 deposit() 함수를 추가하시오.
 - > 입금은 최소 1원 이상부터 가능
- 5. 계좌(Account) 클래스에 출금을 위한 withdraw() 함수를 추가하시오.
 - > 계좌잔고가 부족할 경우 출력이 불가능하다는 메시지 표시
- 6. Account 인스턴스에 저장된 정보를 출력하는 display_info() 함수를 추가하시오. (은행이름, 예금주, 계좌번호, 잔고 출력)
 - > 잔고는 3자리마다 쉼표를 출력하시오. (ex. 10,000)
- 7. 입금 횟수가 5회가 될 때 잔고를 기준으로 1%의 이자가 잔고에 추가되도록 (4)의 함수를 변경하시오.

```
입금 : 32000
잔고 : 92000
입금 : 20000
잔고 : 112000
입금 : 1000
잔고 : 113000
입금 : 15000
잔고 : 128000
출금 : 10000
잔고 : 119280.0
은행이름 : 신한은행
예금주 : 조세연
계좌번호 : 425-51-087831
잔고 : 119,280.0
```

예제3 - 문제풀이

```
from random import random
import random
```

```
class Account:
```

```
    def __init__(self, name, amount):
        self.deposit_count = 0
```

```
        self.name = name
```

```
        self.amount = amount
```

```
        self.bank = "신한은행"
```

```
        num1 = random.randint(0, 999)
```

```
        num2 = random.randint(0, 99)
```

```
        num3 = random.randint(0, 999999)
```

```
        num1 = str(num1).zfill(3)
```

```
        num2 = str(num2).zfill(2)
```

```
        num3 = str(num3).zfill(6)
```

```
        self.account_number = num1 + "-" + num2 + "-" + num3
```

```
    def deposit(self, amount):
        self.amount += amount
```

```
        self.deposit_count += 1
```

```
        if self.deposit_count % 5 == 0:
```

```
            self.amount = (self.amount * 1.01)
```

```
    def withdraw(self, amount):
```

```
        if self.amount > amount:
```

```
            self.amount -= amount
```

```
    def display_info(self):
```

```
        print("은행이름: ", self.bank)
```

```
        print("예금주: ", self.name)
```

```
        print("계좌번호: ", self.account_number)
```

```
        print("잔고: ", f"{self.amount:,}")
```

```
seyeon = Account("조세연", 10000)
```

```
seyeon.deposit(50000)
```

```
seyeon.deposit(32000)
```

```
seyeon.deposit(20000)
```

```
seyeon.deposit(1000)
```

```
seyeon.deposit(15000)
```

```
seyeon.withdraw(10000)
```

```
seyeon.display_info()
```


감사합니다:)