

# 프로젝트 제출

## 넓이 구하기

```
# 사각형 넓이를 구하는 클래스 완성!
class Square:
    def __init__(self):
        self.square = int(input('넓이를 구하고 싶은 사각형의 숫자를 써주세요.\n 1.직사각형 2.평행사변형 3.사다리꼴 \n'))

        if self.square == 1:
            print('직사각형 함수는 rect()입니다.')

        elif self.square == 2:
            print('평행사변형 함수는 par()입니다.')

        elif self.square == 3:
            print('사다리꼴 함수는 trape()입니다.')

        else:
            print('1, 2, 3 중에서 다시 입력해주세요')

    def rect(self):
        width, vertical = map(int, input('가로, 세로를 입력하세요. 예시 : 가로,세로\n >>>').split(','))
        area = width * vertical
        result = '직사각형의 넓이는 : ' + str(area)
        return result

    def par(self):
        width, height = map(int, input('가로, 세로를 입력하세요. 예시 : 가로,세로\n >>>').split(','))
        area = width*height
        result = '평행사변형의 넓이는 : ' + str(area)
        return result

    def trape(self):
        width,width_2, height = map(int, input('윗변, 아랫변,높이를 입력하세요. 예시 : 윗변,아랫변,높이\n >>>').split(','))
        area = 0.5*(width+width_2)*height
        result = '사다리의 넓이는 : ' + str(area)
        return result

a = Square() # 객체 생성 & 2, 3을 각각 입력해 봅시다.
```

## 2.키오스크

```
# 문제 3-3. 클래스 업데이트
# 앞에서 구현했던 메서드들을 Kiosk 클래스에 추가합니다. 직접 코드를 작성해 보세요!

class Kiosk:
    def __init__(self):
        self.menu = menu
        self.price = price

    # 메뉴 출력 메서드
    def menu_print(self):
        for i in range(len(self.menu)): # 메뉴의 개수만큼 반복해서
            print(i + 1, self.menu[i], ' : ', self.price[i]) # 메뉴와 가격을 출력합니다.

    # 주문 메서드
    def menu_select(self):
        self.order_menu = [] # 주문 리스트
        self.order_price = [] # 가격 리스트
```

```

n = 0
while n < 1 or len(menu) < n:
    n = int(input("음료를 번호를 입력하세요 : ")) # 음료 번호 입력

    # 메뉴판에 있는 음료 번호일 때
    if 1 <= n & n <= len(menu):
        self.order_price.append(self.price[n - 1]) # 가격 리스트에 추가합니다.
        self.price_sum = self.price[n - 1] # 합계 금액

        # 메뉴판에 없는 번호일 때
    else:
        print("없는 메뉴입니다. 다시 주문해 주세요.")

# 음료 온도 물어보기
t = 0 # 기본값을 넣어주고
while t != 1 and t != 2: # 1이나 2를 입력할 때까지 물어보기
    t = int(input("HOT 음료는 1을, ICE 음료는 2를 입력하세요 : "))
    # 문제 2-3의 답을 입력하세요.
    if t == 1:
        self.temp = 'hot'
    elif t == 2:
        self.temp = 'cold'
    else:
        print("1과 2 중 하나를 입력하세요.\n")

self.order_menu.append(self.temp + ' ' + self.menu[n - 1]) # 주문 리스트에 추가합니다.
# 문제 2-2의 답을 입력하세요.
print(self.temp, self.menu[n-1], ' : ', self.price[n-1], '원') # 온도 속성을 추가한 주문 결과를 출력

# 추가 주문 또는 지불
while n != 0: # 지불을 선택하기 전까지 반복합니다.
    print() # 줄 바꾸면서
    n = int(input("추가 주문은 음료 번호를, 지불은 0을 누르세요 : ")) # 추가 주문 또는 지불
    if n > 0 and n < len(self.menu) + 1:
        # 추가 음료 온도
        t = 0 # 기본값을 넣어주고
        while t != 1 and t != 2: # 1이나 2를 입력할 때까지 물어보기
            t = int(input("HOT 음료는 1을, ICE 음료는 2를 입력하세요 : "))
        # 문제 2-3의 답을 입력하세요.
        if t == 1:
            self.temp = 'hot'
        elif t == 2:
            self.temp = 'cold'
        else:
            print("1과 2 중 하나를 입력하세요.\n")
        # 문제 2-4. 추가 음료의 온도를 입력받고, 가격 리스트, 주문 리스트, 합계 금액을 업데이트해보세요.
        if 1 <= n & n <= len(menu):
            self.order_price.append(self.price[n - 1]) # 가격 리스트에 추가합니다.
            self.price_sum += self.price[n - 1] # 합계 금액

        print(
            '추가 주문 음료', self.temp, self.menu[n-1], ' : ', self.price[n-1], '원\n', '합계 : ',
        )
    else:
        if n == 0: # 지불을 입력하면
            print("주문이 완료되었습니다.")
            print(self.order_menu, self.order_price) # 확인을 위한 리스트를 출력합니다.
        else: # 없는 번호를 입력할 때
            print("없는 메뉴입니다. 다시 주문해 주세요.")

# 지불
def pay(self):
    np=0

```

```

        while np < 1 or np > 2:
            np = int(input('현금은 1번 카드는 2번'))
            if np == 1:
                print('직원 호출 ㄱㄷ')
            elif np == 2:
                print('ic 방향 맞게 잘 꽂아라')
            else:
                print('다시 결제 시도하셈')

# 주문서 출력
def table(self):
    for i in range(len(self.order_menu)):
        print(str(self.order_menu[i]) + ':' + str(self.order_price[i]) )

    print('합계금액:' + str(self.price_sum))

menu = ['americano', 'latte', 'mocha', 'yuza_tea', 'green_tea', 'choco_latte']
price = [2000, 3000, 3000, 2500, 2500, 3000]

a = Kiosk() # 객체 생성
a.menu_print() # 메뉴 출력
a.menu_select() # 주문
a.pay() # 지불
a.table() # 주문표 출력

```