Programming Languages Assignment6-A

2021312738 소프트웨어학과 김서환

```python
def main():
    store = Store(1235, "HomeStore", "123 Main St", "123-456-7890")
    print(f"create new store ! Name : {store.Name} / ID : {store.ID} / Address : {store.Address} / Tel : {store.Tel}")

    customer1 = Customer("102155222554", "Park JiMin", "321 Elm St", 0, "987-654-3210", ["yogiyo", "bbq"])
    customer2 = Customer("202155333554", "Jung SooJin", "654 Birch St", 0, "456-789-1234", [])
    print(f"create new staff ! SSN : {customer1.SSN} / Name : {customer1.Name} / Address : {customer1.Address} / Tel : {customer1.Tel} / Memberships : {customer1.Membership}")
    print(f"create new staff ! SSN : {customer2.SSN} / Name : {customer2.Name} / Address : {customer2.Address} / Tel : {customer2.Tel} / Memberships : {customer2.Membership}")

    staff1 = Staff(1, "111-22-3333", "Kim YoungJae", "456 Oak St", "Cashier", 30000)
    staff2 = Staff(2, "222-33-4444", "Lee MinHo", "789 Pine St", "Manager", 50000)
    print(f"create new staff ! ID : {staff1.ID} / SSN : {staff1.SSN} / Name : {staff1.Name} / Address : {staff1.Address}")
    print(f"create new staff ! ID : {staff2.ID} / SSN : {staff2.SSN} / Name : {staff2.Name} / Address : {staff2.Address}")

    products = [
        Product("013231788393", "PRODUCT", "Description 1", 8.16, 2),
        Product("012784545789", "PRODUCT", "Description 2", 5.56, 1),
        Product("007855114259", "MILK", "Description 3", 3.58, 1),
        Product("007874237152", "PRODUCT", "Description 4", 2.24, 1)
    ]

    order = Order(store, customer1, staff1)
```

메인 함수에서 Assignment6-A.docx에서 요구되었던 store 객체 1개, customer 객체 2개, staff 객체 2개, product 객체 4개, order를 store, customer, staff를 인자로 넘겨서 생성했다.

Product는 ProductCode, Name으로 구분되며, Assignment6-A.docx에 있던 RECEIPT 예시 중에서 아래의 사진처럼 ProductCode, Name 이 2개의 값이 아예 같은 product가 존재해서 따로 013231788393의 ProductCode를 가진 product를 만들어줬다.

```
PRODUCT      012784545789    8.16
PRODUCT      012784545789    5.56
```

그리고 order로 store, customer, staff, product, quantity를 넘겨줘야하는데 아직 물건을 산게 없어서 넘기지 않았다. (생성자에서 None으로 default값을 설정해줌, 추후에 addProduct로 구매한 물건과 물건 수량을 넘겨줄 예정)

```python
print("\nAvailable Products:")
for i, product in enumerate(products, 1):
    print(f"{i}. {product}")
```

그 후 살 수 있는 물품들을 보여준다.

```
while True:
    choice = input("\nEnter product number to add (or 'q' to finish): ")
    if choice.lower() == 'q':
        break
    product_idx = int(choice) - 1
    if 0 <= product_idx < len(products):
        qty = int(input(f"Enter quantity for {products[product_idx].Name}: "))
        order.addProduct(products[product_idx], qty)
    else:
        print("Invalid product number!")
print()
order.printReceipt()
```

그리고 숫자를 입력받아 사고 싶은 물품의 번호와 물품 수량을 입력받아서 order.addProduct(product, qty)로 넘겨준다.

만약 그만 구매하고 싶다면 q를 입력하면 종료된다.

또한, 숫자가 valid한 범위에 있지 않으면 "Invalid product number!" 메시지를 출력한다.

order.printReceipt()를 호출하면 Assignment6-A.docx에 있던 RECEIPT 예시처럼 출력된다.

```
Welcome to HomeStore
Staff: Kim YoungJae
Customer ID: 102155222554

RECEIPT
06/14/2025
14:38:02
ST # 1235
ST # 1235

ProductName     ProductCode     Price   Q
PRODUCT         007874237152    11.20   5
PRODUCT         012784545789    5.56    1
MILK            007855114259    17.90   5

TOTAL
$34.66
# ITEMS SOLD 11

TOTAL POINTS: 11

***CUSTOMER COPY***
```

<클래스 구조>

Store, Staff, Customer, Product 클래스는 그냥 문제에서 주어진 인자들을 전달받을 수 있는 생성자와 property(getter), setter, __str__을 각각 구현한 것 외에는 특별한 것은 없다.

Order 클래스는 다음과 같다.

```
class Order:
    def __init__(self, Store_object, Customer_object, Staff_object, Product_objects=None, Quantity=None):
        self.Store_object = Store_object
        self.Customer_object = Customer_object
        self.Staff_object = Staff_object
        self.Product_objects = Product_objects if Product_objects is not None else []
        self.Quantity = Quantity
        self.product_quantities = defaultdict(int)

    def addProduct(self, product, quantity):
        self.Product_objects.append(product)
        self.product_quantities[product] += quantity

    def printReceipt(self):
        total = 0
        total_points = 0
        self.Quantity = 0

        print(self.Store_object)
        print(self.Staff_object)
        print(self.Customer_object)
        print("\nRECEIPT")
        print(datetime.now().strftime("%m/%d/%Y \n%H:%M:%S"))
        print(f"ST # {self.Store_object.ID}\n")
        print("ProductName".ljust(15) + "ProductCode".ljust(15) + "Price".ljust(8) + "Q")

        for product, qty in self.product_quantities.items():
            subtotal = product.Price * qty
            total += subtotal
            total_points += product.Points * qty
            self.Quantity += qty
            print(f"{product.Name.ljust(15)}{product.ProductCode.ljust(15)}{f'{subtotal:.2f}'.ljust(8)}{str(qty).ljust(5)}")

        print(f"\nTOTAL\n${total:.2f}")
        print(f"# ITEMS SOLD {self.Quantity}")
        print(f"\nTOTAL POINTS: {total_points}")
        print("\n***CUSTOMER COPY***")

        self.Customer_object.Purchasing_Points = self.Customer_object.Purchasing_Points + total_points
```

Product와 Quantity를 None으로 지정해두고, addProduct를 통해서 구매한 물건과 수량을 넘겨준다. addProduct에서는 주문한 product들을 dictionary형태로 관리하여 같은 product를 구매하면 기존에 구매한 수량에 더해진 값을 저장할 수 있도록 계속 갱신했다.

구매를 마치고 q를 입력해서 빠져나오면 printReceipt()가 호출되는데, dictionary에서 하나씩 꺼내와서 해당 product와 수량을 출력해주고, total Price를 출력하고 Quantity를 order list의 total quantity로 나타내어 출력한다. 그 후 Point를 출력해주고, customer의 Perchasing_Points에 더해줘서 포인트를 적립한다. (물품 별 포인트는 제가 임의로 정했습니다.)

전체 실행에 대한 출력 결과는 다음과 같다.

```
create new store ! Name : HomeStore / ID : 1235 / Address : 123 Main St / Tel : 123-456-7890
create new staff ! SSN : 102155222554 / Name : Park JiMin / Address : 321 Elm St / Tel : 987-654-3210 / Memberships : yogiyo bbq
create new staff ! SSN : 202155333554 / Name : Jung SooJin / Address : 654 Birch St / Tel : 456-789-1234 / Memberships :
create new staff ! ID : 1 / SSN : 111-22-3333 / Name : Kim YoungJae / Address : 456 Oak St
create new staff ! ID : 2 / SSN : 222-33-4444 / Name : Lee MinHo / Address : 789 Pine St

Available Products:
1. PRODUCT        013231788393   8.16
2. PRODUCT        012784545789   5.56
3. MILK           007855114259   3.58
4. PRODUCT        007874237152   2.24

Enter product number to add (or 'q' to finish): 4
Enter quantity for PRODUCT: 3

Enter product number to add (or 'q' to finish): 5
Invalid product number!

Enter product number to add (or 'q' to finish): 2
Enter quantity for PRODUCT: 1

Enter product number to add (or 'q' to finish): 4
Enter quantity for PRODUCT: 2

Enter product number to add (or 'q' to finish): 3
Enter quantity for MILK: 5

Enter product number to add (or 'q' to finish): q

Welcome to HomeStore
Staff: Kim YoungJae
Customer ID: 102155222554

RECEIPT
06/14/2025
14:38:02
ST # 1235

ProductName    ProductCode    Price   Q
PRODUCT        007874237152   11.20   5
PRODUCT        012784545789   5.56    1
MILK           007855114259   17.90   5

TOTAL
$34.66
# ITEMS SOLD 11

TOTAL POINTS: 11

***CUSTOMER COPY***
```