Assignment 6A: Store Receipt

Programming Language SWE3006_41

2022310853 Younwoo Park

Dept. of System Management Engineering

Sungkyunkwan University

June 11, 2025

Overview

This program implements a simplified selling software system for a retail store chain, following the specifications provided in the assignment. The primary objective was to model the interactions between a store, its staff, customers, and products, culminating in the generation of a detailed receipt for customer purchases. The implementation was done using Python, incorporating object-oriented programming principles such as inheritance, encapsulation, and abstraction.

Structure and Classes

To ensure data encapsulation and controlled access, all instance variables are private and are accessed via properties and setter methods. Abstract classes are properly utilized to enforce class structure and reusability, following best practices in object-oriented design.

1. Citizen, Customer, Staff

The core structure of the application is composed of several classes. The `StoreChain` abstract base class defines common attributes such as store name and address. It is extended by the `Store` class, which includes additional information such as store ID and telephone number. The `Citizen` abstract base class represents individuals in the system and encapsulates shared fields such as name, social security number, and address. This class is extended by both the `Customer` and `Staff` classes. The `Customer` class introduces customer-specific attributes such as customer ID, telephone number, purchasing points, and a list of membership types. The `Staff` class defines a staff ID, job title, and salary.

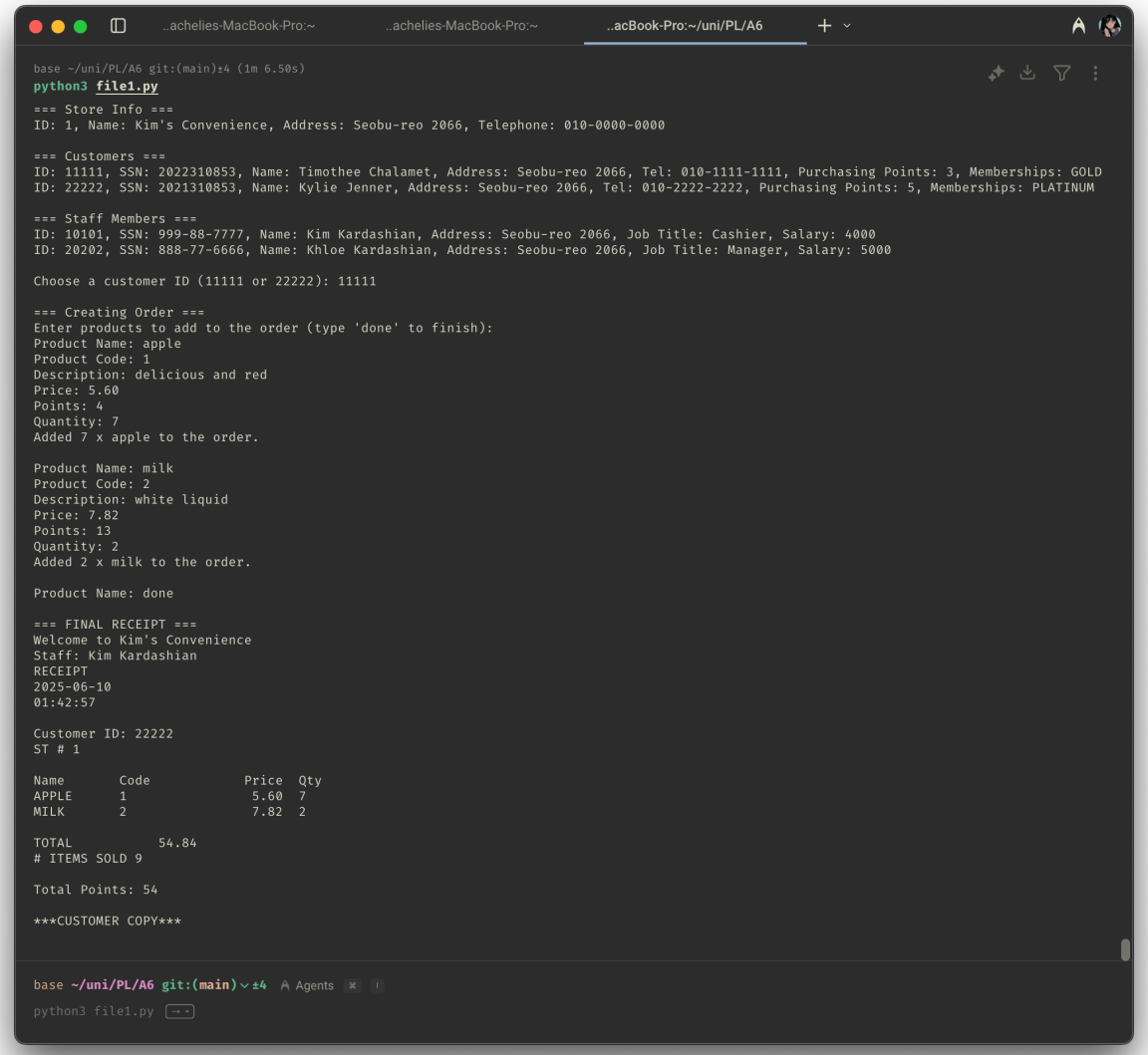
2. Product, Order

The `Product` class encapsulates all necessary information about an item sold by the store, including a product code, name, description, price, and reward points. Each `Order` object is a composite that ties together a store, customer, staff member, a list of products, and their corresponding quantities. The `Order` class includes methods for adding products to the order and generating a receipt, which presents detailed information including store ID, customer ID, itemized product listings with price and quantity, total cost, and points earned.

3. Console Appliactions (i.e. `main()`)

The `main()` function serves as the entry point of the application. It creates a store instance, two customer objects, and two staff members. It then allows the user to select a customer by entering a valid customer ID. The system prompts the user to input product information interactively. Once all products are added to the order, the program prints a formatted receipt that closely resembles the sample provided in the assignment.

Program Output



```
base ~/uni/PL/A6 git:(main)±4 (1m 6.50s)
python3 file1.py
=== Store Info ===
ID: 1, Name: Kim's Convenience, Address: Seobu-reo 2066, Telephone: 010-0000-0000

=== Customers ===
ID: 11111, SSN: 2022310853, Name: Timothee Chalamet, Address: Seobu-reo 2066, Tel: 010-1111-1111, Purchasing Points: 3, Memberships: GOLD
ID: 22222, SSN: 2021310853, Name: Kylie Jenner, Address: Seobu-reo 2066, Tel: 010-2222-2222, Purchasing Points: 5, Memberships: PLATINUM

=== Staff Members ===
ID: 10101, SSN: 999-88-7777, Name: Kim Kardashian, Address: Seobu-reo 2066, Job Title: Cashier, Salary: 4000
ID: 20202, SSN: 888-77-6666, Name: Khloe Kardashian, Address: Seobu-reo 2066, Job Title: Manager, Salary: 5000

Choose a customer ID (11111 or 22222): 11111

=== Creating Order ===
Enter products to add to the order (type 'done' to finish):
Product Name: apple
Product Code: 1
Description: delicious and red
Price: 5.60
Points: 4
Quantity: 7
Added 7 x apple to the order.

Product Name: milk
Product Code: 2
Description: white liquid
Price: 7.82
Points: 13
Quantity: 2
Added 2 x milk to the order.

Product Name: done

=== FINAL RECEIPT ===
Welcome to Kim's Convenience
Staff: Kim Kardashian
RECEIPT
2025-06-10
01:42:57

Customer ID: 22222
ST # 1

Name       Code            Price  Qty
APPLE      1                5.60  7
MILK       2                7.82  2

TOTAL           54.84
# ITEMS SOLD 9

Total Points: 54

***CUSTOMER COPY***
```

The program also handles invalid input gracefully using exception handling and input validation. Date and time stamps are included on the receipt using Python's `datetime` module. The output structure and formatting were designed to match the example provided, ensuring compliance with the assignment's user interface expectations.

Conclusion

Overall, the application meets all functional requirements described in the assignment. It adheres to object-oriented design principles, produces correct and user-friendly output, and supports future extension by maintaining a clean class structure. This implementation successfully demonstrates how basic software design concepts can be applied to model real-world systems like retail management.