James Costello | A00326601

Object Oriented Programming 2

Assignment

OOP 2 Project Report

Introduction

For the OOP2 project, we reused the OOP1 code base. This application is based on a restaurant. Where the customer can view a range of menus from vegan, kids, main, and special menu.

Where the customer can place an order and pay. The restaurant also has employees such as waiters and managers along with contractors and delivery drivers.

For the OOP2 project we added a rewards card and offers class to the project.

List Of User Stories

Below is a list of user stories assigned as part of the assignment and completed.

Code	Descriptions	Status
[01_00P2]	Lambdas. for example: Consumer, Predicate, Supplier, Functio etc.	Done

[02_00P2]	Streams terminal operations such as.
	min(), max(), count(), findAny(), findFirst(), allMatch(), anyMatch(), noneMatch(), forEach()
	collect() - Collectors.toMap(), Collectors.groupingBy() and Collectors.partitioningBy()
		Done

[03_00P2]	Streams intermediate operations e.g. filter(), distinct(), limit(), map() and sorted()	Done
[04_00P2]	Switch expressions and pattern matching.	Done
[05_00P2]	Sealed classes and interfaces.	Done
[06_00P2]	Date/Time API.	Done
[07_00P2]	Records.	Done
[A1_00P2]	Collections/generics use of Comparator.comparing() for sorting.	Done
[A2_00P2]	Concurrency e.g. using ExecutorService to process Callables.	Done
[A3_00P2]	NIO2.	Done

Evaluation

There were no major problems doing the assignment. There were a couple of challenges, these are as follows.

- From the original UML design and the additional UML added for OOP2, which is on my GIT repo, this helps to make it easier to follow a plan but in trying to accommodate all the requirements and applying to the OOP1 assignment it became harder. It would have been easier to start fresh than trying to reuse the previous assignment.

- A better designed UML design could have made it easier to refactor and add new features. Along with a clearer structure that would have made the code easier to follow. Additionally, focusing on reusability whether by adding extra methods within a class or creating standalone functional methods would have contributed to cleaner code. This approach would not only enhance readability but also make the code more modular and reusable.

References

GIT Repo -> [https://github.com/jc6310/OOP1-Assignment]

Video -> [OOP2 Assignment Project.mp4]