James Costello | A00326601

Object Oriented Programming 1

Assignment

OOP1 Project Report

Introduction

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.

List Of User Stories

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

Code	Descriptions	Status
[01]	Classes – a) contrast this() and this b) method overloading c) Varargs d) LVTI	DONE
[02]	Encapsulation	DONE
[03]	Interfaces	DONE
[04]	Inheritance – a) overriding and polymorphism b) contrast super() and super	DONE
[05]	Exceptions (checked and unchecked)	DONE
[06]	Enums	DONE
[07]	Arrays	DONE
[08]	Use of Java Core API (String, StringBuilder, List/ArrayList, Date API)	DONE

[A1]	Call-by-value and defensive copying	DONE
[A2]	Private, default and static interface methods	DONE
[A3]	Records	DONE
[A4]	Custom immutable type	DONE
[A5]	Lambdas (Predicate) - a) discussion of 'final' or 'effectively final' b) method references	DONE
[A6]	Switch expressions and pattern matching	DONE
[A7]	Sealed classes and interfaces	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, which is on my GIT repo, which makes it easier to follow a basic plan to get started but in trying to accommodate all the requirements for the assignment it gradually drifted off the original design completely. It would have been better if I had reviewed and got a better understanding of the requirements for the assignment. As then the UML design would have been a better fit. And less drifting off the original design.

- A better UML design, which would have helped with making the code cleaner in terms of more readable and easier-to-follow the code. With code been made more readable also the code could have been written with reusability in mind. That could mean to add extra methods in a class or just as a functional method. This could have helped with readable and making it easier to follow the code.

- I think that the waiter, manager, and employee classes maybe a little tightly coupled between the 3. This may make the code harder to maintain and hard to modify as there depend too much on each other.

References

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

Video -> [.]