UBER -CAB -BOOKING APPLICATION

DOCUMENTATION

Hardik Jain (2020A7PS0102P) **Sarvesh Gupta (2020A7PS0093P)**

Critical Analysis of the OOP Principles:

1)Encapsulate what varies

Getter and setter methods are implemented for classes Landmark , cab state ,cab id ,changing cab state , setting new location ,getting Location.

objects in classes are made private .(expect driver.x and y coordinate)

2)Classes should be open for extension and closed for modification

According to the Open-Closed Principle (OCP), software entities (classes, modules, methods, and so on) should be open for extension but closed for modification. In practise, this means developing software entities whose behaviour can be modified without editing and recompiling the code itself.

Yes Classes (Location) are used as extension without modifying them.

3)Depend on abstraction, do not depend on concrete classes

Abstract Classes were not required so the code mostly is based on concrete classes as we create objects of these classes and use them for calling methods.

4)Strive for loose coupling between objects that interect

More than 90% of the coupling is loosely done .

For Example :- passing just the cab id ; passing just the coordinates as integers and almost every function is loose coupled .

Analysis of Design Pattern

Command Pattern is used . I can state this as:

This is a behivorial design pattern in which the information needed to perform an action is encapsulated , using an object.

1. Command execution logic gets decoupled from the User.

User awareness remains till command creation only. What

It internally is an ***abstraction***.

2. **Open/Closed Principle is followed** ensuring you can

add new commands without breaking code.

3. Single responsibility is maintained by each

Command , Keeping the code readable.