

Assignment #6

MET CS 665 - Software Design and Patterns Clean Your Code

For the purposes of this refactoring assignment, I chose to make some changes to **Assignment 2 - Delivery Application - Notification System**. The original assignment reflected the agreement a consortium of shops had with local independent van and taxi drivers to deliver their products to the customer's destinations. This implements a notification system to broadcast notifications about delivery requests to all the drivers in their system.

Refactor #1

Using the principles of "Generalization 1*" I implemented the Pull up method and moved the `registerDriver()`, `removeDriver()` and `sendDeliveryRequest()` methods to the Subject (`ShopInterface.java`) class to better demonstrate the Observer Pattern as it did not belong in the concrete class as it was originally. I used the refactor feature in IntelliJ for pull up methods.

Refactor #2

Similarly, I pulled up the data to the `DeliveryRequest.java` class to allow us to add more data to a request including location/address and the confirmation number. I used the refactor feature in IntelliJ for pull up methods.

Refactor #3

Implemented an "extract a delegate method" refactor for printing out the delivery request; `printDelivery()`. I changed method headers to reflect this alteration with the required change in parameters. I renamed the original `DeliveryRequest` interface to `DriverInterface` and replaced `driverRequest()` with `update()` in the `Shop` class and files with other occurrences. I made a different `DeliveryRequest` class for abstracting out the data. I used the refactor feature in IntelliJ for delegation.

Renamed driverinterface

3) Extract a delegate method for printing out the delivery request (printDelivery()). We used the IntelliJ feature for extraction.

Changed method headers to reflect this alteration with the required change in parameters.

DeliveryRequest.java - Refactor # 2 - Extracting a delegate class called DR.java.
This allows us to add more data to the delivery request (location, delivery confirmation number,)

Delegation: using the refactor feature in IntelliJ for the pulling up method and delegation.

Rename driverInterface and replaced driverrequest() with update() in shop.java /otherfiles
recheck

2) Pulling up the data into another class - This allows us to add more data to the
request(location/address, confirmation number)

1) Refactor1: - Pulling up the methods registerDriver() -notifyAll() Pull up method - dealing with
generalization 1 - slide 50

Shop.java - Shop interface needed the reigster(), unregoster()
And notifyAll() for the observer pattern not the concrete class.