Quiz 3

Subject : Inheritence and Polimorphism
Due Date : 26.04.2020 23:59

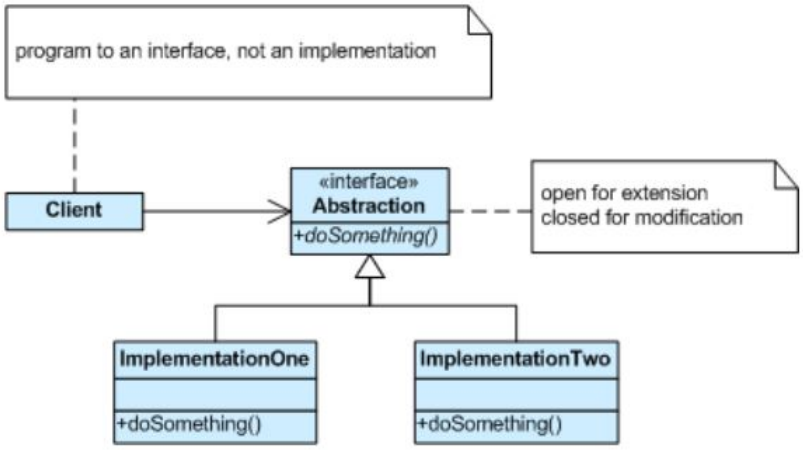
Strategy Pattern

Intent.

- Define a family of algorithms, encapsulate each one, and make them interchangeable. Strategy lets the algorithm vary independently from the clients that use it.
- Capture the abstraction in an interface, bury implementation details in derived classes.

Structure.

One of the dominant strategies of object-oriented design is the "open-closed principle". Figure demonstrates how this is routinely achieved - encapsulate interface details in a base class, and bury implementation details in derived classes. Clients can then couple themselves to an interface, and not have to experience the upheaval associated with change: no impact when the number of derived classes changes, and no impact when the implementation of a derived class changes.



Polymorphism through Strategies

The strategy pattern, also known as the policy pattern, is a behavioral design pattern that lets an object execute some algorithm (strategy) based on external context provided at runtime. This pattern is particularly useful when you have an object that needs to be able to execute a single behavior in different ways at different times. By using the strategy pattern, you can define a set of algorithms that can be dynamically provided to a particular object if/when they are needed. This pattern has a number of benefits, including: encapsulation of particular algorithms in their own classes; isolation of knowledge about how algorithms are implemented;

and, code that is more flexible, mobile, and maintainable. To the last point, you may note that these are the same attributes that result from code that follows the Open/Closed Principle (OCP) and indeed, the strategy pattern is an excellent way to write OCP-adherent code.

Problem

Suppose you are developing an online reservation system that can provide airport transportation service to the users. There are several options for airport transportation including rent a car, city bus, taxi, airport shuttle and limousine service. Making reservation is performed as follows: After the user selects preferred transportation, your system should automatically handle all bookings by using several vendors. Each vendor has its own API. Each transportation method will require communication with a different third party API. First user will choose vendor, then transportation. According to vendors strategies, scenario will be happen. When the bookings are done, the system will output the reservation details containing route and timing information. Since we are using different vendors, there will be different information at the final confirmation page.

I need to implement make reservation functionality to support many different vendors in the future but I do not want a junior developer to mess with the order of actions to be taken when making reservation.

It is expected from you to give a **UML class diagram** showing the pattern usage in case of the pattern structure.

Below, you can find the output for the implementation of this problem.

```
**********Vendor1 Interface**********

Route detail: Preferred transportation Rent a Car Service is selected by user.
Route detail: Rent a Car Service From Aşti to the Airport.
The timing information: 65 minutes.
Route detail: Preferred transportation Taxi Service is selected by user.
Route detail: Taxi Service From Aşti to the Airport.
The timing information: 85 minutes.
Route detail: Preferred transportation Airport Shuttle Service is selected by u
Route detail: Airport Shuttle Service From your home to the Airport.
The timing information: 240 minutes.
Route detail: Preferred transportation Limousine Service is selected by user.
Route detail: Limousine Service From Kızılay to the Airport.
The timing information: 230 minutes.
Route detail: Preferred transportation City Bus Service is selected by user.
Route detail: City Bus Service From YHT to the Airport.
The timing information: 130 minutes.

**********Vendor2 Interface**********

Route detail: Preferred transportation Rent a Car Service is selected by user.
Route detail: Rent a Car Service From Aşti to the Airport.
The timing information: 80 minutes.
Route detail: Preferred transportation Taxi Service is selected by user.
Route detail: Taxi Service From Aşti to the Airport.
The timing information: 100 minutes.
Route detail: Preferred transportation Airport Shuttle Service is selected by u
Route detail: Airport Shuttle Service From your home to the Airport.
The timing information: 315 minutes.
Route detail: Preferred transportation Limousine Service is selected by user.
Route detail: Limousine Service From Kızılay to the Airport.
The timing information: 305 minutes.
Route detail: Preferred transportation City Bus Service is selected by user.
Route detail: City Bus Service From YHT to the Airport.
The timing information: 160 minutes.
```

Notes

- Do not miss the submission deadline.
- Save all your work until the quiz is graded.
- You can ask your questions via Piazza and you are supposed to be aware of everything discussed on Piazza.
- You must submit your work with the file hierarchy as stated below:

→ <student id.zip>
  – report.pdf