Software Design Pattern

Assignment 1

Submitted by :

Fabliha Anber (Roll 56)

Assumptions of the classes:

VehicleDescription:

Attributes:
manufacturingDate : String, model : String, year : Integer, manufacturer : String.

Methods:
getManufacturingDate() : String,getModel() : String, getYear() : Integer,getManufacturer() : String ,setManufacturingDate(manufacturingDate : String) , setModel(model : String), setYear(year : Integer), setManufacturer(manufacturer : String)

Assumption: The vehicleDescription class contains the details of the car set by the owner during registration details. It has the methods of getters and setters of these attributes to set the car details and to get the details to print in Vehicle class.

Vehicle:

Attributes:
vehicleID : Integer, vehicleLicensePlate : Integer

Contructor:
Vehicle(vehicleDescription : VehicleDescription)

Methods:
addVehicleDetails(), showVehicleDetails()

Assumption:
The Vehicle class has the ID of the vehicle of the owner and vehicle details can be added and shown in this class. Every vehicle has one vehicle description, so it has a 1-1 relationship with VehicleDescription class.

Owner:

Attributes:
ownerID : Integer, ownerName : String, ownerAge : Integer, owneremail : String, ownerAddress : Integer, ownerCity : Integer, ownerVehicleID : Integer

Constructor:
Owner(vehicle : Vehicle)

Methods:
showOwnerDetails(), addOwnerDetails(),provideVehicleDetails()

Assumption:
The Owner class has the attributes relating to car owner description and it is assumed that every owner has one vehicle. So, there is 1-1 relationship with Owner class and Vehicle class.

VehicleRegistration:

Attributes:
registrationID : Integer, registrationName : String, registrationDate : String, registrationType : String, cityOfRegistration : String, countryOfRegistration : String

Constructor:
VehicleRegistration(vehicle : Vehicle), VehicleRegistration(owner : Owner)

Methods:
showVehicleRegistrationDetails(), searchVehicleRegistrationDetails(), addVehicleRegistrationDetails(), renewVehicleRegistrationDetails(), updateVehicleRegistrationDetails(), checkCityChange(), checkValidRegisterUser(), provideUpdatedDetails()

Assumption:
The VehicleRegistration class contains the attributes relating to the registration and the city in which the register was done is also stored. This class shows the vehicle registration details and interacts with the database to store new register information as well as renewed register information. Every VehicleRegistration class can have one or many Owners and Vehicles and the relationship 1 to many is drawn. The VehicleRegistration class cannot exist without Owner and Vehicle, so composition relation is used. This class checks if the register is valid everytime an actor comes to renew his/her register and check their city of registration and updates accordingly in the database. This class searches for registration details and adds vehicle registration for new registers.

Actor:

Methods:
newRegistrationApply(), renewRegistrationApply(), provideUpdateddetails(), provideOwnerDetails(), provideCarDetails(), isCityChanged() : Boolean, addOwnerDetails(), addVehicleDetalils(), checkCityChange()

Assumption:
This class is the person who interacts with the registration class. The actor provided vehicle details and owner details for new registration and updated information (along with the city of registration) for renewed registration

Database:

Attributes:
updatedCityOfRegistration : String

Methods:
registrationSuccessful(), updateRegistrationSuccessful(), isValidRegisterUser() : Boolean, addVehicleRegistrationDetails()

Assumption:
This Database class stores and updates registration information about the actor and sends registration successful or updated registration successful information.