**Raditya Surya Pratama**

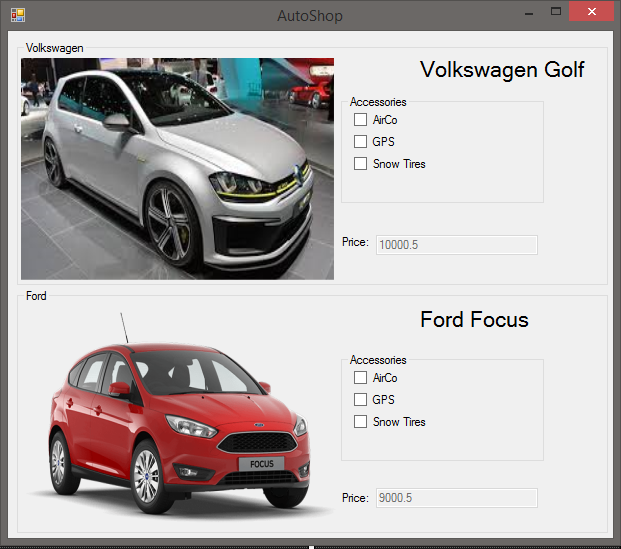
**Radu Alexandru Stoica**

AUTO SHOP

Decorator Pattern

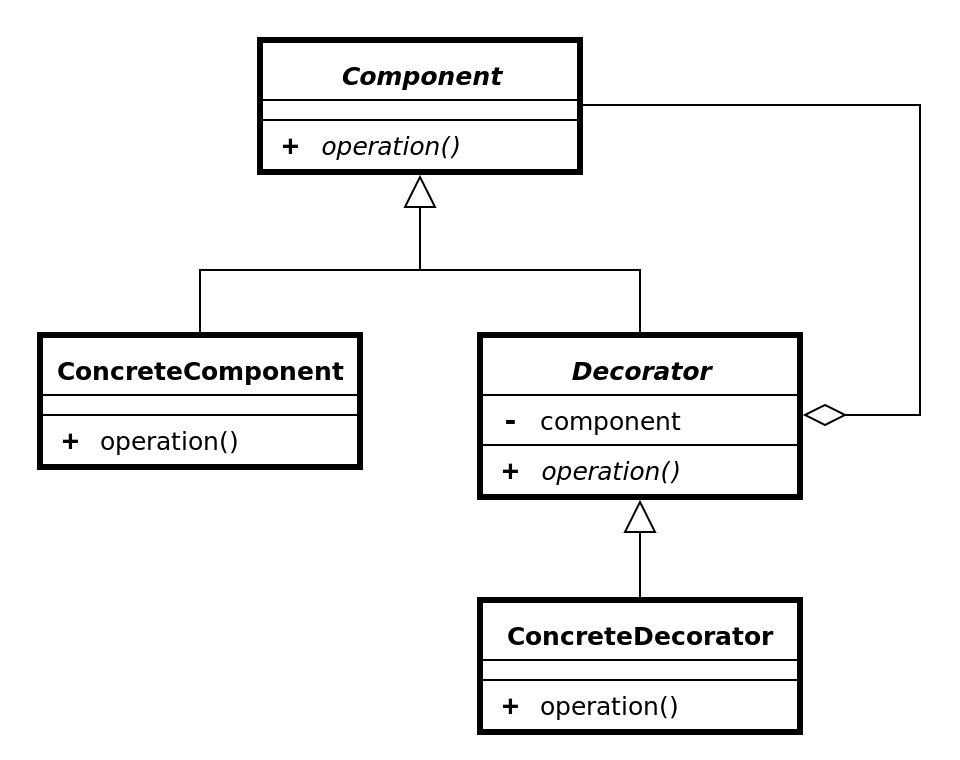
INTRODUCTION

The third assignment for DPR is to implement the decorator pattern by making an auto shop application. This application is working to simulate how the car default price could be overridden if the buyer choose to add more accessories.



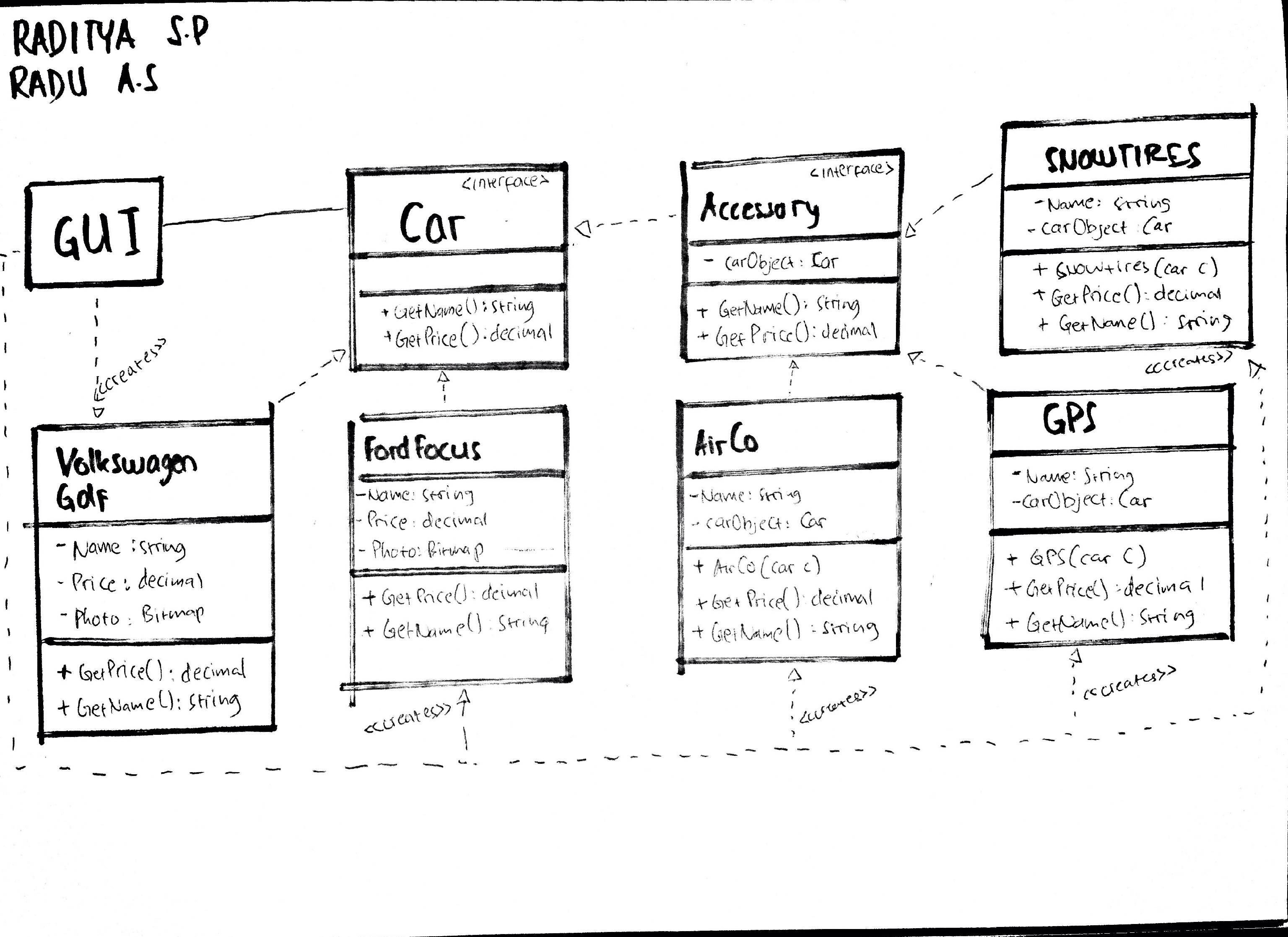
DECORATOR PATTERN

The decorator pattern is a software design pattern that allows behavior to be added to an individual object, either statically or dynamically, without affecting the behavior of other objects from the same class. Decorator provide a flexible alternative to sub class for extending functionality.



Source: <https://en.wikipedia.org/wiki/Decorator_pattern>.

UML DIAGRAM



REUSABILITY

The reusability of our code is represented by the classes that available in the application could be used or reused by another component.

In our application, the Accessory decorator (interface) made it possible if another application or class wants to use the accessory method. It is only possible if the other class or application is related into car accessories, otherwise it won’t work unless the code is modified.

EXTENSIBILITY

For the extensibility, we can extend the application by extending other classes. In our application, we could add more classes in decorator class, such as adding decorator class to add car with a Turbo.

MAINTAINABILITY

For the maintainability matters in the application, the classes could be easily changed or updated. If the user does not want to offer car with Snow tires, we could just remove the decorator class and the application would not include it in the price calculation.

PATTERN DOWNSIDE

There are a couple of disadvantages of using the decorator pattern:

* Create many small objects in the design and overuse can be complex
* Can cause issues if the client relies heavily on the components concrete type
* Complicate the process of instantiating the component
* Hard to keep track of other decorators

Source: [https://neillmorgan.wordpress.com/2010/02/07/decorator-pattern-pros-and-cons/.](https://neillmorgan.wordpress.com/2010/02/07/decorator-pattern-pros-and-cons/.%20)

UNIT TEST

The unit test is included in the project solution.

Two cars object were created, we test the price of the car before and after adding a new accessories. After running the test, all the test is passed.