Software Design I
Season 2024-I
Workshop No. 3 — Structural Design Patterns

Eng. Carlos Andrés Sierra, M.Sc.
Computer Engineering
Universidad Distrital Francisco José de Caldas

Welcome to the third workshop of Software Design I course, congratulations for keep pushing.

As you remember, you had been hired as software engineer in a vehicle constructor company. You are checking the current version of the internal tool used to manage the vehicles created by the company, and you think *Class Diagram* (Figure 1) is not the best way to represent the vehicles and their attributes.

Thus, you must deliver a new version of this internal tool, improving anything you could.

Also, some requirements have been added after first final-user interactions. Changes are described as follows:

1. Application is still consuming a lot of memory, so you must reduce it as much as possible.

2. An authentication system is needed, so you must create a simple login system where you could register users and authenticate them.

3. You should differentiate two types of users: *Admin* and *User*. Admins could create, update, and delete vehicles, while Users could only see and search the vehicles.

4. You should log all the actions made by the users, so you could track both the changes made in the vehicles and searches performed.

5. Separate your project in different subsystems, and a made a simple interface to interact with them.

Carlos Andrés Sierra, Computer Engineer, M.Sc. on Computer Engineering, Titular Professor at Universidad Distrital Francisco José de Caldas.

Any comment or concern related to this document could be send to Carlos A. Sierra at e-mail: *cavirguezs@udistrital.edu.co*

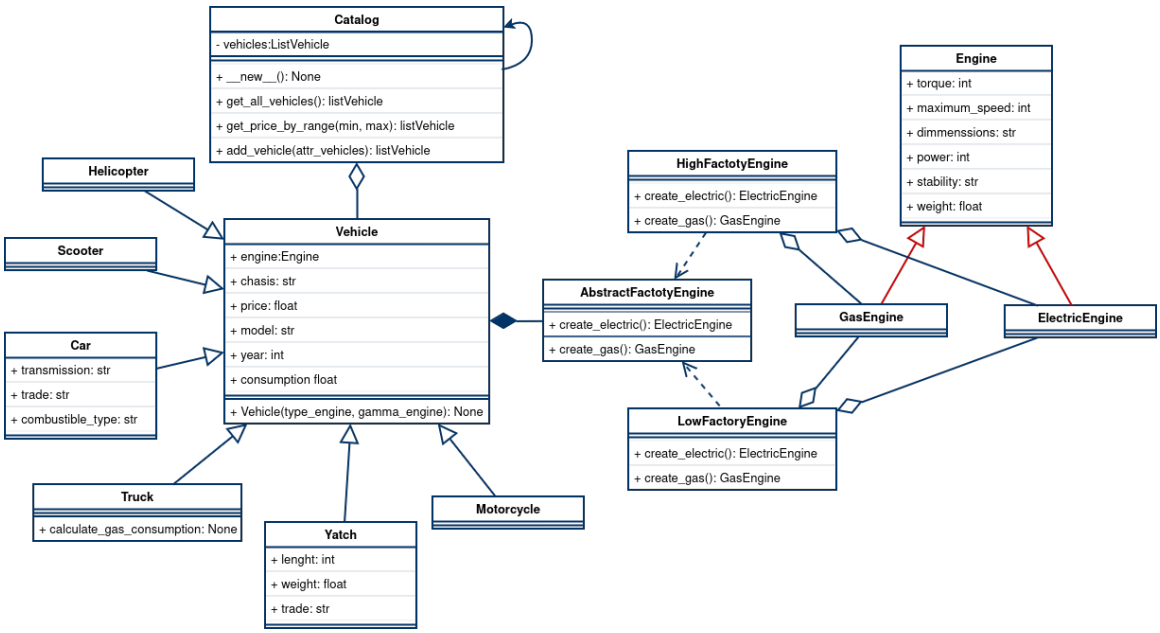

Figure 1. Current Class Diagram v2.

6. In addition, you must create a monitoring system to check the memory consumption and time execution of the searches in the application, and save stats in a *performance_log* file.

7. To increase performance, you should implement a cache system to store the last five search results over vehicles.

8. Verify if you could add encapsulation to increase the security of the application.

You must deliver a technical report where a Class Diagram of your solution is provided, also all related updated additional documentation; here it is recommended to think in components, define a diagram por each component where connections with other components will be absolutely clear. Also, you must write about technical concerns and decisions you make to create the architecture you are proposing; this includes SOLID implementation analysis of your model, coupling, cohesion, and other software engineering principles.

You must create the code and provide a simple *client interface* with a menu to create vehicles, show all vehicles registered, and make searches by maximum speed, type of combustion, and range of years, showing vehicles and them specific information independently of the type.