

# Software Design I

## Season 2024-I

### Workshop No. 2 — Creational Design Patterns

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

Welcome to the second workshop of Software Design I course, congratulations to survive to first challenge.

As you remember, you had been hired as software engineer in a vehicle constructor company. You delivered a first version for the catalog of vehicles, and it was fine. However, the CTO of the company thinks it is possible to get a new catalog version increasing quality and applying better software practices.

Thus, he is asking you to 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. Cars must have next attributes: transmission, chassis, year, trade, model, combustible type, and engine.
2. Yacht must have next attributes: length, weight, year, trade, model, engine.
3. Trucks and Motorcycles will keep same attributes as before. However, just for trucks you need to calculate the gas consumption due to regulations of both USA and Europe.
4. You need to add next vehicles options, and you must define the appropriate attributes they need: Helicopter, Scooter.
5. Create engines in the platform is not a good idea due to possible human errors. So, it is a better idea to have some predefined engines per each one of the vehicles types, divider in gas and electric engines. Also, per each vehicle you should define low gama and high gama engine.

---

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: *cavir-guezs@udistrital.edu.co*

6. All engines must have next attributes: stability, power, weight, dimensions, torque, and maximum speed.
7. You must reduce memory as much as possible, so check where you could reduce the creation of duplicates objects to avoid. Be careful with memory references.

You must deliver a technical report where a Class Diagram of your solution is provided; 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 .

You must create the code and provide a simple 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.

- Sorry, something had been missed: every vehicle must have a price, and in the searches the users would like to search by price range.