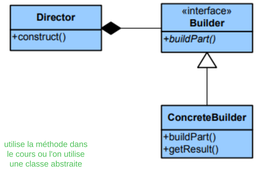
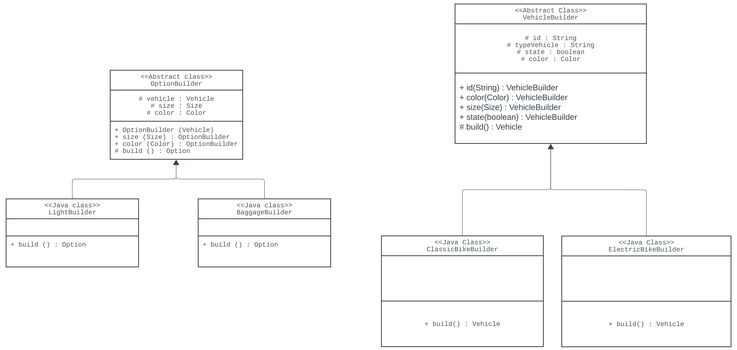
Builder

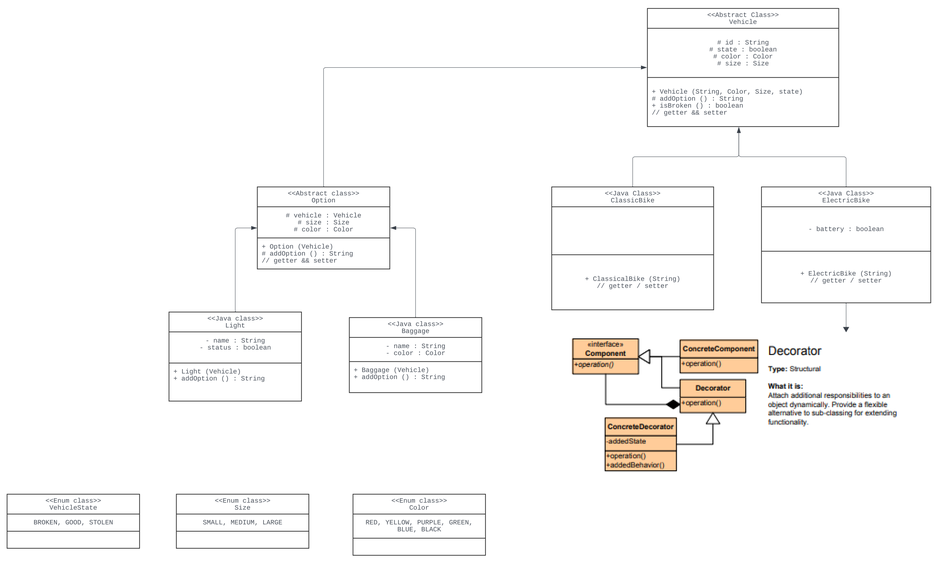
Type: Creational

What it is:
Separate the construction of a complex object from its representing so that the same construction process can create different representations.



utilise la méthode dans le cours ou l'on utilise une classe abstraite pour les builder





Decorator

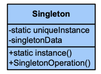
Type: Structural

What it is:
Attach additional responsibilities to an object dynamically. Provide a flexible alternative to sub-classing for extending functionality.

Singleton

Type: Creational

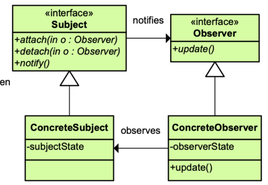
What it is:
Ensure a class only has one instance and provide a global point of access to it.

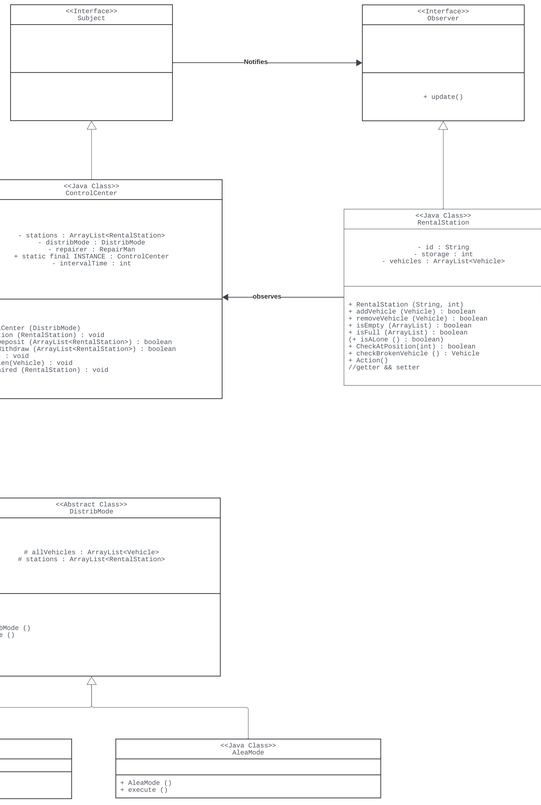


Observer

Type: Behavioral

What it is:
Define a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.

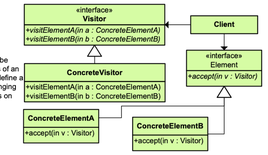




Visitor

Type: Behavioral

What it is:
Represent an operation to be performed on the elements of an object structure. Lets you define a new operation without changing the classes of the elements on which it operates.



Strategy

Type: Behavioral

What it is:
Define a family of algorithms, encapsulate each one, and make them interchangeable. Lets the algorithm vary independently from clients that use it.