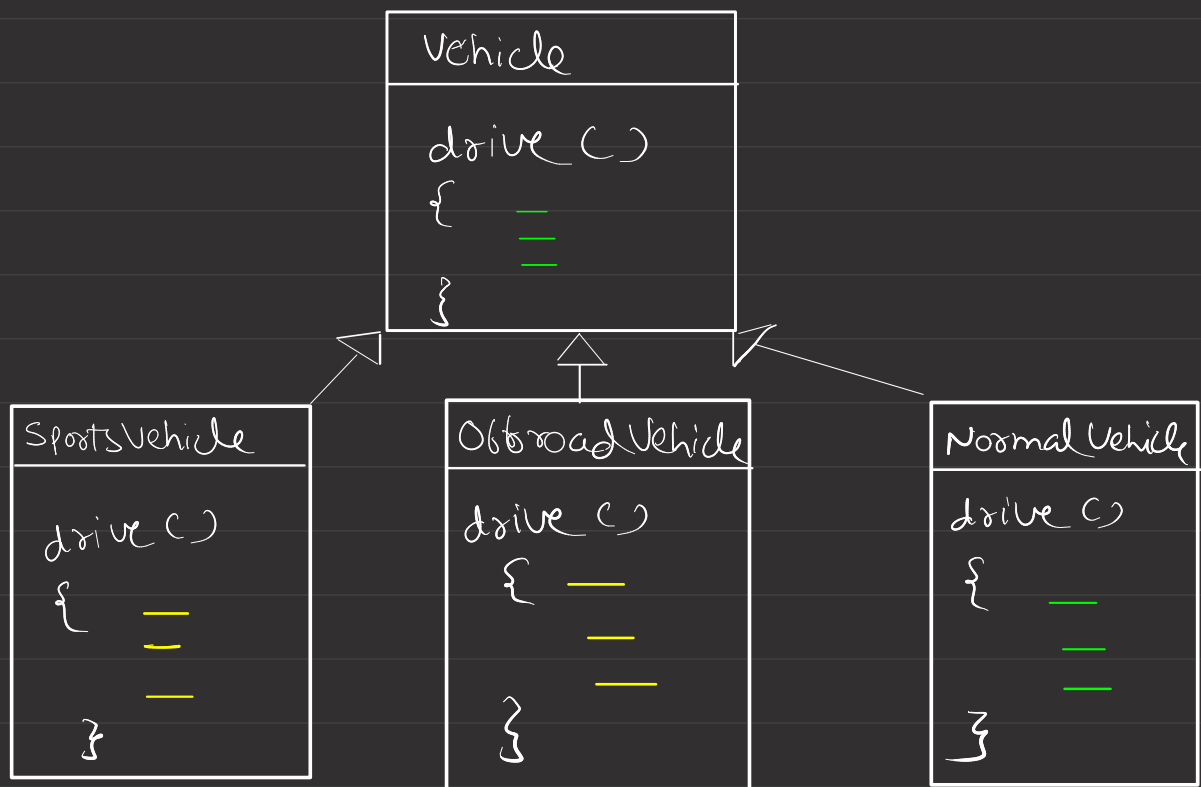# Strategy Design Pattern

This design pattern helps in increasing code reusability among different classes which are child of same parent class

Eg:-
        ——▷ is-a relationship
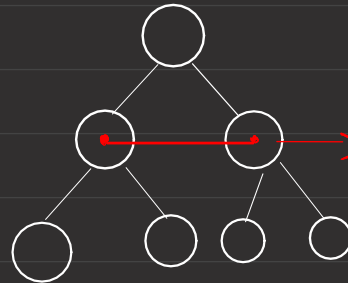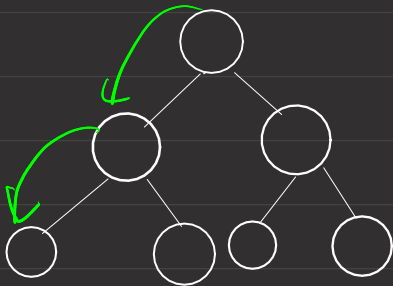   ——→ has-a relationship

let's say we have a system of vehicles.

In the above scenario, we have a Vehicle class, with a driv method in it. 3 classes, inherits it as shown above.

Now the problem with above is that, the SportsVehicle & OffroadVehicle classes have same implementation for drive, which is problematic. Coz if these methods/fields ↑ then
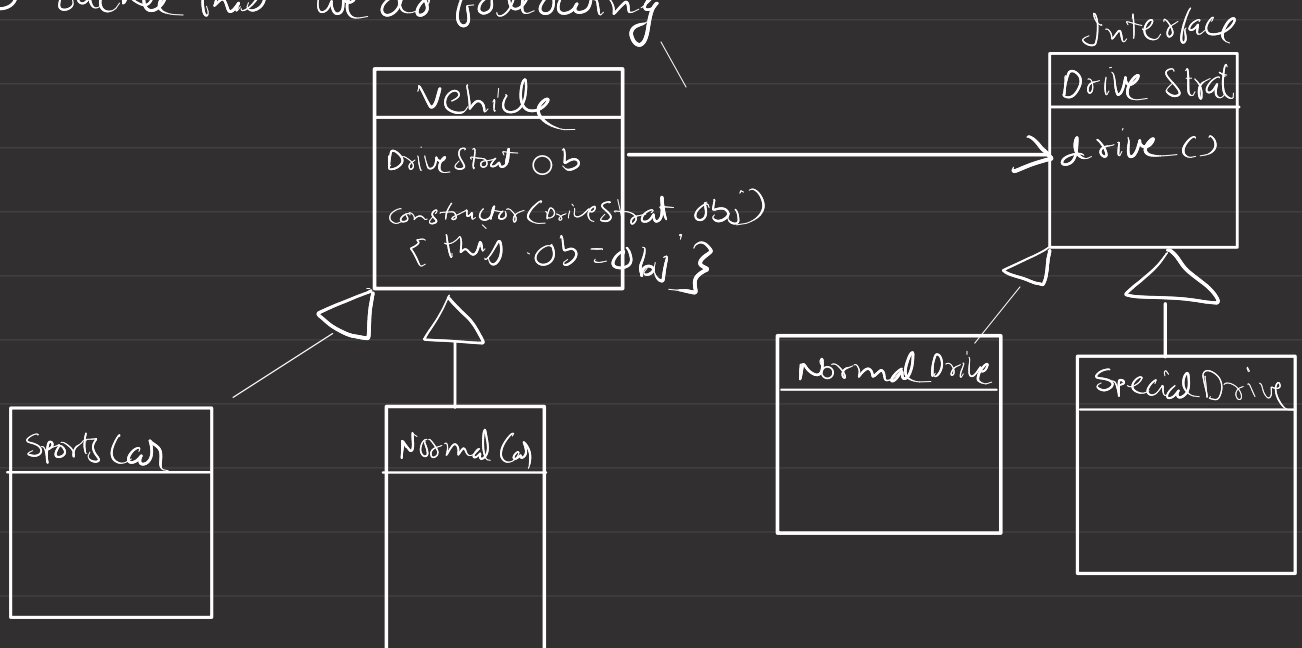
duplicate code would also increase.

It is something like below

→ Same
Implementation

→ This code duplication
is problematic

To tackle this we do following

Vehicle

DriveStrat ob
constructor(DriveStrat obj)
{ this ·ob = obj }

Interface
Drive Strat
drive ()

Sports Car

Normal Car

Normal Drive

Special Drive

So in this scenario, we will be having an interface for our
functionalities and we willuse this interface in
our main class to add the functionality.

This way we do not duplicate code for some funch
onality.