~~Task(15.~~1)

Strategy Pattern

Requirements:

Implement a simple Strategy Design Pattern for calculating vehicle speed based on different car brands.

1.Abstract Interface:

- Define an interface IStrategy with a pure virtual method calculateSpeed(). This will act as a base for different strategies representing how vehicle speed is determined.

2.Concrete Strategy Classes:

- Create two classes StrategyBMW and StrategyMini that inherit from IStrategy. Implement the calculateSpeed() method in each class, where:
  - StrategyBMW should print "BMW Strategy".
  - StrategyMini should print "Mini Cooper Strategy".

3.Vehicle Class:

- Create a class Vehicle that accepts an IStrategy* in its constructor. This strategy will define how the vehicle calculates its speed.
- The Vehicle class will have a VehicleSpeed() method that calls the strategy's calculateSpeed() method.

4.Main Function:

- Create two instances of Vehicle, one using StrategyBMW and the other using StrategyMini.
- Call the VehicleSpeed() function on each instance to see the result of the strategy in action.

5.Create a class diagram for all the classes using https://app.diagrams.net/

Thank You