



Client: "Build my car!"

«Interface»  
VehicleBuilder

+ buildVehicleColor() : void  
+ buildVehicleMake() : void  
+ buildVehicleType() : void  
+ buildVehicleOwner() : void  
+ buildVehiclePrice() : void  
+ getVehicle() : Vehicle

# CarBuilder:

## Builder Design Pattern Structure

«Implemented Class»  
OldVehicleBuilder

- vehicle : String  
- color : String  
- make : String  
- type : String  
- owner : String  
- price : int

+ oldVehicleBuilder(color : String  
make : String  
type : String  
owner : String)  
+ buildVehicleColor() : void  
+ buildVehicleMake() : void  
+ buildVehicleType() : void  
+ buildVehicleOwner() : void  
+ buildVehiclePrice() : void  
+ getVehicle() : Vehicle

«Implemented Class»  
VehicleEngineer

- vehicleBuilder : vehicleBuilder

+ vehicleEngineer(vehicleBuilder :  
vehicleBuilder) : void  
+ getVehicle() : Vehicle  
+ makeVehicle() : void

«Interface»  
VehiclePlan

+ setColor(color : String) : void  
+ setMake(make : String) : void  
+ setType(type : String) : void  
+ setOwner(owner : String) : void  
+ setPrice(price : int) : void

«Implemented Class»  
Vehicle

- vehicleColor : String  
- vehicleMake : String  
- vehicleType : String  
- vehicleOwner : String  
- vehiclePrice : int

+ getVehicleColor() : String  
+ setColor(color : String) : void  
+ getVehicleMake() : String  
+ setMake(make : String) : void  
+ getVehicleType() : String  
+ setType(type : String) : void  
+ getVehicleOwner() : String  
+ setOwner(owner : String) : void  
+ getVehiclePrice() : int  
+ setPrice(price : int) : void

