Laboratory 3

Behavioral Design Pattern

Domain: Car Factory Shop

There was implemented 1 behavioral DPs:

- State- Alter an object's behavior when its state changes

1.State- in the BuildEngine method at DecoratorEngine class we send a class of type EngineType, that stores in the EngineType property.

In the Clone method in dependency of the property state engineType, the value of `vehicleBuilder.vehicle["engine"]` changes by adding (“forced”/”echo”)

```
class DecoratorEngine
    {
        protected VehicleBuilder vehicleBuilder;

        public EngineType engineType;

        public DecoratorEngine(VehicleBuilder vehicleBuilder)
        {
            this.vehicleBuilder = vehicleBuilder;
        }

        public void BuildEngine(EngineType type)
        {
            engineType = type;

        }

        public VehicleBuilder Clone()
        {
            if (vehicleBuilder != null)
            {
                if (engineType.GetType().Name == "ForcedEngine")
                {
                    vehicleBuilder.vehicle["engine"] = vehicleBuilder.vehicle["engine"] +
" forced";
                }
                else
                    if (engineType.GetType().Name == "EchoEngine")
                        vehicleBuilder.vehicle["engine"] =
vehicleBuilder.vehicle["engine"] + " echo";

                return vehicleBuilder;
            }
            else return null;
        }

    }
```