**The Adapter Pattern**

**Object Oriented Goal**: Favoring object composition over inheritance. Bind the client to an interface and to implementation.

**Definition**: The Adapter Pattern converts the interface of a class into another interface the clients expect. Adapter lest classes work together that couldn’t otherwise because of incompatible interfaces.

**Implemented on classes**:

1. Payment
2. CardPaymentTarget (Target Interface)
3. PayPalAdapter
   1. PayPal
4. VisaCardAdapter
   1. VisaCard
5. MasterCardAdapter
   1. MasterCard

**Design Snapshot**:



**Design Explanation:**

In our application, there are multiple ways for the user to pay for the order like Credit Card, PayPal and Cash on Delivery and we also accept Master Card and Visa Card both. Thus the classes for Master Card, Visa Card and PayPal differ in method implementations and also have different interfaces. We need a way for our card payment option to adapt with different types like Master Card, Visa and PayPal.

In this situation, we can add the adapter pattern for the clients to easily use the adapters for payment rather than going for the implemented classes to perform functions. The Adapter Pattern implemented in our design is explained as follows:

* The Payment class is our client class which will only know about the CardPaymentTarget interface and will only implement the pay and authorize methods using the Target interface as the type of different objects like MasterCard, Visa and PayPal.
* The CardPaymentTarget is the target interface which has methods like pay and authorize which needs to be implemented by the respective adapters.
* PayPal, MasterCard and VisaCard are the Adaptee classes which will get the calls that Payment will make on Target Interface.
* PayPalAdapter, VisaCardAdapter and MasterCardAdapter are the class which compose the PayPal, VisaCard and MasterCard classes respectively. The adapter classes implements all the methods from the target interface and in those methods are the method calls to the particular classes like PayPal, VisaCard and MasterCard.
* The client Payment invokes method on the target (CardPaymentTarget) interface. The PayPalAdapter, VisaCardAdapter and MasterCardAdapter gets the method call and it delegates the calls to their respective classes.