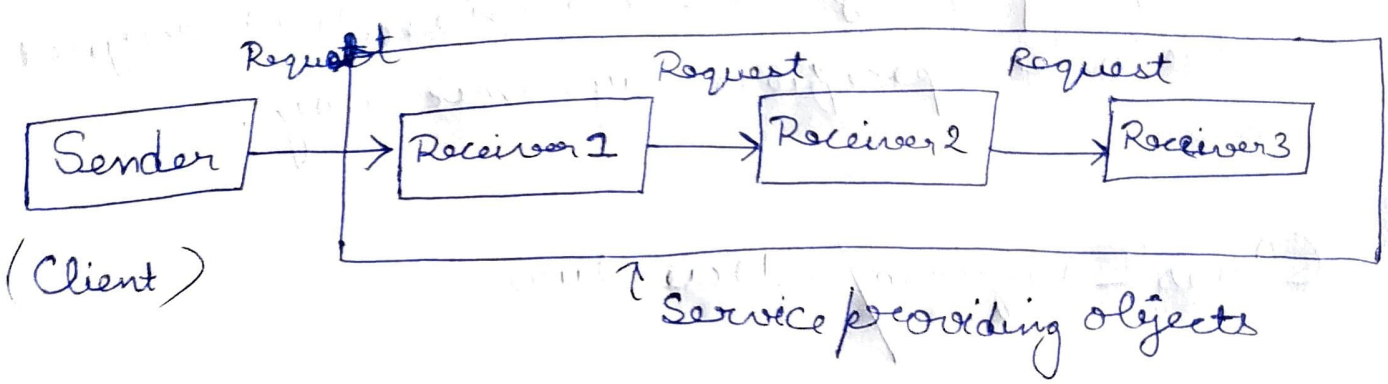
Chain of responsibility design pattern



first of all sender will send a request. It will be received by receiver 1. Now receiver 1 will check if it can process the request. If it cannot process the request it will forward it to receiver 2 and so on similar procedure will be followed by it. If a receiver can process a request it will process it by itself.

So basically sender sends a request to a chain of objects. The request can be handled by any object in the chain.

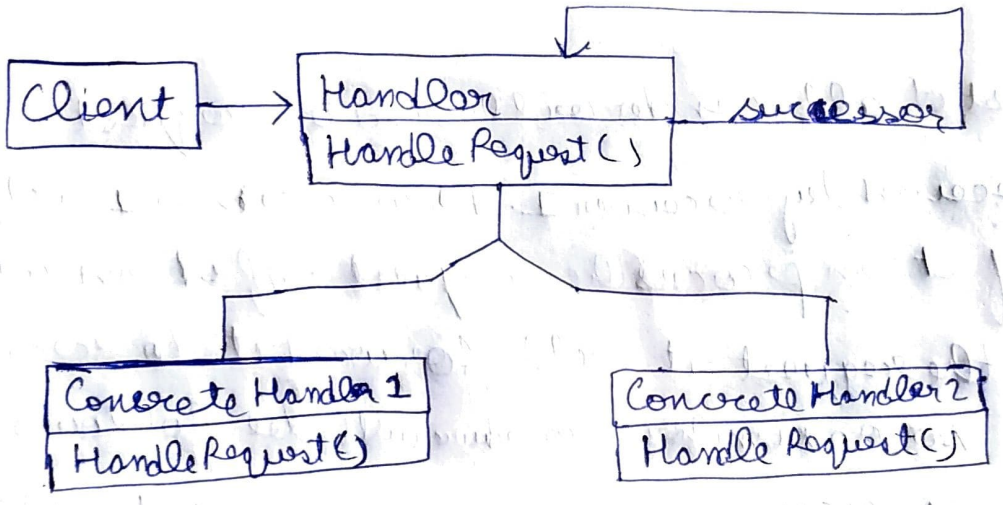
It avoid coupling the sender of a request to its receiver by giving multiple objects a chance to handle the request

Ex → Real life example of this is in ATM machine Suppose you want to get 500Rs from ATM. Now suppose Firstly your request is sent to 2000Rs object so it will not process it & send it to object of 500Rs now here this request will be processed & it will get executed

So It efficiently processes the request without hard-wiring handler relationships.

Chain of responsibility design pattern is used when the group of objects that can handle the request are to be specified in dynamic way.

Class Diagram / Structure



Handler → Object which processes the request. It defines interface for handling the request

Concrete Handler → Handle a particular request