

first of all senderwill send a request. It will be received by receiver I. Now receiver I will check if it can process the request. If it connot process the request it will forward it to receiver 2 and so on similar processes will see followed by it. If a receiver can process a request it will process it by itself.

So lousically sender sends a request to a Chain of objects The request can be Ronaled by only 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

Suppose you want to transact 500Rs from ATM Now suppose firstly your request is sent to 2000Rs object 20 get will not process it to sent to object 9,500Rs mow her this request will be processed to it will get exocula 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 les specified in dynamic way. Class Diagram/Structure Missil. Client Handlor successor Concecte Handler 2 Conecete Handla 1
Handle Request () Hardle Roquest() Handler > Olegect which processes the request. It defines interface for Randling the request Concrete Handler -> Handle à particular request of as produced to de more, by any and the product a man to be a fill the season of lay to the state of ing in the second of the second of the second of the form of the state of the state of the state of the training of the second winds ing in the form of the standard and the trade in the

properties freedom of consisted in the offer of