CHAIN OF RESPONSIBILITY

The chain of responsibility is a behavioral design pattern. Here the request from the client is passed to the chain of objects to process them. The object in the chain will decide if the request is to be processed by them or to handover it to the next object.

The main advantage of this design pattern is we can achieve the Single responsibility principle. Also open/closed principle is achieved because we can add additional classes if required without breaking the existing code. The order of chaining can be decided by us.

Pattern :

Handler:

This is the main interface. Every concrete handlers should implement this. Typically this interface consists of two methods, one is to handle request and the other is to chain the request to another object.

Concrete Handlers:

These classes are the actual implementations of hander interface. It handles the request or it chain the request to the next object.

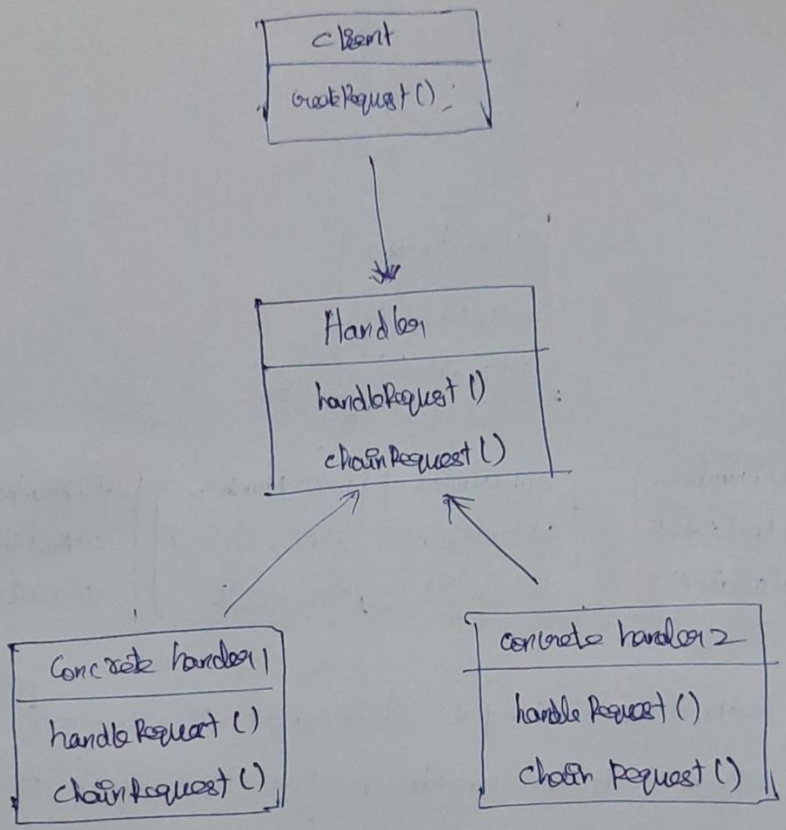
Client :

This the place from where the request arises.The request from client is handled by handlers.

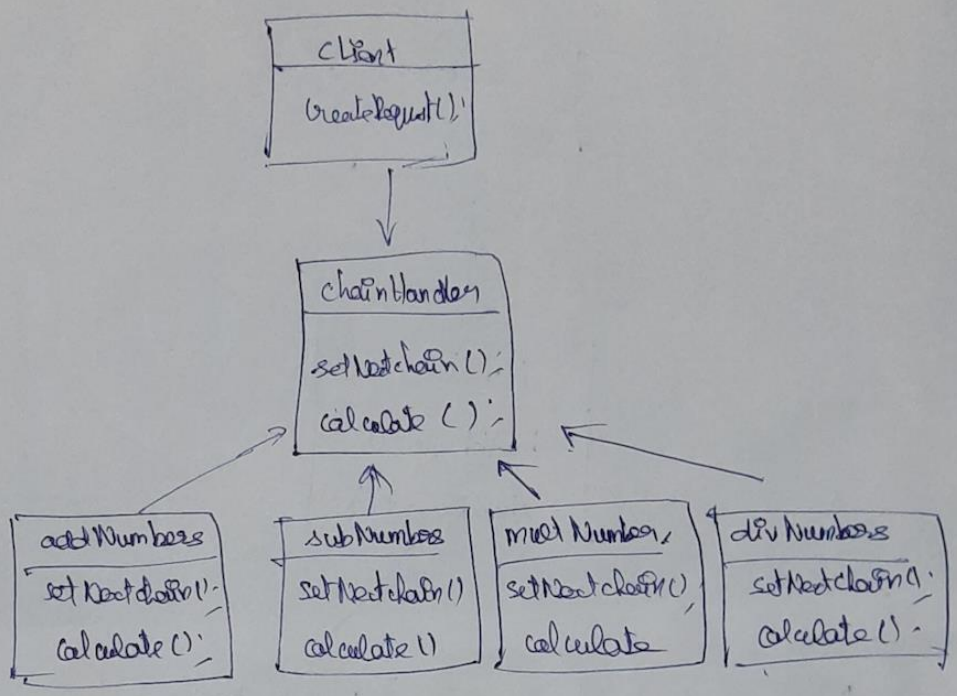
Once the request is received , the request is sent to the chain and that request is processed based on certain conditions to find wheather the request belongs to that particular chain or not.

Chain Of Responsibility design pattern and example which used the Chain Of responsibility design pattern is attached below.

CHAIN OF RESPONSIBILITY



Lets see Arithmatic Operation using Chain of Responsibility.



where setNextChain(); accepts Chain object as argument;

calculate() decides whether to handle the request or to execute the calculate() method of next chained object.

Usually there will be a Main class which will be used for to set the chain and start the chain process.