> Chain of Responsibility Design Pattern:-

→ chain of Responsibility pattern is used to achieve loosinge coupling in software design where a request from the client is passed to a chain of objects to process them.

→ Later, the object in the chain will decide themselves who will be processing the request and whether the request to be sent to the next object in the chain or not.

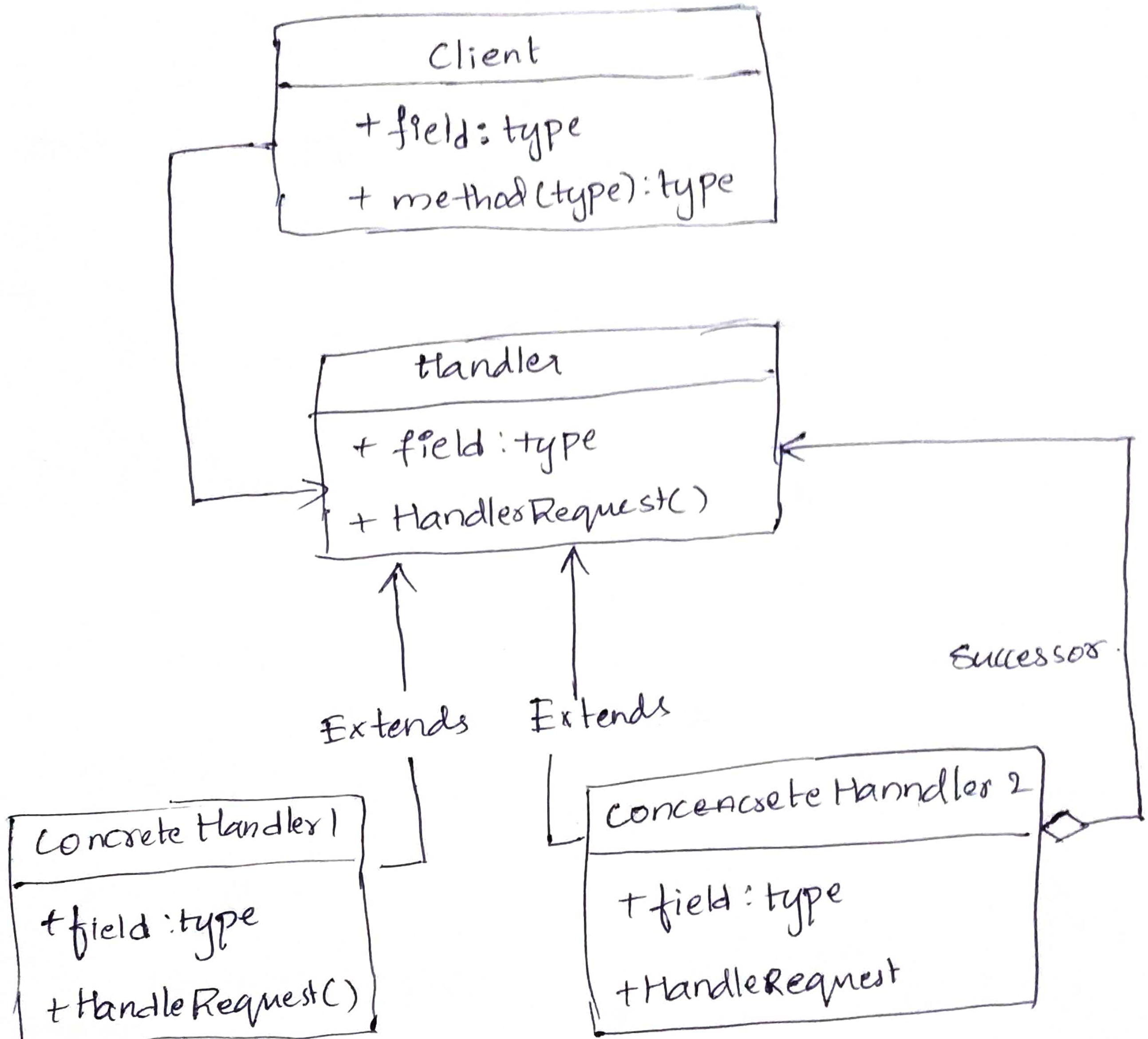
> Chain of Responsibility pattern is applicable:-

1. When you want to decouple a request's sender and receiver
2. Multiple objects, determined at runtime, are candidates to handle a request.
3. When you don't want to specify handlers explicity in your code.
4. When you want to issue a request to one of several objects without specifying the receiver explicitly.

> Example:-

> This pattern is recommed when multiple objects can handle a request and the handler doesn't

> have to be a specific object. Also, the handler is determined at runtime. Please note that a request not handled at all by any handler is a valid use case.



> Handler:- This can be an interface which will primarily receive the request and dispatches the request to chain of handlers. It has reference of only first handler in chain and doesn't know anything about rest of the handlers.

> Concrete handlers:- This are actual handlers of the request chained in some sequential order.

> Client:- Originator of request and this will access the handler to handle it.