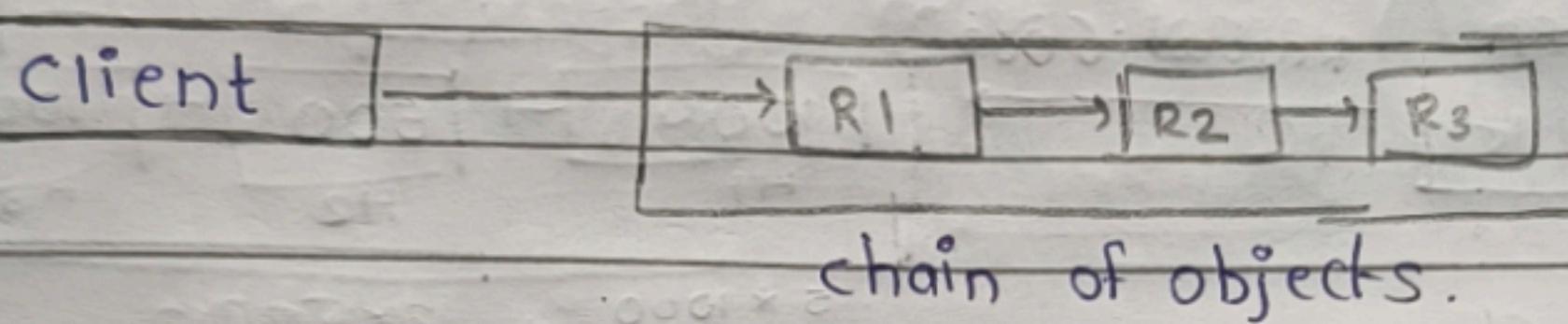


Lecture 22: Chain of Responsibility Pattern

Page No.	
Date	

Introduction

- Use case specific pattern
- Assume karo client job request send kar raha chain of objects ko
- Toh ismein kya hota hai client request kr raha R1 ko, R1 check krega kya woh client ki request fulfill kar skta hais. If yes then req. complete hogi, if not, it sends R2 and go on. In same way request comes back with response.

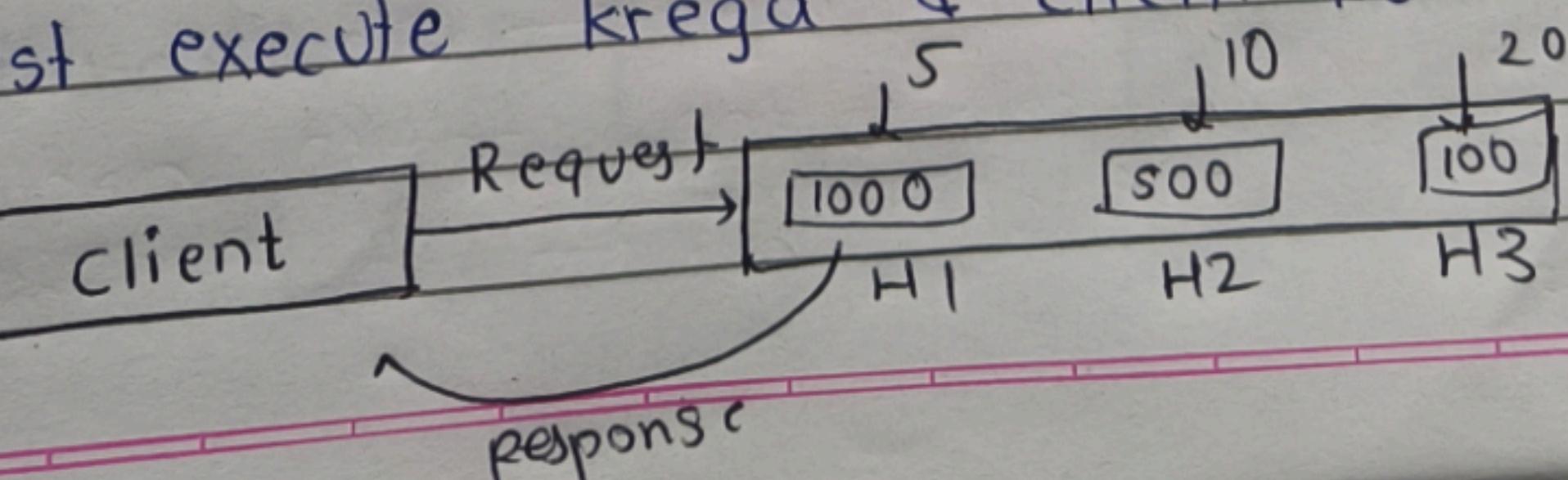


- Yeh linkedlist DS jaisa hai but kuch differences hai

Flow for Cash Dispensing

- ATM machine ke pass multiple handles hote hai jinke pass cash store hota hai
- Example: Client ko 5000 rbona uski request woh H1 ko bhejta hai then H1 check krta hai kya woh req. fulfill kar skta hai.

- H1 ke pass 5 notes of 1000 \Rightarrow 5000 hai so woh request execute krega & client ko de dega.

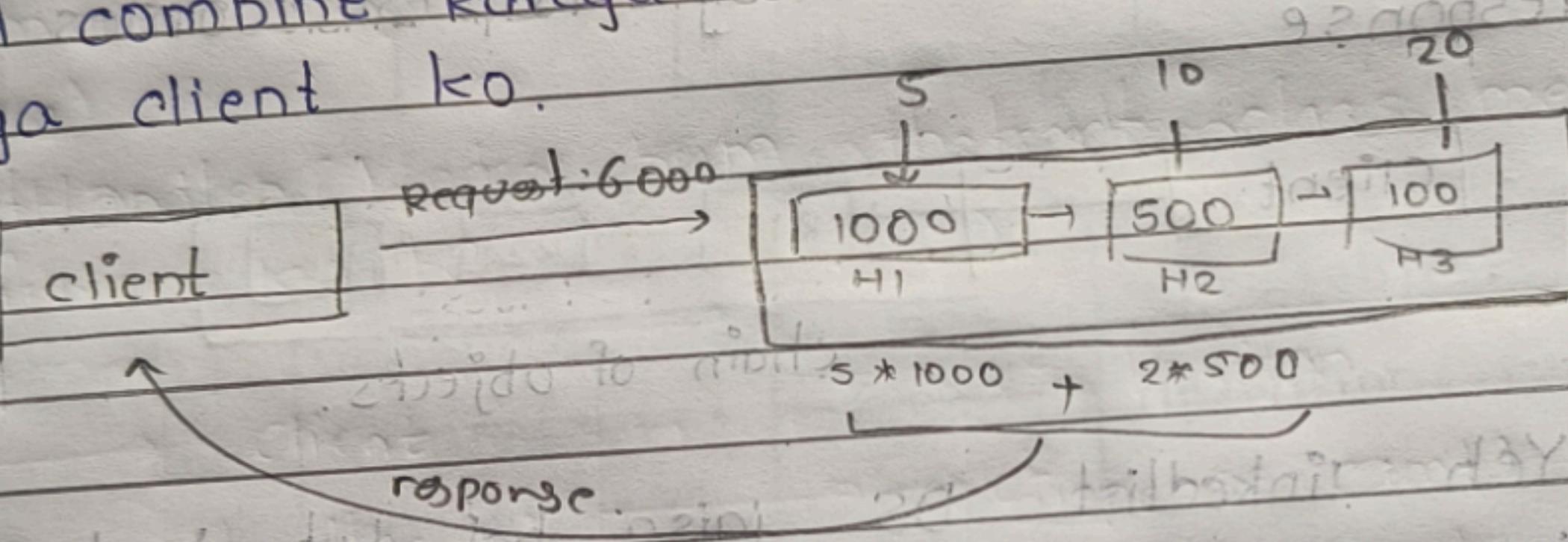


UML Diagram

- Ab client ko 6000 Rs hona usne request ki, go H1 check krega can be handle the request joh uske pass 1000 ke 5 notes \Rightarrow 5000 hai so he can't to ab woh H2 ko bolega ki next part of request i.e. 1000 rs tu handle kar.

- H2 check krega can be handle that request. \rightarrow haa \rightarrow so woh 500 ke 2 notes return krega -

- H1 combine krega dono response ko & bhej dega client ko.



Now client asks for Rs 150.

- Ab humare pass aisa koi handler hi nahi hai joh 250 ko handle kare

- So, we have two options here

- We can return only Rs 100 note from H3

- We can cancel the whole transaction

\rightarrow This depends on use case

- Suppose pass diff ko handle

- Tb client hum pehl fulfill k

- Chain reference

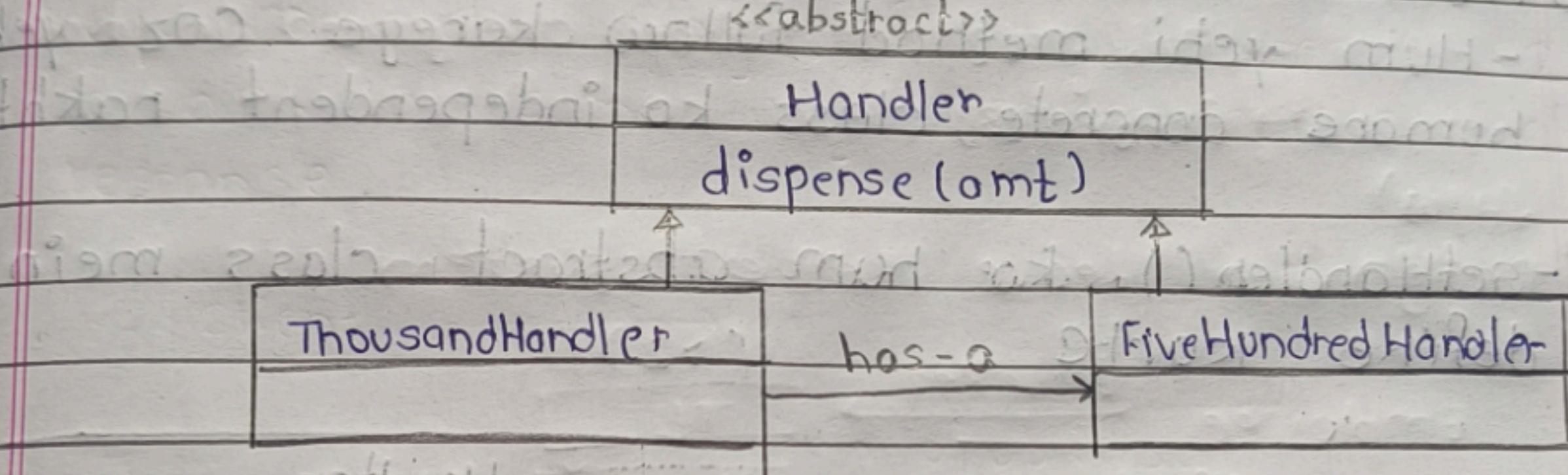
We ha relation

① From

UML Diagram and its Description

- suppose humare pass handler hai and uske pass different sections hai jo different amount ko handle kartad hai

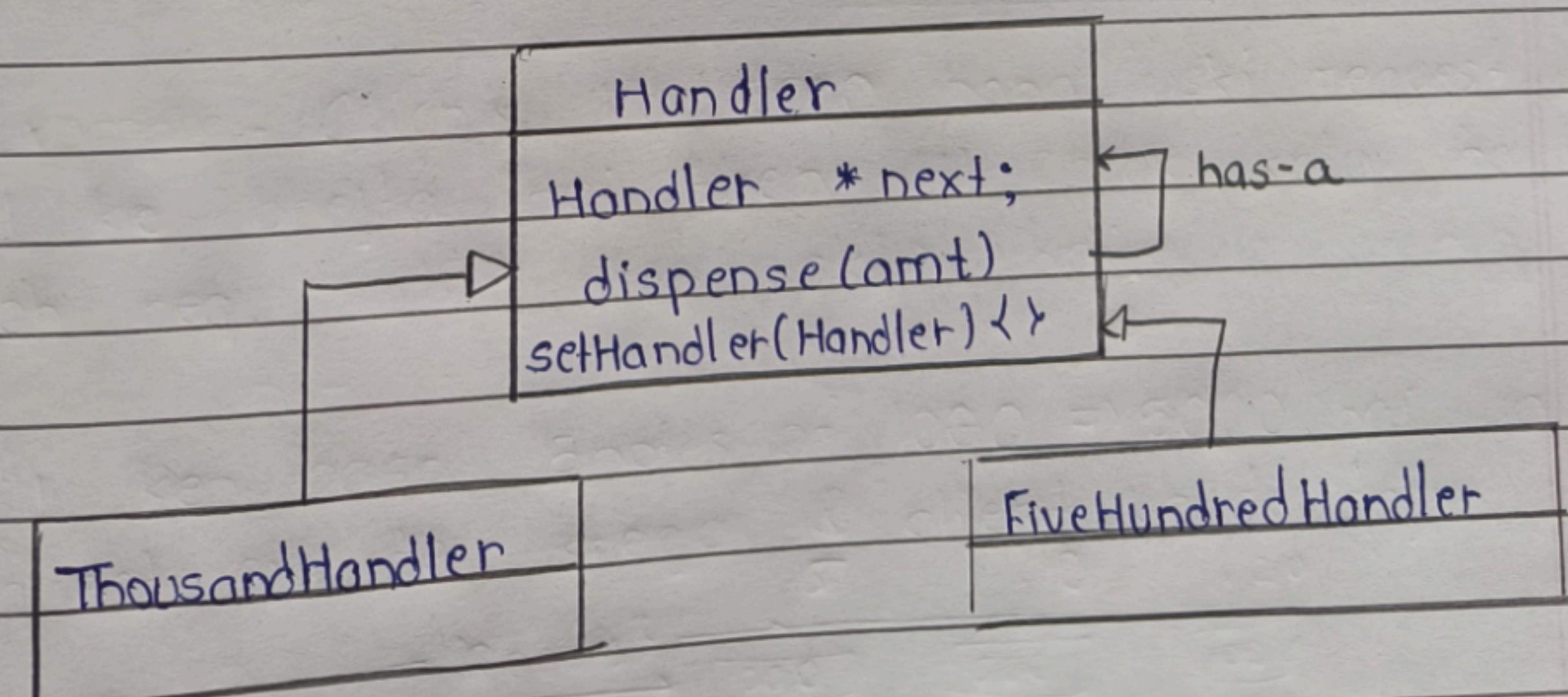
- If client / user request karega money tab hum pehle higher amount then lower amount fulfill karega.



- Chain of responsibility means every handler has reference of next handler.

We have two ways for showing reference relation -

① From concrete object to concrete object



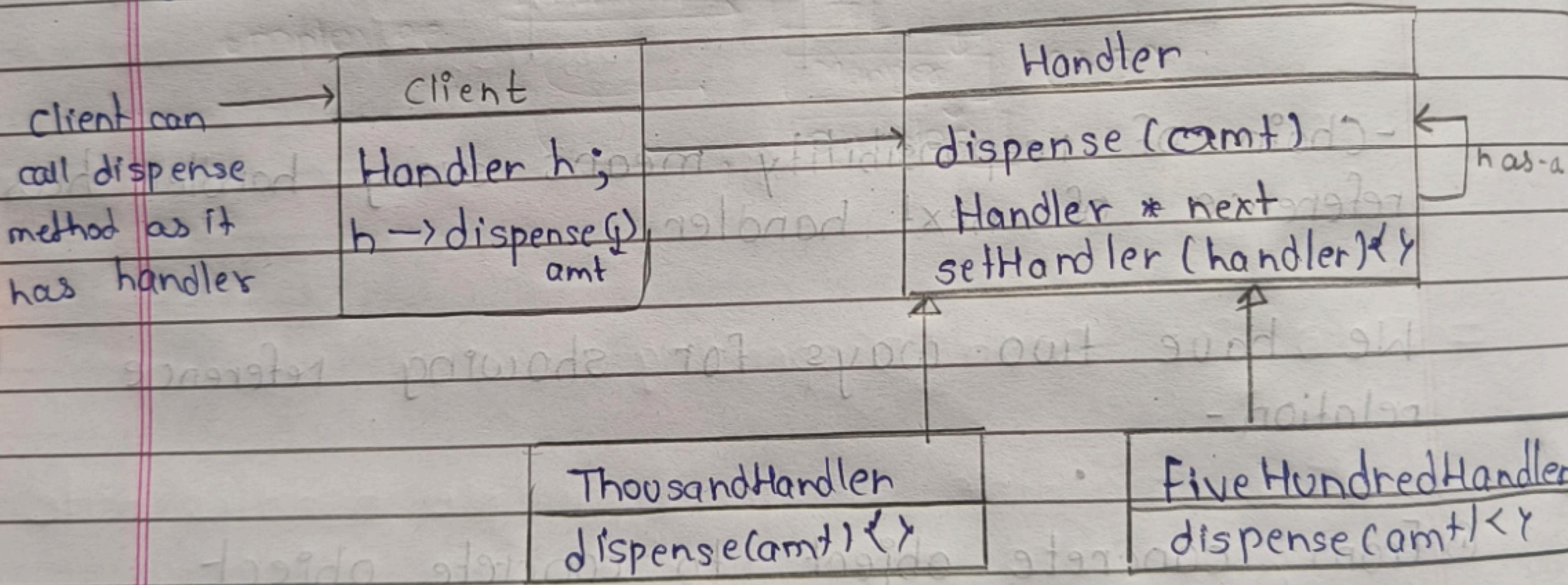
- But yaha humne barcode kar diya ki 1000 ke baad 500 and then 100 & so on hi ayeega.
 What if in future we don't have 500 notes.
 So tab 1000 ke baad 100 aaana chahiye but to make these humne humare main code mein changes krne hoga.

→ This makes it tightly coupled.

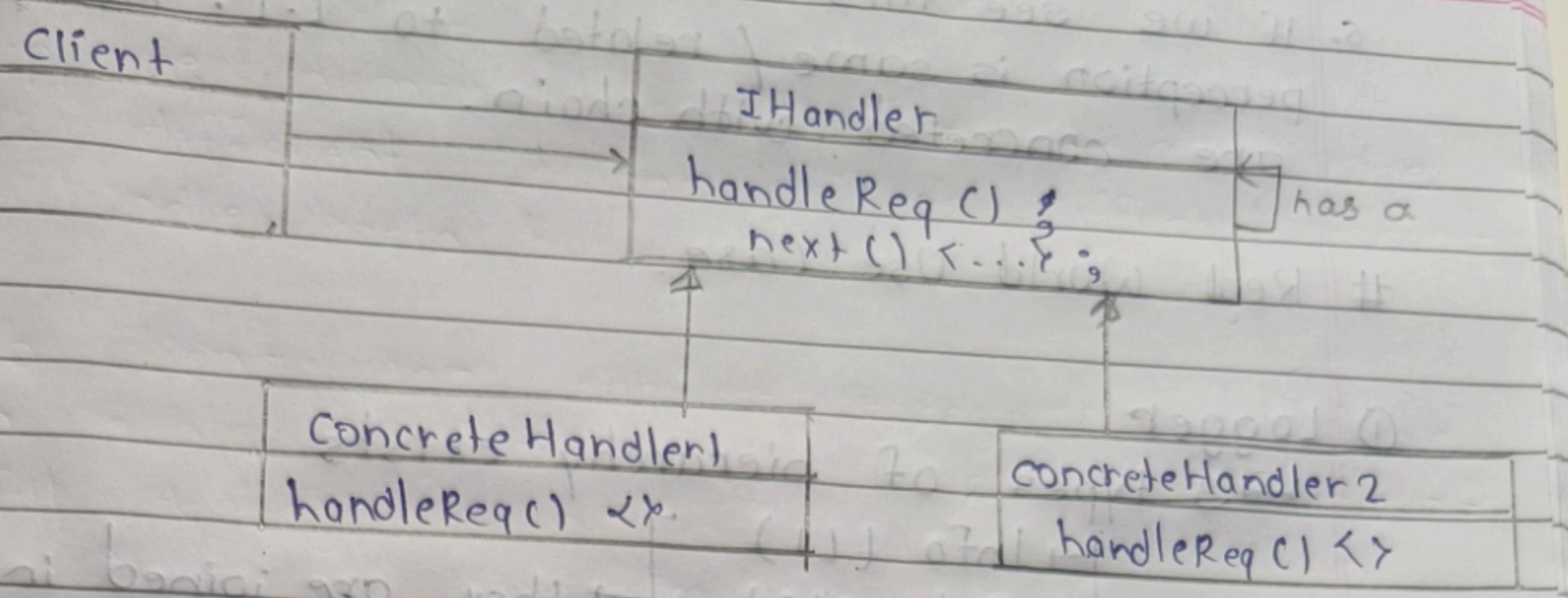
② From Interface to Interface

- Hum yehi method follow karege coz yeh humare concrete class ko independent rakhta hai.

- setHandler() ko hum abstract class mein define karlete



Standard UML



Standard Definition

It allows an object to pass a request along a chain of potential Handler. Each Handler in the chain decides either to process the request or pass it to next Handler.

Linked List VS Chain of Responsibility (COR)

1. At the end of every comparison its all about intent.
2. Linked list ek DS jiska intent hai to store data by making multiple nodes which are linked to each other.
3. COR also links one object to other but intent is to handle request not to store data.
4. LL refers to its own type means point to same object.
5. But COR, we have one interface which has multiple concrete class & one class refers to other

6. If we see this in simplified way, then its ideal perception is same / related to L.L. Multiple objects are concreted with chain.

Real world Use Case

① Logger

→ 3 types of hierarchy exists

- Info (L1)
- Debug (L2)
- Error (L3)

→ they are joined in chain

② Design a Leave Request system

