

Instructions



How is it???

0 responses



**Tell me if you have problems with
practical tasks**

Agenda

- Recall what was learned
- Look at current topic practice
- Check current topic knowledge

Total Recall



What is the primary function of the Proxy Pattern?

- To store data for quick access later ✖
- To simplify the interface of an existing object ✖
- To add an intermediary layer between a client and an actual object ✓
- To replace the original object with a different one ✖

What is the primary function of the Adapter design pattern in software development?

- A) It allows classes with incompatible interfaces to work together by wrapping its own interface around that of an already existing class.
- B) It ensures that a class only has one instance and provides a global point of access to it.
- C) It provides an interface for creating families of related or dependent objects without specifying their concrete classes.
- D) It allows for the dynamic addition of new operations to objects without modifying the classes.



What is the main purpose of using the Decorator?

- a) To add new methods to an object dynamically ✗
- b) To modify existing behavior of an object dynamically ✓
- c) To enhance the performance of an application ✗
- d) None of the above ✗

Which of the following scenarios best suits the use of the Bridge design pattern?

- a) When you need to generate multiple versions of a component without affecting its clients. ✗
- b) When the implementation of a component needs to be chosen or switched at runtime. ✓
- c) When you have several classes that differ only in their implementation details. ✗
- d) All of the above. ✗



Difference

- Adapter converts an interface into one a client expects.
- Decorator dynamically adds behavior to objects.
- Proxy controls access to an object.
- Bridge decouples an abstraction from its implementation, allowing them to vary independently.
- All are structural design patterns in object-oriented programming, aimed at improving code scalability, flexibility, and



Discuss current topic

Coding Is A Left-brain Activity

thanks, brain

CTV

Have you looked at todays' materials?

0



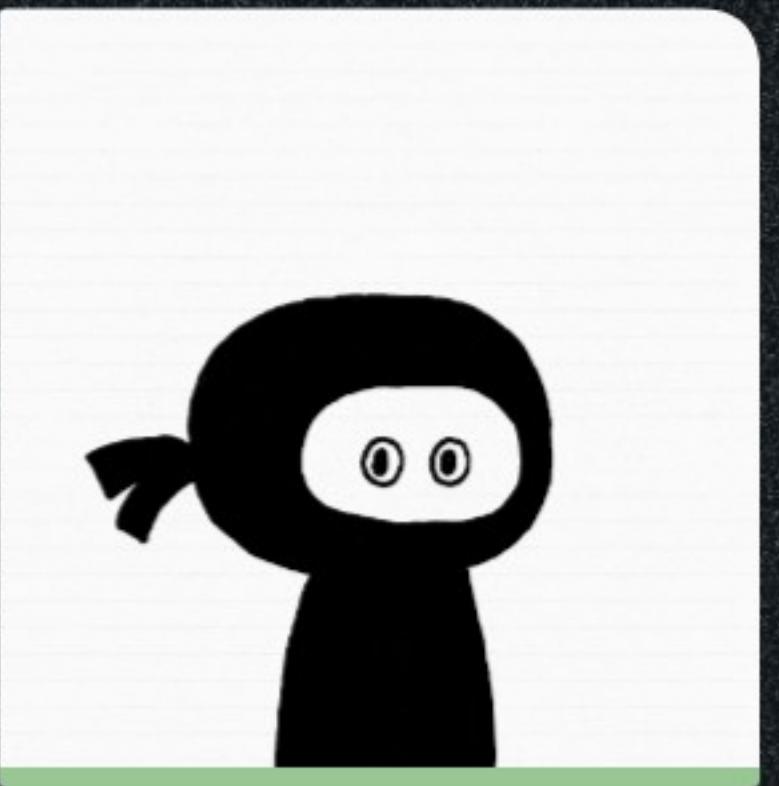
yes

0



no

0



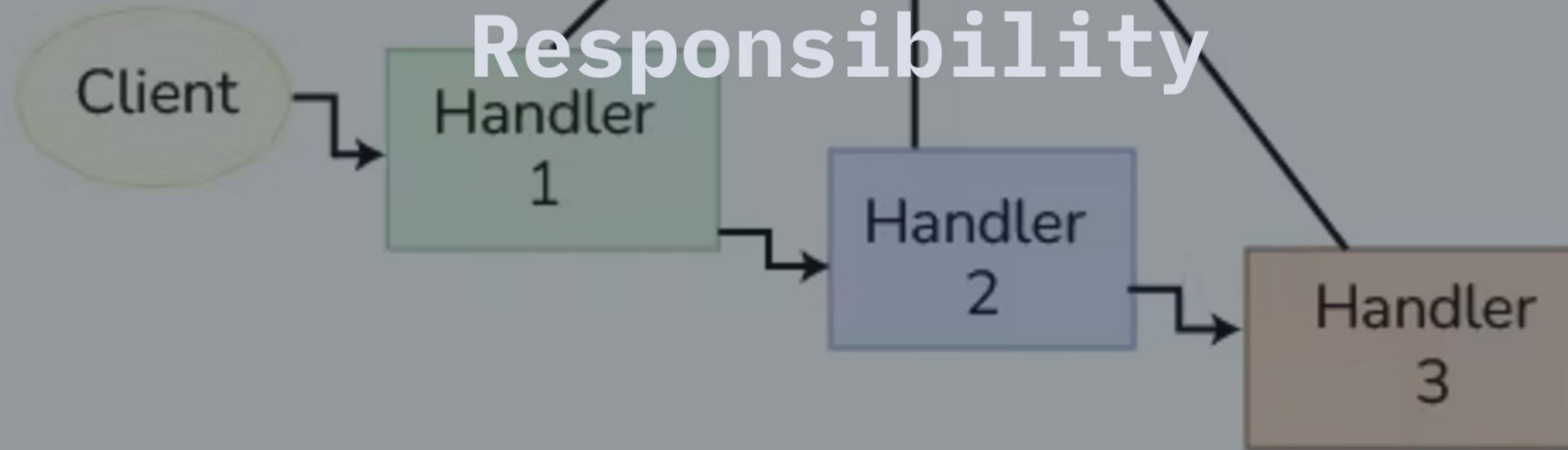
partially

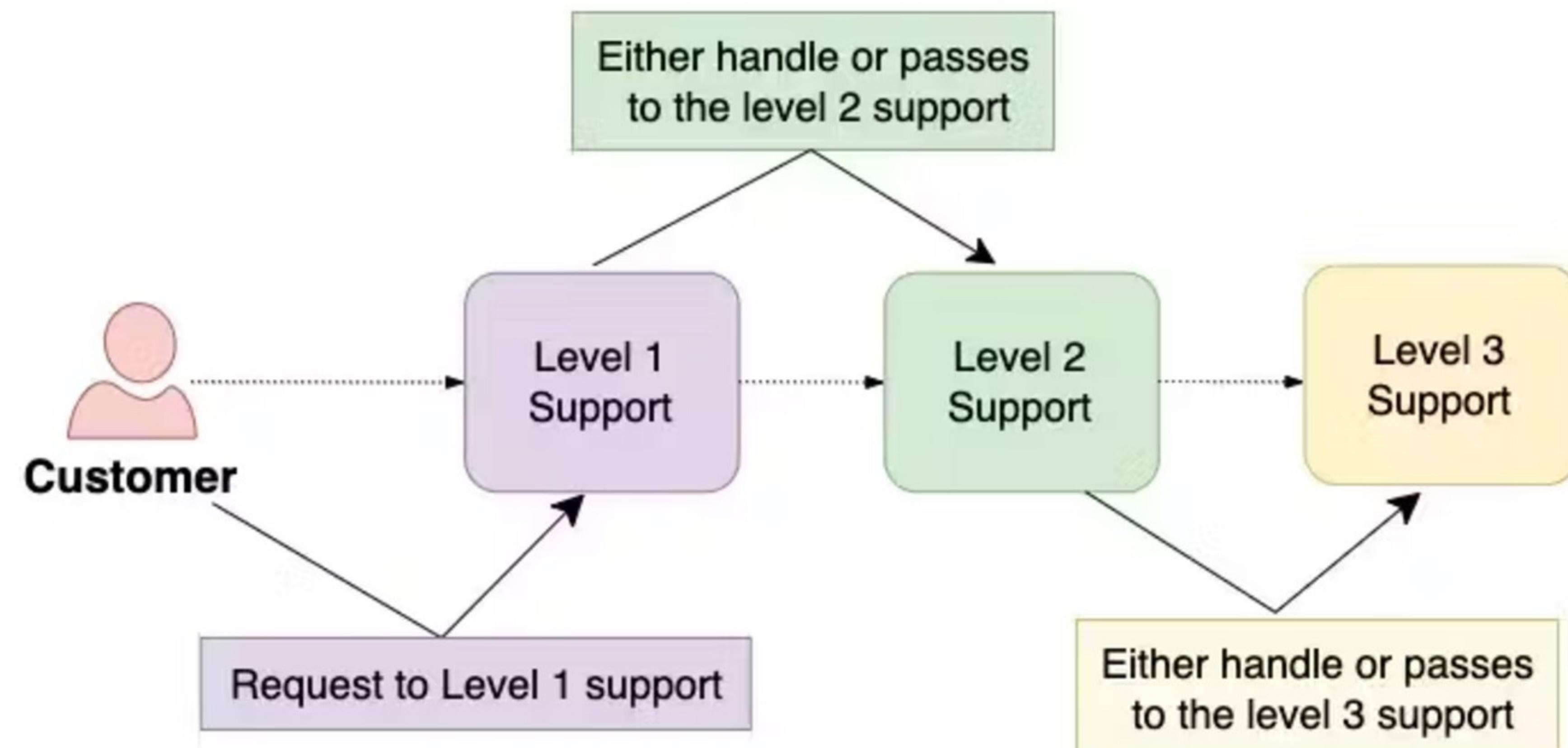
0



i'm the author

Advanced Patterns: Chain of Responsibility





Chain Of Responsibility Design Pattern

Descriptive real-world example

Rules \ Principles

- Each processor in the chain will have its implementation for processing a command
- Every processor in the chain should have reference to the next processor
- Each processor is responsible for delegating to the next processor so beware of dropped commands
- Processors should not form a recursive cycle
- Only one processor in the chain handles a given command

Benefits

- Decoupling of Sender and Receiver
- Flexibility in Assigning Responsibilities
- Enhanced Scalability
- Promotion of the Single Responsibility Principle
- Dynamic Reconfiguration
- Improved Maintainability

Disadvantages

1. Mostly, it can get broken easily:
 - a. if a processor fails to call the next processor, the command gets dropped
 - b. if a processor calls the wrong processor, it can lead to a cycle
2. It can create deep stack traces, which can affect performance
3. It can lead to duplicate code across processors, increasing maintenance

What are the key components of the chain of responsibility pattern?

- Client
- Handler Interface
- Server
- Concrete Handlers



Select all usage scenarios in which Chain of Responsibility can be particularly beneficial:

- Event Handling in UI Frameworks
- Approval Processes
- Command Processing
- Memory Optimization



What does the Chain of Responsibility design pattern primarily help achieve in software design?

- A) Encapsulation of method invocations. ✖
- B) Mutual coupling of classes. ✖
- C) Decoupling of senders and receivers by giving multiple objects an opportunity to handle a request. ✓
- D) Creation of object clones. ✖

In the Chain of Responsibility, if an object can handle a request, what should it do next?

- A) Pass the request to all successor handlers irrespective of its own ability to handle it.
- B) Handle the request and decide whether to pass it to the next handler in the chain.
- C) Modify the request and send it back to the sender.
- D) Terminate the program.

Which of the following scenarios is most suitable for using the Chain of Responsibility pattern?

- A) When exactly one object must handle a specific type of request.
- B) When operations need to be carried out in a strict sequence.
- C) When multiple objects can handle a request, and the handler doesn't have to be a specific object.
- D) When operations are highly dependent on the type of data and specific algorithms.



Q&A part

0 questions
0 upvotes