SDET Course

Design Patterns - Mediator

3 Types of Design Patterns

VERISOFT

- Creational
  - Singleton
  - Builder
  - Prototype
  - Factory Method
  - Abstract Factory
- Structural
  - Adapter
  - Composite
  - Proxy
  - Flyweight
  - Bridge
  - Facade
  - Decorator
- Behavioral
  - Strategy
  - Observer
  - Command
  - Memento
  - State
  - Template Method
  - **Mediator**
  - Chain of Responsibility
  - Interpreter
  - Visitor
  - Iterator

Agenda

- Description
- Diagram
- Code sample (Java)
- Use cases

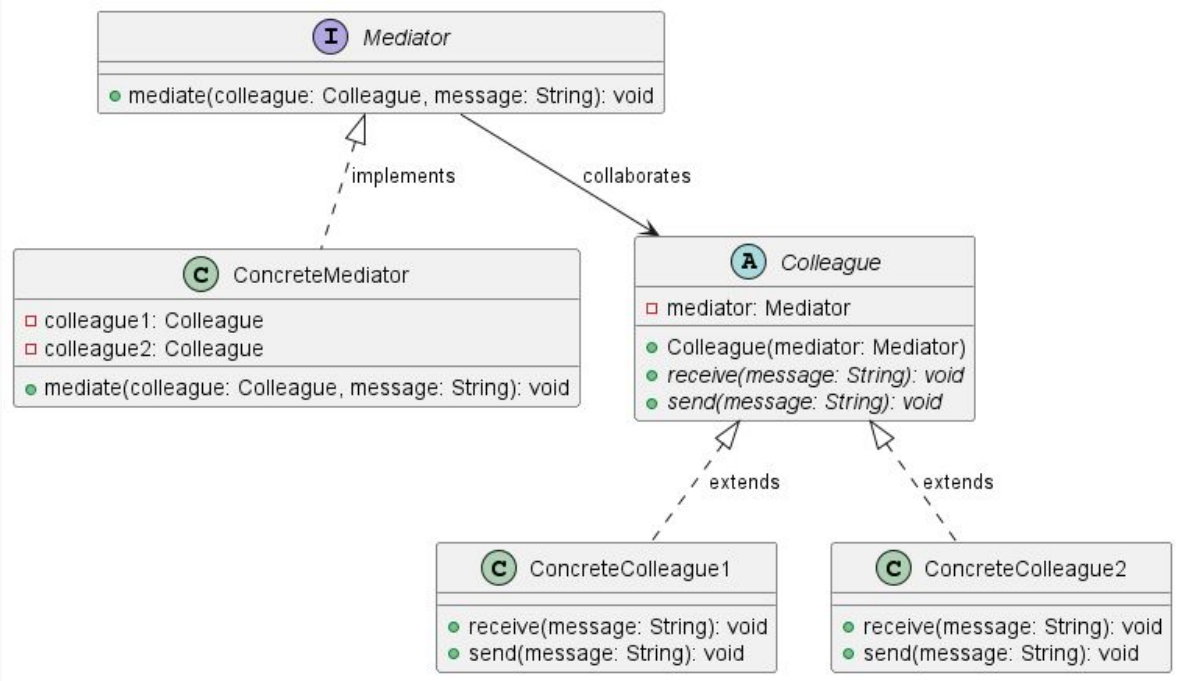
VERISOFT

Descripti
on

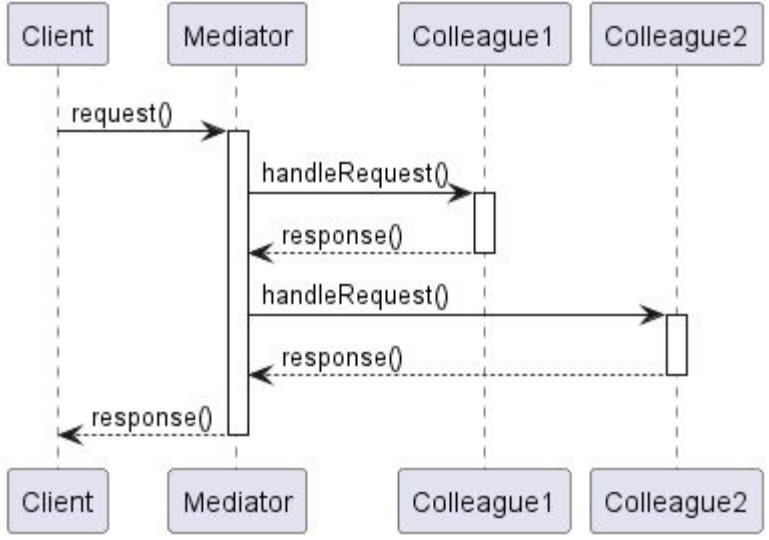
Description

The Mediator Design Pattern is a behavioral design pattern that promotes the reduction of direct communication between objects to decrease their interdependencies. Instead, objects interact with each other through a mediator object. This mediator centralizes complex communications and control logic between objects in a system, making it easier to understand, maintain, and modify. By decoupling objects from each other, the Mediator pattern allows for more reusable and modular components. This pattern is especially useful in scenarios where an application involves a complex set of interactions between different components, such as in user interface design or when coordinating actions among various system components. The goal of the Mediator pattern is to simplify system design by centralizing interaction in a way that reduces the direct linking of objects, thereby lowering the system's overall complexity.

Class Diagram



Sequence Diagram



Code Sample

Use cases

- General
  - Chat Applications
  - Event Management Systems
  - Smart Home Systems
  - Financial Trading Platforms

- In Test Automation
  - Test Suite Management
  - Test Data Preparation
  - Environment Configuration and Cleanup

Happy Coding