

# **Command**

# **Observer**

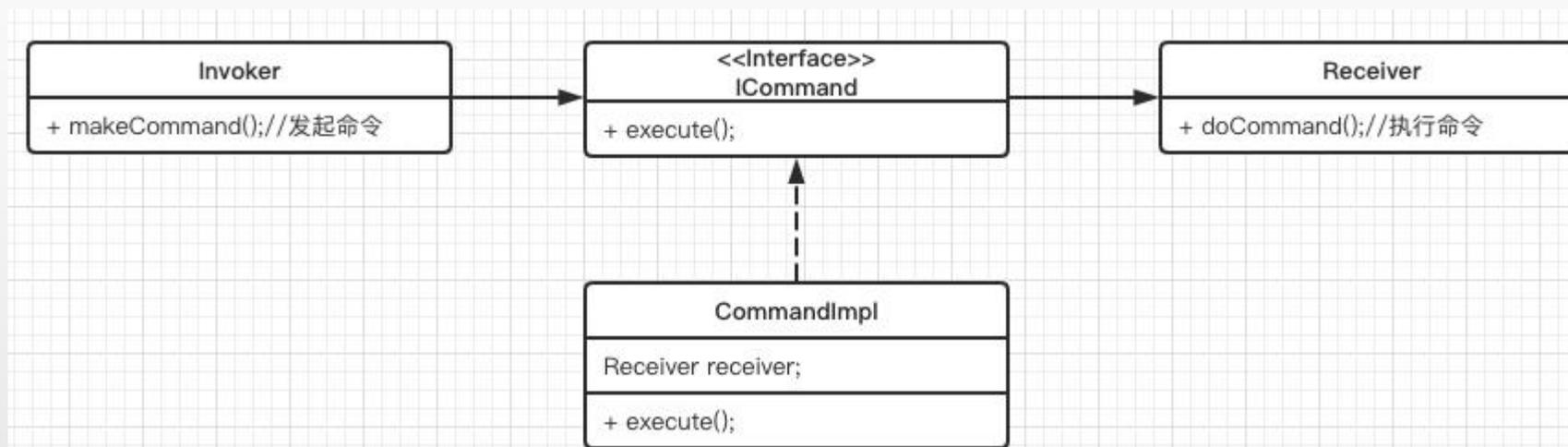
# **Chain Of responsibility**

# **Mediator**

Design Pattern 1

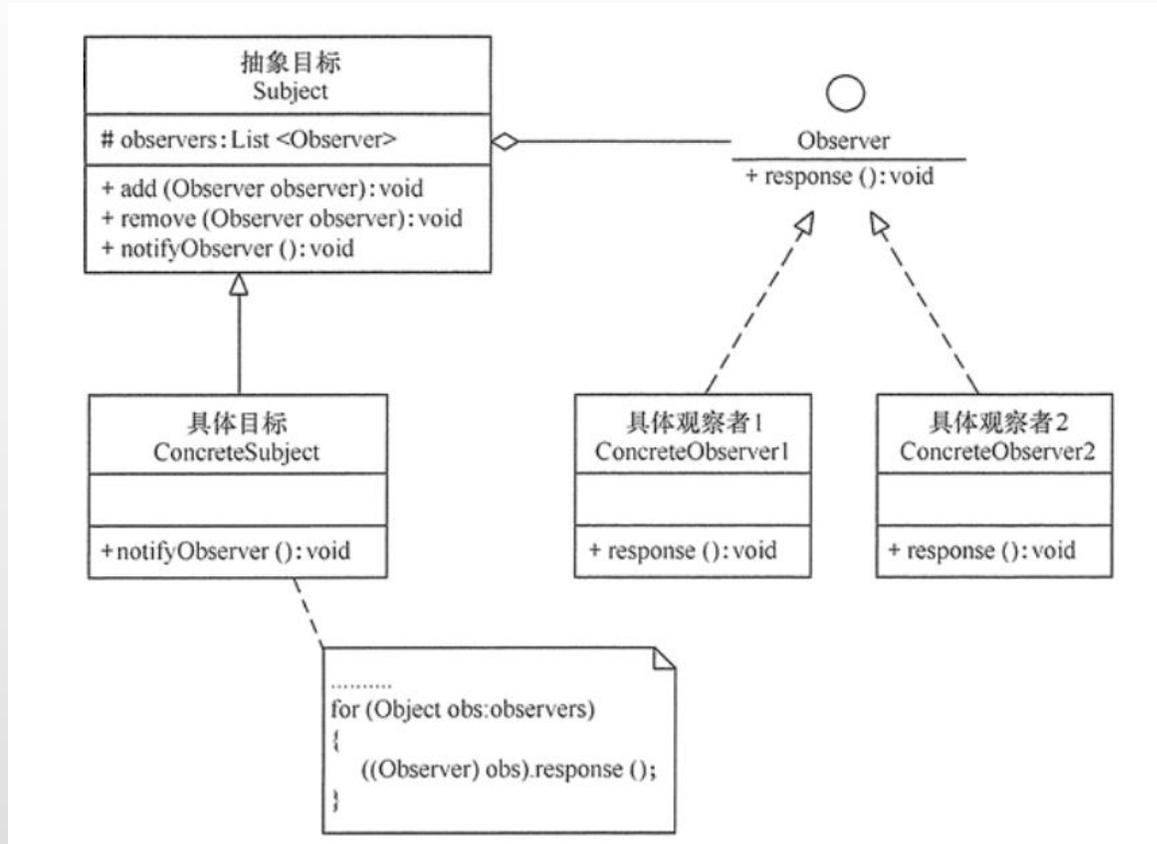
# Command

- Command 将一个请求封装为一个对象,使发出请求的责任和执行请求的责任分割开。这样两者之间通过命令对象进行沟通,这样方便将命令对象进行储存、传递、调用、增加与管理



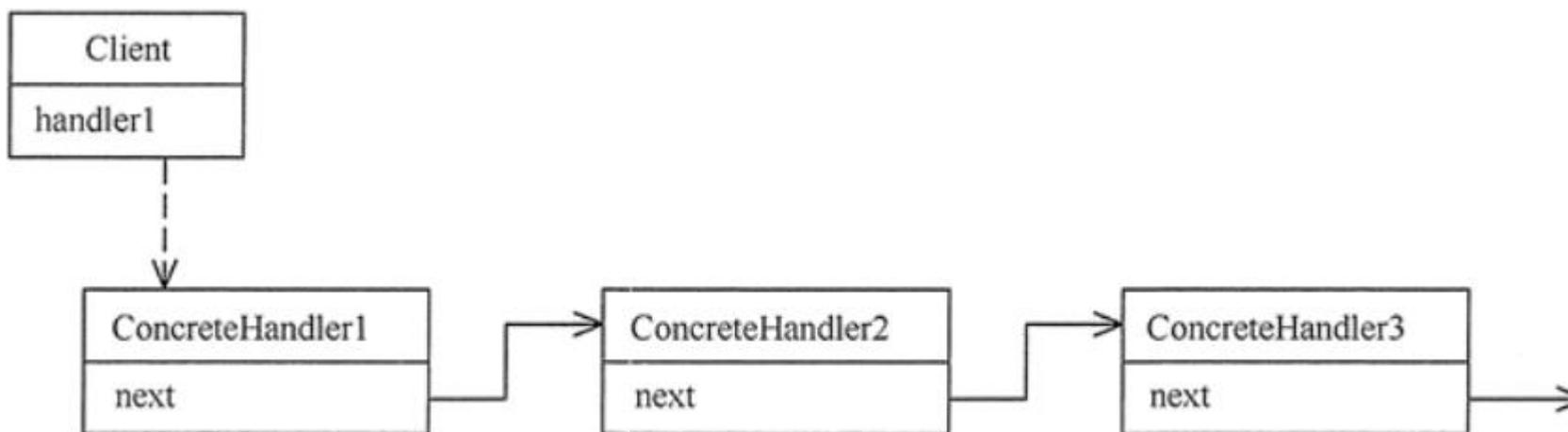
# Observer

- 指多个对象间存在一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并被自动更新。这种模式有时又称作发布-订阅模式、模型-视图模式，它是对象行为型模式



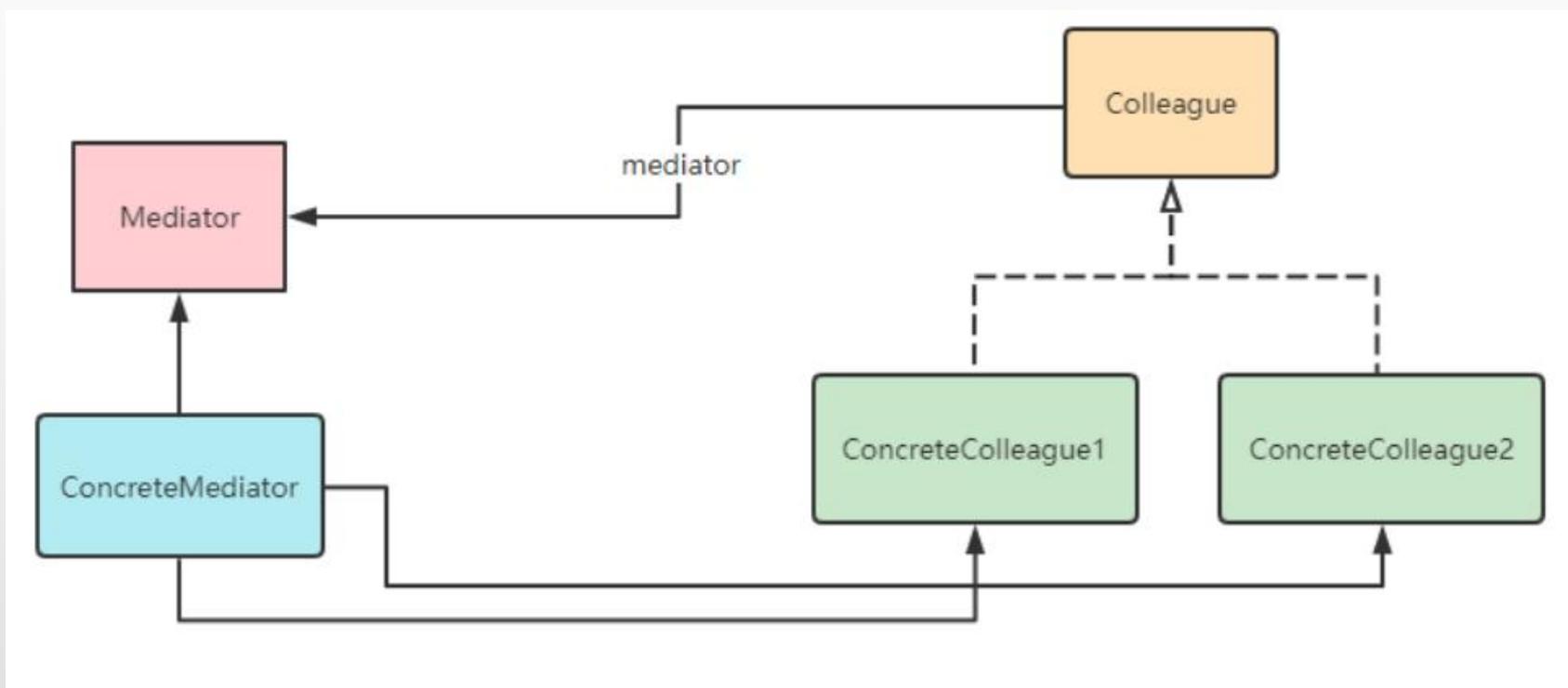
# Chain Of Responsibility

- 为了避免请求发送者与多个请求处理者耦合在一起，于是将所有请求的处理者通过前一对象记住其下一个对象的引用而连成一条链；当有请求发生时，可将请求沿着这条链传递，直到有对象处理它为止



# Mediator

用一个中介对象封装一系列对象的交互过程。中介使各对象不需要显示引用，从而降低了耦合，可以独立的改变它们的交互。



# Level

	Class	Component	Module
Command			√
Observer		√	√
Chain Of Responsibility		√	
Mediator	√	√	√