**Padrão de Projeto: Chain of Responsability**

**O que é?**

O padrão de projeto Chain of Responsibility é um padrão comportamental que permite construir uma cadeia de objetos que são responsáveis por lidar com uma solicitação. Cada objeto na cadeia possui uma chance de lidar com a solicitação ou passá-la para o próximo objeto na cadeia.

Basicamente, quando um objeto recebe uma solicitação, ele decide se processa a solicitação ou a passa para o próximo objeto na cadeia. Isso continua até que a solicitação seja tratada por um dos objetos ou chegue ao final da cadeia sem ter sido manipulada.

Esse padrão é útil quando há múltiplos objetos que podem tratar uma solicitação, mas não se sabe qual será capaz de lidar com ela no momento em que o código é escrito. Isso oferece flexibilidade para adicionar, remover ou reorganizar os objetos na cadeia sem alterar o cliente que faz a solicitação.

Em resumo, o padrão Chain of Responsibility estabelece uma sequência de objetos que têm a oportunidade de tratar uma solicitação, permitindo que ela seja processada por um desses objetos de acordo com a lógica estabelecida na cadeia.

**Como funciona?**

Claro, o padrão Chain of Responsibility funciona criando uma cadeia de objetos que são capazes de processar uma solicitação. Aqui está uma explicação passo a passo de como ele opera:

**Construção da cadeia**: Você começa definindo uma interface (ou classe abstrata) que declara métodos para processar solicitações e definir um link para o próximo objeto na cadeia. Cada classe concreta implementa essa interface e possui um campo que aponta para o próximo objeto na cadeia.

**Recebimento da solicitação**: Quando uma solicitação é feita, ela é passada para o primeiro objeto na cadeia.

**Verificação e processamento**: O objeto verifica se pode lidar com a solicitação. Se puder, ele a processa e termina. Caso contrário, passa a solicitação para o próximo objeto na cadeia.

**Iteração na cadeia**: Esse processo continua até que um objeto seja capaz de processar a solicitação ou até que a cadeia chegue ao final, sem que nenhum objeto seja capaz de lidar com a solicitação.

**Flexibilidade e modificação da cadeia**: A cadeia pode ser modificada dinamicamente, seja alterando a ordem dos objetos ou adicionando e removendo novos objetos, sem afetar o cliente que faz as solicitações.

**Tratamento de falhas ou situações não processadas**: No final da cadeia, se nenhuma classe conseguir lidar com a solicitação, pode-se definir um comportamento padrão ou enviar uma mensagem de erro.

Em resumo, o padrão Chain of Responsibility permite que uma solicitação percorra uma série de objetos em uma cadeia até que um desses objetos possa processá-la, oferecendo flexibilidade e desacoplamento entre o emissor da solicitação e os objetos que a manipulam.

**Qual problema resolve?**

O padrão Chain of Responsibility resolve o problema de desacoplar remetentes de receptores de solicitações, permitindo que uma solicitação seja processada por diversos objetos sem que o remetente precise conhecer detalhes sobre quem ou como irá tratá-la. Isso oferece flexibilidade para que diferentes objetos possam lidar com uma solicitação, com a opção de alterar dinamicamente a ordem ou os destinatários sem impactar o remetente.

Além disso, o Chain of Responsibility também ajuda a evitar o acoplamento forte entre o emissor da solicitação e os objetos que a processam, facilitando a manutenção, extensão e reutilização do código, já que novos objetos podem ser adicionados à cadeia sem modificar o código do remetente.