**Padrão de Projeto: Decorator**

**O que é?**

Decorator é um padrão estrutural de projeto que permite você inserir novos comportamentos a objetos envolvendo-os em outros objetos (wrappers) que têm esses comportamentos.

**Como funciona?**

A partir de uma classe base, criamos classes decoradoras que oferecem comportamentos adicionais a esta classe, passando-a como parâmetro a essas classes.

Este padrão envolve a criação de classes adicionais que envolvem objetos existentes e fornecem funcionalidades extras sem alterar a estrutura do objeto original. Essas classes decoradoras implementam a mesma interface que o objeto original, permitindo que sejam usadas de maneira intercambiável. Isso promove a extensibilidade do código, pois novas funcionalidades podem ser adicionadas sem modificar as classes existentes.

**Qual problema resolve?**

Com o padrão Decorator, podemos flexibilizar a herança de classes. Quando herdamos uma classe, esta herança é estática, ou seja, não podemos alterar um determinado comportamento de um objeto em tempo de execução. Você precisa substituir um objeto inteiro por outro criado a partir de uma subclasse diferente. Além disso, na maioria das linguagens de programação, as classes filhas podem ter apenas uma classe pai, não podemos herdar várias classes ao mesmo tempo.

Com classes auxiliares que possuem, cada uma, comportamentos extras para uma classe pai, podemos adicionar comportamentos diferentes a uma determinada classe em tempo de execução através dos seus decoradores.