Modelagem de Sistemas

Prof. Dr. Antonio Marcos Selmini

SISTEMAS DE INFORMAÇÃO **ESPM**



Padrão de Projeto Estrutural



Padrão de Projeto Estrutural Composite

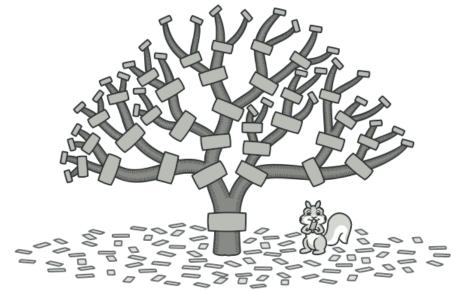
Segundo GoF:

Compor objetos em estruturas de árvore para representarem hierarquias partes-todo. **Composite** permite aos clientes tratarem de maneira uniforme objetos individuais e composições de objetos.

Usar o padrão **Composite** faz sentido apenas quando o modelo central de sua aplicação pode ser representado como uma árvore. **Composite** é um objeto com uma composição de um ou mais objetos semelhantes e que cada objeto possui funcionalidade também semelhante.



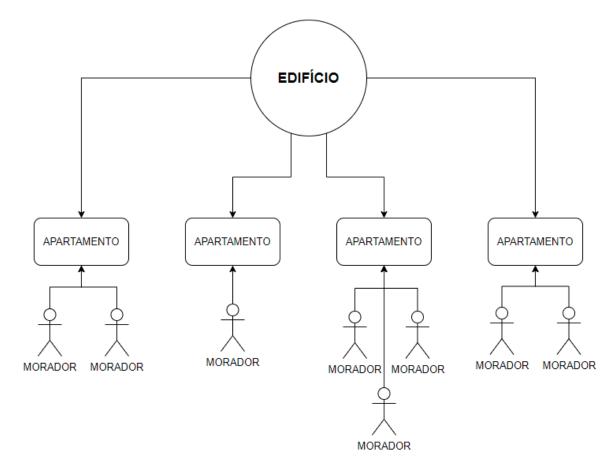
Fonte: https://climaco.medium.com/design-pattern-composite-7fcc39c08ff2. Acesso em 05/10/2024.



Fonte: https://refactoring.guru/pt-br/design-patterns/composite. Acesso em 05/10/2024.

Exemplo ilustrativo: imagine um edifício com vários apartamentos, onde em cada apartamento moram pessoas. Este cenário pode ser modelado usando o padrão **Composite**, no qual os apartamentos são considerados "compostos" que contêm várias pessoas. Cada componente do sistema, seja um apartamento ou uma pessoa, deve ser tratado de maneira uniforme pelo sistema de gerenciamento do edifício. Assim, operações como listar todos os moradores ou verificar quantas pessoas estão em cada apartamento podem ser realizadas de forma simples e recursiva.

Dessa forma, tem-se: o edifício seria um Composite pois é composto de um grupo de apartamentos, já apartamento seria um Composite também, pois compõe-se de um ou mais moradores e o morador seria uma unidade isolada, vejamos mais sobre essa explicação juntamente com diagrama UML.

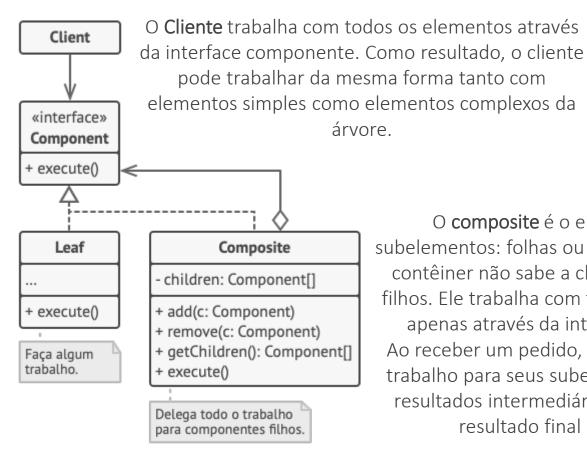


Fonte: https://climaco.medium.com/design-pattern-composite-7fcc39c08ff2. Acesso em 05/10/2024.

Padrão Estrutural Composite – estrutura

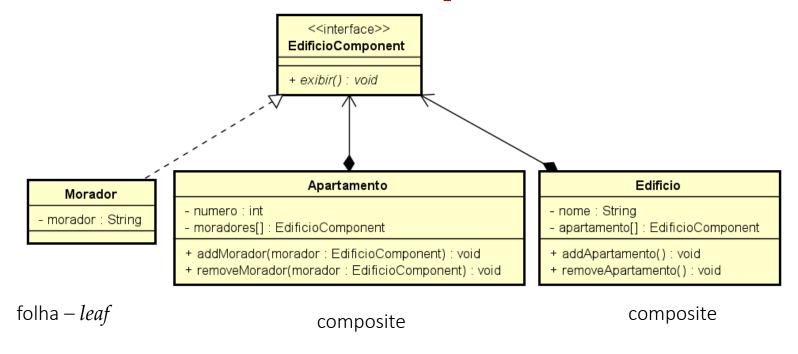
A interface **componente** descreve operações que são comuns tanto para elementos simples como para elementos complexos da árvore.

A folha é um elemento básico de uma árvore que não tem subelementos. Geralmente, componentes folha acabam fazendo boa parte do verdadeiro trabalho, uma vez que não tem mais ninguém para delegá-lo.



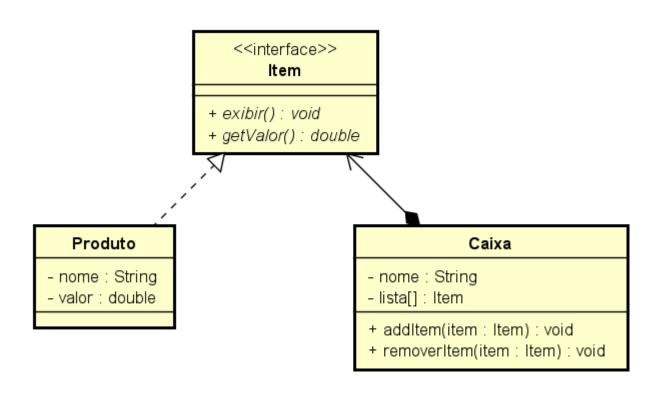
O composite é o elemento que tem subelementos: folhas ou outros contêineres. Um contêiner não sabe a classe concreta de seus filhos. Fle trabalha com todos os subelementos apenas através da interface componente. Ao receber um pedido, um contêiner delega o trabalho para seus subelementos, processa os resultados intermediários, e então retorna o resultado final para o cliente.

Padrão Estrutural Composite – exemplo



Como sugestão de exercício, você pode implementar a estrutura apresentada. Após a implementação e o teste, você poderá fazer uma melhoria adicionando a classe **Andar** com os métodos **addApartamento()** e removeApartamento(). Ajuste a classe **Edificio** para conter andar e não apartamento. Acrescente o método **getTotalMorador()** na interface e faça os ajustes nas classes derivadas. Acrescente o método **pesquisarMorador(String)** para retornar um valor *boolean*.

Padrão Estrutural Composite – exercício



Após codificar as classes mostrada no diagrama ao lado, codifique a classe **Main** para gera alguns produtos e coloca-los em duas caixas diferentes. Uma caixa deve estar "dentro" da outra.

Imprima os dados dos produtos armazenados e também o valor total dos produtos armazenados na caixa.

Padrão Estrutural Composite – aplicação

Sistemas de Arquivos: diretórios (composites) podem conter arquivos e outros diretórios.

Catálogos de Produtos: produtos individuais e pacotes de produtos podem ser tratados de maneira semelhante.

Interface Gráfica: Componentes gráficos como janelas e botões, que podem conter outros componentes.

Sistemas de Pedidos: gerenciamento de itens individuais e pacotes de itens.

Padrão Estrutural Composite – prós e contras

Vantagens do Padrão Composite

- Simplicidade: permite que o código cliente trate objetos simples e compostos de forma uniforme.
- 2. Facilidade de expansão: adicionar novos componentes (folhas ou compostos) é simples e não requer mudanças no código existente.
- Flexibilidade: o padrão facilita o uso de estruturas hierárquicas, como árvores, e a manipulação de objetos complexos com diferentes níveis de composição.

Desvantagens do Padrão Composite

- 1. Complexidade: pode tornar o sistema mais complexo quando a hierarquia de objetos é muito grande ou diversificada.
- 2. Manutenção: se não bem implementado, a manutenção pode se tornar complicada, pois todos os objetos devem seguir a mesma interface.

SISTEMAS DE INFORMAÇÃO **ESPM**