

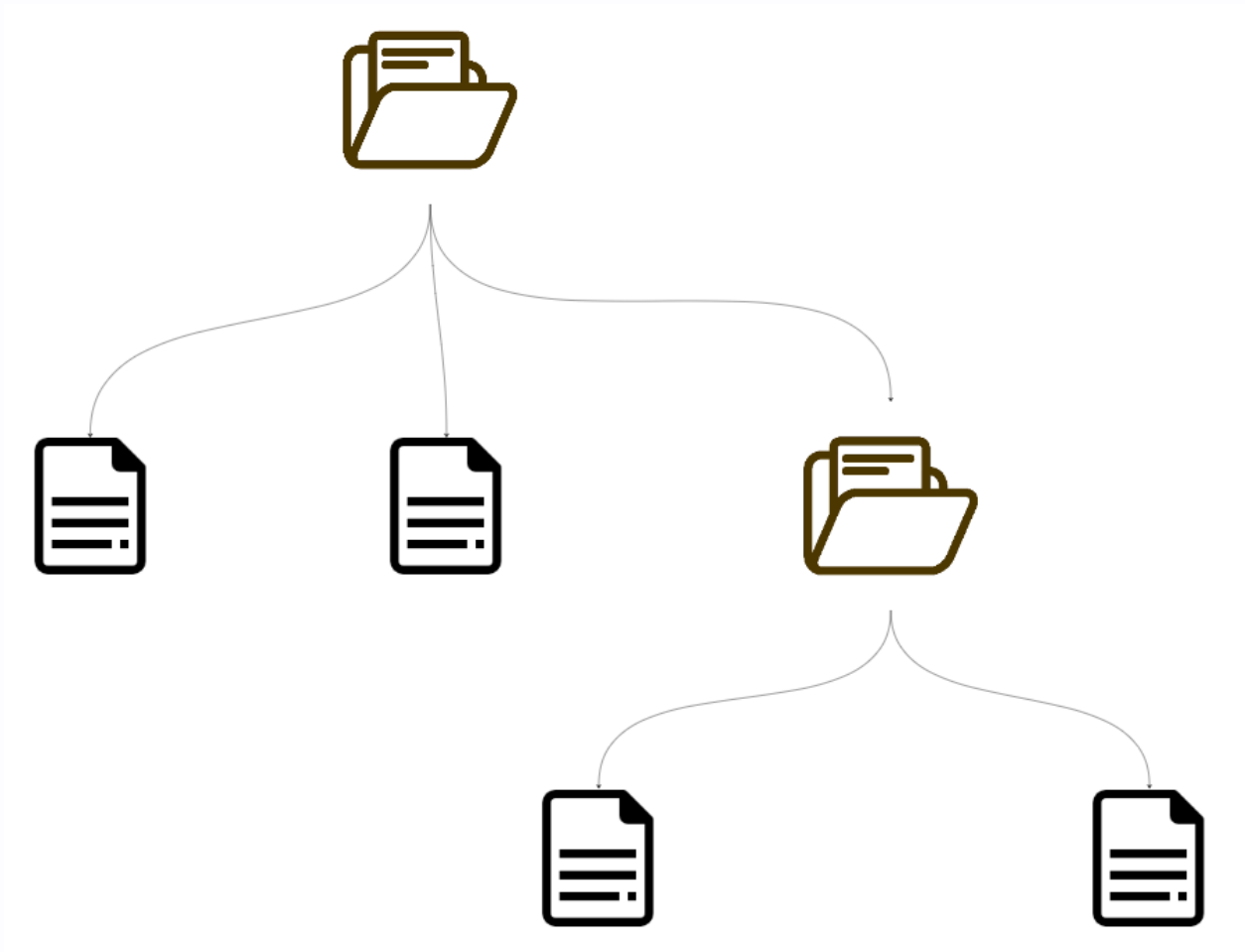
# Composite

Análise e Desenvolvimento de Sistemas - 6º Período

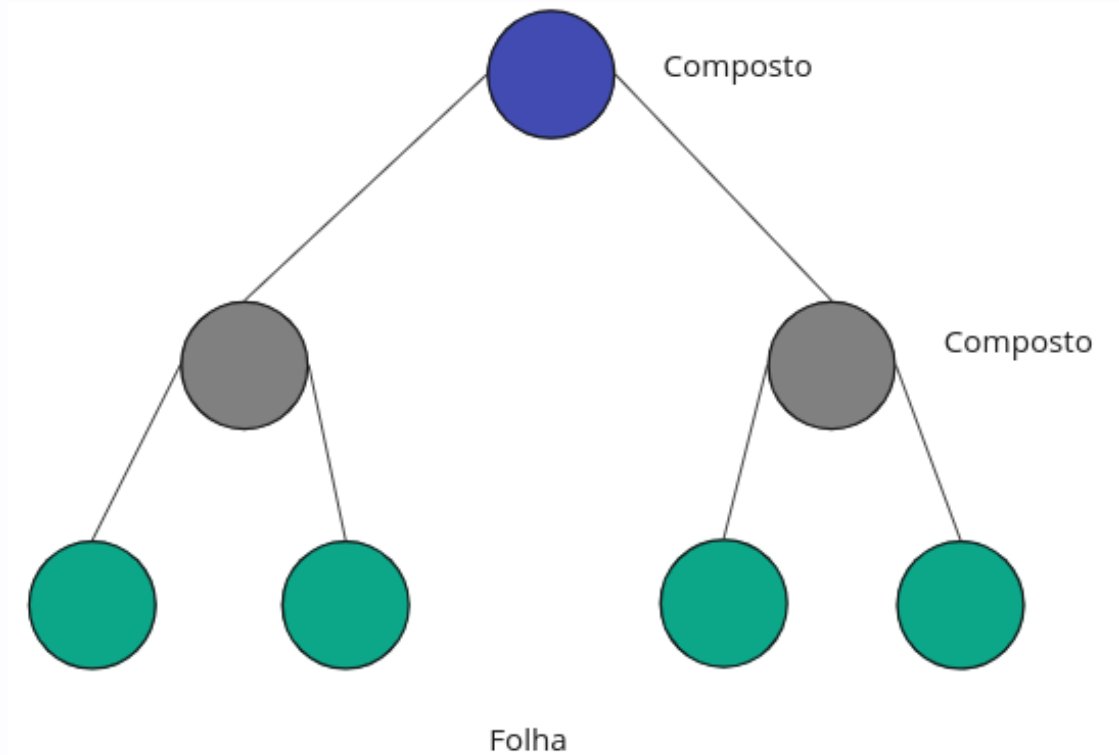
# O padrão Composite

- Padrão estrutural que permite a criação de objeto usando uma estrutura de árvores/hierarquia
- Os objetos criados aqui podem ser utilizados como se fossem objetos individuais

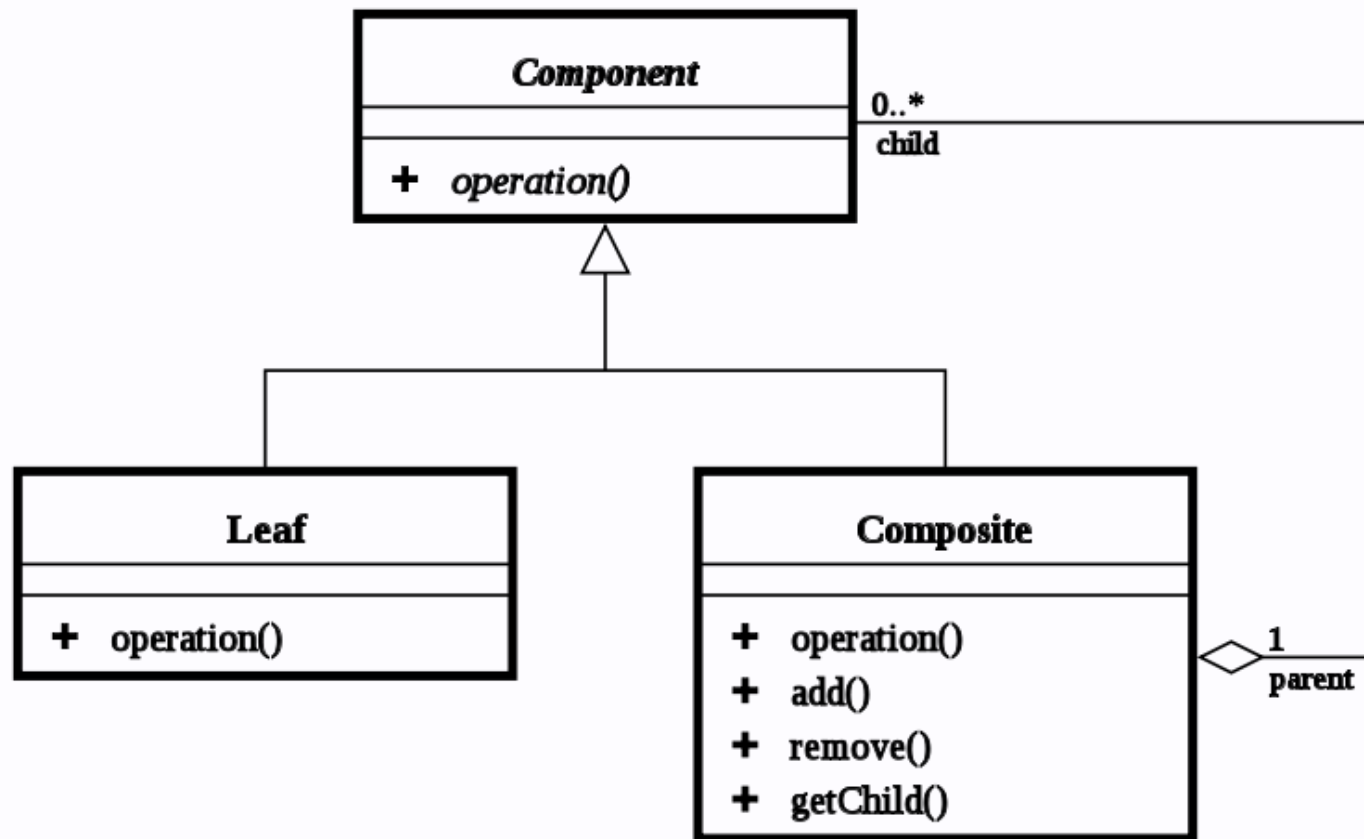
# Hierarquia de objetos



- Podemos fazer uma analogia com uma árvore binária
- Os nós que possuem filhos são os "compostos". As folhas são os objetos "simples"



# O diagrama do Composite



# Os 3 elementos do Composite

- A interface (abstração). É a interface entre os objetos simples e compostos;
- As "folhas", ou objetos simples. Implementações concretas da interface e que vão realizar as tarefas;
- E o "composite", ou objeto composto. É quem vai delegar as execuções para as folhas

# Interface

- O componente que será interface para os objetos simples e compostos;
- Ambos herdam dessa classe e a implementa

# Folhas

- São as classes concretas "folhas";
- Não possuem classes filhas;
- Geralmente são elas que executam as tarefas, delegadas pelo objeto composto



# Composite (objeto)

- Objeto que delega as tarefas para as "folhas", por meio do relacionamento de composição com a interface;
- Não possui relacionamento direto com as folhas;

# Composite (objeto)

- Esse objeto precisa de um "container" que armazena seus outros objetos;
- Esse container geralmente é uma lista ou outra estrutura de dados parecida;
- Podem ser objetos simples ou compostos

# Composite (objeto)

- O composite precisa de uma função para adicionar novos objetos ao seu container e outro para remover

# Tipo de relacionamento no Composite

- No padrão Composite, as classes estabelecem um relacionamento de composição, que é uma forma de associação entre objetos que permite que um objeto contenha outros objetos como parte de sua estrutura.

# Motivações de uso

- Criação de objetos em hierarquias;
- Quando queremos que o código interaja com objetos simples e também compostos;
- Facilidade em criar objetos complexos

# Motivações de uso

- Recursividade: O padrão Composite facilita a implementação de operações recursivas em estruturas de árvore. Por exemplo, percorrer todos os elementos de uma árvore e executar uma operação específica em cada um deles.
- Composição dinâmica. Conseguimos adicionar novos objetos (compostos ou simples) e removê-los;