

# Padrões de Projeto

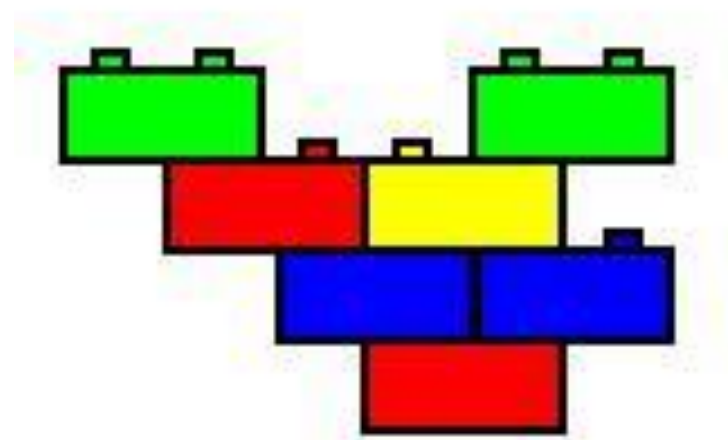
Prof. Adilson Vahldick

Departamento de Engenharia de Software

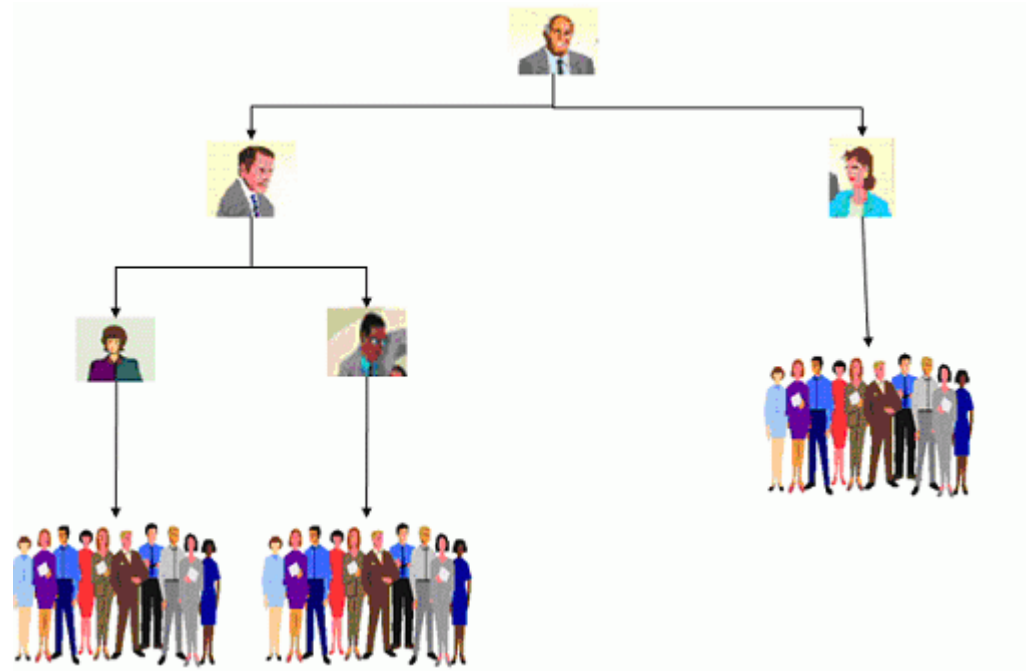
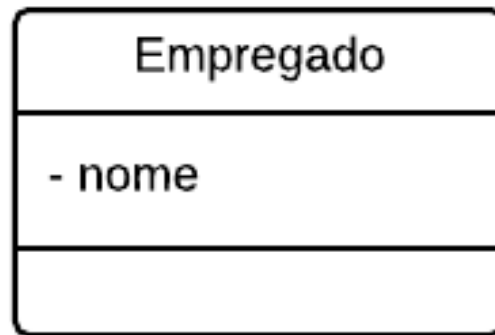
Udesc Ibirama

# Objetivos da aula

- Conhecer e aplicar o padrão
  - Composite

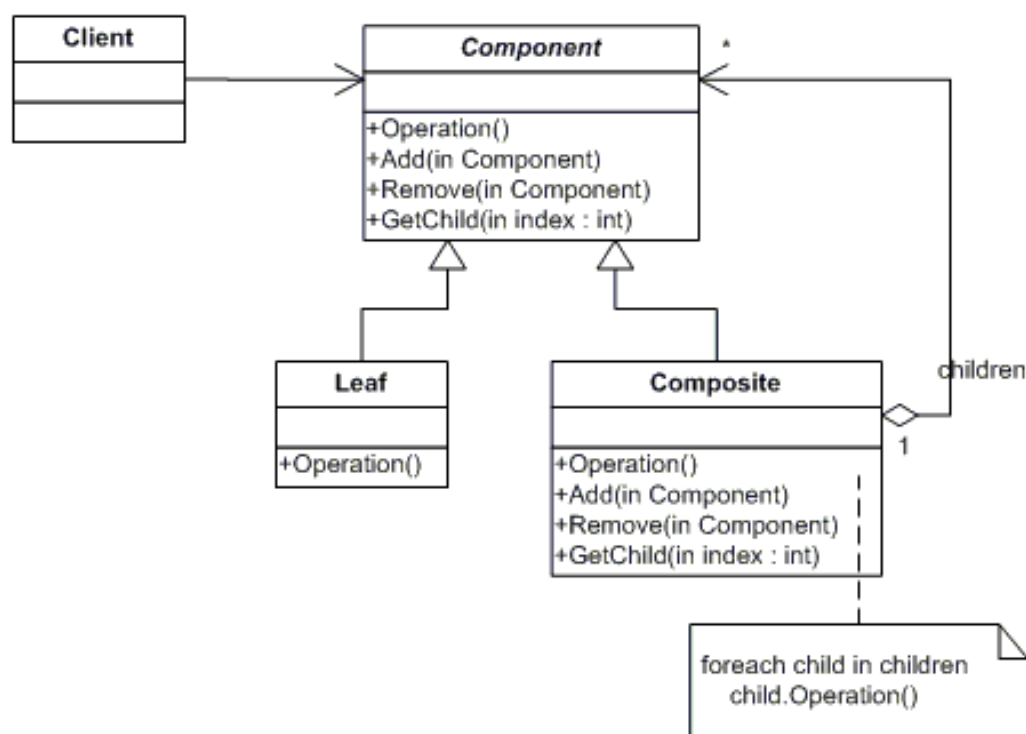


# Problema (1)

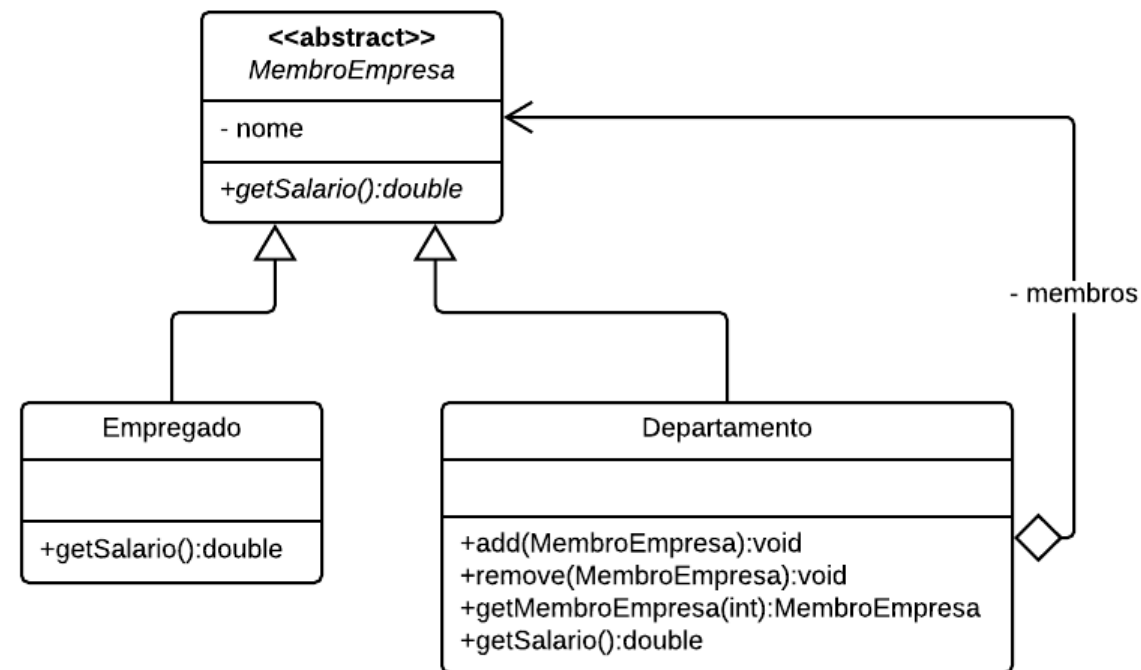
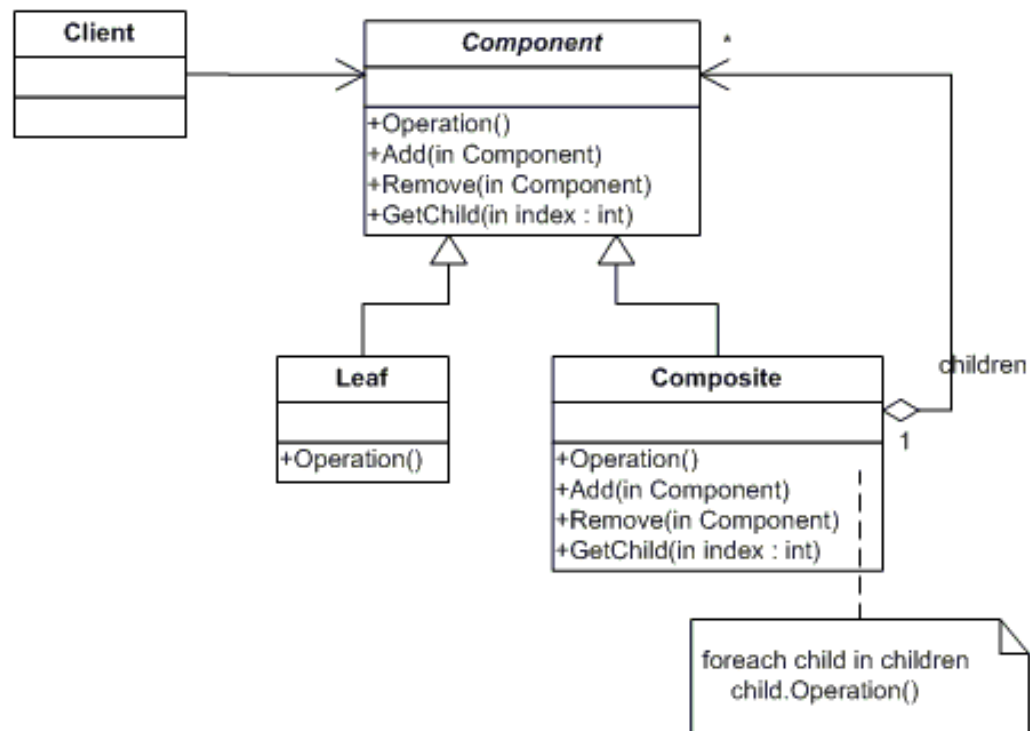


# Solução (1)

- **Composite:** compor objetos em estruturas de árvore para representar hierarquia partes-todo. Permite aos clientes tratarem de maneira uniforme objetos individuais e composições de objetos.



# Solução (2)



# Solução (3) – composite1

```
public abstract class MembroEmpresa {

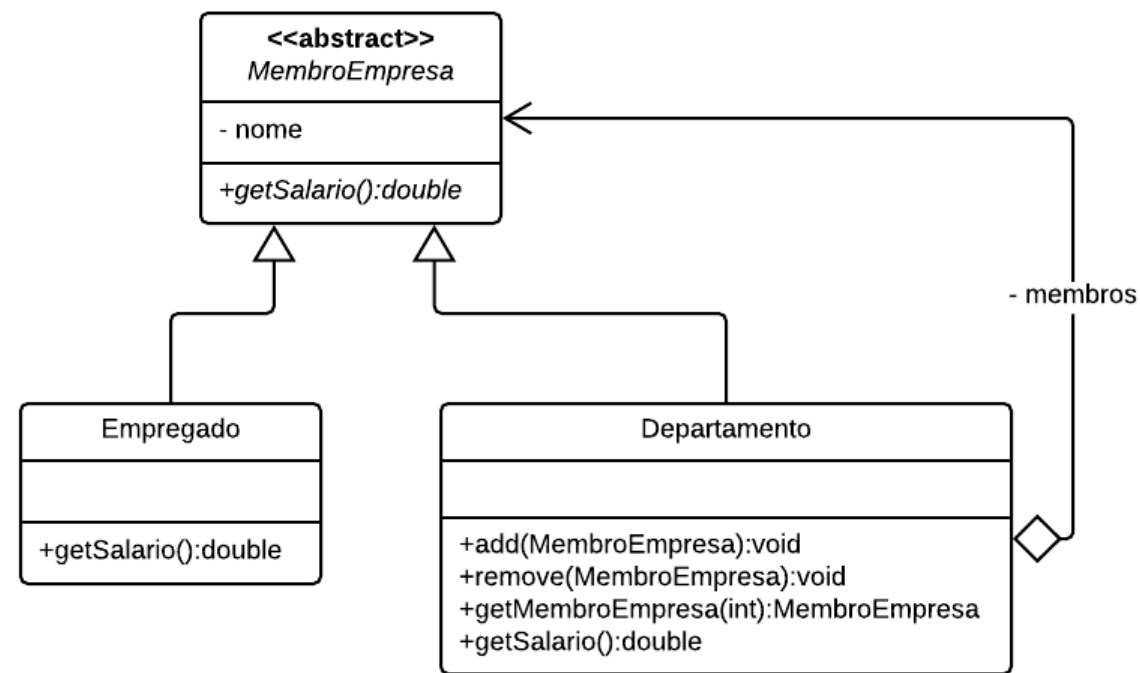
    private String nome;

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public abstract double getSalario();

}
```



# Solução (4) – composite1

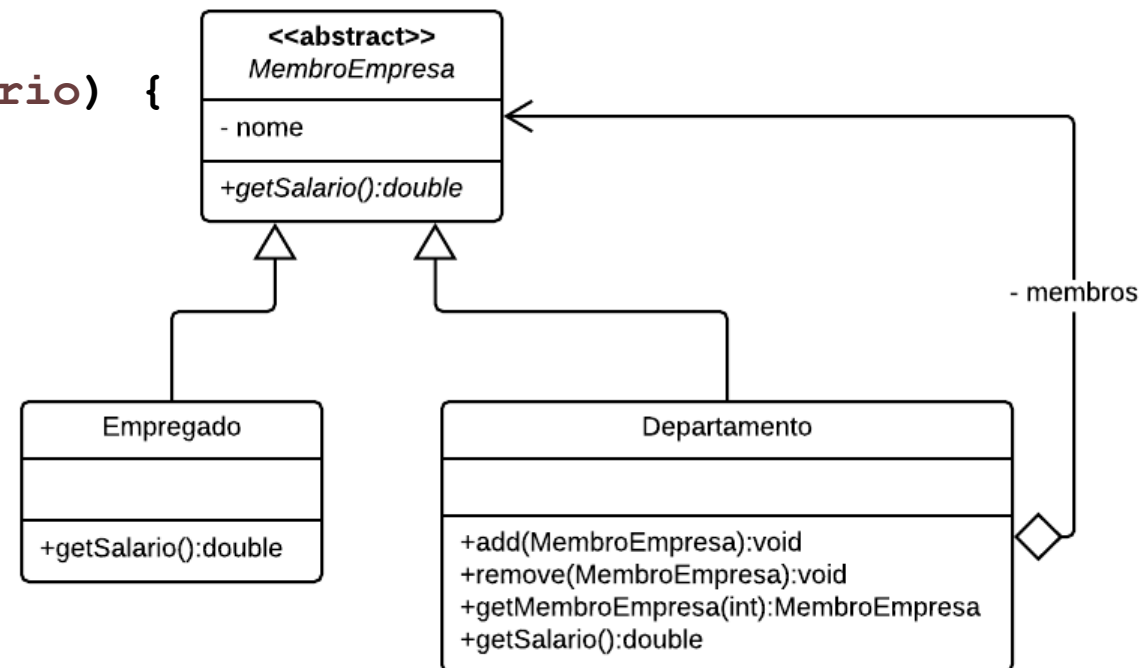
```
public class Empregado extends MembroEmpresa {

    private double salario;

    public Empregado(String nome, double salario) {
        this.setNome(nome);
        this.salario = salario;
    }

    public double getSalario() {
        return salario;
    }

}
```



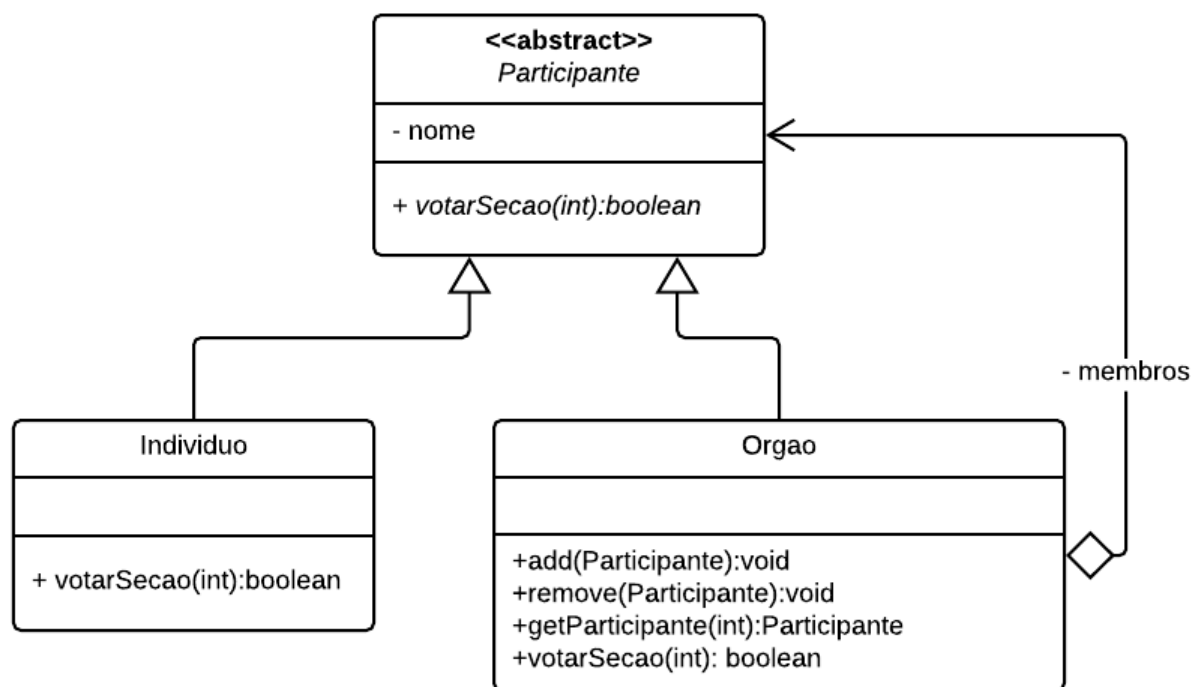
# Solução (5) – composite1

```
public class Departamento extends MembroEmpresa {  
  
    private List<MembroEmpresa> membros = new ArrayList<>();  
  
    public void add(MembroEmpresa membro) { membros.add(membro); }  
  
    public void remove(MembroEmpresa membro) { membros.remove(membro); }  
  
    public MembroEmpresa getMembroEmpresa(int index) { return membros.get(index); }  
  
    public double getSalario() {  
        double total = 0;  
        for (MembroEmpresa m:membros)  
            total += m.getSalario();  
        return total;  
    }  
}
```



# Exercício 1 – composite2

- Falta implementar o método votarSecao da classe Orgao. O Orgao vota conforme a decisão da maioria simples.



# Exercício 2

