

Instituto Federal Catarinense (*Campus Blumenau*)

Professor: Ricardo de La Rocha Ladeira

Matéria: Padrões de Projeto

Nomes: Gabrielli Danker

Turma: BCC 2025.1

Data de entrega: 10 de Abril de 2025

Exercícios

- 1) Você foi contratado para desenvolver um sistema de gerenciamento de menus para um restaurante. O sistema deve permitir a criação de um menu hierárquico, onde cada item do menu pode ser um prato individual ou uma categoria de pratos que contém outros itens. Implemente uma solução em Java (ou em outra linguagem orientada a objetos de sua preferência) utilizando o padrão *Composite* para representar essa estrutura de menu. Para resolver esse exercício, siga os seguintes passos:

- Defina uma interface `ItemCatalogo` com um método `exibirDetalhes()`.
- Crie uma classe `Produto` que represente um item individual do menu e implemente `ItemCatalogo`.
- Crie uma classe `Categoria` que pode conter múltiplos `ItemCatalogo` (pratos e outras categorias).
- No método `main`, construa um menu com pelo menos duas categorias e alguns pratos. Depois, exiba a estrutura do menu chamando `exibirDetalhes()` na raiz.

RESPOSTA:

```
import java.util.ArrayList;
import java.util.List;

interface ItemCatalogo{
    void exibirDetalhes(String prefixo);
}

class Produto implements ItemCatalogo{
    private String nome;
    private double preco;

    public Produto(String nome, double preno){
        this.nome = nome;
        this.preco = preco;
    }
}
```

```

        @Override
        public void exibirDetalhes(String prefixo){
            System.out.println(prefixo + "- " + nome + " (R$ " +
preco + ")");
        }
    }

class Categoria implements ItemCatalogo{
    private String nome;
    private List<ItemCatalogo> itens = new ArrayList<>();

    public Categoria(String nome){
        this.nome = nome;
    }

    public void adicionarItem(ItemCatalogo item){
        itens.add(item);
    }

    public void removerItem(ItemCatalogo item){
        itens.remove(item);
    }

    @Override
    public void exibirDetalhes(String prefixo){
        System.out.println(prefixo + " + " + nome);
        for(ItemCatalogo item : itens){
            item.exibirDetalhes(prefixo + " ");
        }
    }
}

public class MenuRestaurante{
    public static void main(String[] args){
        //Bebidas
        Produto cocacola = new Produto("Coca-cola 2L",
14.00);
        Produto pepsi = new Produto("Pepsi 2L" + 12.00);
        Produto sucoNatural = new Produto("Suco Natural",
10.00);

        //Pratos principais
        Produto lasanha = new Produto("Lasanha", 25.00);
        Produto bifeComFritas = new Produto("Bife com
fritas", 28.00);
    }
}

```

```

        Produto parmegiana = new Produto ("Parmegiana",
30.00);

        //Sobremesas
        Produto pudim = new Produto("Pudim", 10.00);
        Produto sorvete = new Produto("Sorvete", 8.50);
        Produto acai = new Produto("Açaí", 12.00);

        //Categorias
        Categoria bebidas = new Categoria("Bebidas");
        bebidas.adicionarItem(cocacola);
        bebidas.adicionarItem(pepsi);
        bebidas.adicionarItem(sucoNatural);

        Categoria pratosPrincipais = new Categoria("Pratos
Principais");
        pratosPrincipais.adicionarItem(lasanha);
        pratosPrincipais.adicionarItem(bifeComFritas);
        pratosPrincipais.adicionarItem(parmegiana);

        Categoria sobremesas = new Categoria("Sobremesas");
        sobremesas.adicionarItem(pudim);
        sobremesas.adicionarItem(sorvete);
        sobremesas.adicionarItem(acai);

        Categoria menu = new Categoria("Menu Restaurante");
        menu.adicionarItem(bebidas);
        menu.adicionarItem(pratosPrincipais);
        menu.adicionarItem(sobremesas);

        menu.exibirDetalhes("");

    }
}

```

O código se encontra na pasta Composite -> Questão 1

Esse código retorna:

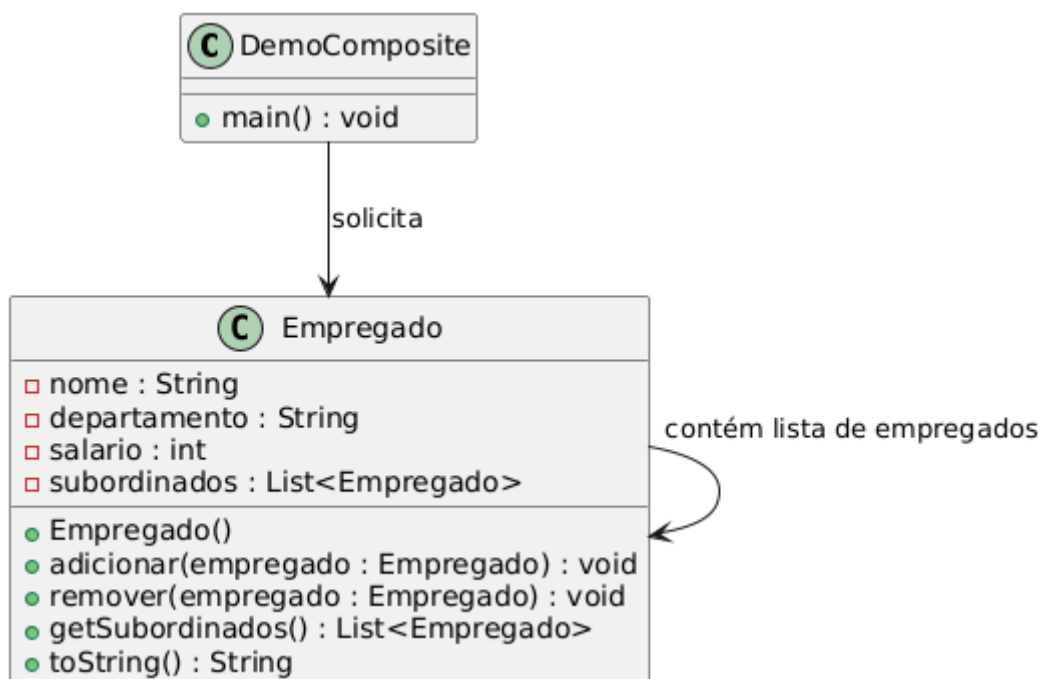
```

PS C:\Users\Blabl\Documents\site-simples\PAP> java MenuRestaurante
+ Menu Restaurante
+ Bebidas
- Coca-cola 2L (R$ 14.0)
- Pepsi 2L (R$ 12.0)
- Suco Natural (R$ 10.0)
+ Pratos Principais
- Lasanha (R$ 25.0)
- Bife com fritas (R$ 28.0)
- Parmegiana (R$ 30.0)
+ Sobremesas
- Pudim (R$ 10.0)
- Sorvete (R$ 8.5)
- Açaí (R$ 12.0)
PS C:\Users\Blabl\Documents\site-simples\PAP>

```

- 2) Use o padrão *Composite* para criar um exemplo que demonstre a hierarquia organizacional dos empregados de uma empresa, com base no diagrama de classes fornecido na Figura 2.

Figura 2. Diagrama de classes do exercício 2.



Fonte: elaborado pelo autor.

RESPOSTA:

```

import java.util.ArrayList;
import java.util.List;

class Empregado {
    private String nome;
    private String departamento;
    private int salario;
    private List<Empregado> subordinados;

    public Empregado(String nome, String departamento, int salario) {
        this.nome = nome;
        this.departamento = departamento;
        this.salario = salario;
        this.subordinados = new ArrayList<>();
    }

    public void adicionar(Empregado empregado) {
        subordinados.add(empregado);
    }

    public void remover(Empregado empregado) {
        subordinados.remove(empregado);
    }

    public List<Empregado> getSubordinados() {
        return subordinados;
    }

    @Override
    public String toString() {
        return "Empregado: " + nome + ", Departamento: " + departamento
+ ", Salário: R$" + salario;
    }
}

```

```

public class DemoComposite {
    public static void main(String[] args) {
        // Criando objetos Empregado
        Empregado ceo = new Empregado("Ana", "Diretoria", 10000);
        Empregado gerenteTI = new Empregado("Bruno", "TI", 8000);
        Empregado dev1 = new Empregado("Carlos", "TI", 5000);
        Empregado dev2 = new Empregado("Diana", "TI", 5000);
        Empregado gerenteRH = new Empregado("Elisa", "RH", 7500);
    }
}

```

```

    Empregado rh1 = new Empregado("Fábio", "RH", 4000);

    // Montando a hierarquia
    ceo.adicionar(gerenteTI);
    ceo.adicionar(gerenteRH);

    gerenteTI.adicionar(dev1);
    gerenteTI.adicionar(dev2);

    gerenteRH.adicionar(rh1);

    // Exibindo a hierarquia
    exibirHierarquia(ceo, 0);
}

public static void exibirHierarquia(Empregado emp, int nivel) {
    String prefixo = "  ".repeat(nivel);
    System.out.println(prefixo + emp.toString());
    for (Empregado subordinado : emp.getSubordinados()) {
        exibirHierarquia(subordinado, nivel + 1);
    }
}
}

```

O código está na pasta Composite -> Questão 2.

Esse código retorna:

```

PS C:\Users\Blabl\Documents\site-simples\PAP> java DemoComposite
Empregado: Ana, Departamento: Diretoria, Salário: R$10000
  Empregado: Bruno, Departamento: TI, Salário: R$8000
    Empregado: Carlos, Departamento: TI, Salário: R$5000
    Empregado: Diana, Departamento: TI, Salário: R$5000
  Empregado: Elisa, Departamento: RH, Salário: R$7500
    Empregado: Fábio, Departamento: RH, Salário: R$4000
PS C:\Users\Blabl\Documents\site-simples\PAP>

```