

Disciplina: POO

Sistema de Pedidos em uma Cafeteria

Você foi contratado para desenvolver um sistema simples de pedidos para uma cafeteria. O sistema deve registrar pedidos de diferentes tipos de bebidas, como cafés e chás, e calcular o preço total dos itens. A cafeteria oferece opções como cafés com diferentes tipos de leite e chás com sabores variados. As bebidas podem ser servidas em diferentes tamanhos e com ou sem adicionais.

Requisitos:

1. Classe abstrata Bebida:

- Deve ter os seguintes atributos:
 - String nome (nome da bebida)
 - double precoBase (preço base da bebida)
 - String tamanho (tamanho da bebida, podendo ser "Pequeno", "Médio" ou "Grande")
- Método abstrato:
 - public abstract double calcularPreco(): para calcular o preço final da bebida, que pode ser diferente com base no tamanho ou em adicionais.

2. Classe concreta Cafe (herda de Bebida):

- o Atributo adicional:
 - String tipoLeite (pode ser "Integral", "Desnatado", "Vegetal", ou "Sem Leite").
- Implementar o método calcularPreco(), onde o preço final depende do tamanho do café e se há um acréscimo para tipos de leite diferentes (exemplo: "Vegetal" custa mais).

3. Classe concreta Cha (herda de Bebida):

- Atributo adicional:
 - String sabor (pode ser "Camomila", "Hortelã", "Chá Preto" ou "Chá Verde").
- Implementar o método calcularPreco(), onde o preço final depende do tamanho do chá e se há um acréscimo para sabores especiais.

4. Classe Pedido:

- Deve conter uma lista de bebidas (ArrayList<Bebida>) para armazenar os pedidos.
- Métodos:
 - adicionarBebida(Bebida bebida): para adicionar uma bebida ao pedido.
 - calcularTotal(): para calcular o valor total de todas as bebidas no pedido.

5. Polimorfismo:

 A lista de bebidas no pedido deve permitir tanto instâncias de Cafe quanto de Cha. O cálculo do preço final de cada bebida será feito usando polimorfismo (chamando o método calcularPreco() de forma dinâmica).

Exemplo:

```
import java.util.ArrayList;

// Classe abstrata Bebida
abstract class Bebida {
   protected String nome;
   protected double precoBase;
   protected String tamanho;

   public Bebida(String nome, double precoBase, String tamanho) {
      this.nome = nome;
   }
}
```

```
this.precoBase = precoBase;
        this.tamanho = tamanho;
    }
   public abstract double calcularPreco();
   public String getNome() {
        return nome;
    }
}
// Classe Cafe
class Cafe extends Bebida {
   private String tipoLeite;
    public Cafe(String nome, double precoBase, String tamanho,
String tipoLeite) {
        super(nome, precoBase, tamanho);
       this.tipoLeite = tipoLeite;
    }
   @Override
   public double calcularPreco() {
        double precoFinal = precoBase;
        if (tamanho.equals("Médio")) {
            precoFinal += 2.0;
        } else if (tamanho.equals("Grande")) {
```

```
precoFinal += 4.0;
        }
        if (tipoLeite.equals("Vegetal")) {
            precoFinal += 1.5;
        }
        return precoFinal;
}
// Classe Cha
class Cha extends Bebida {
    private String sabor;
    public Cha(String nome, double precoBase, String tamanho,
String sabor) {
        super(nome, precoBase, tamanho);
        this.sabor = sabor;
    }
    @Override
    public double calcularPreco() {
        double precoFinal = precoBase;
        if (tamanho.equals("Médio")) {
            precoFinal += 1.5;
        } else if (tamanho.equals("Grande")) {
            precoFinal += 3.0;
```

```
if (sabor.equals("Chá Preto")) {
            precoFinal += 1.0;
        }
        return precoFinal;
   }
}
// Classe Pedido
class Pedido {
   private ArrayList<Bebida> bebidas = new ArrayList<>();
   public void adicionarBebida(Bebida bebida) {
        bebidas.add(bebida);
    }
   public double calcularTotal() {
        double total = 0.0;
       for (Bebida bebida : bebidas) {
            total += bebida.calcularPreco();
        return total;
}
```

```
// Classe principal
public class Main {
    public static void main(String[] args) {
        Pedido pedido = new Pedido();
        Cafe cafe1 = new Cafe("Café Expresso", 5.0, "Grande",
"Vegetal");
        Cha cha1 = new Cha("Chá Verde", 4.0, "Médio", "Chá
Verde");
        pedido.adicionarBebida(cafe1);
        pedido.adicionarBebida(cha1);
        System.out.println("Total do pedido: R$ " +
pedido.calcularTotal());
    }
}
```



Tarefas:

- 1. Crie as classes seguindo a estrutura proposta.
- 2. Adicione mais tipos de bebidas (por exemplo, cafés gelados, diferentes tipos de chás).
- 3. Expanda o sistema permitindo que o cliente adicione adicionais (exemplo: adoçante, chantilly) e ajuste os preços de acordo.
- 4. Teste o sistema com diferentes combinações de bebidas e tamanhos.