

Exercício - Sistema de Pagamento

Enunciado

Você vai implementar um sistema para simular diferentes formas de pagamento em um e-commerce. O sistema deve permitir a realização de pagamentos utilizando cartão de crédito, boleto e Pix. Além disso, deve permitir que o cliente faça o pagamento com parcelas ou em uma única vez.

Requisitos:

1. **Classe Abstrata `FormaDePagamento`:

- Deve ter um método abstrato `realizarPagamento(double valor)` para processar o pagamento.
- Pode ter um construtor que inicializa o nome da forma de pagamento.

2. **Classe `CartaoDeCredito`:

- Deve herdar de `FormaDePagamento`.
- Deve implementar o método `realizarPagamento(double valor)`.
- Deve ter um construtor que recebe o número do cartão e o nome do titular.
- Deve ter uma sobrecarga do método `realizarPagamento` para permitir o pagamento parcelado, ou seja, com um parâmetro extra indicando o número de parcelas.

3. **Classe `Boleto`:

- Deve herdar de `FormaDePagamento`.
- Deve implementar o método `realizarPagamento(double valor)`.
- Deve ter um construtor que recebe o código de barras do boleto.

4. **Classe `Pix`:**

- Deve herdar de `FormaDePagamento`.
- Deve implementar o método `realizarPagamento(double valor)`.
- Deve ter um construtor que recebe a chave Pix.

5. **Classe `TestePagamento`:**

- Classe principal com o método `main` para criar instâncias de `CartaoDeCredito`, `Boleto`, e `Pix`, e realizar pagamentos com diferentes valores e modos de pagamento.

Dicas:

- Utilize a palavra-chave `super` para chamar o construtor da classe `FormaDePagamento` nas subclasses.
- Para sobrecarregar métodos, crie o método `realizarPagamento()` com diferentes assinaturas.

Código de Exemplo

```
```java
```

```
// Classe Abstrata FormaDePagamento
```

```
abstract class FormaDePagamento {
```

```
 protected String nome;
```

```
 public FormaDePagamento(String nome) {
```

```
 this.nome = nome;
```

```
 }
```

```
 public abstract void realizarPagamento(double valor);
```