

Curso Superior de Tecnologia em Sistemas Para Internet
Disciplina de Programação Orientada a Objetos
Prova Prática – Avaliação 01 do Semestre (Peso: 4,0 | Valor: 100 pontos)
Professor: Everton Fernandes

Nome:

Turma:

Data:

Instruções Gerais

- A prova é individual.
 - Você terá **2h15min** para a execução.
 - É **proibido o uso de internet** e de **ferramentas de IA** como ChatGPT, GitHub Copilot, Cursor etc. O uso resultará em **anulação da prova e nota 0**.
 - O **uso de celular é proibido** durante toda a avaliação.
 - A consulta é permitida apenas ao conteúdo armazenado em **pen drive pessoal**, incluindo materiais da disciplina e também matérias de livre escolha.
-

Objetivo

Desenvolver um **sistema de e-commerce orientado a objetos**, simulando a estrutura de grandes plataformas como Amazon ou Mercado Livre, utilizando:

- Encapsulamento com atributos privados
- Getters e Setters (usar somente quando necessário)
- Construtores e sobrecarga
- Enums com e sem valores
- Agregação e composição entre objetos
- Listas + forEach
- Regras de negócio e lógica condicional

Estrutura Obrigatória do Sistema

Você deverá implementar as seguintes **classes e enums**:

♦ Enum **CupomDesconto** (Enum com valores)

Tipos de cupons e seus valores:

- **NENHUM** → R\$ 0
- **CUSTOFRETE** → R\$ 15
- **FRETEGRATIS** → R\$ -15 (frete grátis se valor total > R\$ 100)
- **DESCONTO15** → R\$ 15
- **DESCONTO20** → R\$ 20
- **DESCONTO50** → R\$ 50
- **DESCONTO70** → R\$ 70

Cada cupom deve conter um **valor em reais** acessível por meio de um **getter**.

♦ Enum **FormaPagamento** (Enum simples sem valores)

- **PIX**
 - **DINHEIRO**
 - **CREDITO**
 - **DEBITO**
 - **BOLETO**
-

♦ Enum **CategoriaProduto** (Enum simples sem valores)

- **ELETRONICO**
- **ROUPA**
- **LIVRO**
- **ACESSORIO**

♦ Classe **Produto**

Atributos obrigatórios:

- **nome**

- `preco`
 - `categoriaProduto`
-

♦ Classe **Cliente**

Atributos obrigatórios:

- `nome` (via Scanner)
 - `endereço` (via Scanner)
 - `formaPagamento` (informado manualmente no construtor)
-

♦ Classe **Pedido**

Atributos obrigatórios:

- `cliente` (agregação)
 - `listaDeProdutos` (composição)
 - `cupomDesconto` (opcional)
 - `valorTotal`
-

Métodos obrigatórios:

`adicionarProduto(...)`

Adicione um produto à lista.

`exibirDetalhesDaCompra()`

Imprima:

- Nome e endereço do cliente
 - Lista de produtos e com seus nomes e preços (usar `forEach`)
 - Forma de pagamento
 - Valor final após aplicar as **regras de desconto abaixo**
-

Regras de Negócio a serem aplicar quando o método `exibirDetalhesDaCompra()` for chamado:

Descontos aplicáveis:

1. **Cupom `DESCONT070` ->** Só pode ser usado com às seguintes condições:
 - Se tiver mais de 6 produtos na lista de produtos do pedido
 - Se o valor total do pedido for maior ou igual à R\$ 350
 - Se pelo menos um produto da lista de produtos for da categoria `ELETRONICO`
 - Se o pedido do cliente tiver a forma de pagamento: `PIX`
 - **Se todas as condições acima forem verdadeiras, aplicar R\$ 70 de desconto.**
2. **Cupom `DESCONT050` ->** Só pode ser usado com às seguintes condições:
 - Se tiver mais de 5 produtos na lista de produtos do pedido
 - Se o valor total do pedido for maior que R\$ 350
 - Se pelo menos um produto da lista de produtos for da categoria `ELETRONICO`
 - **Se todas as condições acima forem verdadeiras, aplicar R\$ 50 de desconto no valor total da compra.**
3. **Cupom `DESCONT020` ->** Só pode ser usado com às seguintes condições:
 - Se tiver mais de 3 produtos na lista de produtos do pedido
 - Se o valor total do pedido for maior que R\$ 150
 - **Se todas as condições acima forem verdadeiras, aplicar R\$ 20 de desconto no valor total da compra.**
4. **Cupom `FRETEGRATIS`**
 - **Se o valor total do pedido for maior que R\$ 100, aplicar R\$ 15 de desconto como frete grátis.**
5. **Cupom `DESCONT015` ou `CUSTOFRETE`**
 - Sem regras específicas. Podem ser usados livremente.

6. Cupom **NENHUM**

- **IGNORAR TODAS AS REGRAS**, aplicando nenhum desconto.

ATENÇÃO: Apenas um cupom será aceito por pedido. Caso o cupom informado não atenda aos requisitos, mostre na tela **o valor total do pedido sem nenhum desconto aplicado**.

Mensagens Finais de Pagamento

Forma de Pagamento	Mensagem Final
PIX / DINHEIRO	Compra efetuada com sucesso! Entrega no endereço: [endereço do cliente]
CREDITO / DEBITO	Compra em processamento. Entrega após confirmação no endereço: [endereço do cliente]
BOLETO	Aguardando pagamento do boleto. Entrega após compensação bancária.

Classe **Main**

Deve conter:

- Leitura via Scanner:
 - Para obter o nome do cliente
 - E obter o endereço do cliente
- Instanciar o cliente
- Instanciar produtos manualmente
- Criar um pedido usando cupom de desconto **DESCONT070**
- Criar outro pedido com apenas um produto sem nenhum cupom de desconto informado (**utilize sobrecarga no construtor**)
- Adicionar produtos ao pedido

- Chamar o método **exibirDetalhesDaCompra()** dentro da classe pedido, ao final mostrar os detalhes da compra na tela
-

Critérios de Avaliação (Total: 100 pts)

Critério	Pontos
Encapsulamento + Getters/Setters bem aplicados	5 pts
Construtores + Sobrecarga de métodos(mínimo 1 uso obrigatório)	5 pts
Enum com e sem valores corretamente utilizados	5 pts
Associação e composição entre objetos	10 pts
Uso de List + forEach para exibir produtos	10 pts
Clareza ao exibir detalhes da compra, organização e boas práticas	20 pts
Regras de negócio + lógica condicional + validações	45 pts
Total	100 pts

Observações finais:

- Todos os atributos devem ser **private**.
- Use setters somente quando necessário(evite modelos anêmicos).
- O código deve ser testável e bem estruturado, certifique-se de criar e usar corretamente os enums e objetos na classe main corretamente.