### **Projeto-1 Java**

**Objetivo:** Implementar um sistema de vendas de produtos utilizando conceitos de programação orientada a objetos: métodos, encapsulamento, construtores, herança, interfaces, classes abstratas e tratamento de exceções.

### **Descrição:**

Você deverá criar um sistema para gerenciar diferentes tipos de produtos. O sistema deve ser capaz de gerenciar informações sobre eletrônicos, roupas e alimentos, exibindo detalhes específicos para cada tipo de produto. Além disso, o sistema deve permitir a adição, remoção e listagem de produtos, bem como a emissão de um pequeno relatório em arquivo de texto.

#### **1. Criação de Classes e Interfaces**

**Interface Produto**

Métodos:

public String getTipo();

public String getNome();

public double getPreco();

public void setNome(String nome);

public void setPreco(double preco);

public void setQuantidade(int quatidade);

public int getQuantidade();

public String exibirDetalhes();

**Classe abstrata ProdutoBase**

Implementa a interface Produto.

Atributos:

private String nome;

private double preco;

private int quantidade;

Métodos:

Implementar todos os métodos da interface Produto.

Métodos concretos:

public ProdutoBase(String nome, double preco, int quantidade)

public ProdutoBase()

Métodos abstratos:

public abstract String exibirDetalhes();

**Classe Eletronico**

Herda de ProdutoBase.

Atributos adicionais:

private int garantiaMeses;

Métodos:

Construtores:

public Eletronico(String nome, double preco, int garantiaMeses, int quantidade)

public Eletronico()

public int getGarantiaMeses();

public void setGarantiaMeses(int garantiaMeses);

Implementar o método exibirDetalhes(), exibindo e retornando todas as informações do eletrônico.

public String getTipo(), retornando "Eletrônico".

**Classe Roupa**

Herda de ProdutoBase.

Atributos adicionais:

private String tamanho;

private String cor;

Métodos:

Construtores:

public Roupa(String nome, double preco, String tamanho, String cor, int quantidade)

public Roupa()

public String getTamanho();

public void setTamanho(String tamanho);

public String getCor();

public void setCor(String cor);

Implementar o método exibirDetalhes(), exibindo todas as informações da roupa.

public String getTipo(), retornando "Roupa".

**Classe Alimento**

Herda de ProdutoBase.

Atributos adicionais:

private String dataValidade;

Métodos:

Construtores:

public Alimento(String nome, double preco, String dataValidade, int quantidade)

public Alimento();

public String getDataValidade();

public void setDataValidade(String dataValidade);

Implementar o método exibirDetalhes(), exibindo todas as informações do alimento.

public String getTipo(), retornando "Alimento".

**Classe CarrinhoDeCompras**

Atributos:

private List<Produto> itens;

demais atributos para escrita de arquivo de texto.

Métodos:

Construtor:

public CarrinhoDeCompras(), inicializando a lista de itens.

public void adicionarItem(Produto produto), para adicionar um produto ao carrinho.

public void removerItem(Produto produto), para remover um produto do carrinho.

public void listarItens(), para exibir os detalhes de todos os produtos no carrinho.

public List <Produto> exportarListaProdutos(), para devolver a lista de produtos do carrinho.

public double calcularTotal(), para calcular o valor total dos produtos no carrinho.

Obs.: para excluir um item da lista de produtos do carrinho, talvez seja necessário buscar pelo seu nome e quantidade, visto que não temos um identificador único para esse caso.

public void gerarArquivoTexto(String nomeArquivo).

Esse método deverá gravar um arquivo de texto com o formato similiar ao exemplo abaixo:

item: Qtd: Nome: Preço: SubTotal:

1 3 Smarthphone 2999.0 8997.0

2 6 T-Shirt 89.99 539.93

3 4 Arroz 9.99 39.96

Total: 9576,90

Essa classe, deverá ter suas potenciais exceções tratadas.

**Classe Principal**

Método main:

Criar duas instâncias de CarrinhoDeCompras.

Na primeira adicionar um Eletronico, uma Roupa, e um Alimento.

Na segunda instância adicionar ou mais itens quaisquer.

Listar os produtos e o total de ambos os CarrinhoDeCompras.

Gravar os arquivos referentes a cada carrinho de compras.

### **Requisitos Adicionais:**

Utilize encapsulamento para proteger os atributos das classes.

Utilize herança para compartilhar a lógica comum entre as classes Eletronico, Roupa e Alimento.

Utilize métodos sobrecarregados (opcional) para fornecer diferentes maneiras de criar instâncias de produtos.

A implementação deve ser modular e seguir boas práticas de programação orientada a objetos.

O arquivo de saída (.txt) será gerado na raiz do projeto, caso outro local não seja especificado.