

Tema 2 – Sistema básico para gerenciar estoque – Ponto de vista do comerciante

Classe produto: classe abstrata, guarda dados básicos de um produto

Atributos:

- Código;
- Nome;
- Fabricante;
- Quantidade (estoque);
- Valor unitário;

Métodos:

- `public double vender(int quantidade)` – remove a quantidade pedida do estoque, e retorna o valor de venda equivalente – caso não haja quantidade suficiente, retorne -1;
- `public void comprar(int quantidade)` – adiciona a quantidade pedida ao estoque;
- `public String imprimir()` – retorna todos os dados do produto;

Construtor: apenas um, com todos os dados;

Gets e sets: gets para todos os atributos, quantidade sem set;

Classe Eletrônico: herda de produto, e adiciona os seguintes atributos:

- Voltagem (110, 220, bivolt);
- Potência;
- Classificação de eficiência (A, B, C, D, E);

Deve ter um construtor que receba todos os parâmetros novos e os necessários à classe produto, além de modificar o método imprimir para que os novos dados também sejam retornados.

Classe móvel: herda de produto, e adiciona os seguintes atributos:

- Tipo de material;

Deve ter um construtor que receba todos os parâmetros novos e os necessários à classe produto, além de modificar o método imprimir para que os novos dados também sejam retornados.

Classe perecível: herda de produto, e adiciona os seguintes atributos:

- Validade (utilize um objeto do tipo `LocalDate`);
- Refrigerado (variável booleana que indica se o produto deve ser refrigerado);

Deve ter um construtor que receba todos os parâmetros novos e os necessários à classe produto, além de modificar o método imprimir para que os novos dados também sejam retornados.

Classe gerenciarProdutos – responsável por guardar todos os produtos desse comércio;

Atributo:

- ArrayList de objetos do tipo Produto;

Métodos:

- public void adicionar(Produto p) – adiciona um produto ao ArrayList;
- public boolean remover (int codigo) – busca o produto com o código correspondente e o remove do ArrayList, retornando true – se não encontrar o produto, retorna false;
- public Produto buscarPorCodigo(int codigo) – busca o produto dado pelo código, retornando-o se encontrado. Caso não encontre, retorne null;
- public String listarEletronicos() – lista todos os produtos eletrônicos, guardando seus dados em uma String e depois retornando essa String;
- public String listarPereciveis() – lista todos os produtos perecíveis, guardando seus dados em uma String e depois retornando essa String;
- public String listarMoveis() – lista todos os produtos que sejam móveis, guardando seus dados em uma String e depois retornando essa String;
- public String listarEstoqueMenorQue(int quantidadeMinima) – lista todos os produtos que estejam com estoque menor que o indicado pelo valor quantidadeMinima, guardando os dados em uma String e depois retornando essa String;
- public String listarEstoqueVazio() – lista todos os produtos que estejam com quantidade igual a zero, guardando os dados em uma String e depois retornando essa String;
- public String listarPorFabricante(String fabricante) – lista todos os produtos feitos por um certo fabricante, guardando os dados em uma String e depois retornando essa String;
- public String listarPorValidade(LocalDate validade) – lista todos os produtos que vencerão na data especificada por parâmetro, guardando os dados em uma String e depois retornando essa String;
- public double venderProduto (int codigoProduto, int quantidade) – procura o produto indicado pelo código e vende a quantidade pedida, retornando o valor da venda. Caso o produto não exista, retorne -1;
- public boolean comprarProduto(int codigoProduto, int quantidade) – procura o produto indicado pelo código e adiciona a quantidade passada ao estoque daquele produto, retornando true. Caso não encontre o produto, retorne false;
- public String listarTudo() - lista todos os produtos, guardando os dados em uma String e depois retornando essa String;

Método main – Menu de acesso para todas as funções da classe de gerencia. Lembre-se que é possível adicionar produtos de três tipos, e isso deverá ser pedido ao usuário.