

Trabalho APS

Deverá ser apresentada em sala de aula.

Exercício: Controle de Estoque de <Nome da sua Empresa>

Neste exercício, você será desafiado a criar um sistema de controle de estoque para uma <empresa>, utilizando conceitos de herança, polimorfismo, interfaces, exceções, abstração e Javadoc em Java. Você deve criar classes que representam diferentes tipos de produtos que uma <empresa>, pode vender, permitir a venda desses produtos e lidar com exceções relacionadas à venda. Além disso, é fundamental que você forneça documentação usando Javadoc para explicar as classes, métodos e atributos do seu código.

Instruções:

Crie uma classe abstrata chamada Produto que represente um produto genérico em uma <empresa>. A classe deve conter os seguintes atributos:

- a) nome para o nome do produto.
- b) preco para o preço unitário do produto.
- c) quantidadeEmEstoque para a quantidade disponível em estoque.

Esta classe deve ser abstrata, pois não desejamos instanciar objetos Produto diretamente.

Crie uma interface chamada Vendavel com um método abstrato vender que permitirá a venda de produtos. Esta interface deve ser implementada por todas as classes de produtos que podem ser vendidos na <empresa>.

Crie duas classes que herdam de Produto, uma chamada Alimento e outra chamada Brinquedo, para representar diferentes tipos de produtos vendidos na <empresa>. Cada uma dessas classes deve adicionar atributos específicos e implementar o método vender da interface Vendavel.

Na classe Alimento, adicione um atributo adicional chamado marca para representar a marca do alimento.

Na classe Brinquedo, adicione um atributo adicional chamado material para representar o material do brinquedo.

Crie uma classe de exceção personalizada chamada `EstoqueInsuficienteException` que será lançada quando a tentativa de venda exceder a quantidade disponível em estoque.

Na classe que implementa a interface `Vendavel`, ao vender um produto, verifique se a quantidade a ser vendida é maior do que a quantidade em estoque. Se for, lance uma `EstoqueInsuficienteException`. Caso contrário, atualize a quantidade em estoque de acordo com a venda.

Crie uma classe principal chamada `<empresa>` que contenha o método `main` para simular o controle de estoque. Crie uma lista de objetos do tipo `Vendavel` e adicione alguns produtos de exemplo, incluindo tanto produtos do tipo `Alimento` quanto produtos do tipo `Brinquedo` ao estoque.

Utilize um loop para percorrer a lista de produtos no estoque e exiba as informações de cada produto, incluindo nome, preço, quantidade em estoque e, se aplicável, os atributos específicos (marca para alimentos e material para brinquedos).

Permita a venda de produtos do estoque. No método `main`, demonstre a venda de alguns produtos do estoque, mas verifique se você está tratando corretamente as exceções lançadas ao tentar vender produtos com estoque insuficiente.

Documentação Javadoc: Certifique-se de fornecer documentação usando Javadoc para explicar as classes, métodos e atributos do seu código. Cada classe e método deve ser adequadamente documentado, incluindo descrições, parâmetros, retornos e possíveis exceções lançadas.

Observações:

Utilize a abstração ao criar uma classe abstrata `Produto` para representar os atributos comuns a todos os produtos vendidos na `<empresa>`.

Certifique-se de usar herança e interfaces para garantir que os produtos possam ser vendidos de forma genérica, independentemente de seu tipo.

Utilize o polimorfismo ao tratar objetos de diferentes subclasses como objetos do tipo `Vendavel`.

Certifique-se de manipular corretamente as exceções lançadas ao tentar vender produtos com estoque insuficiente.

Este exercício visa demonstrar como você pode utilizar conceitos de orientação a objetos, herança, polimorfismo, interfaces, exceções, abstração e Javadoc para criar um sistema de controle de estoque de uma `<empresa>` em Java. Lembre-se de manter o código organizado e seguir as melhores práticas de programação.