

PRÁTICA SOBRE COLEÇÕES – LISTA COM ORDENAÇÃO

O dono de um supermercado precisa de um sistema para controlar o estoque de produtos que ele possui na sua loja. Seguindo as boas práticas em programação orientada a objetos, implemente em Java a parte desse sistema responsável pelo cadastro e geração de um relatório dos produtos do estoque.

Nesse sistema, o estoque é caracterizado pelo seu tipo (que pode ser estoque de fluxo ou sazonal) e pela lista de produtos que o compõem. Já o produto é caracterizado pelo seu nome, pela sua marca e quantidade disponível.

Para facilitar a localização dos produtos no relatório a ser gerado por esse sistema, deseja-se que os produtos fiquem ordenados de acordo com o nome dos mesmos. Se houver mais de um produto com o mesmo nome, para esse produto deve-se utilizar como segundo critério de ordenação a marca do mesmo. Em ambos os casos a ordenação deve ser realizada em ordem crescente. Vale observar também que nunca haverá no estoque mais de um produto caracterizado pelos mesmos valores para a tupla {nome, marca}. Segue um exemplo desse relatório que deve ser gerado e impresso na tela pelo sistema (seguindo exatamente essa formatação):

Tipo do estoque: fluxo

Nome	Marca	Quantidade
Arroz	Prato Fino	250
Arroz	Tio Joao	125
Sabonete	Dove	50
Sabonete	Nivea	25
Sabonete	Palmolive	40

Vale observar que a classe que contém o método *main* deverá ser responsável apenas por instanciar todos os objetos necessários e chamar os métodos para cadastrar os produtos do estoque e gerar o relatório de estoque exemplificado anteriormente. Além disso, essa classe não deverá ter qualquer código para impressão de dados na tela.

Os seguintes requisitos também devem ser atendidos para essa implementação:

- Faça uso da classe *ArrayList* e seus métodos *add* e *sort*.
- Faça uso da classe *Comparator* para definir o critério de ordenação dos objetos da lista de produtos. Além disso, implemente um método denominado *getComparator* que retorne o objeto do tipo *Comparator* que será utilizado pelo método *sort* para definir o critério de ordenação dos objetos contidos na lista.
- Não usar inferência de tipo nem expressão lambda na sua implementação.

Obs.: Não há necessidade de solicitar qualquer dado para o usuário do sistema.

ANEXO

```
Comparator<Objeto> comparador = new Comparator<Objeto>( ) {  
    @Override  
    public int compare(Objeto o1, Objeto o2){  
        ...  
    }  
};
```