

### EXERCÍCIO DE FIXAÇÃO Nº 3

1) Dentro de um mercado deseja-se projetar um sistema de controle de vendas. Para isso, a princípio, foram definidas as seguintes classes:

A classe **Produto** possui como atributos o nome do produto, seu código e o preço unitário dele. Seu construtor deve exigir esses três atributos como parâmetros.

Existe uma classe **ItemVenda** que contém como atributos um objeto da classe **Produto** e a sua quantidade. Existem dois construtores para esta classe: uma somente com o objeto de **Produto** e outra com o objeto de **Produto** e sua quantidade. A quantidade, quando não informada, deve ser considerada com o valor 1 (um).

A classe **Cliente** possui como atributos nome e código, ambos **String**. E um único construtor que recebe sempre os dois como parâmetros.

Existe também uma classe **Mercado** que possui uma coleção de objetos da classe **Cliente**, que representa os seus clientes e uma coleção de objetos da classe **Produto**, que representa os produtos existentes no mercado. Seu construtor apenas inicializa esta coleção (*ArrayList*). Deve possuir métodos para adicionar um cliente e outro para removê-lo da sua coleção, assim como também métodos equivalentes para manipular a coleção de produtos.

E uma classe **Venda**, que possui como atributos um objeto de **Mercado**, outro de **Cliente** e uma coleção de objetos da classe **ItemVenda**. O seu construtor recebe o objeto mercado e o código do cliente, inicializando a coleção (*ArrayList*) de objetos de **ItemVenda**. Se o código do cliente informado no construtor não existir na coleção de clientes do objeto da classe **Mercado**, essa venda não pode ser instanciada. Caso exista o código na coleção de clientes, a venda deve ser instanciada e o objeto **Cliente** da classe deve ser recuperado pelo código informado.

A classe **Venda** deve possuir um método que adicione um objeto **ItemVenda** à sua coleção de itens de venda. Para isso, deve receber como parâmetro o código do produto e, opcionalmente, a quantidade. A adição de um objeto **ItemVenda** só será possível se o código informado existir na coleção de produtos do objeto da classe **Mercado**. Caso não exista, a adição do objeto **ItemVenda** não deve acontecer. Caso exista, deve ser criado e adicionado o objeto correspondente de **ItemVenda** à coleção de itens da venda atual.

E, finalmente, a classe **Venda** deve também possuir um método que remova um item de venda (pelo código do objeto de **Produto** ou pela ordem do item na venda) e mais um método que retorne o valor total da venda.

Todas as classes citadas devem possuir todos seus atributos como *private* e disponibilizar métodos acessores (*get* e *set*) para todos eles.

2) Faça também uma classe **Exec** que servirá só para a execução de código que manipulará as classes definidas. Esta classe só terá o método *main* executável, que terá o seguinte funcionamento:

- a) Pergunta as informações necessárias para cadastrar todos os clientes do mercado. O cadastro vai se realizando continuamente até que se digite o código do cliente em branco.
- b) Pergunta as informações necessárias para cadastrar todos os produtos do mercado. O cadastro vai se realizando continuamente até que se digite o código do produto em branco.
- c) Inicie uma venda, informando apenas o código do cliente. Se válido (conforme instruções anteriores), prossegue a venda informando o código de cada produto e sua quantidade. Produtos inválidos (conforme instruções anteriores) não devem ser adicionados como item de venda.
- d) A venda conclui quando o código do produto é informado em branco. Quando isso acontecer, a venda se encerra e é exibida a seguir a lista dos itens de venda (um item por linha) contendo os valores de código do produto, nome do produto, quantidade, preço unitário e preço total do item. O valor total da venda deve ser exibido ao final.

ATENÇÃO: Utilize a classe Scanner para entrada de dados.