

Dojô: Compras

1. Enunciado

Para este exercício, deve-se implementar uma classe "Cliente", que possuirá nome, dinheiro e produtos comprados.

O cliente então "comprará" alguns produtos, listados no array. Ao fazer isso, o dinheiro que ele possui deverá mudar para o troco que ele receberá, caso o preço do produto seja menor ou igual ao montante que ele possui. Essa compra será feita por meio do método "CompraProduto", que deverá receber o dinheiro do cliente, o preço e o nome do produto e deverá retornar o troco, caso o preço do produto seja menor ou igual dinheiro que o cliente possui. Caso contrário, deve apenas retornar o dinheiro que o cliente já possuía e mandar uma mensagem dizendo que o produto não pode ser adquirido.

Ao final, deve-se fazer um método para exibir os dados do cliente. Dentre eles, os produtos comprados.

É interessante fazer a verificação dos campos que serão colocados no cliente. Por exemplo, o dinheiro, assim como os preços dos produtos, não pode ser menor do que 0.

Deve-se criar uma classe principal "Mercado", que conterà um array de Strings com os produtos e um array de doubles com os respectivos preços dos produtos. Nesta classe, serão criados os clientes, com nome e dinheiro, e onde também serão feitas as compras. Para isso, deve-se chamar o método "CompraProduto" da classe "Cliente", cujo retorno será passado para o método "setDinheiro", também da classe "Cliente", visto que o troco da compra passará a ser todo o dinheiro que o cliente terá .

2. Casos de Teste

Entrada	Saída
<pre>String[] produtos = {"Banana","Maçã", "Pão","Arroz","Refrigerante"}; double[] precos = {2.00,5.23,8,12.5,1.9}; Cliente c= new Cliente("Maria", 4); c.setDinheiro(c.CompraProduto(c.getDinheiro(), precos[0],produtos[0])); c.setDinheiro(c.CompraProduto(c.getDinheiro(), precos[3],produtos[3]));</pre>	O produto (Banana) foi comprado com sucesso! Não é possível comprar o produto (Arroz), pois o cliente não tem dinheiro o suficiente! O cliente Maria efetuou as seguintes compras (cronologicamente): Banana

c.imprimirDados();	Lhe restaram 2.0 reais.
--------------------	-------------------------

3. Objetivos

Exercitar a validação da entrada dos dados, a utilização de arrays e a modelagem de entidades.

4. Observações

Não há.