

Testr 2.0

Supermercado em Java

 [Voltar](#)

Deadline: Dia 1 de dezembro de 2025 às 00:00 hora

Linguagem de Programação:

Descrição:

Considere um supermercado em que usuários podem realizar várias compras. Cada compra possui um ou mais produtos adquiridos em diferentes quantidades. Cada produto possui nome, preço e quantidade em estoque. O mesmo produto pode aparecer em várias compras. Além dos produtos e suas quantidades, as compras registram o cliente que fez a compra. Cada cliente possui nome e CPF. Para simular este contexto, faça um programa em Java que leia e execute uma sequência de comandos até que seja digitado o comando ENCERRAR. Os comandos possíveis são:

- REGISTRAR_PRODUTO : adiciona o produto com o preço indicado e 0 itens no estoque.
- REGISTRAR_CLIENTE : adiciona um cliente.
- ADQUIRIR_PRODUTO : incrementa a quantidade de produto no estoque.
- INICIAR_COMPRA : inicia o processo de uma nova compra da pessoa cujo cpf foi indicado. Os próximos comandos serão necessariamente ADD_ITEM ou FINALIZAR_COMPRA.
- ADD_ITEM : adiciona um produto na compra com a quantidade indicada. Se o produto já existir, incrementa a quantidade.
- FINALIZAR_COMPRA: finaliza a compra. Mostra na tela o valor total da compra. As quantidades de produtos vendidas devem ser decrementadas do estoque.
- REVENUE: mostra a soma dos valores de todas as compras feitas no supermercado até o momento.
- MAIS_VENDIDO: mostra o nome do produto com mais itens vendidos

considerando todas as compras.

- MELHOR_CLIENTE: mostra o nome do cliente que fez mais compras.
- EM_FALTA: lista os produtos com menos de 5 itens em estoque.

REGRA: Para tornar o projeto mais divertido, não é válido armazenar como atributos das classes o número de compras feitas por cada cliente e o número de itens vendidos de cada produto. As quantidades dos relatórios devem ser computadas a partir das compras realizadas.

REGRA: Pode ser útil utilizar o tipo [Pair](#) para armazenar pares de valores durante o processo de gerar os relatórios.

Tente organizar o código e isolar as responsabilidades das classes o máximo que você conseguir. Por exemplo, crie uma classe Sistema que realize as interações com usuário e, dependendo do comando, invoque métodos das outras classes.

Arquivos

Não existem arquivos

Enviar Resposta

Nenhum arquivo selecionado.

Entradas e Saídas

Entrada 1

Saída 1

| | |
|----------------------------|--------|
| REGISTRAR_PRODUTO arroz | 170.00 |
| 10.0 | 170.00 |
| ADQUIRIR_PRODUTO arroz 20 | arroz |
| REGISTRAR_CLIENTE joao 111 | joao |
| INICIAR_COMPRA 111 | arroz |
| ADD_ITEM arroz 17 | |
| FINALIZAR_COMPRA | |
| REVENUE | |
| MAIS_VENDIDO | |
| MELHOR_CLIENTE | |
| ----- | |

Entrada 2

| | |
|------------------------------|--------|
| REGISTRAR_PRODUTO arroz | 44.00 |
| 5.0 | 10.00 |
| REGISTRAR_PRODUTO feijao | 25.00 |
| 7.0 | isabel |
| REGISTRAR_CLIENTE maria 765 | feijao |
| ADQUIRIR_PRODUTO feijao 10 | arroz |
| REGISTRAR_PRODUTO agua 3.0 | agua |
| REGISTRAR_CLIENTE isabel 123 | 79.00 |
| ADQUIRIR_PRODUTO agua 7 | |
| ADQUIRIR_PRODUTO feijao 5 | |
| ----- | |

Saída 2