Descrição

A Black Friday costuma ser uma das melhores épocas do ano para quem deseja comprar smartphones, devido à grande variedade de descontos oferecidos pelas lojas.

Neste problema, você terá que simular a venda de smartphones de uma loja virtual.

Para fazer essa simulação de vendas, você terá que implementar duas classes: Celular e Venda. OBS: O arquivo venda.hpp não deve ser modificado.

Além disso, as seguintes situações deverão ser consideradas:

Situação 1: aplicação de desconto em todos os celulares de um determinado fabricante. Por exemplo: desconto de 10% em todos smartphones da Motorola.

Situação 2: remoção de um determinado modelo da lista de smartphones disponíveis à venda.

Situação 3: reposição de estoque de um modelo que já está à venda.

Situação 4: a quantidade de celulares em uma única venda, não pode ser maior que a quantidade atual em estoque do modelo.

A classe Celular deverá ter um construtor e armazenar as informações a seguir:

| **Celular** |
| --- |
| ---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------- |
| + cod: int |
| + peso (em kg): double |
| + fabricante: string |
| + modelo: string |
| + armazenamento: int |
| + memoria: int |
| + cor: string |
| + quantidade: int |
| + valor\_unitario: float |
| Celular(string modelo, string fabricante, int armazenamento, int memoria, double peso, string cor, int qtd, float valor) |

OBS: O primeiro celular inserido deverá receber o código 1, o segundo celular o código 2 e assim sucessivamente. Como esse código não é passado como parâmetro, seu valor deve ser atribuído na implementação do construtor da classe.

A classe Venda deverá ter uma lista de celulares como membro e os seguintes métodos:

| **Venda** |
| --- |
| ---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------- |
| - celulares: std::list<Celular> |
| + adicionaCelular(const Celular& celular): void |
| + recarregaEstoque(int cod, int qtd): void |
| + efetuaVenda(int cod, int qtd): void |
| + aplicaDesconto(string fabricante, float desconto): void |
| + removeModelo(int cod): void |
| + imprimeEstoque(): void |
| + ordena(): void |

Breve descrição:

* adicionaCelular(const Celular& celular); //adiciona um novo celular à lista de celulares disponíveis à venda.
* recarregaEstoque(int cod, int qtd); //repõe o estoque de um determinado modelo de celular.
* efetuaVenda(int cod, int qtd); //efetua a venda de um determinado modelo de celular.
* aplicaDesconto(string fabricante, float desconto); // aplica um desconto (em porcentagem) em todos os celulares de um mesmo fabricante.
* removeModelo(int cod); //remove um determinado modelo da lista de celulares disponíveis à venda.
* imprimeEstoque(); // imprime os modelos dos celulares em ordem alfabética, somente daqueles que possuem pelo menos um item no estoque.
* ordena(); // ordena a lista de celulares (utilizando a composição fabricante + modelo)

Help

<https://www.cplusplus.com/reference/list/list/>

<https://www.cplusplus.com/reference/list/list/sort/>

<https://www.cplusplus.com/reference/locale/toupper/>

<https://www.cplusplus.com/reference/string/string/>

<http://www.cplusplus.com/forum/beginner/37166/>

<http://www.cplusplus.com/forum/beginner/222475/>

<https://www.cplusplus.com/doc/tutorial/classes/>

Formato de entrada

Cada método da classe Venda é associado a um comando. Esses comandos são executados de forma aleatória, com exceção dos métodos ordena() e imprimeEstoque() que são chamados uma única vez após o processamento de todos os comandos. Os parâmetros de entrada de cada comando, seguem a mesma ordem de declação dos seus respectivos métodos. Por exemplo, para o comando sell a ordem os parâmetros são: codigo e quantidade. OBS: Os atributos de entrada do comando insert seguem a seguinte ordem: modelo, fabricante, cor, armazenamento, memoria, peso, qtd e valor.

Associações dos comandos aos métodos da classe Venda:

| **Comando** | **Método** |
| --- | --- |
| insert | adicionaCelular(const Celular& celular); |
| sell | efetuaVenda(int cod, int qtd); |
| add | recarregaEstoque(int cod, int qtd); |
| del | removeModelo(int cod); |
| discount | aplicaDesconto(string fabricante, float desconto); |

Formato de saída

Os métodos ordena() e imprimeEstoque() da classe Venda são executados e todos os modelos de celulares, com pelo menos uma unidade em estoque, deverão ser exibidos em ordem alfabética. Notem que é utilizado a composição dos atributos "fabricante + modelo" como critério de ordenação.

Exemplo de entrada

insert

Galaxy M22, Samsung, 128, 4, 0.186, preto, 10, 1349

insert

Moto G50, Motorola, 128, 4, 0.206, azul, 10, 1799

insert

iPhone 12 Pro, Apple, 512, 6, 0.206, branco, 10, 6974

sell

2 5

sell

1 6

sell

3 4

Exemplo de Saída

Apple iPhone 12 Pro, 512GB, 6GB, 2060 gramas, Branco, 6 restantes, R$ 6974.00

Motorola Moto G50, 128GB, 4GB, 2060 gramas, Azul, 5 restantes, R$ 1799.00

Samsung Galaxy M22, 128GB, 4GB RAM, 1860 gramas, Preto, 4 restantes, R$ 1349.00

Depurando localmente

Caso você queira testar seu VPL localmente, você pode baixar os arquivos necessários [aqui](https://drive.google.com/file/d/1mV8CLIJ4CbFYqY2iJINJBXGME-kEmX2t/view), e usá-los com diferentes arquivos de entrada, assim:

g++ \*.cpp -o main

./main < input.txt