Universidade de Coimbra

Programação Orientada a Objetos

Projeto - Gestor de compras online

Francisco da Silva Rosendo – 2020217697

Vasco Seabra Mota Henriques de Gouveia – 2020218407

Índice

Introdução:

O objetivo deste trabalho é o desenvolvimento de uma aplicação para a gestão das compras online de um supermercado. Esta deve permitir que os utilizadores registados façam login através do seu email, visualizem a lista de produtos disponíveis, adicionem produtos ao carrinho de compras, listem os descontos disponíveis, finalizem a compra e analisem o seu histórico de compras.

Estrutura:

Interface de utilizador:

É pedido ao utilizador um email para o efetuar o login. Caso o email não esteja associado a nenhum cliente, este é pedido de novo.

Uma vez feito o login. O utilizador escolhe a opção que deseja realizar devolvendo um número de 1 a 8 na consola. As operações restantes são escolhidas de modo análogo.

```
Options:
1 - Listar produtos
2 - Adicionar produto
3 - Listar descontos
4 - Listar carrinho de compras
5 - Finalizar compra
6 - Historico de compras
7 - Logout
8 - Sair

Option:
```

Output do menu de operações

O programa só termina se o utilizador der ordem específica para tal, escolhendo a opção 8.

Classes:

Product:

Superclasse onde são definidas as variáveis nome, identificador, preço por unidade e stock disponível. Depois, dependendo do produto em questão existem várias subclasses:

Cleaning:

Subclasse para produtos de limpeza, onde é adicionada a variável toxicidade.

Food:

Subclasse para produtos alimentares, adicionando as variáveis calorias (por cada 100g) e gordura (em percentagem).

Furniture:

Subclasse para itens de mobiliário, onde as variáveis são o peso e uma classe dimensão. Esta última armazenando a altura, largura e profundidade.

Nesta subclasse dá-se *override* ao método *is_mobilia(),* devolvendo um booleano com valor verdade. Isto permite diferenciar os produtos que são itens de mobiliário dos restantes, visto que é necessário contabilizar o peso para o cálculo da taxa de entrega.

Cliente:

Classe onde são armazenados os dados de cada cliente, sendo este o nome, a morada, o email, o telefone, uma classe com a data de nascimento, um booleano que indica se é ou não cliente frequente e, por fim, uma lista com as encomendas que fez previamente.

Clientes:

Classe onde é inicializada uma lista com todos os clientes que contém os métodos *login()* e *getByEmail().*

getbyEmail()*:* Método que itera por todos os índices da lista de clientes comparando o email de cada instância com o email passado por argumento. Se houver correspondência, é devolvida a instância com os dados do cliente, caso contrário devolve null.

login(): Lê o input do utilizador com o email, e, recorrendo ao método supramencionado imprime uma mensagem de boas-vindas, caso contrário imprime uma mensagem de erro e volta a pedir um email válido.

Discount:

Esta classe armazena o um booleano com o tipo de desconto (verdadeiro para “leve 4 pague 3”, falso para o desconto percentual) e o identificador do produto.

Discounts:

Inicializa uma lista com os descontos.

FileManager:

A classe *FIleManager* tem como objetivo a leitura e escrita de ficheiros. Esta vários métodos para o efeito.

read_clients(): Realiza a leitura do ficheiro .txt com os dados dos clientes, armazenando-os numa lista, que é posteriormente escrita num ficheiro do tipo .obj, através do método

writethisClass(). Antes de os novas instâncias serem adicionadas à lista, os emails da instância a adicionar e os das já presentes são comparados, de modo a prevenir repetições.

writethisClass(): Cria e escreve num ficheiro .obj. Recebe como argumento uma String com o nome a dar ao ficheiro e o objeto a escrever.

Os métodos *read_products()* e *read_discounts()* funcionam de modo análogo ao *read_clients()*, mas são aplicados aos produtos e aos descontos, respetivamente.

Por fim, os métodos *load_clients(), load_products()* e *load_discounts()* têm como objetivo fazer a leitura dos ficheiros .obj e devolver as listas neles contidas.

Inventário:

A classe Inventário encarrega-se da gestão dos produtos e respetivos descontos. O seu método *getProductfromId()* recebe como argumento um inteiro com o id do produto e devolve todos os dados desse produto.

Order:

A classe order é utilizada para fazer a gestão das encomendas. Tem como variáveis um dicionário, que armazena o produto e a quantidade desejada, o inventário disponível, o cliente que realiza a encomenda e o valor total da compra.

add_product(): Recebe com argumento o identificador e a quantidade do produto desejado. Se este produto já se encontrar no carrinho de compras, a quantidade é incrementada. Se o utilizador especificar uma quantidade negativa, essa vai ser removida da encomenda. Caso não haja quantidade suficiente face ao pretendido pelo utilizador, é adicionado à encomenda todo o stock restante.

process_order(): Este método analisa a todos os artigos pertencentes à encomenda e calcula o preço final, aplicando o custo de envio e as respetivas promoções.

App:

A classe App é o main do nosso programa. Numa primeira fase, começa por ler os três ficheiros de texto fornecidos que contêm as informações dos clientes, produtos e descontos, respetivamente e organiza a sua informação em ficheiros objeto. De seguida, os ficheiros objeto são lidos, e, com a sua informação é feita a criação do inventário.

Caso os ficheiros objeto já existam, não serão criados novos. Isto permite que, de uma utilização para a outra, informações como as encomendas dos clientes estejam atualizadas.

Posto isto, recorrendo à utilização de um scanner, é feito o login com o email fornecido pelo utilizador e é inicializado um booleano com o valor falso. Enquanto este valor permanecer falso é percorrido um ciclo *while* que passa por imprimir as várias oito opções disponíveis e consoante a escolha do utilizador, chamar as respetivas funções. Caso o utilizador

Considerações Finais:

Na nossa ótica, através da aplicação dos métodos adquiridos na disciplina, os objetivos do trabalho foram alcançados, realizando a aplicação todas as operações desejadas e cumprindo todos os requisitos definidos.