**Projeto POO**

*Gestor de compras online*

Entendendo o projeto:

Como é de esperar, a primeira parte do nosso projeto foi a compreensão do enunciado, começamos por ler individualmente e, após isso, prosseguimos para a discussão de ideias e formas de realizar o trabalho. Não houve muitas situações nas quais discordámos uma vez que o projeto era bastante intuitivo, porém ambos pensamos em maneiras diferentes no que toca aos ficheiros. Um de nós queria começar por fazê-los logo e outro queria fazer com que o programa primeiro funcionasse e só posteriormente adaptá-lo aos ficheiros. A conclusão final foi que seria melhor e mais fácil fazer a opção de adaptar o código no final.

Planeamento do código:

Assim como ensinado nas aulas, a primeira fase foi ver a estrutura que iriamos ter (diagrama de classes), naturalmente usamos o UML (uma pequena descrição de cada classe) para a realização da mesma. Esta parte tornou-se relativamente fácil, ambos concordámos nas classes que iriam ser necessárias. Após isso vinha a parte difícil, o código.

Código:

Nós não nos conhecíamos, anteriormente ao trabalho, logo decidimos começar por uma parte simples para nos habituarmos a programar juntos e ao ritmo de cada um. Começámos pela classe “Client”, uma classe banal. Prosseguimos para a classe “Product” e para as suas classes filho, “Food”, “Clean” e “Furniture”. Embora mais trabalhosas estas classes foram relativamente fáceis de se fazer.

Passámos então para a classe “Promotion”, esta classe determina as datas entre as quais um produto está em promoção, logo tivemos de criar a classe “Date”. A classe “Promoção” tem duas classes filhos. A classe “Pay3Take4” faz exatamente o que é dito. Por outro lado, a classe “PayLess” causou-nos um bocadinho de confusão, havia duas interpretações que tivemos, a primeira interpretação possível era que na compra de 3 unidades, por exemplo, todas elas teriam um desconto de 10% (Se cada unidade custasse 1€ o cliente iria pagar 2,70€ pelos 3) o outro ponto de vista, e o que optamos por fazer, era que usando o mesmo exemplo, a primeira unidade teria um valor de 100%, a segunda 95% e a terceira 90%, ficando assim um total de 2,85€.

Na função “main” focámo-nos em criar a interface, inicialmente (como ainda não tínhamos feito a parte dos ficheiros usámos o main como criador de clientes, produtos e promoções. O main tem diversos “cases” em que cada um tem uma função, sendo que os mais trabalhosos foram o case 2, que realiza uma compra, e o case 4 que a finaliza. Os restantes cases foram relativamente intuitivos.

A classe “Market” tem como sentido as operações regulares de uma loja. Diferencia os clientes regulares dos frequentes. Verifica e adiciona unidades de produtos. Para além disso ainda aplica as promoções.

A classe “Purchase” tem o objetivo de gerir as compras feitas, nesta classe é onde se vai buscar o preço dos produtos com a função “getPrice” e onde se forma uma lista das compras realizadas.

Passando então para os ficheiros, foi possivelmente a parte mais difícil, até nos questionámos se deveríamos ter começado a implementá-los desde o início. Inicialmente tivemos duas visões diferentes para como esta parte funcionaria, a primeira visão era que deveria haver um único ficheiro de texto, este ficheiro receberia tanto as informações dos clientes, produtos e promoções. Para este método seria apenas precisa uma classe. A outra visão era fazer 3 ficheiros de texto separados, um para os clientes, outro para os produtos e outro para as promoções. Decidimos ir com a primeira opção, era uma solução mais elegante e menos trabalhosa. Para esse fim utilizamos a classe “Files”

Com este projeto, para além de ganharmos experiência a programar, ambos percebemos a importância de uma boa comunicação, não só porque é essencial estarmos de acordo com o que é melhor fazer, mas também porque uns podem ter ideias melhores e mais simples que os outros.

Trabalho realizador por:

Carlos Miguel Matos Soares – 2020230124

Pedro Gonçalves Vaz - 2020225812