IPS Instituto Politécnico de Setúbal
Escola Superior de Tecnologia de Setúbal

PA 2021/22

Programação Avançada

Enunciado de Laboratório

8

Padrões de Software – Factories e Strategy (avaliado)

Objetivos:

- Implementação e utilização dos padrões de software Factory e Strategy
- Criação de testes unitários

Introdução:

Os padrões factory enquadram-se nos padrões de desenho criacionais, fornecendo vários mecanismos de criação de objetos, por forma a aumentar a flexibilidade e reutilização de código. O padrão Strategy enquadra-se nos padrões de desenho comportamentais e, tal como o nome sugere, permite que um objecto troque de estratégia de comportamento em tempo de execução.

Neste laboratório pretende-se criar uma pequena aplicação para gerir uma lista de produtos, adquiridos em lojas diferentes.

Comece por clonar o projeto existente no GitHub no seguinte endereço:
https://github.com/estsetubal-pa-geral/Lab8Template.git

Nível I

1. Crie a classe abstrata **LojaInformatica** que implementa a interface **LojaInformaticaFactory**, com o atributo IVA, do tipo *double*:

2. Crie a classe **LojaInformaticaContinente** que extende de **LojaInformatica** e que representará uma loja de informática localizada em Portugal continental. Esta loja tem o IVA a 23%. Adicionalmente, deverá fornecer os seguintes produtos:

Referência	Produto	Preço (sem IVA)
100	Processador Intel	250
101	Processador AMD	180
102	Placa gráfica nVidia	500
103	Placa gráfica ATI	450

3. Crie o teste unitário que valida que a exceção **ProdutoInexistente** é lançada, quando uma referência não existe.

4. Adicione a lógica necessária no método *main()* que permita funcionar com esta loja. Verifique se os valores estão corretos, com e sem IVA.

Nível II

1. Crie a classe **LojaInformaticaAcores** que representará uma loja localizada no arquipélago dos Açores. Esta loja utiliza o IVA a 18%. Para além de conter todos os produtos que a **LojaInformaticaContinente** possui (sugere-se uma extensão à classe, e não à duplicação de código), contém adicionalmente os seguintes produtos:

Referência	Produto	Preço (sem IVA)
201	Monitor 24 polegadas	200
202	Webcam	30
203	Auscultadores	50

2. Crie um teste unitário que valide que na **LojaInformaticaAcores** é possível obter os produtos da **LojaInformaticaContinente**

Nível III

1. Pretende-se obter o valor médio de uma determinada lista de produtos, com e sem IVA. Comece por instanciar no método *main()* duas listas de produtos, do tipo *ArrayList*, e adicione a cada uma 4 artigos à sua escolha. Na primeira lista deve utilizar produtos da **LojaInformaticaContinente**, e na segunda produtos da **LojaInformaticaAcores**.
2. Crie uma interface **Strategy**, com o método *double compute(List<Produto> produtosList);*
    - Crie duas classes **PrecoMedioSemIvaStrategy** e **PrecoMedioComIvaStrategy**, que implementem esta interface, e implemente o respetivo método *compute*, de acordo com a estratégia a implementar.
3. Atualize o código do *main()* e execute as estratégias para cada uma das listas criadas, verificando se os resultados são os expectáveis.

Nível IV

Pretende-se agora implementar o uso da estratégia na classe abstrata **LojaInformatica**.

1. Adicione um atributo do tipo *Strategy* à classe abstrata **LojaInformatica**, e respetivo *setter*.
2. Crie agora o método *determinarPreco (List<Produto> produtosList)*, que devolve um *double* e executa a *strategy* utilizada na instância em questão.
3. Atualize o código do *main()* e verifique o *output* da invocação das duas estratégias, nas diferentes *factories*.

Nível V

1. Pretende-se encapsular a prática de promoções em qualquer loja que exista. Para tal, crie a classe abstrata **Promocoes**, que implementa a interface **LojaInformaticaFactory** e que contém o método abstrato *void aplicarDesconto(Produto p)*. Esta classe possui um atributo do tipo *LojaInformatica* (e respetivo *getter*), aos produtos da qual serão aplicados os descontos.

2. Crie a classe **BlackFriday**, que será uma concretização de **Promocoes**.

    - O respetivo construtor deverá inicializar o atributo *loja*, da qual serão obtidos os produtos;
    - O método *obterProduto(String referencia)* deverá obter o produto da loja especificada e aplicar o desconto;
    - Na BlackFriday existe um desconto de 40%.

3. Adicione a lógica necessária no método *main()* para simular este novo cenário.