PRÁTICA LABORATORIAL 12

Objetivos:

- Herança
- Overriding
- Casting de Objetos

EXERCÍCIOS

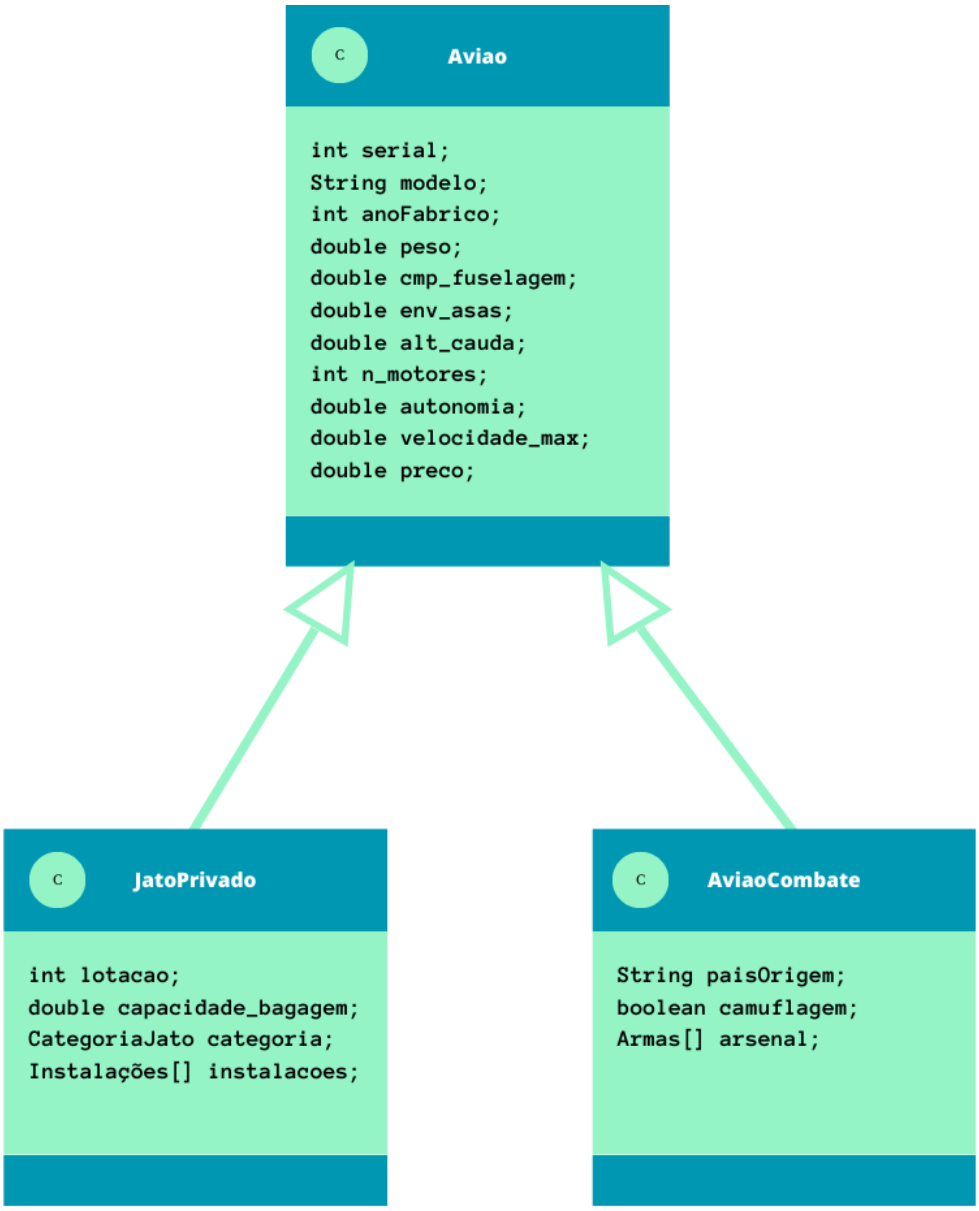
Parte 1

1. Desenvolva uma pequena aplicação que permita armazenar informação relativa a um conjunto de aviões que uma empresa especializada dispõe para venda. A empresa comercializa essencialmente aviões de dois tipos: jatos particulares e aviões de combate.
    - Os aviões possuem: número de série, modelo, ano de fabrico, peso (kg.), comprimento da fuselagem (m.), envergadura das asas (m.), altura da cauda (m.), número de motores, autonomia (km.), velocidade máxima (km/h) e preço.

    Para além dos dados já mencionados, cada tipo de avião possui características próprias:
    - Jato Particular: Lotação, Capacidade de Bagagem (cm3), Categoria [Light Jet, Midsize Jet, Heavy Jet], Instalações [ ] (por exemplo: WC, Cinema, Suite, Chuveiro, Tomadas, Cozinha, Escritório).
    - Avião de Combate: País de Origem, Camuflagem (boolean), Armas[ ] (por exemplo: Metralhadoras, Misseis, Foguetes, Torpedos, Bombas) num conjunto máximo de 3 Armas.

(Continua na próxima página)

No projeto FichaPratica_11, com um package AirplaneStore, implemente o código necessário para representar o seguinte diagrama de classes:



- Deve garantir o encapsulamento de todas as Classes criadas.
- Para as coleções Instalações e Armas, deve criar na devida classe (JatoPrivado ou AviaoCombate) os métodos de acesso controlado às coleções. Nomeadamente: inserção, remoção e listagem de elementos na consola.
- Crie uma nova Classe: Catalogo, que terá como único atributo um ArrayList de Aviões.
- Crie os métodos necessários para adquirir e vender aviões do catálogo.
- Crie o método calcularTotal que calcule todo o valor (€) do catálogo.
- Crie o método listarCatalogo que imprima todos os aviões disponíveis para venda.

  Consultar casting de objetos. Função instanceOf( )

2. Transforme a Classe Carro numa Classe denominada Veículo. De seguida, crie a classe Carro, a classe Mota e a classe Camião devendo herdar os atributos e comportamentos de Veículo.
    - Classe Veículo: Deve ter atributos de Marca, Modelo, Ano de Fabrico, Potência, Cilindrada, TipoCombustivel e litros100Km. Os métodos deverão ser os previamente implementados, ou seja, ligar( ), corrida( ) e calcularConsumo( ).
    - Classe Carro: Para além de se assumir um veículo, deve ter o atributo quantidadePassageiros. Elabore o método calcularCusto que com base no tipoCombustivel deve apresentar em € o custo da viagem (pode fazer uso de outros métodos).
      (GASOLINA = 2.10€/L, DIESEL = 1.95€/L, GPL = 1.15€/L, ELETRICO = 0.12€/L).
    - Classe Mota: Para além de se assumir um veículo, deve também ter o método imagem que imprime o conteúdo do ficheiro mota.txt na consola.
    - Classe Camião: Para além de se assumir um veículo, deve ter o atributo capacidadeCarga (em Kg). Elabore também o método viagem que recebe como parâmetro a distância e a carga (em Kg), e avalie se o camião tem capacidade para essa carga:
        - Caso ultrapasse a capacidade deve recusar a viagem.
        - Caso a carga esteja dentro da capacidade, deve calcular o consumo e custo da viagem (considere que todos os camiões usam DIESEL), sabendo que cada 100Kg de carga aumentam 0,1L/100Km ao consumo.

Teste as classes instanciando um veículo de cada tipo.
Faça uma corrida entre dois carros e imprima os dados do vencedor.
Faça uma corrida entre uma mota e um camião e imprima os dados do vencedor.
Calcule o custo de uma viagem do carro (por exemplo para 150Km.).
Efetue uma viagem válida e outra inválida (excesso peso) para o camião.

3. **(Exercício Pizzaria – Parte 1)** Desenvolva uma aplicação que permita gerir Pizzas da Pequena Pizzaria®. Para isso considere os seguintes requisitos funcionais:
    - Cada pizza é caracterizada pelo seu código, nome, descrição, preço, tamanho [Pequena, Média, Grande] e coleção de ingredientes que representam a composição da pizza.
    - Cada pizza disponibilizada pelo restaurante é composta por um conjunto de ingredientes associados (no máximo cada pizza terá 5 ingredientes) e respetivas quantidades.
    - Cada ingrediente é identificado pelo seu código, nome, unidade de medida [Gramas, Litros, Unidades] e número de calorias por unidade de medida (por exemplo: 5Kcal por grama ou 15Kcal por Litro ou 35 Kcal por unidade).

    Para apoiar o desenvolvimento desta aplicação deve mo projeto FichaPratica_11, com um package Pizzaria, criar as classes necessárias para responder aos requisitos do problema. Nomeadamente:
    - Garantir o encapsulamento de todas as classes criadas.
    - Num novo package Pizzaria.Enums, criar a enumeração para representar a unidade de medida dos ingredientes [Gramas, Litros, Unidades].
    - No package Pizzaria.Enums criar a enumeração para representar o tamanho das pizzas [Pequena, Média, Grande].
    - Na Classe Pizza deve adicionar métodos que permitam:
        - Adicionar novos ingredientes a uma Pizza até um máximo de 5.
        - Editar a quantidade de um ingrediente que pertença à coleção de uma pizza.
        - Remover um ingrediente (identificando o ingrediente pelo seu id).
        - Determinar o número de calorias de uma Pizza.
        - Apresentar uma descrição detalhada da Pizza, assim como os seus ingredientes, como por exemplo:

```
> ***** Pizza Portugal *****
> Código: P1991
> Descrição: Pizza tradicional com ingredientes nacionais que promete levar quem a come
> numa viagem pela cultura de Portugal.
> Preço: 19,90€
> Tamanho: Grande
> Ingrediente 1: [ I9922 | Massa Alta | Gramas | 1.5 Kcal ] : 200 g.
> Ingrediente 2: [ I3476 | Queijo de Cabra | Gramas | 2.8 Kcal ] : 75 g.
> Ingrediente 3: [ I4445 | Rodela Chouriça de Mirandela | Unidades | 35 Kcal ] : 10 uni.
> Ingrediente 4: [ I0015 | Molho Tomate | Litros | 325 Kcal ] : 0.09 L.
> Ingrediente 5: [ I0900 | Fatia Pimento Verde | Unidades | 4 Kcal ] : 6 uni.
```

4. **(Exercício Pizzaria – Parte 2)** A aplicação que desenvolveu para a Pequena Pizzaria® foi um sucesso e, como tal, foi contactado novamente para desenvolver novas funcionalidades. Após reunião com o CEO, chegaram a novos requisitos funcionais:
    - Recorde-se que cada Pizza era composta pelo código, nome, unidade de medida [Gramas, Litros, Unidades] e número de calorias associadas. No entanto é necessário que cada Pizza possua (pelo menos) dois ingredientes:
        - Base: que descreve o tipo de massa utilizado (Massa Alta ou Massa Fina).
        - Cobertura da Pizza: engloba alguns tipos de ingredientes e os seus tipos:
            - Queijo: Mozzarela, Serra, Cabra, Ovelha, Brie, Cheddar, etc…
            - Carne: Porco, Vaca, Chouriço, Frango, etc…
            - Vegetal: Tomate, Cebola, Pimento, Cogumelos, Milho, Ananás, etc…
            - Frutos do Mar: Camarão, Lagosta, Atum, etc…
    - Como os clientes da Pequena Pizzaria® são muito exigentes em relação aos produtos utilizados para confecionar uma Pizza, qualquer tipo de ingrediente que faça parte da cobertura da Pizza deve possuir um atributo que permite identificar a sua origem [Nacional, Importado].

Implemente as alterações necessárias de modo a refletir, em conjunto com o exercício anterior, todos os requisitos apresentados. Tenha por base o seguinte diagrama para auxilia na resolução dos exercícios:

Com a estrutura de classes construída, deverá atender os seguintes pontos:

- Realize as alterações necessária para que a unidade de medida, por defeito, dos ingredientes do Tipo Base seja obrigatoriamente gramas.
- Na classe Pizza, adicione/altere os métodos de forma que:
    - Seja apenas possível adicionar ingredientes do tipo Topping quando a Pizza já possuir um ingrediente do tipo Base.
    - Não deverá ser possível ter mais que um ingrediente do tipo Base.
    - No máximo, a pizza deverá possuir 4 ingrediente do tipo Topping.
    - Seja possível definir o tipo da Pizza com base nos ingredientes que esta possui, considerando a seguinte classificação:
        - Pizza de Carne: Contém apenas Toppings do tipo Carne.
        - Pizza do Mar: Contém apenas Toppings do tipo Fruto do Mar.
        - Pizza Vegetariana: Contém apenas Toppings do tipo Vegetal.
        - Pizza Completa: Contém, pelo menos, um Topping de cada tipo.
        - Pizza Mista: Não cumpre nenhum dos anteriores requisitos de classificação.

Teste a aplicação desenvolvida criando diversas Pizzas e testando todos os métodos implementados

Elabore o JavaDoc para o projeto.

Bom trabalho! ☺