

Prática 8 Classes e Interfaces

Exercício 8.1

Considere as seguintes entidades e características representativas de veículos motorizados:

- Motociclo, caracterizado por matrícula, marca, modelo, cilindrada, tipo (desportivo ou estrada).
- Automóvel ligeiro, caracterizado por matrícula, marca, modelo, cilindrada, número do quadro, capacidade da bagageira.
- Táxi, que tem as características de um automóvel ligeiro e, adicionalmente, o número da licença.
- Pesado de mercadorias, caracterizado por matrícula, marca, modelo, cilindrada, número do quadro, peso, carga máxima.
- Pesado de passageiros, caracterizado por matrícula, marca, modelo, cilindrada, número do quadro, peso, número máximo de passageiros.
- Empresa de aluguer de viaturas, caracterizada um nome, código postal, email e por um conjunto de viaturas, que pode incluir qualquer um dos elementos anteriores.

Todas as viaturas devem implementar a seguinte interface:

```
public interface KmPercorridosInterface {  
    void trajeto(int quilometros);  
    int ultimoTrajeto();  
    int distanciaTotal();  
}
```

- a) Analise o problema cuidadosamente e modele as interfaces e classes necessárias, as suas associações (herança, composição) bem como todos os atributos e métodos. Construa a classe principal, a função *main*, e escreva código para testar diversas possibilidades com as entidades anteriores. Por exemplo:
- Crie uma empresa com um ou mais veículos de cada tipo.
 - Teste vários métodos (*setters*, *getters*, *toString*, *equals*, métodos de *KmPercorridosInterface*, ..).
 - Indique qual a viatura com maior quilómetros percorridos.
- b) (Exercício aberto) Acrescente outras funcionalidades que lhe pareçam interessantes (por exemplo, validar a matrícula, verificar formato de email, ordenar viaturas por cilindrada, etc.).

Exercício 8.2

Considere as seguintes entidades e características representativas de alimentos:

- Carne, tem variedade (vaca, porco, peru, frango, outra), proteínas (double), calorias (double), peso (double).
- Peixe, tem tipo (congelado ou fresco), proteínas (double), calorias (double), peso (double).

- Cereal, tem nome (String), proteínas (double), calorias (double), peso (double). É um alimento vegetariano.
 - Legume, tem nome (String), proteínas (double), calorias (double), peso (double). É um alimento vegetariano.
 - Prato, tem um nome (String) e composição (conjunto de alimentos)
 - PratoVegetariano, tem um nome (String) e composição (conjunto de alimentos vegetarianos).
 - PratoDieta, tem um nome (String) e composição (conjunto de alimentos) e limite máximo de calorias (double).
 - Ementa – tem um nome (String), um local (String) e uma lista de pratos.
- c) Analise o problema cuidadosamente e modele as interfaces e classes necessárias, as suas associações (herança, composição) bem como todos os atributos e métodos. Implemente todas as classes necessária, seguindo as seguintes considerações:
- A unidade de calorias e de proteínas é relativa a 100gr.
 - Para cada prato deve ser possível obter informações sobre alimentos, peso total, calorias, proteínas,...
 - Implemente os métodos *hashCode()*, *equals()*, *toString()* em todas as classes.
 - Os pratos devem respeitar a interface *java.lang.Comparable* para permitir comparação com base em calorias.
- b) Teste a implementação com programa A08E02.java disponível no elearning. Verifique se obteve um resultado semelhante a (a ordem pode variar):

```

ERRO: Não é possível adicionar 'Carne frango, Proteínas 22.3, calorias 345.3, Peso 300.0' ao -> Dieta (0.0 Calorias) Prato 'Prato N.0 (Dieta)' composto por 0
Ingredientes
ERRO: Não é possível adicionar 'Carne frango, Proteínas 22.3, calorias 345.3, Peso 300.0' ao -> Vegetariano Prato 'Prato N.3 (Vegetariano)' composto por 0 Ingredientes
ERRO: Não é possível adicionar 'Peixe congelado, Proteínas 31.3, calorias 25.3, Peso 200.0' ao -> Vegetariano Prato 'Prato N.3 (Vegetariano)' composto por 0 Ingredientes
ERRO: Não é possível adicionar 'Peixe congelado, Proteínas 31.3, calorias 25.3, Peso 200.0' ao -> Vegetariano Prato 'Prato N.3 (Vegetariano)' composto por 1 Ingredientes
ERRO: Não é possível adicionar 'Peixe congelado, Proteínas 31.3, calorias 25.3, Peso 200.0' ao -> Vegetariano Prato 'Prato N.3 (Vegetariano)' composto por 1 Ingredientes
ERRO: Não é possível adicionar 'Carne frango, Proteínas 22.3, calorias 345.3, Peso 300.0' ao -> Vegetariano Prato 'Prato N.3 (Vegetariano)' composto por 1 Ingredientes
ERRO: Não é possível adicionar 'Carne frango, Proteínas 22.3, calorias 345.3, Peso 300.0' ao -> Vegetariano Prato 'Prato N.4 (Vegetariano)' composto por 1 Ingredientes
ERRO: Não é possível adicionar 'Carne frango, Proteínas 22.3, calorias 345.3, Peso 300.0' ao -> Vegetariano Prato 'Prato N.4 (Vegetariano)' composto por 1 Ingredientes
ERRO: Não é possível adicionar 'Peixe congelado, Proteínas 31.3, calorias 25.3, Peso 200.0' ao -> Vegetariano Prato 'Prato N.5 (Vegetariano)' composto por 1 Ingredientes
ERRO: Não é possível adicionar 'Carne frango, Proteínas 22.3, calorias 345.3, Peso 300.0' ao -> Dieta (0.0 Calorias) Prato 'Prato N.9 (Dieta)' composto por 0
Ingredientes
Dieta (84.2 Calorias) Prato 'Prato N.0 (Dieta)' composto por 2 Ingredientes
Prato 'Prato N.1' composto por 2 Ingredientes
Prato 'Prato N.7' composto por 2 Ingredientes
Vegetariano Prato 'Prato N.2 (Vegetariano)' composto por 2 Ingredientes
Prato 'Prato N.8' composto por 2 Ingredientes
Vegetariano Prato 'Prato N.3 (Vegetariano)' composto por 2 Ingredientes
Vegetariano Prato 'Prato N.5 (Vegetariano)' composto por 2 Ingredientes
Dieta (84.2 Calorias) Prato 'Prato N.6 (Dieta)' composto por 2 Ingredientes
Dieta (71.28 Calorias) Prato 'Prato N.9 (Dieta)' composto por 2 Ingredientes
Vegetariano Prato 'Prato N.4 (Vegetariano)' composto por 2 Ingredientes

```

- c) (Exercício opcional) Construa um programa Ementa que permita as seguintes operações:
- Ingrediente
 - Adicionar Carne
 - Adicionar Peixe
 - Adicionar Cereal
 - Adicionar Legume
 - Prato
 - Cria Prato
 - Apaga Prato
 - Selecciona Prato
 - Adiciona Ingrediente
 - Remove Ingrediente
 - Ementa
 - Adiciona Prato
 - Remove Prato
 - Imprime Ementa