
PROGRAMAÇÃO ORIENTADA AOS OBJECTOS (em JAVA5/6)

LEI - LCC 2º ANO/2º SEMESTRE – 2008/2009

Exame de Recurso – 14H00 - 9 de Julho de 2009
Cotação: 20 valores - Duração: 120 m

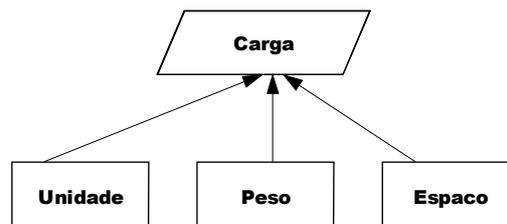
Nota 1: Responda a cada parte em folhas separadas.

Nota 2: Cada parte e questão tem indicada a respectiva cotação.

Nota 3: Leia o texto até ao fim antes de começar a resolver.

PARTE I (6 valores)

Considere a seguinte hierarquia de classes que representa os vários tipos de *serviços de transporte de carga* que uma empresa de transporte rodoviário vende aos seus clientes:



Sobre qualquer *Carga* pretende-se ter a seguinte informação: código único, total de Kms do transporte, código do cliente, destino, nome do produto e preço final (calculado cf. regras apresentadas a seguir).

A empresa fornece, de momento, 3 tipos distintos de serviços de *Carga*.

Carga à unidade, em que o que conta é o número de itens transportados, *ao peso*, em que o que conta é o número de Kg transportados, e *ao espaço*, em que o que conta são os m³ transportados e um valor de risco do transporte do produto (com valor entre 0.0 e 1.0).

Sobre qualquer tipo de serviço pretende-se ter a capacidade de calcular o seu *preço final para o cliente*.

Para cada tipo de *Carga* a empresa possui um *preço base fixo* seja por unidade, por Kg ou por m³, e possui também um *preço fixo por Km* de transporte. Porém, todos estes valores podem variar ao longo do ano.

Caso não consiga respeitar os requisitos genéricos anteriores quanto aos preços, associe a cada instância de cada tipo de *Carga* criada os respectivos preços base de produto e de Km.

Codifique as classes **Carga** e **Espaco**, tendo em atenção que não se pretendem métodos *set* e *get*, mas que são obrigatórios os construtores e os métodos **equals()**, **clone()** e **hashCode()**. (4.5)

Escreva ainda o código do método **public double orcamento(Carga c)** que deve calcular e devolver o custo actual do serviço de *Carga* dado como parâmetro (1.5).

PARTE II (7 valores)

Esta empresa de transportes possui uma frota de camiões em permanente funcionamento. Cada transporte realizado por um camião consiste de uma lista de cargas de produtos, cargas que podem ser dos vários tipos possíveis (Unidade, Peso, Espaço, etc.).

Um **transporte** é um camião e a sua carga total. Escreva o código de uma classe **Transporte** em que cada instância representa a matrícula do camião, todas as cargas que foram carregadas nesse camião para serem distribuídas e a data de transporte (1.5). Codifique os construtores desta classe e ainda os seguintes métodos de instância:

- a) Método que adiciona uma nova carga ao camião (1.0);
- b) Método que adiciona uma lista de cargas ao camião (1.5);
- c) Método que devolve os nomes de todos os destinos das cargas transportadas (1.5);
- d) Método que determina o valor total de tal transporte (1.5).

PARTE III (7 valores)

Reutilizando a classe **Transporte**, considere agora uma classe **MapaTransportes** que representa para a empresa a informação sobre todos os **transportes** realizados durante o ano por cada um dos seus camiões. Cada camião é identificado inequivocamente pela sua matrícula.

- a) Codifique estruturalmente, ou seja, em termos de atributos, a classe **MapaTransportes**, bem como os construtores da classe e o método **getMapaTransportes()** (2.0);
- b) Codifique um método de instância **public int pesoTotal(String mat)** que determine o peso total da carga do tipo **Peso** transportada pelo camião de matrícula dada (1.5);
- c) Codifique um método que devolva as matrículas de todos os camiões que transportaram carga para o destino dado como parâmetro (1.5);
- d) Codifique um método que determine a matrícula do camião que mais quilómetros percorreu durante o ano corrente (2.0).

Prof. F. Mário Martins