

# Exame de Paradigmas da Programação IV e Programação Orientada aos Objectos 2ª chamada

2006.07.08

Duração: **2h**

*Leia o teste com muita atenção antes de começar.*

RESPONDA A CADA PARTE EM FOLHAS SEPARADAS. SE UTILIZAR A NOTAÇÃO DO JAVA 1.5 POR FAVOR INDIQUE-O EM CADA PERGUNTA.

---

## PARTE I - 8 VALORES

1. Considere que se pretende desenvolver uma classe para representar matrizes quadradas de floats.
  - (a) Defina as variáveis de instância e escreva os construtores vazio e com parâmetros (a definir pelo aluno).
  - (b) Escreva o método para inserir um valor na matriz, `void insere(int x, int y, float val) throws ???Exception`
  - (c) Defina o método para somar duas matrizes: `Matriz soma(Matriz m)`;
  - (d) Defina o método `equals` assumindo que duas matrizes são iguais se forem proporcionais, i.e. se existir um número real tal que  
 $m1 = a * m2$   
sendo `m1` e `m2` instâncias de `Matriz`
2. Assuma agora que se pretende estender o sistema criando uma classe de matrizes triangulares inferiores, `MatrizTriangular`. Defina a hierarquia de classes e, se necessário, apresente a redefinição dos métodos anteriores.
3. Escreva um construtor para a classe `MatrizTriangular`, que receba como parâmetro uma `Matriz`, indicando claramente as opções tomadas na implementação.

## PARTE II - 6 VALORES

4. Relembre a classe `Percurso`, constante do enunciado do trabalho prático da disciplina. Considere que a classe `Percurso` contém como variável de instância um `ArrayList` `locs`, cujos valores são instâncias de `Localidade`.
  - (a) Escreva o método `boolean equals(Object o)` que permite comparar percursos, atendendo a que dois percursos são iguais desde que contenham as mesmas localidades.

- (b) Escreva um método que permita adicionar uma localidade a um percurso, `void add(Localidade l) throws ???Exception;`
- (c) Escreva um método `diferença` que dado um percurso como parâmetro, devolva uma `Collection` de `Localidade` que pertencem ao percurso recebido com parâmetro, mas que não pertencem ao percurso receptor da mensagem. A assinatura do método pretendido é `Collection diferença(Percurso p) throws ???Exception;`
- (d) Implemente o método `toString()` da classe `Percurso`.

### PARTE III - 6 VALORES

5. Considere agora a Agência de Viagens *Vá Lá* que organiza visitas a percursos. A informação da Agência *Vá Lá* está contida numa classe `Agência` que tem como variável de instância um `Map` cujas chaves representam nomes de percursos e os valores são instâncias de percursos. A agência precisa de obter as seguintes informações do sistema:

- (a) Para cada localidade quantos percursos a visitam? Devolve um `Map` em que as chaves são os nomes das localidades, e o valor associado é o número de ocorrências no conjunto de todos os percursos.  
`Map locEmPercurso();`
- (b) Para uma dada localidade quais os percursos (pretende-se obter uma lista de nomes de percursos) que a visitam?  
`List quePercursos(Localidade aLoc);`
- (c) Qual a lista dos nomes de percursos que contém localidades que só aparecem num percurso?  
`List perComLocalidadeUnica();`