

Universidade da Beira Interior

Programação Orientada a Objectos

Cursos: Matemática / Informática, Ensino da Informática, Engenharia Informática

Exame - 2006/02/07

Duração: 90 minutos

SEM CONSULTA

(8 valores)

Antes de responder ao teste leia TODO o enunciado e TODAS as questões.

1 - Pretende-se construir uma aplicação, muito simples, para uma pequena empresa que quer ter o registo informático dos seus empregados. Cada empregado tem um número único e sequencial (atribuído automaticamente), um nome, desempenha uma função e ganha um salário mensal. A empresa tem um número de identificação fiscal, uma designação e nela trabalha um número variável de empregados. A empresa mantém o registo dos empregados que deixam de trabalhar na empresa para eventualmente os contratar novamente mais tarde. Os dados de um empregado que entra para a empresa pela 2ª vez já não são registados novamente.

a) Diga quais as classes, apenas o nome e o que representam, que construída para a aplicação.

b) Para **cada uma das classes** anteriores:

- i) – indique quais os atributos
- ii) – defina um construtor à sua escolha.
- iii) – construa o método toString (que deve sobrepor ao método toString da classe Object).
- iv) – construa o método equals (que deve sobrepor ao método equals da classe Object).

c) Explique o que é a classe Object e o que significa sobrepor um método da classe Object.

d) Nas questões que se seguem deverá começar por **indicar qual a classe onde irá definir o método pedido**. Suponha ainda que para cada classe estão definidos os *getters* e *setters* de cada atributo.

- i) Construa um método para inserir um novo empregado na empresa. **Note que**, inserir um empregado pode significar registar o empregado pela primeira vez ou assinalar que aquele trabalhador volta a ser funcionário da empresa.
- ii) Construa um método que permita aumentar o salário de um empregado em 10%.
- iii) Construa um método que mostre todos os dados dos empregados que ganham um salário superior a 1000€.
- iv) Construa um método que calcule o montante gasto pela empresa em salários no corrente mês.

2 – Explique o que é e como funciona o mecanismo de herança.

API da classe `java.util.Vector`:

```
Vector() // construtor vector vazio, dimensão inicial zero.  
Vector(int capacidadeInicial) // construtor vector vazio, com dimensão inicial.  
void addElement(Object elemento) // adiciona o elemento especificado ao final do vector.  
void insertElementAt(Object obj, int indice) // insere o elemento especificado na posição indice.  
void removeElementAt(int indice) // remove o elemento na posição indice.  
void setElementAt(Object obj, int indice) // substitui o elemento da posição indice pelo objecto dado.  
Object elementAt(int indice) // devolve o componente presente no indice.  
void clear() // remove todos os elementos do vector.  
Object clone() // devolve uma cópia do vector.  
boolean contains(Object elemento) // verifica se o objecto especificado é um componente deste vector  
Object firstElement() // devolve o primeiro componente (indice 0) do vector.  
Object lastElement() // devolve o último componente do vector.  
int indexOf(Object elemento) // procura o índice da 1ª ocorrência de elemento (utiliza o método equals)  
int indexOf(Object elemento, int indice) // inicia a procura anterior na posição indice.  
boolean isEmpty() // verifica se o vector não tem componentes  
int size() // devolve a dimensão actual.  
...
```