

— Exame —  
Desenvolvimento de Sistemas de Informação

LESI/LMCC  
Chamada 2 - 2002/03

09/07/2003

Duração máxima: 2h00  
**Leia as questões com atenção.**  
**Responda a cada parte em folhas separadas.**

### Grupo I

1. Considere o seguinte código Java:

```
public class A implements I {
    private Vector bs;    // vector com inst. de B
    private B boss;

    public A() {
        ...
    }

    public void setBoss(B b) {
        bos = b;
    }

    public void insere(B b) {
        bs.add(b.clone());
    }

    public B get(int i) {
        B bAux;

        bAux = search(i);
        return bAux.clone();
    }

    private B search(int i) {
        for (int i=0; i<bs.size(); i++) {
            B b = (B)bs.get(i);
```

```

        if (b.valido(i))
            return b;
    }
    return new B();
}

public interface I {
    ...
}

public class B {
    private int min, max;

    public boolean valido(int i) {
        boolean res;

        if (i>min && i<max)
            res = true;
        else
            res = false;

        return res;
    }
}

public class B1 extends B {
    ...
}

```

- (a) Construa um **Diagrama de Classes** para o código apresentado (será prestada especial atenção à correcta utilização da composição e da agregação).
- (b) Construa um **Diagrama de Sequência** para o método get da classe A.
2. Considere um relógio com três botões: modo, horas e minutos. O botão “modo” altera o modo de funcionamento do relógio. O relógio tem os seguintes modos:
- 1 – alarme ligado;
  - 2 – alarme desligado;
  - 3 – alteração da hora do relógio;
  - 4 – alteração da hora do alarme.

Os dois outros botões permitem incrementar a hora e os minutos. Quando o modo é 3 este dois botões permitem alterar a hora e os minutos do relógio. Quando o modo é 4 permitem alterara hora e os minutos do alarme. Se o modo for 1 ou 2, estes botões não têm qualquer acção associada.

Construa um **Diagrama de Estados** que represente o modo de funcionamento do relógio descrito.

## Grupo II

Considere a seguinte descrição de um sistema para gestão de submissões a conferências a ser instalado na secretaria do departamento de informática:

O software deverá suportar todo o processo de gestão da informação necessária ao correcto processamento das submissões às conferências.

Sobre cada conferência deverá ser registado o seu título, data de realização (início/fim), lista dos organizadores e lista das submissões.

De momento, deverão ser considerados dois tipos de submissões: artigos completos e artigos curtos. Todas as submissões possuem obrigatoriamente um código atribuído automaticamente pelo sistema, um título, uma lista de autores, o endereço do ficheiro onde ela está armazenada e uma tabela associando a cada um dos revisores da submissão a classificação por ele atribuída (de 1 a 5 — 0 caso ainda não tenha sido atribuída uma classificação). Os artigos completos possuem, ainda, uma lista de temas e um resumo. É previsível que outros tipos de submissões venham a ser consideradas.

Tanto os organizadores das conferências, como os revisores e os autores das submissões devem estar, obrigatoriamente, registados como utilizadores do sistema. Os utilizadores são identificados pelo seu endereço de email. Deverá ainda ser guardada informação relativa ao seu nome e à sua instituição de origem. O sistema deverá manter uma tabela de todos os utilizadores registados no sistema.

Para além de permitir o registo de novos utilizadores, conferências e submissões, o sistema deverá permitir a seguinte funcionalidade:

- Determinar a lista ordenada por classificação das submissões a uma dada conferência;
- Determinar a lista dos emails dos revisores que ainda não enviaram as suas revisões para uma dada conferência.

Responda às seguintes questões:

1. Construa um **Diagrama de Use Case** que reflecta a descrição dada. Forneça uma descrição para cada um dos *use case*, descrevendo o seu comportamento e eventuais pré-condições.
2. Construa um **Diagrama de Classes** para o sistema (não necessita indicar os métodos de cada classe).
3. Considerando ainda o diagrama de classes anterior, construa um **Diagrama de Sequência** para o método `Vector emFalta(String titConf, int cod)` que determina a lista dos emails dos revisores que ainda não enviaram as classificações relativas ao artigo com código `cod` da conferência com título `titConf`.