— Exame —
Desenvolvimento de Sistemas de Informação

LESI/LMCC
Chamada 1 - 2003/04

??/06/2003

Duração máxima: 2h00
Leia o exame com atenção.

Grupo I

Considere o seguinte excerto de código Java:

```
public class T extends Observable {
    private Map t;

    public T() { this.t = new HashMap(); }

    public void addA(A a) {
        String num= a.getNumero();
        this.turma.put(num, a);
        this.setChanged();
        this.notifyObservers();
    }

    public Aluno get(String num) throws TException { ... }

    public void del(String num) throws TException { ... }

    public int quantosPassam() {
        Enumeration e = this.t.elements();
        int tot = 0, m;

        while (e.hasMoreElements()) {
            A a = (A)e.nextElement();
            m = a.getMedia();
            if (m>=10)
               tot++;
        }
        return tot;
    }
}

public abstract class A implements Cloneable {
    private String nome;
    private String numero;
    private int notaT, notaP;

    public A(String numero,String nome,int notaT,int notaP) { ... }
    }

    public abstract int getMedia();

    public boolean equals(Object o) {
        ...
```

```
    }

    public Object clone() {
        ...
    }
}

public class Areg extends A implements Cloneable {
    public int getMedia() {
       int t = this.getNotaT();
       int p = this.getNotaP();
       return (t+p+0.1*(t-p))/2;
    }
}

public class Ate extends A implements Cloneable {
    public int getMedia() {
       int t = this.getNotaT();
       return t;
    }
}
```

1. Desenhe um **Diagrama de Classes** para o código apresentado (será prestada especial atenção à correcta definição das associações entre as classes).

2. Desenhe um **Diagrama de Sequência** para o método `int quantosPassam()` da classe `T`.

Grupo II

Relembre o trabalho prático:

> A LusoPerímetro pretende montar um sistema de gestão de informações sobre eventos desportivos. Para tal pediu já a potenciais interessados que procedessem a uma primeira análise e implementação de um protótipo de um dos módulos do sistema. Tratou-se do módulo de prestação de informações via SMS. Esse módulo recolhe informação sobre eventos ocorridos durante as provas e encaminha essa informação para clientes que nela tenham manifestado interesse.
>
> A LusoPerímetro pretende agora que seja desenvolvido um novo módulo para o tratamento estatístico da informação relativa ao serviço. O módulo em causa estará disponível para os gestores da empresa e permitirá em cada momento saber o número de mensagens já enviadas (total global, total por modalidade ou total por tipo de evento). Deverá ainda permitir saber rápidamente qual a modalidade mais popular/impopular e qual o tipo de evento mais popular/impopular (quem tem mais/menos inscrições). Adicionalmente, deverá permitir saber em cada momento qual o número de mensagens a aguardar cobrança.
>
> A informação necessária para o tratamento estatístico é fornecida, quer pelo módulo de prestação de informações via SMS (eventos enviados), quer pela facturação (eventos cobrados).

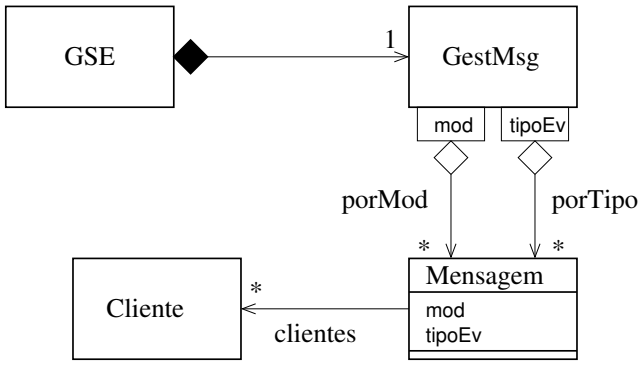
Responda às seguintes questões:

1. Responda a **apenas uma** das duas questões seguintes:

   (a) Quem alterações teria que efectuar no diagrama de Use Case que desenvolveu no trabalho prático tendo em vista acomodar este novo módulo? Responda a esta questão desenhando um diagrama com o que seria necessário acrescentar.

   (b) Escreva um **Diagrama de *Use Case*** que reflicta a descrição dada.

   Em qualquer dos casos, forneça uma descrição para cada um dos *use case* do diagrama, descrevendo de forma abreviada o seu comportamento e eventuais pré-condições.

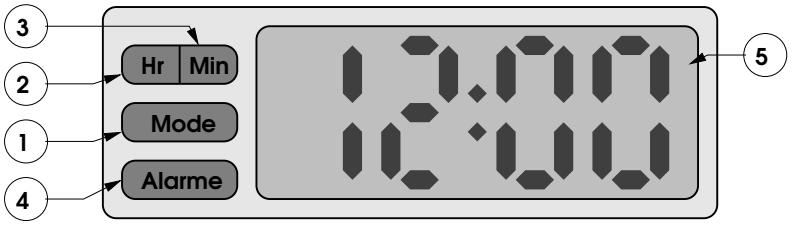


2. Considere a proposta de arquitectura de classes para o novo módulo apresentada na figura. Tendo em vista melhorar o desempenho do processamento estatístico, a proposta organiza as mensagens em duas tabelas: numa as mensagens estão indexadas por modalidade e na outra por tipo de evento.

   Com base na arquitectura proposta, desenhe um **Diagrama de Sequência** para o método `List verificaMod()` (da classe GSE) que verifica se todas as mensagens presentes nas listas da tabela porMod estão também presentes em alguma lista da tabela porTipo. O método deverá devolver uma lista com todas as mensagens que não verifiquem a condição. (Considere que as tabelas são implementadas com Map e as listas com List)

Grupo III

Considere o seguinte excerto de um manual de instruções para um relógio despertador:



① Botão "Mode" — prima este botão para percorrer repetidamente os diferentes modos de funcionamento do relógio:

Apresentação da hora actual ➔ Regulação da hora actual ➔ Regulação da hora do alarme ➔ ... (volta a Apresentação da hora actual)

② Botão "Hr (hora)" — prima este botão para incrementar a hora quando em modo de regulação de hora actual/alarme (não produz qualquer efeito em modo de apresentação da hora actual).

③ Botão "Min (minutos)" — prima este botão para incrementar os minutos quando em modo de regulação de hora actual/alarme (não produz qualquer efeito em modo de apresentação da hora actual).

④ Botão "Alarme" — prima este botão para activar/desactivar o alarme (se está activo desactiva e vice-versa). O relógio começa a tocar quando o alarme está activo, o relógio está em modo de apresentação da hora, e a hora actual é igual à hora do alarme (o alarme desliga-se automaticamente após 3 minutos).

⑤ Écran — apresenta a hora correspondente ao modo em que o relógio se encontra.

Desenhe um Diagrama de Estados que represente o funcionamento do relógio tal como descrito. Para além dos eventos que provocam as transições de estado, inclua as acções que o relógio terá que realizar.