

— Exame —  
Desenvolvimento de Sistemas de Informação

LESI/LMCC  
Chamada 2 - 2003/04

02/07/2004

Duração máxima: 2h00  
**Leia as questões com atenção.**

### Grupo I

Considere o seguinte código Java:

```
public class JCalculadora extends JFrame implements Observer{

    private Calculadora calc;
    private JButton jButton0, jButton1, jButton2, jButton3, jButton4;
    private JButton jButton5, jButton6, jButton7, jButton8, jButton9;
    private JButton jButtonClear, jButtonDiv, jButtonEquals;
    private JButton jButtonMinus, jButtonPlus, jButtonTimes;
    private JLabel screen;

    public JCalculadora() {
        initComponents();
        this.calc = new Calculadora();
        this.calc.addObserver(this);
    }

    private void initComponents() { ... }

    private void clear_press(ActionEvent evt) {
        this.calc.clear();
    }

    private void digit_press(ActionEvent evt) {
        String cmd = evt.getActionCommand();
        this.calc.processa(Integer.parseInt(cmd));
    }

    public static void main(String args[]) {
        new JCalculadora().show();
    }

    public void update(Observable o, Object arg) {
        this.screen.setText(((Integer)arg).toString());
    }
}

public class Calculadora extends Observable {

    private double val_ant, val_act;
    private boolean new_number;
    private char opr;
```

```

public Calculadora() {
    this.val_ant = this.val_act = 0;
    this.new_number = true;
    this.opr = '=';
    this.setChanged();
    this.notifyObservers(new Integer((int) this.val_act));
}

public void processa(int d) {
    if (this.new_number) {
        this.val_act = d;
        this.new_number = false;
    } else {
        this.val_act = this.val_act*10+d;
    }
    this.setChanged();
    this.notifyObservers(new Integer((int) this.val_act));
}

public void clear() { ... }
}

```

Construa um **Diagrama de Classes** para o código apresentado (utilize cardinalidades para representar quantos JButtons são necessários; será prestada especial atenção à correcta indicação de todos os métodos de cada classe).

## 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 correspondente ao software a instalar nos telemóveis dos clientes. O referido software deverá receber as mensagens enviadas através da *SMS gateway* e efectuar o seu processamento. Sempre que uma nova mensagem relativa a eventos desportivos é recebida tal facto deve ser comunicado ao utilizador do telemóvel. Nessa altura o utilizador deverá poder optar por consultar imediatamente a mensagem, ou guardar a mensagem para consulta posterior. O software deverá ainda permitir consultar mensagens anteriormente guardadas. Adicionalmente, o software deverá permitir a consulta do histórico das mensagens recebidas quer por tipo de evento, quer por modalidade.

A proposta de arquitectura de classes para o novo módulo é apresentada na figura ??:

- *aLer* é a tabela das mensagens ainda não lidas (indexada por código de mensagem);
- *lidas* é a lista de todas as mensagens já lidas (ordenada cronologicamente);
- *porMod* é uma tabela para acesso rápido (por modalidade) às mensagens da lista *lidas*;
- *porTipo* é uma tabela para acesso rápido (por tipo de evento) às mensagens da lista *lidas*.

Responda às seguintes questões:

1. Escreva um **Diagrama de Use Case** que reflecta a descrição dada.

Forneça uma descrição para cada um dos *use case*, descrevendo de forma abreviada o seu comportamento e eventuais pré-condições.

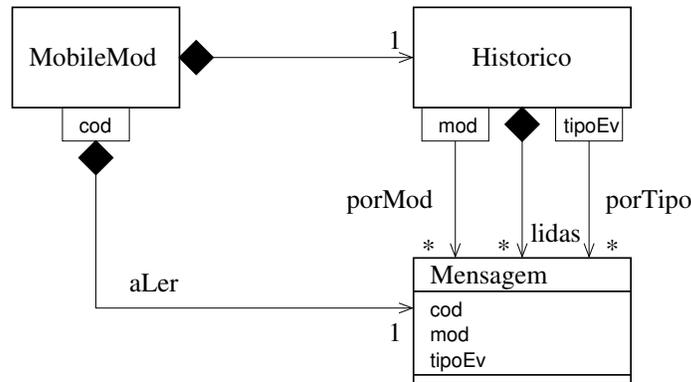


Figura 1: Arquitectura

2. Sabendo que as tabelas são implementadas com Map e as listas com List, escreva o código Java correspondente ao diagrama de classes da figura ??.
3. Construa um **Diagrama de Sequência** para o método `void ler(int cod)` (da classe `MobileMod`) que passa a mensagem com código `cod` da tabela `aLer` para o histórico (actualizando adequadamente as várias estruturas de dados lá existentes).

### Grupo III

Considere a seguinte descrição de um processo de autorização de despesas com deslocações:

Um formulário de autorização de viagens é utilizado nas empresas para aprovação de despesas de viagem dos seus funcionários. Normalmente o funcionário preenche um formulário e envia-o ao director da sua unidade para ser assinado.

Se a quantia for pequena (menos de EUR300.00), o director de serviço assina o formulário e envia-o para a contabilidade para ser processado. Ao receber o formulário a contabilidade emite um cheque a favor do funcionário e arquiva o formulário.

Se a quantia for elevada (EUR300.00 ou mais), o director de serviço assina o formulário e envia-o para o director financeiro para aprovação. O director financeiro assina-o e envia-o para a contabilidade para ser processado.

Obviamente tanto o director de serviço como o director financeiro podem rejeitar a autorização de deslocação se não considerarem as despesas aceitáveis. Neste caso o funcionario pode decidir alterar o formulário de modo a incluir mais informação sobre a despesa, ou desistir do pedido de aprovação.

Escreva um **Diagrama de Actividade** que represente o processo descrito. O diagrama deverá apresentar pistas para cada actor envolvido no processo, bem como os fluxos relativos ao formulário.