— Exame —
Desenvolvimento de Sistemas de Informação

LESI/LMCC
Chamada 1 - 2002/03

23/06/2003

Duração máxima: 2h00
Leia as questões com atenção.

Grupo I

1. Considere o seguinte código:

```
public class Main {
  private A a;
  private Vector bs;

  void split() {
    this.a = new A(bs);
  }
}

public class A {
  private Vector bAprovados;
  private Vector bRecusados;
  private int some;

  void A(Vector bAll) {

    while (b in bAll) {
      if (!b.aceite())
        bRecusados.add(b);
      else {
        bAprovados.add(b);
        int aux = b.getValor();
        soma += aux;
      }
    }
  }
}

public class B {
  private int valor, aval;

  public boolean aceite() {
    boolean res;
```

```
    if (aval>10)
      res = true;
    else
      res = false;

    return res;
  }
}
```

Construa um **Diagrama de Sequência** para o método split.

2. Baseado na seguinte descrição de parte da sintaxe de uma linguagem de programação, construa um **Diagrama de Classes** que ilustre a estrutura dos programas escritos na referida linguagem:

   > Um programa consiste numa colecção de módulos — um desses módulos será identificado como o módulo principal. Um módulo consiste numa colecção de items. Um item pode ser uma declaração de variável, uma rotina, ou um módulo aninhado. Rotinas são constituidas por duas partes: a parte declarativa e uma sequência não vazia de instruções. A parte declarativa consiste numa colecção de declarações das variáveis passadas como parâmetros e numa colecção de declarações das variáveis locais. As instruções podem ser ciclos, condicionais, ou atribuições — sendo que as atribuições contêm uma referência à declaração da variável a atribuir.

3. Considere o seguinte extracto de código Java:

```
public class Lista {
    // variáveis de instância
    private Vector lista;

    // métodos de instância
    public void ordena() {
      int i=lista.size()-1, j, min;
      Comparable c,
      Object o;

      while(i>0) {
        j=0;
        while (j<i) {
          c = (Comparable)lista.get(j);
          o = lista.get(i);
          if (c.compareTo(o)==1) {
            lista.set(i, c);
            lista.set(j, o);
          }
          j++;
        }
        i--;
      }
    }
    ...
}
```

Escreva um **Diagrama de Actividade** que descreva o algoritmo do método `ordena`.

Grupo II

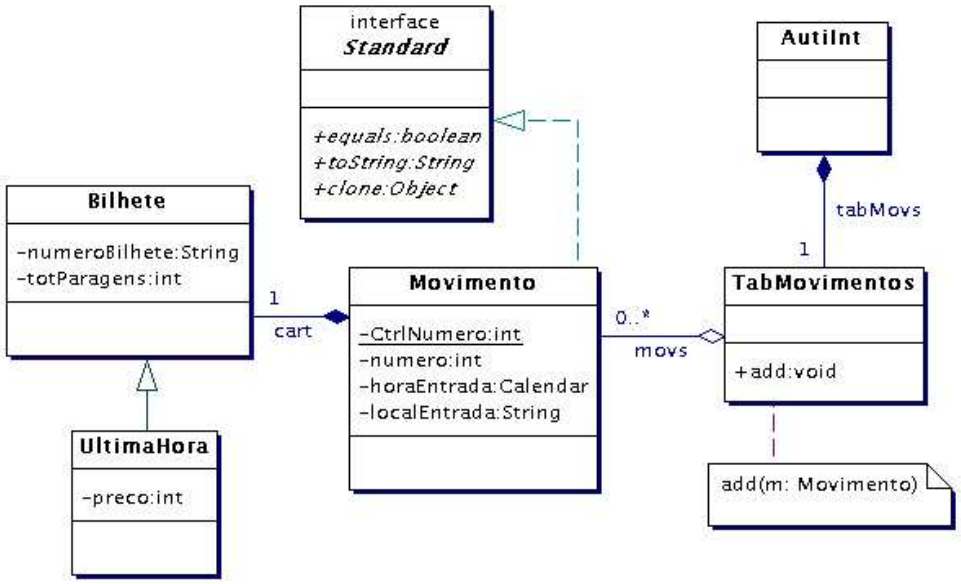
Considere a seguinte descrição do software a desenvolver para os autocarros inteligentes a serem utilizados pela TUF (relembre o trabalho prático):

> O software deverá suportar, de forma automática, todo o processo de gestão da informação a comunicar às paragens, bem como fazer o processamento de eventuais avarias.
>
> Quando entram, os passageiros são identificados pelo seu bilhete. Quando saem, a saída dos passageiros é também registada. Passageiros sem bilhete podem utilizar uma máquina presente à entrada do autocarro para obter um bilhete de última hora.
>
> Quando o condutor fecha as portas do autocarro este envia informação à paragem sobre o hora da chegada e sobre as entradas e saídas (incluindo bilhetes emitidos). A informação enviada sobre cada bilhete inclui: o número do bilhete em questão, a hora e local de entrada e o número de paragens percorridas. Para os bilhetes de última hora é também enviada informação sobre o preço pago pelo cliente (para tornar o processo de entrada no autocarro mais rápido não é devolvido troco na compra de bilhetes última hora).
>
> Em caso de avaria, o condutor limita-se a premir um botão no painel e o sistema deverá comunicar à central a avaria e a localização actual do autocarro.

Responda às seguintes questões:

1. Escreva um **Diagrama de *Use Case*** que reflicta a descrição dada. Forneça uma breve descrição para cada um dos *use case* identificados.

2. Considere o diagrama de classes apresentado na figura:



   Escreva o código Java de uma implementação possível para este diagrama.

3. Considerando ainda o diagrama de classes da figura anterior, escreva um **Diagrama de Sequência** para o método `Vector menosDe(int n)`, da classe AutoInt, que determina a lista de todos os bilhetes que fizeram viagens com um número de paragens inferior a um valor n dado.