

— Exame —
Desenvolvimento de Sistemas de Informação

LESI/LMCC
2ª Chamada - 2006/07

30/06/2007

Duração máxima: 2h00

Leia o exame com atenção e responda utilizando UML 2.0.

Cada grupo deve ser respondido em folhas diferentes.

Grupo I

1. Considere que os pontos que a seguir se apresentam documentam o processo de avaliação que decorre no final do semestre para avaliação da disciplina de DSI:
 - (a) Os alunos tem de entregar o trabalho prático (assuma que tem duas fases obrigatórias), sendo que sem entregar ambas as fases são considerados não avaliados.
 - (b) A primeira fase de entrega ocorre antes dos exames, e a segunda no período entre os exames da época normal e o exame de recurso. Ao trabalho é atribuída uma única nota.
 - (c) A época normal é constituída por duas chamadas, mas cada aluno só pode realizar um exame, devendo inscrever-se para tal nos serviços académicos. Esta inscrição é obrigatória.
 - (d) À época de recurso podem ir alunos que tenham reprovado na época normal. Podem ainda realizar este exame os alunos que tendo obtido uma classificação positiva num exame e queiram melhorar a sua nota. Neste caso a nota final da componente teórica será o máximo entre as duas notas. Em ambas as situações é necessário preceder à inscrição nos serviços académicos, sendo que a época de inscrição ocorre depois do fim da época normal.
 - (e) Para a nota final ser positiva, o que implica passar à disciplina, tal carece de aproveitamento ($nota \geq 10$) a ambas as componentes teórica e prática.
 - (f) Alunos que não obtenham aproveitamento a uma das componentes estão reprovados independentemente da média obtida.

- (g) Finalmente, para os trabalhadores estudantes existe ainda mais uma oportunidade de realizar a componente teórica na época especial, sendo a época de inscrições posterior ao exame de recurso.

Construa o **Diagrama de Actividades**, que permite representar o processo descrito.

2. Considere um sistema de gestão de parques de estacionamento, cujo domínio do problema já conhece do trabalho prático. Considere que neste caso, se pretende garantir que:
- (a) O sistema deve permitir gerir vários parques, sendo que os clientes mensais podem estacionar em qualquer dos parques.
 - (b) Para um cliente mensal, o sistema guarda a informação sobre o seu nome, morada e matrícula do seu carro.
 - (c) A qualquer momento o sistema mantém o registo sobre a ocupação de cada um dos parques, organizado por parque e simultâneamente por cliente.
 - (d) O sistema tem ainda um registo histórico com todas as ocupações que tiveram lugar no passado.
 - (e) O sistema guarda informação não personalizada sobre os clientes esporádicos.
 - (f) Estes registo são completos e identificam o parque, o cliente (no caso dos clientes mensais), as datas envolvidas e os montantes pagos.

Construa um **Diagrama de Classes** que reflecta a informação descrita.

3. Um voo antes de decolar precisa de realizar as seguintes operações: realizar testes de segurança, abastecer e embarcar passageiros e carga. Os testes de segurança e o abastecimento podem ser realizados por qualquer ordem, mas sempre com o avião vazio (sem passageiro ou carga). Se o avião falhar os testes de segurança, o voo correspondente é destruído (do sistema informático!). Os passageiros só embarcam depois de finalizar o embarque da carga. Após o embarque dos passageiros estar completo o comandante deve inscrever o voo para decolagem e aguardar para autorização da torre para se deslocar para a pista e decolar. Uma vez no ar o avião tem de aterrar, e descarregar passageiros e carga, altura em que o registo do voo é colocado num histórico.

Aleatoriamente um avião pode ser destacado para um teste adicional de segurança, podendo este teste ser realizado em qualquer altura e tendo prioridade sobre todas as outras operações. Ao finalizar este teste o voo deve retornar à operação que estava a realizar antes de iniciar o teste.

Construa um **Diagrama de Estados** para o registo de voo.

Grupo II

Considere a seguinte descrição do software a desenvolver para os autocarros inteligentes a serem utilizados por uma nova empresa de transportes urbanos, os TUM (Transportes da UM):

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.

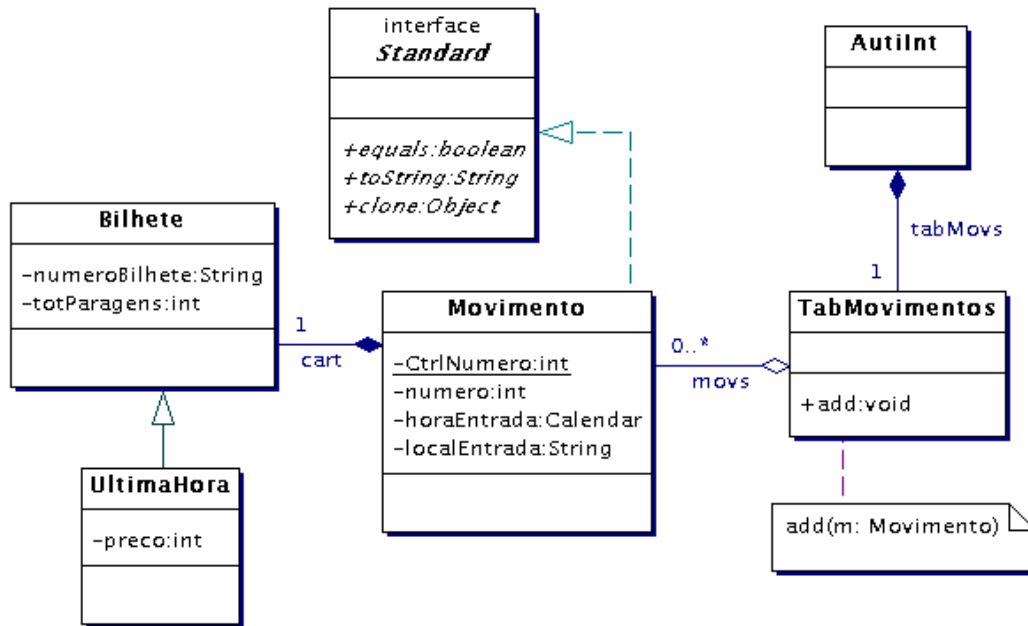
Quando o condutor fecha as portas do autocarro este envia informação à paragem sobre o hora da chegada e sobre as entradas e saídas (incluindo bilhetes emitidos). A informação enviada sobre cada bilhete inclui: o número do bilhete em questão, a hora e local de entrada e o número de paragens percorridas. 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 reflecta a descrição dada. Forneça uma descrição para cada um dos *use case* identificados, descrevendo o seu comportamento e eventuais pré-condições.

2. Considere o diagrama de classes apresentado na figura:



Escreva o código Java (declarações de classe e de métodos) 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 `ArrayList inícioEm(String local)`, da classe `Autilnt`, que determina a lista de todos os bilhetes que fizeram viagens com início num dado local.