

Desenvolvimento de Sistemas Software

LEI - 2009/10

19/01/2010

Duração máxima: 2h00

Leia com atenção e responda utilizando UML 2.0.

Responda a cada grupo em folhas separadas.

PARTE I - 12 valores

Considere que a equipa de projecto que esteve encarregue da modelação da GERECOMSABER tem agora um novo desafio: criar o sistema software de uma cadeia de farmácias *self-service*. O objectivo é dotar uma rede de farmácias de terminais onde os clientes, através de uma interface com o utilizador, podem comprar medicamentos de venda livre.

O dono da farmácia tem como associados nesta rede de farmácias, designada de *Help Yourself*, os fornecedores de medicamentos de venda livre e os bancos que permitem que a compra seja feita nos terminais da farmácia. Os fornecedores de medicamentos disponibilizam um catálogo de produtos (e respectivo preço) que a gestão da farmácia escolhe e encomenda de acordo com as suas necessidades. Sobre cada preço de medicamento a farmácia adiciona uma comissão de acordo com as suas expectativas de venda do produto. Esta comissão, bem assim como a escolha dos fornecedores, é efectuada pela equipa de Gestores de Produto. É obrigação de cada fornecedor ter mais do que um medicamento com o mesmo princípio activo, por forma a que seja possível substituir um medicamento por outro com igual eficácia clínica. Por uma questão de obrigatoriedade imposta pelo regulador do sector das farmácias é também necessário que exista um medicamento genérico por cada medicamento comercializado.

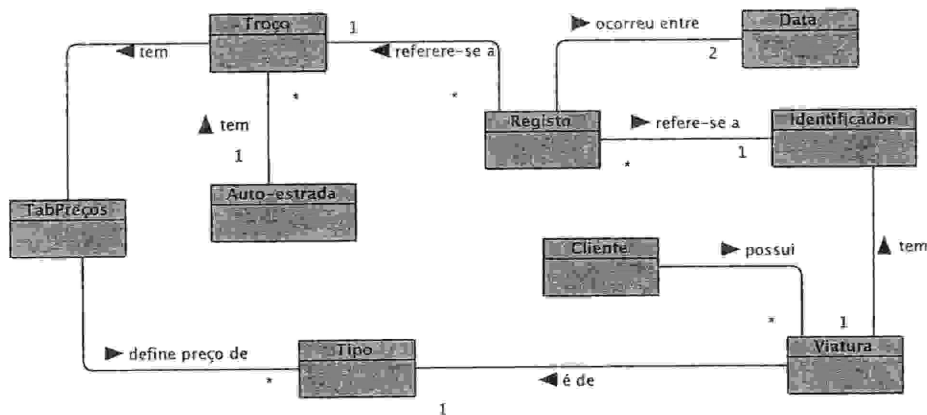
Sempre que um cliente compra um medicamento é-lhe colocada a hipótese de escolher um medicamento genérico. Em situações de rotura de stock, o sistema deve tentar sugerir um medicamento substituto. Caso não exista stock do medicamento substituto, ou o cliente não queira a substituição, o sistema regista a falha para posterior monitorização por parte da gerência.

A farmácia utiliza a modalidade de pagamento electrónico e tem acordo com alguns bancos, permitindo que os clientes desses bancos possam utilizar a farmácia. Caso o cliente não tenha conta num dos bancos identificados como parceiros não será possível efectuar a compra. Como decorre do acordo entre os bancos e a farmácia, o banco cobra uma percentagem, a acordar entre as partes e definida pelo Gestor Financeiro da Farmácia, do valor da compra.

Além de efectuarem compras, os clientes podem consultar o catálogo de medicamentos, bem como ter acesso às últimas compras que efectuaram no sistema. Os donos da farmácia tem também acesso a um conjunto alargado de relatórios entre os quais se incluem o valor total de vendas, valores de comissões cobradas/pagas a fornecedores e a bancos, bem como determinar o top 5 de medicamentos mais vendidos.

Tendo em conta a descrição efectuada, responda às seguintes questões:

1. Desenvolva, utilizando a notação dos Diagramas de Classe, um Modelo de Domínio que considere ser entendível por projectistas e donos da *Help Yourself* e que represente a informação essencial do problema;
2. Construa um Diagrama de Use Case que capture os requisitos existentes na descrição dada;



Sabendo que:

- todas as associações derivadas do Modelo de Domínio deverão ser agregações;
- a camada de negócio (classe ViaVerde) possui (por composição) Maps com objectos Tipo, Cliente, Registo e Auto-Estrada;
- a associação entre Auto-Estrada e Troço captura a ordem dos troços na auto-estrada;
- a direcção a utilizar nas associações deverá ser definida em função do método pedido na alínea (1b).

Construa o **Diagrama de Classes** que permite capturar a informação fornecida. Seja o mais completo possível na construção do mesmo, identificando as relações, nomeando-as e colocando as multiplicidades associadas.

- (b) Construa o **Diagrama de Sequência** para a operação da classe ViaVerde que, dado um identificador e um intervalo de datas, determina o montante a pagar pelo cliente nesse intervalo de datas.

2. Considere a seguinte descrição de um processo de inscrição de alunos numa UCE (Unidade Curricular de Especialização) de mestrado:

Após ter sido proposta, a UCE pode ser aceite ou rejeitada. Se for rejeitada, o processo termina sem sucesso. Se for aceite, fica a aguardar que sejam abertas as inscrições. Quando as inscrições são abertas, é executada a operação `setCapacidade(n)` (indica quantos alunos se poderão inscrever). Note que esta operação só deve ser executada na abertura das inscrições.

Após as inscrições serem abertas a UCE aceita inscrições até às estas serem fechadas. Enquanto existirem vagas, a operação `insereAluno(aluno)` é executada em resposta a cada evento `inscrição(aluno)`. Se a capacidade é esgotada (o método `esgotado()` pode ser utilizado para testar essa condição), a UCE fica cheia. Nesse caso, passa a ser executada a operação `adicListaEspera(aluno)` em resposta aos eventos `inscrição(aluno)`.

Enquanto as inscrições não forem fechadas é possível a UCE ser cancelada. Também enquanto as inscrições não forem fechadas, é possível, a um aluno inscrito, desistir. Nesse caso, se a UCE estiver cheia, é executada a operação `repesca()` que vai buscar um aluno à lista de espera (se não existir lista de espera a vaga fica disponível).

Quando as inscrições são fechadas é executada a operação `notificaDocente()` e o processo termina com sucesso.

Construa o correspondente **Diagrama de Estados**. Sempre que apropriado indique quais as operações executadas em resposta aos eventos.