

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
ESCOLA AGRÍCOLA DE JUNDIAÍ  
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS  
COMPONENTE: TAD0009 – PROG. ORIENTADA A OBJETOS  
PROF. JOSENALDE OLIVEIRA

Exercícios Propostos – Lista 3 – Unidade II

1. Elabore um sistema no paradigma OO que implemente as responsabilidades de uma empresa que entrega botijões de gás (suponha funcionar 24 horas).

Crie a classe Pedido, descubra seus atributos e métodos.

Implemente a classe Principal para um ArrayList de objetos pedido, onde seja possível testar os seguintes requisitos funcionais:

- a) O usuário seleciona no menu a opção "Fazer pedido" e o sistema solicita e insere no novo pedido a hora da compra e o endereço de entrega.
- b) O sistema informa os dados do pedido ao usuário e solicita que ele os confirme ou altere, se for o caso. Em caso de alteração, permite alterar apenas o endereço de entrega, altera no pedido e o exibe. Em caso de confirmação, o sistema solicita ao usuário que digite a quantidade de botijões que deseja e insere no pedido.
- c) O sistema calcula e insere no pedido o total da compra (pesquise o valor médio do preço do botijão na sua região para exibir ao usuário o preço unitário) e a hora de entrega para 6 horas corridas após a hora da compra (verificar mudança de dia), insere no pedido e informa estes dados.
- d) O sistema solicita o número do cartão de crédito e o insere no pedido como forma de pagamento. Em seguida marca como "confirmado" o status do pedido e exibe o código do pedido (número sequencial gerado automaticamente).
- e) Quando o pedido é entregue, o atendente seleciona a opção do menu "Confirmar entrega" que busca o pedido pelo código e, se encontrado, altera o seu status para "entregue". Se não for encontrado, o sistema informa "Pedido não localizado".
- f) A qualquer momento, o usuário pode selecionar no menu as opções "Ver pedidos confirmados" ou "Ver pedidos entregues" para consultar todos os pedidos em aberto ou os atendidos, respectivamente.

2. Elabore um sistema no paradigma OO que implemente o problema de um robô andando em uma sala. Para tanto:

- a) Declare uma classe Robo contendo os atributos linha, coluna e passo, onde linha e coluna representam a posição atual do robô (coordenadas), e passo representa de quantos em quantos pontos o robô se locomove a cada vez. Implemente ainda os métodos (com seus respectivos parâmetros e tipos de retorno): mostrarPosicaoAtual, andarFrente, andarTras, andarDireita e andarEsquerda.
- b) Considere que a sala na qual o robô está tem tamanho 20x40, instancie 1 objeto Robô: R1 (na posição 0,0) e mostre a sala (espaço vazio com robô – 1 – na posição atual) conforme ilustração a seguir:

c) Realize o deslocamento do robô de acordo com a escolha do usuário (1 - Andar para Frente, 2 - Andar para Trás, 3 - Andar para Direita, 4 - Andar para Esquerda). A cada escolha, o sistema deve deslocar o robô e mostrar a sala novamente. Considere que o robô não poderá ultrapassar as fronteiras da sala. Caso isso esteja prestes a ocorrer, o robô deverá se deslocar até a posição imediatamente anterior a fronteira da sala.

3. Desenvolva um Jogo de Dados no paradigma OO atendendo às seguintes regras de negócio:

- a) O jogo admite até 11 jogadores diferentes. Ler o nome e o valor que deseja apostar (entre 1 e 12)
- b) O jogo não admite divisão do prêmio, portanto, não é possível duplicidade de apostas no mesmo número
- c) O jogo possui dois dados, numerados de 1 a 6 cada.
- d) Os dados são então lançados e os números surgem randomicamente na face de cada dado. Estes valores são então somados e exibe-se na tela o número sorteado.
- e) O jogo verifica se alguém venceu ou se a “máquina” venceu!
- f) Se alguém vencer, terá seus dados registrados em arquivo com a indicação de uma vitória
- g) Se a mesma pessoa jogar novamente e ganhar, o registro é atualizado para incrementar as vitórias deste jogador
- h) Na primeira tela do jogo (ao carregar) deve ser possível ver o ranking (caso haja) do TOP FIVE, com o nome e número de vitórias de cada ganhador. E ao apertar alguma tecla passa para a “tela” de coletar dados dos jogadores.
- i) Considere modelar o problema em UML com classes Dado, Player, Game etc.

4. Elaborar um programa OO que:

- a. Implemente uma classe abstrata C1;
- b. Implemente duas classes concretas C2 e C3, ambas herdando de C1;
- c. Implemente duas classes concretas C4 e C5, ambas herdando de C2;
- d. Implemente duas interfaces, I1 (com 1 método, no mínimo) e I2 (com dois métodos, no mínimo);
- e. Considere que a classe C3 implementa as interfaces I1 e I2;
- f. Considere que a classe C5 implementa a interface I1;
- g. Considere que todas as classes devem ter pelo menos um método e um atributo próprios;
- h. Demonstre no exercício:
  - i. Sobrecarga de construtores;
  - ii. Sobrecarga de métodos;

iii. Sobreposição de métodos.

i. Implemente a classe Principal, para testar todos os métodos das demais classes, contendo pelo menos um objeto de cada classe concreta.