Universidade Federal do Rio Grande do Norte
Escola Agrícola de Jundiaí
Tecnologia em Análise e Desenvolvimento de Sistemas
Programação Orientada a Objetos
Profa. Alessandra Mendes

LISTA 2 – 31/08/2023

1. Sejam A e B dois vetores contendo 10 elementos inteiros. Elabore um algoritmo que:

   - Leia A e B.
   - Calcule a soma dos elementos de A.
   - Crie o vetor C contendo a soma dos elementos de mesma posição dos vetores A e B.
     Refaça o mesmo exercício utilizando arraylists no lugar de vetores.

2. Elabore um programa em Java que implemente as responsabilidades de uma empresa que entrega botijões de gás 24 horas.

   Crie a classe Pedido, descubra seus atributos e métodos e implemente a classe Principal para um arrayList de objetos pedido:

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, o sistema pede novamente a hora da compra e 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) e) O sistema calcula e insere no pedido o total da compra (o botijão custa R$ 60,00) 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”.

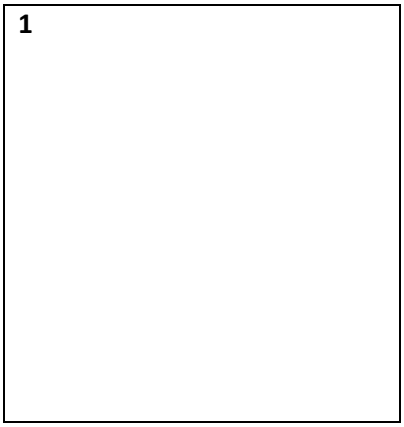
f) A qualquer momento, o usuário pode selecionar no menu as opções “Ver pedidos confirmados” ou “Ver pedidos entregues” para consultar todos os pedidos em aberto ou os atendidos, respectivamente.

3. Elabore um programa em Java que implemente o problema de um robô andando em uma sala. Para tanto:

a) Declare uma classe robô 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.

4. Elabore um programa em Java que implemente as responsabilidades de um cinema utilizando a classe Ingresso de tal forma que:

   Cada ingresso deve ter o código do ingresso (calculado automaticamente), o cpf do cliente, o nome do filme, o número da poltrona (1 a 120) e o valor do ingresso (16 reais inteira ou 8 reais meia). Cada usuário só poderá comprar um ingresso por vez e só pode ser vendido um ingresso para cada poltrona.

   Implemente o ArrayList SALA (de Ingressos) para controlar os ingressos da sala. Para cada ingresso vendido, instancie um objeto ingresso e insira-o no ArrayList. Além de adquirir um ingresso, o usuário poderá ainda alterar o seu próprio ingresso já adquirido ou desistir do ingresso (neste caso o ingresso deverá ser excluído do ArrayList).

   Crie também a classe Principal para testar os métodos da classe Ingresso considerando que este cinema só tem um sala que comporta, no máximo, 120 pessoas.

5. Elabore um programa em Java que simule uma agenda de contatos. Para tanto:

   a) Crie uma classe para representar uma pessoa, com os atributos nome, telefone e e-mail. Crie os métodos públicos necessários para sets e gets e também um método para retornar todos os dados de uma pessoa (estado do objeto).

   b) Crie uma classe Principal que seja capaz de armazenar várias pessoas (na lista Agenda) e seja capaz ainda de realizar as seguintes operações: cadastrar pessoa, remover pessoa, buscar pessoa, informar um dado específico de uma pessoa (nome, telefone ou e-mail) e escrever toda a agenda.

6. Elabore um programa em Java que:

   Implemente a classe Aluno, cujos objetos representam os alunos matriculados em uma disciplina. Cada objeto dessa classe deve guardar os seguintes dados do aluno: matrícula, nome, 2 notas de prova e 1 nota de trabalho. Escreva um construtor que receba todos os valores dos atributos como parâmetros e os seguintes métodos para esta classe:

media	Calcula a média final do aluno (cada prova tem peso 2,5 e o trabalho tem peso 2)
final	Calcula quanto o aluno precisa na prova final (retorna zero se ele não for para a final). Considere apto para a prova final o aluno cuja media for < 5 e aprovado com média >=5
estado	Retorna uma String contendo todo o estado do objeto, incluindo os valores da média e quanto ele precisa tirar na prova final (se for o caso).

   Implemente a classe Disciplina contendo o ArrayList de Alunos (classe Aluno) como atributo, além do código da disciplina, título, carga horária e nome do professor. Esta classe controla a disciplina e seus alunos. Para isto, a classe deve implementar os métodos:

construtor	recebe como parâmetro o código da disciplina (inteiro sequencial)
melhorAluno	retorna o nome do aluno que obteve a maior média
media	retorna a média dos alunos da disciplina
inserirDisciplina	recebe os dados da disciplina
inserirAluno	cadastra um aluno na disciplina
consultarAluno	retorna os dados do aluno cujo código é igual ao enviado como parâmetro
getDados	retorna os dados da disciplina
alterarProfessor	altera o professor da disciplina

   Implemente a classe Principal para testar todos os métodos da classe Disciplina.

7. Elabore um programa OO que escreva uma classe que represente um país.

   Considere que um país tem como atributos o seu nome, o nome da capital, sua dimensão em Km2 e uma lista de países com os quais ele faz fronteira. Represente a classe e forneça os seguintes construtores e métodos:

a) Construtor que inicialize o nome, capital e a dimensão do país;

b) Métodos de acesso (get) para os atributos indicadas no item (a);

c) Um método que permita verificar se dois países são iguais. Considere como países iguais aqueles que tiverem o mesmo nome e a mesma capital. A assinatura deste método deve ser:

```
public boolean equals(Pais outro);
```

d) Um método que defina quais outros países fazem fronteira (note que um país não pode fazer fronteira com ele mesmo);

e) Um método que retorne a lista de países que fazem fronteira;

f) Um método que receba um outro país como parâmetro e retorne uma lista de vizinhos comuns aos dois países.

8. Playlist é um termo inglês que geralmente é utilizado para se referir a uma determinada lista de músicas que podem ser tocadas em sequência ou embaralhadas. Simule uma playlist de acordo com as especificações abaixo:

Implemente em Java a classe Música, inserindo os atributos (private) e métodos (public) pertinentes. Implemente também uma classe Principal. Nesta classe, instancie um vetor ou uma lista de objetos da classe Música chamado Playlist e mostre o MENU abaixo, executando as respectivas funcionalidades de acordo com a escolha do usuário:

PLAYLIST

---

1 – Adicionar uma música
2 – Excluir uma música
3 – Tocar uma música específica (pelo título)
4 – Tocar as músicas de um cantor
5 – Tocar as músicas em sequência
6 – Tocar as músicas embaralhadas (random)
7 – Ver as músicas da playlist
8 – Sair

Observações:

1. Nenhum método da classe Música pode ter o comando print ou suas variações (printf ou println);
2. A Playlist não poderá ter músicas repetidas (mesmo título e cantor);

Considere “tocar” como o ato de escrever na tela o título da música e sua duração decrescendo do tempo total cadastrado (em min e seg, transformado para seg) até que chegue a 0.

Sugestão: use o bloco de comandos abaixo para “simular” a música tocando.

```
// duracaoTotal é um atributo da classe música com o tempo da duração da música em segundos
for(cont=0;cont<m.getDuracaoTotal(); cont++)
 {
        try{
       Thread.sleep(1000); // pausa de 1 segundo
    }catch(InterruptedException e){
       System.out.println("Erro na execução da música: "+e.getMessage());
    }
    System.out.print("|");
}
```

OBS: Implemente outros métodos que julgar necessários nas questões.