

4ª Lista de Exercícios

Parte I- Exercícios teóricos

- 1) Complete as frases a seguir:
 - a. Listas e tabelas de valores podem ser armazenados em _____.
 - b. Um vetor é um grupo de _____ (chamado elementos ou componentes) contendo valores que são todos do mesmo _____.
 - c. O _____ permite que os programadores iterem através dos elementos em um vetor sem precisar usar um contador.
 - d. O número usado para referenciar um elemento particular de um vetor é chamado de _____ do elemento.
 - e. Um vetor que usa dois índices é denominado um vetor _____.
 - f. Argumentos de linha de comando são armazenados em _____.
 - g. Dado o comando **java MyClass test**, o primeiro argumento de linha de comando é _____.
- 2) Qual será o conteúdo dos vetores declarados no código a seguir, ao término da execução do método main?

```
public static void main (String[] args) {  
    double[] valores = {1,2,3,4,5,6};  
    double[] primeiraCopia = valores;  
    double[] segundaCopia = valores;  
    primeiraCopia[1] = 1;  
    segundaCopia[2] = valores[0]+ primeiraCopia[1];  
    primeiraCopia[3] = valores[1]+ segundaCopia[2];  
    valores[4] = primeiraCopia[2]+ segundaCopia[3];  
    valores[5] = segundaCopia[3]+ primeiraCopia[4];  
}
```

Parte II- Exercícios práticos

- 1) (Rafael Santos) Escreva uma aplicação que declare e inicialize um vetor de booleanos (lendo-os do teclado) e calcule quantos elementos são iguais à true.
- 2) (Rafael Santos) Escreva uma classe que encapsule um array de 12 bytes, onde cada elemento do array contém o número de dias no mês correspondente, desconsiderando se o ano é bissexto ou não. Por exemplo, o elemento 0 do array (correspondente à janeiro) deve valer 31. Escreva um método que retorne o valor encapsulado para determinado mês. Por fim, crie uma classe de teste.
- 3) (Rafael Santos) Escreva um programa que crie um array de valores inteiros e preenche o array de forma que o valor do elemento do array seja o dobro de seu índice. Desta forma o quinto elemento do array deverá valer 8 - lembre-se que os índices começam em zero.
- 4) Desenvolva um programa que lê a idade de 10 pessoas, armazena em um ArrayList, lê uma posição e remove do ArrayList a idade localizada na posição informada. Por exemplo, se o usuário informar o número 4, a idade armazenada na posição 4 do ArrayList deve ser removida. Utilize um dos métodos da classe ArrayList para remover um elemento da lista.
- 5) (Deitel) Uma empresa paga vendedores com base em comissão. O vendedor recebe R\$200 acrescido de 9% do total de suas vendas na semana. Por exemplo, um vendedor que vende R\$5000 em uma semana recebe R\$200 mais 9% de R\$5000, totalizando R\$650. Escreva uma aplicação (usando um vetor de contadores) para determinar quantos vendedores receberam salários nas seguintes faixas de valores (admita que o salário de cada vendedor é truncado para um valor inteiro):
 - \$200–299
 - \$300–399
 - \$400–499
 - \$500–599

- \$600–699
 - \$700–799
- 6) (Deitel) Escreva uma aplicação que recebe 5 números entre 10 e 100. A medida que cada número é lido, exiba-o somente se ele não estiver repetido em relação a um número já lido. Trate o pior caso, onde todos os 5 números são diferentes. Utilize o menor vetor possível para resolver esse problema. Exiba o conjunto completo de valores únicos lidos após o usuário introduzir cada novo valor.
 - 7) (Rafael Santos) Escreva uma classe Menu que encapsule um vetor de Strings. Nesta classe, crie o método `escolhe`, que mostra as Strings com um número na frente, espere o usuário digitar um número e retorne a String correspondente àquele número. Dicas: Faça com que o usuário seja impossibilitado de digitar números fora da faixa com um laço `while`, e faça com que os números a serem impressos antes das Strings estejam na faixa 1-N, onde N é o tamanho do array. Desta forma, a escolha fica mais amigável para o usuário.
 - 8) Crie uma classe de banco de palavras que encapsula um vetor de palavras. A classe deve ter um método para sortear uma palavra do vetor. Dica: Utilize o método `nextInt(int limite)` da classe `Random` da API de Java para gerar um número aleatório no intervalo `[0, limite[`. Não esqueça de inicializar seu arquivo com o comando `import java.util.Random;`.
 - 9) Implemente o jogo da forca utilizando `String` e vetor, bem como a entrada e a saída padrão. Para tanto, pense no funcionamento do jogo no mundo real, separando cada passo em métodos. O número máximo de erros que o jogador pode cometer é 6. A visualização dos erros pode ser feita por uma mera contagem ou, preferencialmente, pela exibição do boneco. Utilize o banco de palavras da questão anterior de maneira que, a cada nova partida, o jogo deverá solicitar uma palavra do banco de palavras.
Dicas:
 - Para ler uma letra do teclado, use o método `nextLine` da classe `Scanner`, que armazena a entrada do usuário em uma `String` e, em seguida, o método `charAt(int)` da classe `String`, que retorna o `char` na `String` na posição passada como argumento. Logo, a chamada `char letra = scanner.nextLine().charAt(0);` retorna a primeira letra da palavra lida pelo teclado.
 - Para manipular `String` como vetor, use o método `toCharArray` da classe `String`, que retorna a `String` como um vetor de `char`: `char[] vetor = texto.toCharArray();`.
 - 10) Crie a classe `RepositorioContasVetor`, que gerencia o armazenamento de contas bancárias através de um vetor. A classe deverá conter métodos para: inserir uma conta, remover uma conta, atualizar uma conta, procurar uma conta pelo número, procurar uma conta pelo cliente, listar todas as contas, listar todas as contas de um cliente e verificar se uma conta existe. Crie uma classe `TesteRepositorioContasVetor` para testar as funcionalidades desenvolvidas. Observação: Não deve ser possível armazenar contas com o mesmo número.
 - 11) Com base na questão anterior, implemente uma classe para representar um repositório de clientes. Utilize vetor ou `ArrayList`.
 - 12) (Deitel) Uma pequena companhia aérea comprou um computador para usar seu novo sistema automatizado de reservas. Você deve desenvolver o novo sistema. Para tanto, você deve escrever uma aplicação para identificar assentos em cada voo do único avião da companhia (capacidade de 10 assentos). Sua aplicação deve exibir as seguintes alternativas: “Tecla 1 para Primeira Classe” e “Tecla 2 para Classe Econômica”. Se o usuário digitar 1, sua aplicação deve atribuir um assento na seção de primeira classe (assentos 1-5). Se o usuário digitar 2, sua aplicação deve atribuir um assento na seção econômica (assentos 6-10). Sua aplicação deve então exibir um ticket de viagem para o passageiro, indicando o número do assento e se este está na primeira classe ou na classe econômica. Use um vetor unidimensional do tipo `boolean` para representar o mapa de assentos do avião. Inicialize todos os elementos do vetor com `false` para indicar que todos os assentos estão livres. À medida que cada assento for atribuído, ajuste os elementos correspondentes no vetor para `true` para indicar que o assento não está mais disponível. Sua aplicação nunca deve atribuir um assento que já foi ocupado. Quando a seção econômica está cheia, sua aplicação deve perguntar à pessoa se ela aceita ser alocada na primeira classe (e vice-versa). Se sim, faça a atribuição de assento apropriada. Se não, exiba a mensagem “O próximo voo sai em 3 horas”.
 - 13) (Rafael Santos) Crie uma classe denominada `MatrizDeInteiros` para representar uma matriz de valores do tipo `int`. Essa classe deve conter um método denominado `eIgual` para verificar se duas matrizes são iguais e que, portanto, deve retornar `true` se as matrizes comparadas forem iguais ou `false`, caso contrário. Duas matrizes devem ser consideradas iguais se forem do mesmo tamanho e guardarem os mesmos valores. Por fim, crie uma classe de teste.
 - 14) (Rafael Santos) Escreva o método `eQuadrada` para a classe `MatrizDeInteiros`, que retorna `true` se a matriz for quadrada, isto é, se a matriz tem o mesmo número de linhas e colunas. Não esqueça de testar o novo método.
 - 15) (Rafael Santos) Escreva o método `eQuadradoMágico` para a classe `MatrizDeInteiros`, que retorna `true` se a matriz for um quadrado mágico. Um quadrado mágico é uma matriz quadrada de valores inteiros onde a soma dos valores em cada linha, coluna ou diagonal principal é a mesma, e não existem elementos repetidos. Por exemplo, a matriz apresentada a seguir é um quadrado mágico. Não esqueça de testar o novo método.

2	7	6
9	5	1
4	3	8

- 16) (Rafael Santos) Escreva um programa que declare um array bidimensional chamado tabuada de 10x10 posições e popule os elementos do array com os valores da tabuada da multiplicação para aquele elemento, de forma que, por exemplo, o elemento tabuada[7][9] valha 63.

- 17) Implemente o jogo da velha utilizando matrizes. Para tanto, considere que deve existir uma classe para encapsular a lógica do jogo, representando o tabuleiro. A classe deve ter métodos para fazer uma jogada (indicando as coordenadas), checar se o jogo não acabou ainda, saber qual jogador ganhou ou se deu empate, imprimir o tabuleiro na tela, e outros, conforme necessidade.

Antes de começar o jogo, devem ser definidos os símbolos para cada jogador (X ou O). Durante a partida, cada jogador deve escrever sua jogada (coordenadas) na linha de comando, e o jogo deve imprimir o tabuleiro e esperar pela jogada do próximo jogador.

O programa não deve permitir que o jogador tente marcar uma casa que já esteja marcada, nem que tente jogar em casas que não existam (coordenadas negativas, ou maior que o tamanho do tabuleiro). A classe que encapsula a lógica do jogo deve conter métodos para testar se uma dada jogada é válida, que devem ser chamados antes de efetuar a jogada de fato.

- 18) Implemente o jogo de batalha naval utilizando matrizes. Considere que cada navio ocupa três posições, sendo o tabuleiro de tamanho 7x7. O comportamento do jogo é descrito a seguir.

- Trata-se de uma competição entre 2 jogadores: Ganha o jogo quem destruir os navios do adversário primeiro.
- Primeiramente, cada jogador deve posicionar os navios no tabuleiro. Logo, visualizar o tabuleiro é essencial.
- Dado que os navios foram posicionados, cada jogador faz um palpite para atacar uma determinada posição do tabuleiro, de maneira alternada, começando pelo jogador 1.
- Cada palpite deverá causar um efeito que deverá ser visualizado pelo jogador. Um palpite certo é aquele que atinge um navio (ou parte dele); um palpite é errado, se não atingir navio algum e um palpite elimina um navio somente quando todas as posições que compõem o navio são atacadas.
- Uma posição que já foi atacada não pode ser atacada novamente.
- Posições fora das dimensões do tabuleiro também não podem ser atacadas.
- O jogo só acaba quando um dos jogadores consegue destruir todos os navios do seu adversário ou quando o jogador seleciona a opção de encerrar partida.
- No caso de haver um vencedor, o programa deve mostrar.
- Deverão ser utilizadas a entrada e a saída padrão.

- 19) Crie a classe Carta, cujos objetos possuem um nome e um naipe. Crie uma classe Baralho, que possui 56 cartas no total, sendo 52 cartas de naipe e 4 coringas. O coringa é uma carta cujo nome é “coringa” e cujo naipe é “coringa” também. A classe Baralho deve ter os seguintes métodos:

- embaralhar: Mistura as cartas dentro do baralho. Para tanto, sorteie 2 posições e troque-as. Dica: Utilize o método `nextInt(int limite)` da classe `Random` da API de Java para gerar um número aleatório no intervalo `[0, limite[`. Não esqueça de inicializar seu arquivo com o comando `import java.util.Random;`
- darCarta: Devolve uma carta, retirada do topo do baralho (última posição), caso o baralho não esteja vazio;
- temCarta: Verifica se tem carta no baralho, devolvendo `true` ou `false`.
- imprimirBaralho: Imprime as cartas para verificar como estão dispostas (se estão embaralhadas, por exemplo).

- 20) Crie a classe `ArrayUtilidades` que contenha um método que recebe um array de inteiros e devolve um array de boolean, onde cada posição contém `true` se o elemento da posição correspondente no array de entrada é positivo ou `false`, caso o elemento seja negativo ou zero.

- 21) (Ana Cristina Melo) As informações a seguir se referem a uma aplicação de controle pessoal de horário de remédios. Para cada remédio cadastra-se o nome de quem vai tomar o remédio, a data de início, a quantidade de dias que foi prescrito pelo médico, a quantidade de vezes ao dia, a dosagem e o nome do remédio. Ao cadastrar o remédio, a aplicação sugere todos os horários possíveis para tomar o remédio. O usuário escolhe o melhor horário e a aplicação avisa até quando o remédio deve ser tomado e prepara uma planilha de horários. No início do dia, o usuário seleciona a opção de planilha de horários de remédios do dia. Caso o usuário atrase o horário de tomar o remédio num determinado dia, a aplicação reorganiza os horários daquele dia. Identifique as classes, atributos e métodos desse cenário.

- 22) (Ana Cristina Melo) As informações a seguir se referem a uma aplicação de controle de comanda eletrônica da padaria Doce Sabor do Seu Joaquim. O cliente usa uma comanda eletrônica durante suas compras na Padaria. A cada produto consumido, o atendente registra em sua comanda (que possui uma numeração) o produto e a quantidade. Ao passar no caixa na saída da Padaria, a Caixa lê os gastos da comanda, finalizando a compra. Na leitura da comanda, verifica-se o valor unitário de cada produto a fim de calcular o valor total da compra. Identifique as classes, atributos e métodos desse cenário.
- 23) (Ana Cristina Melo) Carolina controla em Excel uma planilha com sua lista de compras mensal. Ela cadastra o nome do produto, a unidade da compra, a quantidade prevista para um mês, a quantidade que efetivamente será comprada e o preço estimado (atualizado todo mês), como ilustrado a seguir. A quantidade de compra pode variar em virtude de sobra de um mês para o outro, ou da necessidade de um gasto maior no mês. Por exemplo: almoço em família. As compras são feitas pela própria Carolina. Por esse motivo, ela não vê necessidade de relacionar as marcas dos produtos. Mensalmente, Carolina analisa o quanto pagou por cada produto, e se achar necessário, atualiza o preço estimado de cada produto. Identifique as classes, atributos e métodos desse cenário.

Produto	Unidade de compra	Qtd Mês	Qtd compra	Preço estimado
Arroz	Kg	8	7	6,00
Feijão	Kg	6	6	10,00
Açúcar	Kg	3	2	4,50
Carne	Kg	6	7,6	29,50
...				
Total Estimado				500,00

Mais exercícios: Capítulo 7 do livro de Deitel (10ª Edição) e capítulo 11 do livro de Rafael Santos.