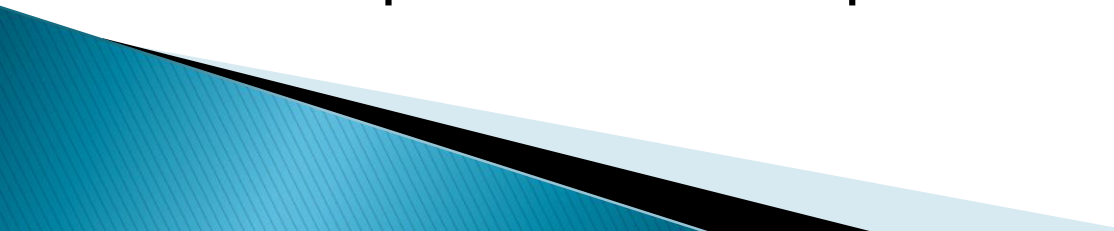


Linguagem de Programação

Linguagem Java
Uso de Métodos (Parte 3)

Passagem de parâmetros

- Na literatura, há 2 tipos básicos de passagem de parâmetros para subprogramas
 - Por valor: uma cópia do valor da variável é transmitida ao subprograma
 - Por referência: uma referência (pode-se considerar o endereçamento de memória) é transmitida ao subprograma
 - Permite alterar o conteúdo da variável que foi passada como parâmetro
- 

Passagem de parâmetros por valor e por referência

- Em java, tipos primitivos (int, long, float, double, char, etc.) são passados por valor
- Tipos referenciais (objetos, tais como StringBuffer, Vector, entre outros) são passados por referência

Exemplo da passagem de parâmetros por valor

```
public class Trocar {
```

```
    public static void main(String[] args) {
```

```
        int a = 10, b = 20;
```

```
        System.out.printf("a = %d\tb= %d\n", a, b);
```

```
        trocar(a, b);
```

```
    }
```

```
    public static void trocar(int a, int b) {
```

```
        int aux = a;
```

```
        a = b;
```

```
        b = aux;
```

```
        System.out.printf("a = %d\tb= %d\n", a, b);
```

```
    }
```

```
}
```

Variáveis simples

```
C:\Sandra\AulasJava\Metodos>javac Trocar.java
```

```
C:\Sandra\AulasJava\Metodos>java Trocar
```

```
a = 10  b= 20
```

```
a = 20  b= 10
```

Exemplo da passagem de parâmetros por referência

```
public class PassagemPorReferencia {  
  
    public static void main(String[] args) {  
        StringBuffer entrada = new StringBuffer("Fatec");  
        System.out.println(entrada);  
        mudarValor(entrada);  
        System.out.println(entrada);  
    }  
  
    public static void mudarValor(StringBuffer entrada) {  
        entrada.append(" Carapicuiiba");  
    }  
}
```

```
C:\Sandra\AulasJava\Metodos>javac PassagemPorReferencia.java  
  
C:\Sandra\AulasJava\Metodos>java PassagemPorReferencia  
Fatec  
Fatec Carapicuiiba
```

Enviando um vetor para um método (1/2)

```
1  import java.util.Random;
2  import java.util.Scanner;
3
4  public class VetorComoParametro {
5
6      public static void main(String[] args) {
7          Scanner entrada = new Scanner(System.in);
8
9          System.out.print("Informe o tamanho do vetor: ");
10         int n = entrada.nextInt();
11         int[] vet = new int[n];
12
13         System.out.print("Informe o valor limite para os elementos do vetor: ");
14         int limite = entrada.nextInt();
15         preencherVetor(vet, limite);
16         imprimirVetor(vet);
17         entrada.close();
18     }
```

O vetor é passado como parâmetro por referência

Enviando um vetor para um método (2/2)

```
19 public static void preencherVetor(int[] vet, int limite) {  
20     Random rand = new Random();  
21     for (int i = 0; i < vet.length; i++) {  
22         vet[i] = rand.nextInt(limite);  
23     }  
24 }  
25  
26 public static void imprimirVetor(int[] vet) {  
27     int i;  
28     System.out.print("Conteudo do vetor: [ ");  
29     for (i = 0; i < vet.length - 1; i++) {  
30         System.out.printf("%d, ", vet[i]);  
31     }  
32     System.out.printf("%d ]\n", vet[i]);  
33 }  
34  
35 }
```

```
C:\Sandra\AulasJava\Metodos>javac VetorComoParametro.java
```

```
C:\Sandra\AulasJava\Metodos>java VetorComoParametro
```

```
Informe o tamanho do vetor: 5
```

```
Informe o valor limite para os elementos do vetor: 50
```

```
Conteudo do vetor: [ 9, 27, 34, 6, 44 ]
```

Enviando uma matriz para um método (1/2)

```
1  import java.util.Random;
2  import java.util.Scanner;
3
4  public class MatrizComoParametro {
5
6      public static void main(String[] args) {
7          Scanner entrada = new Scanner(System.in);
8          System.out.print("Informe o numero de linha e coluna da matriz: ");
9          int linhas = entrada.nextInt(), colunas = entrada.nextInt();
10         int[][] mat = new int[linhas][colunas];
11         System.out.print(
12             "Informe o valor limite para os elementos da matriz: ");
13         int limite = entrada.nextInt();
14         preencherMatriz(mat, limite);
15         imprimirMatriz(mat);
16         entrada.close();
17     }
18 }
```

A matriz é passada como parâmetro por referência

Enviando uma matriz para um método (2/2)

```
18
19 public static void preencherMatriz(int[][] mat, int limite) {
20     Random rand = new Random();
21     for (int i = 0; i < mat.length; i++) {
22         for (int j = 0; j < mat[i].length; j++) {
23             mat[i][j] = rand.nextInt(limite);
24         }
25     }
26 }
27 public static void imprimirMatriz(int[][] mat) {
28     System.out.println("Conteudo da matriz:");
29     for (int i = 0; i < mat.length; i++) {
30         for (int j = 0; j < mat[i].length; j++) {
31             System.out.printf("%d\t", mat[i][j]);
32         }
33         System.out.println();
34     }
35 }
36 }
37 }
```

```
C:\Sandra\AulasJava\Metodos>javac MatrizComoParametro.java
```

```
C:\Sandra\AulasJava\Metodos>java MatrizComoParametro
```

```
Informe o numero de linha e coluna da matriz: 2
```

```
3
```

```
Informe o valor limite para os elementos da matriz: 20
```

```
Conteudo da matriz:
```

```
4      17      11
```

```
19     16      19
```

Vetor e matriz como valor de retorno

- Em Java é possível que um método devolva um vetor ou uma matriz como valor de retorno
- Nesse caso, o retorno não será apenas um único dado (conteúdo de uma variável simples)

Vetor como valor de retorno (1/2)

```
1  import java.util.Random;
2  import java.util.Scanner;
3
4  public class VetorComoRetorno {
5
6      public static void main(String[] args) {
7          Scanner entrada = new Scanner(System.in);
8          System.out.print("Informe o tamanho: ");
9          int n = entrada.nextInt();
10         System.out.print("Informe o valor limite para os elementos do vetor: ");
11         int limite = entrada.nextInt();
12         int[] vet = criarVetor(n, limite);
13         imprimirVetor(vet);
14         entrada.close();
15     }
16 }
```

Vetor como valor de retorno (2/2)

```
17 public static int[] criarVetor(int n, int limite) {
18     Random rand = new Random();
19     int[] v = new int[n];
20     for (int i = 0; i < v.length; i++) {
21         v[i] = rand.nextInt(limite);
22     }
23     return v;
24 }
25 public static void imprimirVetor(int[] vet) {
26     int i;
27     System.out.print("Conteudo do vetor: [ ");
28     for (i = 0; i < vet.length - 1; i++) {
29         System.out.printf("%d, ", vet[i]);
30     }
31     System.out.printf("%d ]\n", vet[i]);
32 }
33
34 }
```

```
C:\Sandra\AulasJava\Metodos>javac VetorComoRetorno.java
```

```
C:\Sandra\AulasJava\Metodos>java VetorComoRetorno
```

```
Informe o tamanho: 10
```

```
Informe o valor limite para os elementos do vetor: 20
```

```
Conteudo do vetor: [ 11, 10, 13, 6, 0, 14, 10, 12, 14, 3 ]
```

Matriz como valor de retorno (1/2)

```
1  import java.util.Random;
2  import java.util.Scanner;
3
4  public class MatrizComoRetorno {
5
6      public static void main(String[] args) {
7          Scanner entrada = new Scanner(System.in);
8          System.out.print("Informe o numero de linha e coluna da matriz: ");
9          int linhas = entrada.nextInt(), colunas = entrada.nextInt();
10         System.out.print(
11             "Informe o valor limite para os elementos da matriz: ");
12         int limite = entrada.nextInt();
13         int[][] mat = criarMatriz(linhas, colunas, limite);
14         imprimirMatriz(mat);
15         entrada.close();
16     }
```

Matriz como valor de retorno (2/2)

```
17
18 public static int[][] criarMatriz(int linhas, int colunas, int limite) {
19     Random rand = new Random();
20     int[][] m = new int[linhas][colunas];
21     for (int i = 0; i < m.length; i++) {
22         for (int j = 0; j < m[i].length; j++) {
23             m[i][j] = rand.nextInt(limite);
24         }
25     }
26     return m;
27 }
28 public static void imprimirMatriz(int[][] mat) {
29     System.out.println("Conteudo da matriz:");
30     for (int i = 0; i < mat.length; i++) {
31         for (int j = 0; j < mat[i].length; j++) {
32             System.out.printf("%d\t", mat[i][j]);
33         }
34         System.out.println();
35     }
36 }
37 }
```

```
C:\Sandra\AulasJava\Metodos>javac MatrizComoRetorno.java
```

```
C:\Sandra\AulasJava\Metodos>javac MatrizComoRetorno.java
```

```
C:\Sandra\AulasJava\Metodos>java MatrizComoRetorno
```

```
Informe o numero de linha e coluna da matriz: 3
```

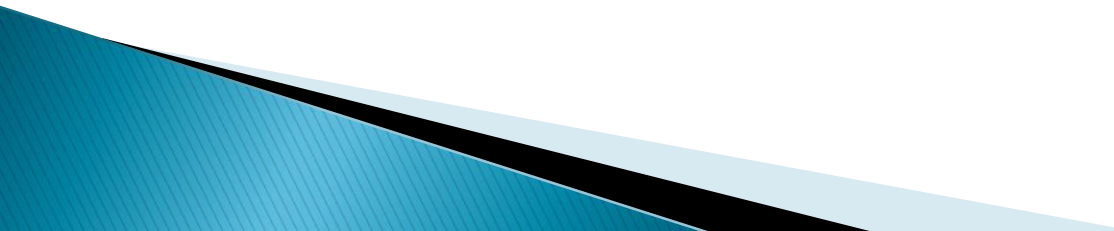
```
4
```

```
Informe o valor limite para os elementos da matriz: 20
```

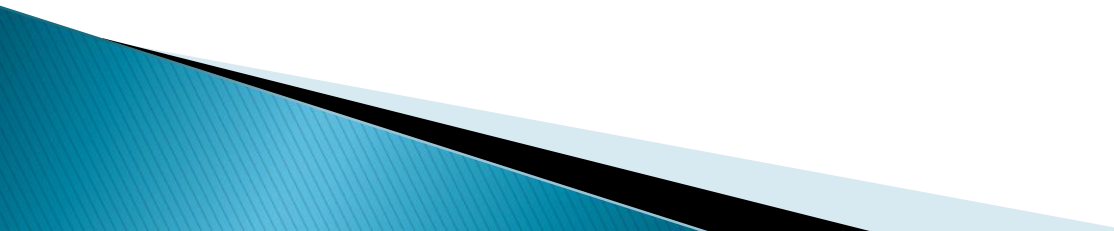
```
Conteudo da matriz:
```

18	0	1	14
18	1	19	8
1	2	6	8

Exercícios

- 9) Faça um programa para armazenar 20 números reais em um vetor chamado Num. Em seguida, chamar um método denominado imprimirPos que recebe o vetor Num por parâmetro e imprime apenas os números positivos contidos no vetor.
- 10) Faça um programa para armazenar 50 valores reais em um vetor chamados Valores. Em seguida, implemente os seguintes métodos para:
- mostrar os valores do vetor;
 - calcular e mostrar a média dos valores do vetor; e
 - encontrar e retornar para o método “main” o maior valor contido no vetor e imprimir o valor recebido.
- 

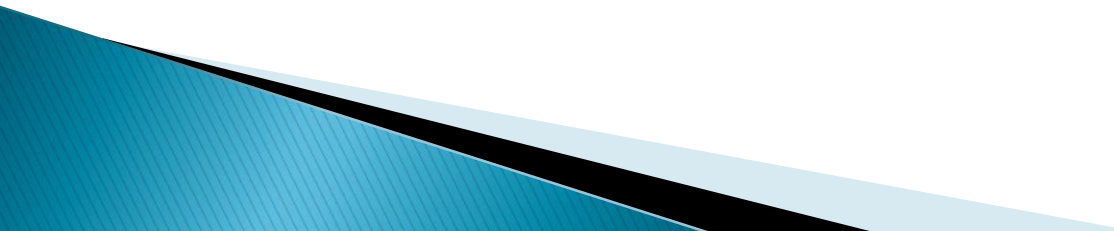
Exercícios

- 11) Faça um programa para armazenar 100 números inteiros em um vetor chamado Numeros. Em seguida chamar um método denominado somarPares que:
- recebe o vetor Numeros por parâmetro;
 - soma os números pares; e
 - retorna o total da soma para a função principal a fim de imprimir esse total.
- 12) Escreva um programa para:
- ler o número inteiro e armazenar esse número em uma variável chamada Opcao.
 - armazenar 30 números inteiros positivos em um vetor chamado Vet.
- 

Exercícios

12) Continuação do exercício

O programa deve possuir quatro métodos denominados f1, f2, f3, e f4, as quais possuirão as seguintes funcionalidades:

- f1: deverá imprimir o vetor;
 - f2: deverá imprimir os números armazenados nas posições ímpares do vetor;
 - f3: deverá imprimir o quadrado dos números armazenados nas posições pares do vetor; e
 - f4: deverá imprimir a soma dos números divisíveis por 4.
- 

Exercícios

12) Continuação do exercício

Obs: a execução dos métodos ocorrerá caso o usuário digite: ou o valor 1, ou o valor 2, ou o valor 3, ou o valor 4. Assim:

Se Opcao = 1 será necessário chamar o método f1;

Se Opcao = 2 será necessário chamar o método f2;

Se Opcao = 3 será necessário chamar o método f3;

Se Opcao = 4 será necessário chamar o método f4;

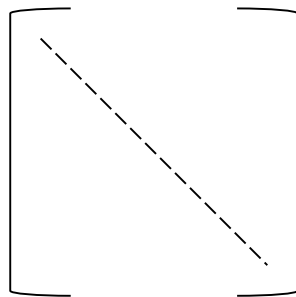
Para outros valores será necessário imprimir “Não existe um método que corresponde a Opcao digitada”



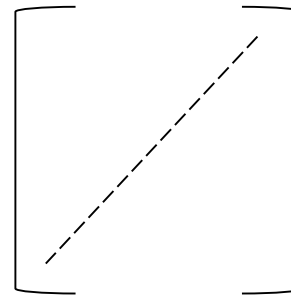
Exercícios

13) Faça um programa para armazenar números inteiros em uma matriz quadrada de ordem 4, ou seja, uma matriz 4x4. Em seguida criar os seguintes métodos:

- Método que recebe a matriz e verifica se a soma de todos os valores contidos na matriz é par;
- Método que recebe a matriz e a imprime; e
- Método que recebe a matriz e imprime os elementos da diagonal principal e diagonal secundária.



Diagonal Principal



Diagonal Secundária

Exercícios

14) Faça um programa para armazenar números inteiros em uma matriz 2x2. Em seguida criar um método chamado calcularDeterminante que irá receber a matriz e calcular e imprimir o determinante da matriz. O determinante é obtido como a definição a seguir:

$$\det A = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix} = (a_1 * b_2) - (a_2 * b_1)$$

Exemplo:

$$\det Mat = \begin{bmatrix} 3 & -2 \\ 4 & 1 \end{bmatrix} = (3*1) - (-2 * 4) = 3 + 8 = 11$$

Exercícios

- 15) Escreva um programa para armazenar números inteiros em duas matrizes de ordem 3, chamadas MatrizA e MatrizB. Além disso o programa deve conter os seguintes métodos:
- lerMatriz: responsável por inserir dados nas matrizes;
 - verificarMatrizIdentidade: que recebe uma matriz de ordem 3 e verifica se essa matriz é uma matriz identidade, em que os elementos da diagonal principal são iguais a 1 e os demais elementos são todos iguais a zero.

Dessa forma, o programa deverá chamar o método lerMatriz para a entrada de dados, em seguida irá chamar o método verificarMatrizIdentidade que recebe a MatrizA e depois chamar novamente o método verificarMatrizIdentidade que recebe a MatrizB.

Referências

- DEITEL, H.M.; DEITEL, P.J.. *Java Como Programar*. 4. ed., Porto Alegre: Bookman, 2002.
- ECKEL, B.. *Thinking in Java*. 3. ed., Prentice Hall, 2002.