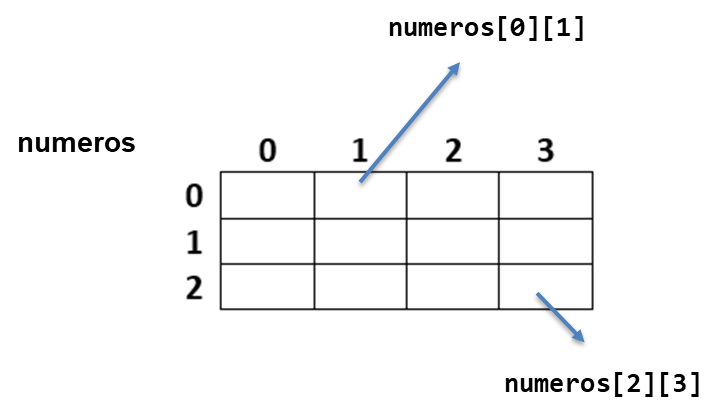
# Aula 10-Matriz

## Introdução

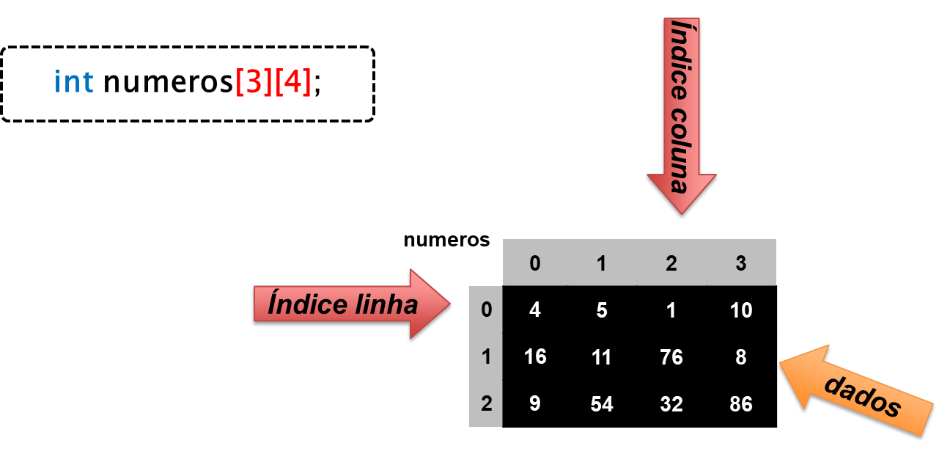
Olá, meus estudantes, sejam bem-vindos à nossa aula de Algoritmo e Programação em Java do Instituto da Oportunidade Social. Nessa aula, você vai conhecer como trabalhar com matrizes.

## Matrizes

Matriz é uma variável homogênea multidimensional e basicamente é um conjunto de variáveis do mesmo tipo, que possuem o mesmo identificador (nome) e são alocadas sequencialmente na memória. Uma matriz precisa de um índice para cada uma de suas dimensões.



Por exemplo, a matriz numeros tem 3 linhas e 4 colunas.



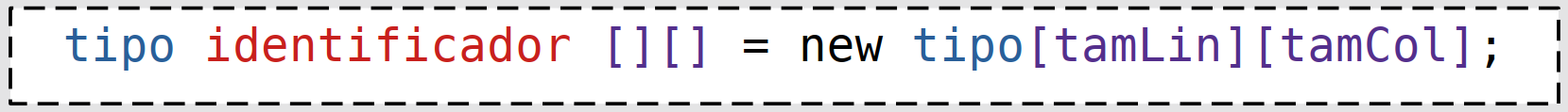
Por exemplo, na memória do computador podemos ter diversas variáveis como mostrado abaixo:

Text, table

Description automatically generated

A variável matriz possui 3 linhas e 5 colunas, ela pode armazenar 15 elementos, sendo que o primeiro elemento está na posição (0, 0). Lembre-se, o índice de uma matriz deve começar no valor zero.

Toda matriz deve ser representada por um tipo de dado, que será armazenado na matriz, um indicador, que o nome da matriz, e o tamanho, que a dimensão (linha e coluna) da matriz. A sintaxe de uma matriz é:



Por exemplo:

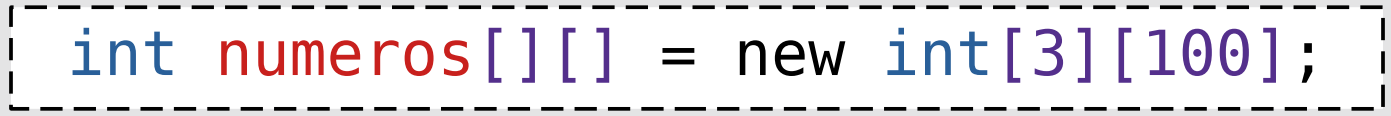


A variável numeros é do tipo double e possui duas linhas e cinco colunas e, com essa declaração o computador separa dez posições de memória sequenciais para armazenar os seus elementos:

A picture containing table

Description automatically generated

Vejamos outro exemplo:



Nesse caso, o computador separa 300 posições de memória para armazenar valores inteiros dentro da matriz numeros.

Table

Description automatically generated

Nós podemos também criar matrizes a partir de uma lista de valores entre chaves { } e separados por vírgula.

A picture containing chart

Description automatically generated

Nesse caso, a vírgula separa os elementos em colunas e cada grupo de chaves { } é uma nova linha. Então, a variável numeros tem 3 linhas e 4 colunas, a variável n2 tem 2 linhas e 2 colunas e a variável n3 tem 3 linhas e 3 colunas.

Nós acessamos os elementos da matriz pelo índice, que indica a posição do dado nela. Sempre quando estamos trabalhando com matriz bidimensional usaremos a linha e a coluna para identificar cada posição de um elemento.

Table

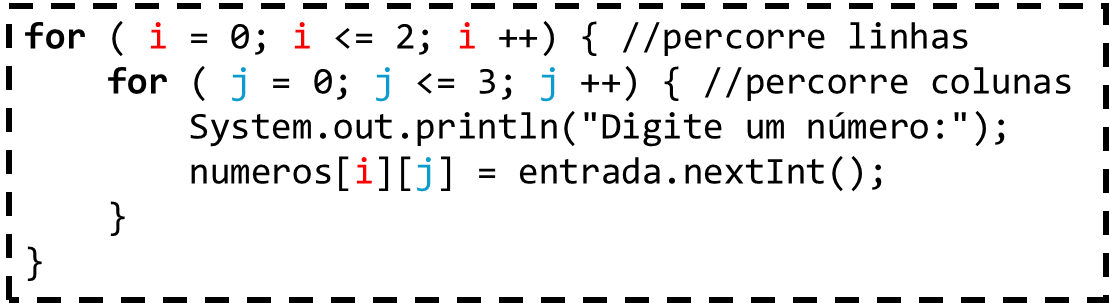
Description automatically generated

A ideia básica, para percorrer os elementos de uma matriz, é fixar a primeira linha e percorre todas as colunas, depois fixar na segunda linha e percorre todas as colunas. Dessa forma, conseguimos percorrer todos os elementos da matriz.

A picture containing diagram

Description automatically generated

Geralmente, para percorrer uma matriz utilizamos dois laços de repetição, o primeiro laço percorre a linha e o segundo laço percorre a coluna.



No primeiro laço **for**, a variável **i** inicia em 0 (que é o índice da primeira linha), então a próxima linha que será executada é o segundo laço **for**. Nesse segundo laço, a variável **j** inicia em zero (que a primeira coluna) e vai até 3 (que é a última coluna). Então, a variável **i** é incrementada para 1, o segundo laço reinicia novamente e a variável **j** inicia em zero (que a primeira coluna) e vai até 3 (que é a última coluna). Por fim, a variável **i** assume o valor 2 (que é a última linha) e a variável **j** inicia em zero (que a primeira coluna) e vai até 3 (que é a última coluna). Assim, percorrendo todos os elementos da matriz numeros.

Precisamos percorrer os elementos da matriz para preenchê-la e para ler os elementos dela caso quisermos imprimi-los, por exemplo.

Text

Description automatically generated

### Propriedade length usada em uma matriz

Vamos ver o seguinte exemplo:

Text, letter

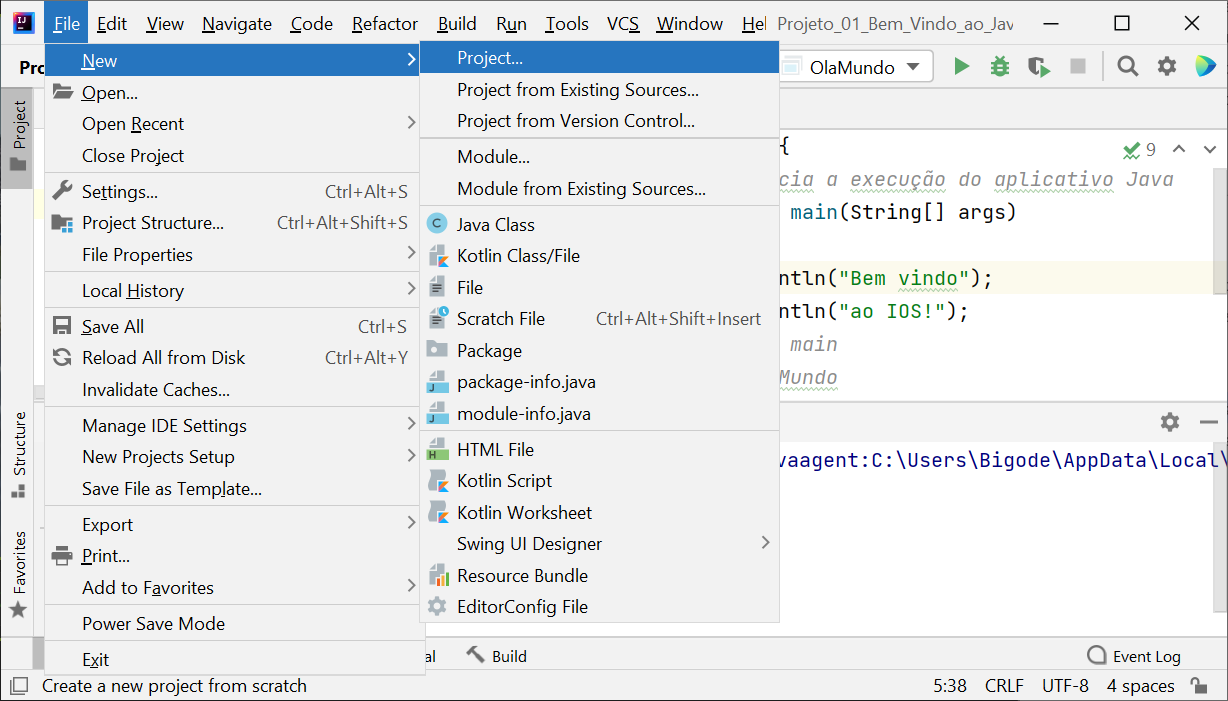
Description automatically generated

Nesse exemplo, a instrução **numeros.length** retorna a quantidade de linhas, que são 4 e a instrução **numeros[i].length** retorna a quantidade de colunas para cada linha.

### Vamos praticar

Vamos fazer um programa para criar uma matriz de inteiros já definidos na declaração da matriz e imprimir os elementos dessa matriz. Siga os passos para criar o projeto:

1. Crie um novo projeto no InterlliJ IDEA. Para isso, acesse o Menu **File**, a opção **New** e depois clique na opção **Project**.



1. As janelas seguintes são as mesmas descritas na criação de um projeto, portanto se tiver alguma dúvida consulte o material didático. Não se esqueça de dar um nome representativo para o projeto, por exemplo: Projeto\_31\_matriz
2. No diretório **src**, crie um arquivo para a classe. Não se esqueça de dar um nome representativo para o projeto, por exemplo: **Exemplo1Matriz** e insira o código mostrado na Figura 1.
3. **public** **class** Exemplo1Matriz {
4. **public** **static** **void** main(String[] args) {
5. // Declarar e inicializar a matriz
6. **int** numeros[][] = {{4, 5, 1, 10},{16, 11, 76, 8},{9, 54, 32, 89}};
7. **int** i, j;
8. **for**(i = 0; i < numeros.length; i++){
9. **for**(j = 0; j < numeros.length; j++){
10. System.out.print(numeros[i][j] + "\t");
11. }
12. System.out.println("");
13. }
14. }
15. }

Figura - Código do programa utilizando matriz.

1. Ao executar o código, o programa imprimir os valores da matriz em suas respectivas linhas e colunas.

Text

Description automatically generated

Na linha 6 da Figura 1, o comando for é usado para percorrer as linhas da matriz e na linha 7 da Figura 1 o segundo for é usado para percorrer as colunas da matriz.

Observe, que o comando **System.out.print** na linha 8 da Figura 1 possui um código de sequência de escape **\t**, que salta um espaço de tabulação separando os elementos da matriz. E na linha 10 da Figura 1, o comando **System.out.println** tem uma string vazia, pois esse comando provoca uma quebra de linha para separar as linhas da matriz.

### Vamos praticar II

Vamos fazer um programa para criar uma matriz de inteiros 2x2, preencher essa matriz com dados digitados no teclado, calcular o somatório das linhas da matriz e de todos os elementos dela e por fim imprimir os elementos digitados. Siga os passos para criar o projeto:

1. Crie um novo projeto no InterlliJ IDEA. Para isso, acesse o Menu **File**, a opção **New** e depois clique na opção **Project**.

Graphical user interface, text, application, email

Description automatically generated

1. As janelas seguintes são as mesmas descritas na criação de um projeto, portanto se tiver alguma dúvida consulte o material didático. Não se esqueça de dar um nome representativo para o projeto, por exemplo: Projeto\_32\_matriz
2. No diretório **src**, crie um arquivo para a classe. Não se esqueça de dar um nome representativo para o projeto, por exemplo: **Exemplo2Matriz** e insira o código mostrado na Figura 2.
3. **import** java.util.Scanner;
5. **public** **class** Exemplo2Matriz {
6. **public** **static** **void** main(String[] args) {
8. Scanner teclado = **new** Scanner(System.in);
10. // Declaração da matriz
11. **int**[][] numeros = **new** **int**[2][2];
12. **int** somaLinhas = 0, total = 0;
14. // Armazenar os dados
15. **for** (**int** i = 0; i < numeros.length; i++){
16. **for** (**int** j = 0; j < numeros[i].length; j++){
17. System.out.println("Digite um número para a posição [" + i + "][" + j + "]: ");
18. numeros[i][j] = teclado.nextInt();
19. }
20. }
22. // Precorrer os dados para calcular o somatório
23. **for** (**int** i = 0; i < numeros.length; i++){
24. somaLinhas = 0; // zera o acumulador de linhas toda vez que inicia uma nova linha
25. **for** (**int** j = 0; j < numeros[i].length; j++){
26. somaLinhas += numeros[i][j]; // acumula os somatório dos elemento de cada linha
27. }
28. System.out.println("A soma da linha [" + i + "] é: " + somaLinhas);
29. total += somaLinhas; // acumula o somatório total dos elementos da matriz
30. }
32. System.out.println("O somatório dos elementos da matriz é: " + total);
34. // Precorrer a matrix para imprimir os dados
35. System.out.println("=================================================");
36. System.out.println("|              Elementos da Matriz              |");
37. System.out.println("=================================================");
38. **for** (**int** i = 0; i < numeros.length; i++){
39. **for** (**int** j = 0; j < numeros[i].length; j++){
40. System.out.print(numeros[i][j] + "\t");
41. }
42. System.out.println("");
43. }
44. teclado.close();
45. }
46. }

Figura - Código do programa utilizando matriz.

1. Ao executar o programa, ele irá solicitar cinco valores inteiros, depois irá procurar o maior valor digitado e imprimi-lo na tela.

Graphical user interface, text, application, email

Description automatically generated

Nesse exemplo, utilizamos três laços de repetição: um para preencher a matriz com os valores digitados pelo usuário, outro para realizar o somatório das linhas e de todos os elementos da matriz e o terceiro para imprimir os valores do vetor.

## Exercícios de revisão

1. Criar e ler uma matriz 4 x 4, contar e imprimir quantos valores maiores que 10 ela possui. Imprimir os elementos da matriz no final.
2. Declare uma matriz 5 x 5. Gere uma matriz identidade, ou seja, preencha com 1 a diagonal principal e com 0 os demais elementos. Imprimir a matriz identidade no final.
3. Para cada conjunto de valores abaixo, escreva o programa, usando laço(s), que preencha um array com os valores:

|  |  |
| --- | --- |
|  |  |
|  |  |

1. Gere uma matriz de 100x100 elementos inteiros positivos menores ou iguais a 100.
   * Imprimir a matriz gerada.
   * Percorrer e matriz e substituir os elementos ímpares por -1 e os pares por 1.
   * Imprimir a matriz após a substituição.
2. **Desafio 01 – Distância entre cidades**: A distância entre várias cidades é dada pela tabela abaixo (em km):

Table

Description automatically generated

* + Implemente um programa que:
    - Armazene a tabela acima em uma matriz.
    - O programa não deve perguntar distâncias já informadas (por exemplo, se o usuário já forneceu a distância entre 1 e 3 não é necessário informar a distância entre 3 e 1, que é a mesma) e também não deve perguntar a distância de uma cidade para ela mesma, que é 0.
    - leia um percurso fornecido pelo usuário em um array unidimensional.
    - Calcule e mostre a distância percorrida.
  + Por exemplo:
    - dado o percurso 1, 2, 3, 2, 5, 1, 4 para a tabela mostrada como exemplo teremos: 15 + 10 + 10 + 28 + 12 + 5 = 80 km.

1. **Desafio 02 – Jogo da Velha**: Implementar o famoso jogo da velha usando uma matriz 3 x 3.
   * As jogadas devem ser alternadas entre o usuário e o computador.
   * O jogo deve solicitar um valor entre 1 e 9 para determinar a casa do tabuleiro que acontecerá a jogada.
   * O programa deve informar se aquela casa já está ocupada, se estiver, deve solicitar novamente a casa. O jogo deve solicitar também o valor ‘X’ ou ‘O’ para a jogada.
   * As jogadas devem ser alternadas e sempre iniciar pelo usuário seguido da jogada do computador.
   * O jogo deve indicar quem é o vencedor da jogada ou se deu velha (empate)
   * As jogadas do computador devem ser geradas de forma aleatória.
   * O programa deve permitir várias jogadas.

Table, calendar

Description automatically generated