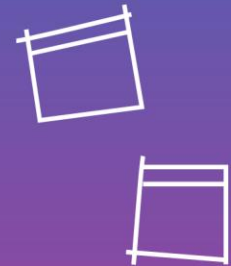




 **VEM SER**
DBC



Vetores, Matrizes e Repetições

Sumário

- Vetores
- Matrizes
- Comando de Repetição
 - For
 - Foreach
 - While

Vetores (Arrays)

- Um *array* é uma coleção de um ou mais objetos, do mesmo tipo, armazenados em endereços adjacentes de memória.

Vetores (Arrays)

- Um *array* é uma coleção de um ou mais objetos, do mesmo tipo, armazenados em endereços adjacentes de memória.
- Cada objeto é chamado de *elemento* do array.

Vetores (Arrays)

- Um *array* é uma coleção de um ou mais objetos, do mesmo tipo, armazenados em endereços adjacentes de memória.
- Cada objeto é chamado de *elemento* do array.
- Da mesma forma que para variáveis simples, damos um nome ao array.

Vetores (Arrays)

- Um *array* é uma coleção de um ou mais objetos, do mesmo tipo, armazenados em endereços adjacentes de memória.
- Cada objeto é chamado de *elemento* do array.
- Da mesma forma que para variáveis simples, damos um nome ao array.
- O tamanho do array é o seu número de elementos.

Vetores (Arrays)

- Um *array* é uma coleção de um ou mais objetos, do mesmo tipo, armazenados em endereços adjacentes de memória.
- Cada objeto é chamado de *elemento* do array.
- Da mesma forma que para variáveis simples, damos um nome ao array.
- O tamanho do array é o seu número de elementos.
- Cada elemento do array é numerado, usando um inteiro chamado de *índice*.

Vetores (Arrays)

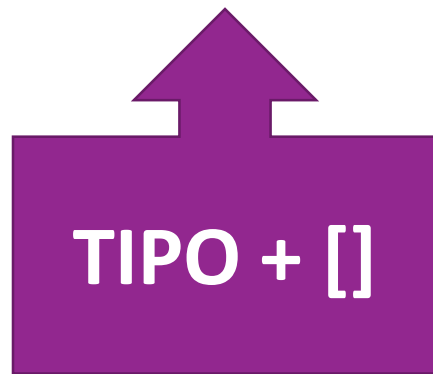
- Um *array* é uma coleção de um ou mais objetos, do mesmo tipo, armazenados em endereços adjacentes de memória.
- Cada objeto é chamado de *elemento* do array.
- Da mesma forma que para variáveis simples, damos um nome ao array.
- O tamanho do array é o seu número de elementos.
- Cada elemento do array é numerado, usando um inteiro chamado de *índice*.
- No Java, o primeiro índice começa em 0 (zero).

Vetores (Arrays)

```
int[] numeros = new int[3];
```

Vetores (Arrays)

```
int[] numeros = new int[3];
```



Vetores (Arrays)

```
int[] numeros = new int[3];
```



Vetores (Arrays)

```
int[] numeros = new int[3];
```

↑
TIPO + []

↑
NOME

↑
ALOCAR ESPAÇO NA MEMÓRIA

Vetores (Arrays)

```
int[] numeros = new int[3];
```

```
numeros[0] = 29;
```

```
numeros[1] = 77;
```

```
numeros[2] = 14;
```

Vetores (Arrays)

```
int[] vet = { 2,3,5,7,11,13 };
```

Let's practice;

Exercício #1

- Crie um vetor de 3 valores inteiros e:
 - Calcule a soma dos valores
 - Calcule a média dos valores

Matrizes

- Matrizes são estruturas numéricas em formato de tabela.

<https://www.stoodi.com.br/blog/matematica/matriz/>

Matrizes

- Matrizes são estruturas numéricas em formato de tabela.
- É um vetor de duas dimensões.

<https://www.stoodi.com.br/blog/matematica/matriz/>

Matrizes

- Matrizes são estruturas numéricas em formato de tabela.
- É um vetor de duas dimensões.
- Constituídas por linhas e colunas.

<https://www.stoodi.com.br/blog/matematica/matriz/>

Matrizes

- Matrizes são estruturas numéricas em formato de tabela.
- É um vetor de duas dimensões.
- Constituídas por linhas e colunas.
- Todos os valores dentro de uma matriz são do mesmo tipo.

<https://www.stoodi.com.br/blog/matematica/matriz/>

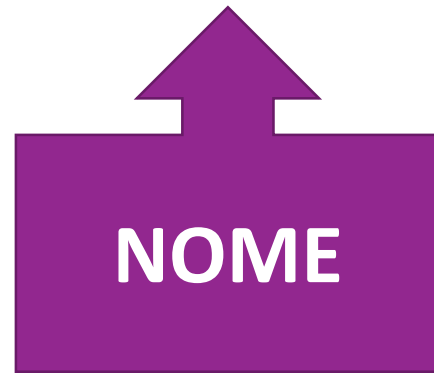
Matrizes

```
int[][] numeros = new int[2][2];
```



Matrizes

```
int[][] numeros = new int[2][2];
```



Matrizes

```
int[][] numeros = new int[2][2];
```

TIPO + [][]

NOME

ALOCAR ESPAÇO NA MEMÓRIA

Matrizes

```
int[][] numeros = new int[2][2];
numeros[0][0] = 29;
numeros[0][1] = 77;
numeros[1][0] = 14;
numeros[1][1] = 67;
```


Matrizes

```
int[][] numeros = new int[2][2];
```

```
numeros[0][0] = 29;
```

```
numeros[0][1] = 77;
```

```
numeros[1][0] = 14;
```

```
numeros[1][1] = 67;
```

x/y	0	1
0	29	77
1	14	67

Let's practice;

Exercício #2

- Crie uma matriz de 2x2, atribua valores e faça as seguintes operações:
 - Calcule a soma de todos os valores
 - Calcule a soma dos valores da primeira linha
 - Calcule a soma dos valores da segunda linha
 - Calcule (soma dos valores da primeira linha - soma dos valores da segunda linha)

Laço de Repetição For

```
for (int i = 0; i < 3; i++) {  
    <comandos>;  
}
```

Laço de Repetição For

variável de controle

```
for (int i = 0; i < 3; i++) {  
    <comandos>;  
}
```

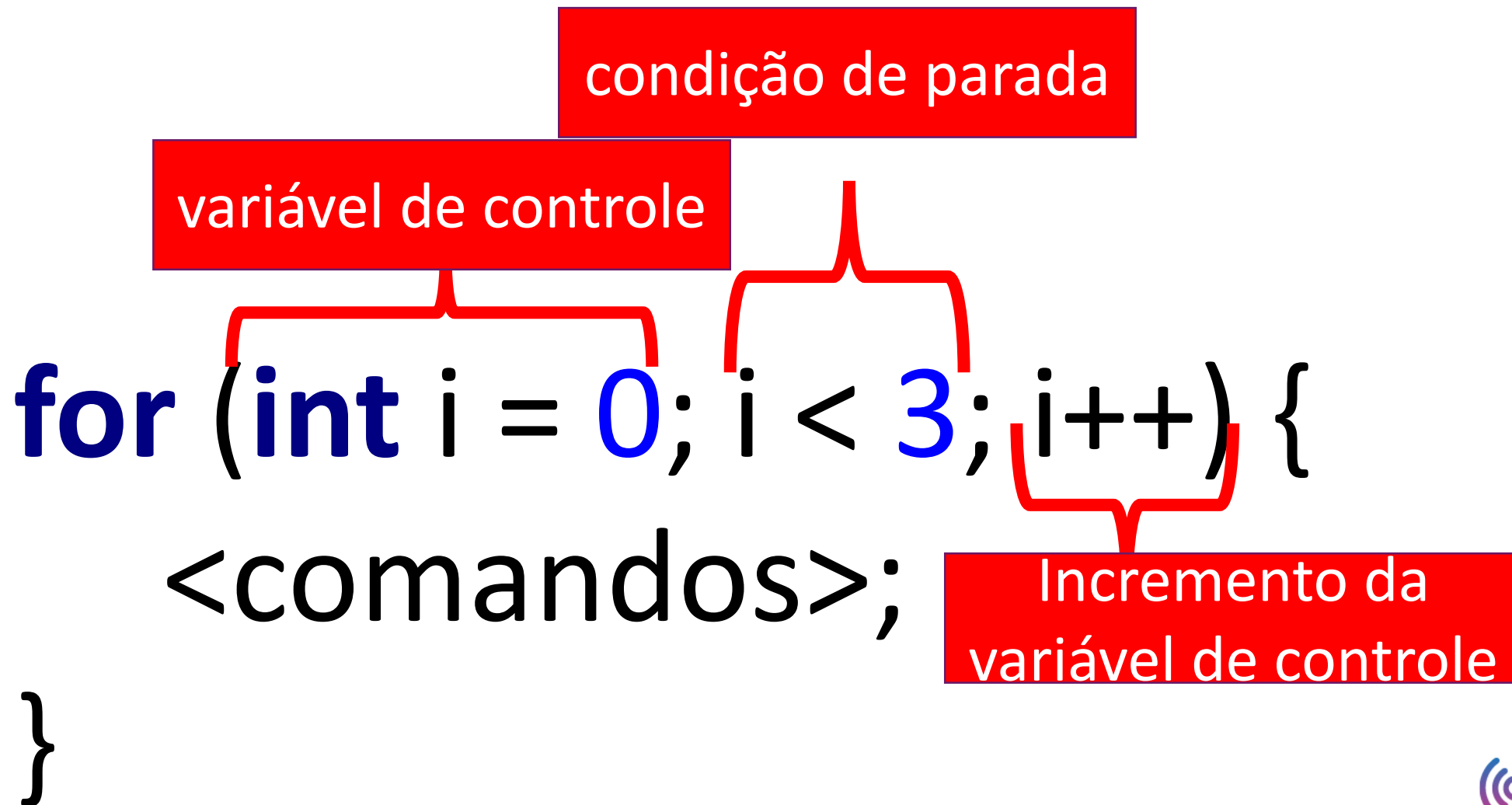
Laço de Repetição For

condição de parada

variável de controle

```
for (int i = 0; i < 3; i++) {  
    <comandos>;  
}
```

Laço de Repetição For



Laço de Repetição For

```
int[] valores = { 2,3,5,7,11, 13 };
for (int i = 0; i < valores.length; i++) {
    System.out.println("posição=" + i + " | valor="+valores[i]);
}
```


Laço de Repetição **Foreach**

```
for (int valor : valores) {  
    <comandos>;  
}
```

Laço de Repetição Foreach

variável de controle

```
for (int valor : valores) {  
    <comandos>;  
}
```

Laço de Repetição Foreach

variável de controle

```
for (int valor : valores) {
    <comandos>;
}
```

Array ou lista que irá ser percorrida

Laço de Repetição **Foreach**

```
int[] valores = {2, 3, 5, 7, 11, 13};  
for (int valor : valores) {  
    System.out.println(valor);  
}
```

Let's practice;

Exercício #3

- Crie um vetor de inteiros com quantidade X de elementos inteiros (solicitada ao usuário) e para cada posição desse vetor, preencha com entradas do usuário.
 - Solicite ao usuário a quantidade de elementos desejada
 - Preencha o vetor
 - Calcule a média dos valores do vetor.

Laço de Repetição **While**

```
while (<condição>){  
    <comandos>;  
}
```

Laço de Repetição **While**

Irá repetir até que a condição seja verdadeira

```
while (<condição>){  
    <comandos>;  
}
```


Laço de Repetição **While**

```

int numero = 0;
while (numero < 3) {
    System.out.println(numero);
    numero++;
}
  
```

Let's practice;

Exercício #4

- Crie um programa que solicite e imprima palavras ao usuário, até que ele digite “fim”
 - Peça uma palavra
 - Imprima a palavra
 - Se a palavra for = “fim” pare de pedir as palavras e encerre o programa

#Task

- Criar um projeto chamado “lista2” na pasta “modulo1” e fazer a lista de exercícios 2