



Alisson RS

[www.alissonrs.com.br](http://www.alissonrs.com.br)

[Moodle](#) 



# Linguagem de programação II



## Estruturas de dados homogêneas - Array e ArrayList

# Estruturas de dados homogêneas - Array

---

- Grupo de variáveis que contem valores do mesmo tipo.
- Grupo com acesso por meio de um único nome.
  - Elementos acessados por índice.
- São objetos
  - Criados com a palavra reservada **new** ou os valores que irá conter.
- Tem tamanho fixo
- Na fonte
  - <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html>



# Estruturas de dados homogêneas - Array

- Declaração:
  - Tipo [] nome; (int [] c;)
  - Tipo nome[]; (int c[]);
- Tipo podem ser valores primitivos ou classes.
  - Double [] valores;
  - Movel [] bolas;
- Criação ou instanciação:
  - nome = new tipo[quantidade de elementos]
    - (bolas = new Movel[3])
  - nome = {item1, item2,item3}
    - (c = {4,5,8,5,9} )

Seja coerente: use os nomes no plural.

# Estruturas de dados homogêneas - Array

```
1 package br.com.alissonrs.aula13;
2
3 public class ExmploArray1 {
4
5     public static void main(String[] args) {
6         int[] valores = new int[4];
7
8         valores[1] = 8;
9
10        valores[3] = 19;
11
12        for (int i = 0; i < valores.length; i++) {
13            System.out.println(valores[i]);
14        }
15
16    }
17
18 }
```

<terminated> ExmploArray1 [Java Application]

0  
8  
0  
19

# Estruturas de dados homogêneas - Array

## Iterador avançado.

- Permite percorrer o array de forma segura.
  - Não ir além dos limites.
- Facilita o entendimento do código.
- **Não** pode ser usado para alterar os valores do array.
- **Não** permite saber o índice do elemento.

```
1 package br.com.alissonrs.aula13;
2
3 public class ExmploArray2 {
4
5     public static void main(String[] args) {
6         int[] valores = new int[4];
7
8         valores[1] = 8;
9
10        valores[3] = 19;
11
12        for (int valor : valores) {
13            System.out.println(valor);
14        }
15
16    }
17
18 }
```

# Estruturas de dados homogêneas - Array

## Array multidimensional

```
1 package br.com.alissonrs.aula13;
2
3 public class ExmploArray2 {
4
5     public static void main(String[] args) {
6         int[][] valores = new int[4][2];
7
8         // Inserir valores
9         for (int i = 0; i < valores.length; i++) {
10             for (int j = 0; j < valores[i].length; j++) {
11                 valores[i][j] = i * 10 + j;
12             }
13         }
14
15         // Mostrar valores
16         for (int i = 0; i < valores.length; i++) {
17             for (int j = 0; j < valores[i].length; j++) {
18                 System.out.printf("Valores[%d][%d]=%0,2d \n",
19                                     i, j, valores[i][j]);
20             }
21         }
22
23     }
24
25 }
```

	Coluna 0	Coluna 1	Coluna 2	Coluna 3
Linha 0	a[ 0 ][ 0 ]	a[ 0 ][ 1 ]	a[ 0 ][ 2 ]	a[ 0 ][ 3 ]
Linha 1	a[ 1 ][ 0 ]	a[ 1 ][ 1 ]	a[ 1 ][ 2 ]	a[ 1 ][ 3 ]
Linha 2	a[ 2 ][ 0 ]	a[ 2 ][ 1 ]	a[ 2 ][ 2 ]	a[ 2 ][ 3 ]

Índice de coluna  
Índice de linha  
Nome do array

```
<terminated> ExmploArray2 [Java Application]
Valores[0][0]=00
Valores[0][1]=01
Valores[1][0]=10
Valores[1][1]=11
Valores[2][0]=20
Valores[2][1]=21
Valores[3][0]=30
Valores[3][1]=31
```



# Estruturas de dados homogêneas - Array

---

## Passagem de array para métodos.

- Array
  - `double horasTrabalhadas = new double[24];`
- Método
  - `void modificaArray(double [] b) {...}`
- Chamada
  - `modificaArray(horasTrabalhadas)`

# Estruturas de dados homogêneas - Array

## Classe auxiliar Arrays

- Define métodos estáticos para manipulação de arrays.
  - sort – ordena o array.
    - `Arrays.sort(valores)`
  - binarySearch – Procura um item no array.
    - `Arrays.binarySearch(valores,19)`
  - equals – compara dois arrays.
    - `Arrays.equals (valores, outroarray)`
  - fill – preenche o array com o elemento indicado.
    - `Arrays.fill(valores, 15)`
  - arraycopy – copia os elementos de uma array para outro array.
    - `Arrays.arraycopy (valores, outroarray)`
- <https://docs.oracle.com/javase/7/docs/api/java/util/Arrays.html>



# Estruturas de dados homogêneas - Array

## Lista de argumentos de comprimento variável

```
1 package br.com.alissonrs.aula13;
2
3 public class ExemploNumeroArgumentosVariado {
4
5     public static void main(String[] args) {
6         double d1 = 10.0;
7         double d2 = 20.0;
8         double d3 = 30.0;
9         double d4 = 40.0;
10        System.out.printf("d1 = %.1f; d2 = %.1f; d3 = %.1f\n", d1, d2, d3);
11        System.out.printf("Media de d1 e d2 é %.1f\n", media(d1, d2));
12        System.out.printf("Media de d1, d2 e d3 é %.1f\n", media(d1, d2, d3));
13        System.out.printf("Media de d1,d2,d3 e d4 é %.1f\n",
14            media(d1, d2, d3, d4));
15    }
16
17    public static double media(double... numeros) {
18        double total = 0.0;
19        for (double numero : numeros) {
20            total += numero;
21        }
22        return total / numeros.length;
23    }
24 }
```

- Somente 1 por método.
- Sempre o último argumento.

# Estruturas de dados homogêneas - ArrayList

## Arrays avançados em Java - ArrayList

- Estruturas de controle de coleções.

```
1 package br.com.alissonrs.aula13;
2
3 import java.util.ArrayList;
4
5 public class ExemploArrayList {
6
7     public static void main(String[] args) {
8
9         ArrayList <String> palavras = new ArrayList<String>();
10
11         palavras.add("Alisson");
12         palavras.add("Gustavo");
13         palavras.add("Elizabeth");
14         palavras.add(0, "Carlos");
15
16         for (String palavra: palavras) {
17             System.out.println(palavra);
18         }
19
20         System.out.println(palavras.toString());
21     }
22
23 }
```

```
<terminated> ExemploArrayList [Java Application] C:\Program Files\Java
Carlos
Alisson
Gustavo
Elizabeth
[Carlos, Alisson, Gustavo, Elizabeth]
```



[www.alissonrs.com.br](http://www.alissonrs.com.br)

**Alisson RS**

