

POO - PROGRAMAÇÃO ORIENTADA A OBJETOS

Vetores e Matrizes(Array's)

Rodrigo R Silva

Instituto Federal de Educação, Ciência e Tecnologia Sul-Rio-Grandense
Campus Bagé

Nesta Aula Veremos...

1 Array

Contextualizando...

- Uma instituição bancária trabalha com um conjunto de contas ao mesmo tempo.
- Como representar um conjunto de contas?
- Podemos criar conjuntos em nossas aplicações com o uso de vetores ou matrizes.
- **Vetores**: conjunto de dados guardados em uma única dimensão.
- **Matrizes**: conjunto de dados guardados em duas ou mais dimensões.
- Vetores e matrizes em Java também são denominados **Arrays** (listas, arranjos de dados).

Array

Definição

- Uma **array** é um container de objetos com um número fixo de elementos de mesmo tipo de dado. O tamanho de uma array é definido no momento da sua criação.
 - O numero de elementos é imutável.
 - Guardam sempre o mesmo tipo de dado.
- **Arrays** também são objetos em Java. Portanto, devem ser declaradas e instanciadas para serem utilizadas.

Declaração

- A declaração de uma array pode ser realizada de duas formas:

Ex:

```
String[] nomes; //Declara uma array de Strings.
```

```
String nomes[];
```

- A segunda forma permite uma única linha de declaração de variáveis do mesmo tipo, mesmo que não sejam arrays.

Ex:

```
String nome, nomes[], cidade;
```

Instanciação

- Após a declaração, a array deve ser instanciada como qualquer outro objeto.
- Na instanciação de uma array definimos o número de elementos que ela conterá.
Ex:
`String[] nomes;`
`nomes = new String[10];`
Ou: `String[] nomes = new String[10];`
- Podemos definir os elementos no momento da declaração:
Ex:
`String[] nomes = "Fulano", "Ciclano", "Beltrano";`

Referência a elementos

- Cada elemento de uma array é referenciado por um índice numérico inteiro o qual representa sua posição no interior do container.
- O primeiro elemento possui índice zero (0).
- O último elemento possui índice $n-1$, onde n é o número de elementos da array.

Ex:

```
String nomes[] = new String[10];  
nomes[0] = "Fulano";  
nomes[1] = "Beltrano";  
nomes[2] = "Ciclano";
```


O atributo length

- O atributo length retorna o número de elementos de uma matriz.

Ex:

```
String nomes[] = new String[10];
```

```
System.out.println("Tamanho: " + nomes.length);
```

- Percorrendo os elementos da matriz:

```
String nomes[] = new String[10];
```

```
for(int x = 0; x < nomes.length; x++)
```

```
System.out.println("nomes[" + x + "]: " + nomes[x]);
```

ForEach

- Podemos percorrer os elementos de um array usando um laço de iteração denominado foreach.
- Trata-se de uma alternativa de utilização do laço **for** para estruturas do tipo listas de dados.

Ex:

```
int x = 0;
```

```
String nomes[] = new String[10];
```

```
for(String nome: nomes)
```

```
System.out.println("nomes[" + x++ + "]: " + nome);
```

A classe Arrays

- Java define uma classe denominada Arrays a qual contém métodos de classe úteis para operações com arrays.
- **Arrays.toString():** podemos usar este método para apresentar os dados de uma array.

Ex:

```
String[] nomes = "Fulano", "Beltrano", "Ciclano";  
System.out.println(Arrays.toString(nomes));
```

- **Arrays.sort():** ordena os elementos de uma array.

Ex:

```
Arrays.sort(nomes);  
System.out.println(nomes);
```

A classe Arrays

- **Arrays.fill():** preenche a array com os valores.

Ex:

```
String[] nomes = new String[10];
```

```
Arrays.fill(nomes, "aluno");
```

- **Arrays.binarySearch():** realiza uma busca por um elemento no interior da array.

Ex:

```
int indice;
```

```
indice = Arrays.binarySearch(nomes1, "Ciclano");
```

```
System.out.println("Posição do elemento: " + indice);
```

OBRIGADO!