

O que são Arrays?

Os **arrays** ou **matrizes**, como são conhecidos pelo Java, fazem parte do pacote `java.util` na coleção da API do **Java**. São objetos de recipientes que contém um número fixo de valores de um único tipo. O comprimento de um **array** é estabelecido quando criado, sendo que após a criação o seu comprimento fica fixo.

Cada item em um **array** é chamado de elemento, e cada elemento é acessado pelo número, o índice. Abaixo é mostrado se dá esse acesso aos seus elementos, lembrando que sempre sua numeração começa em 0.

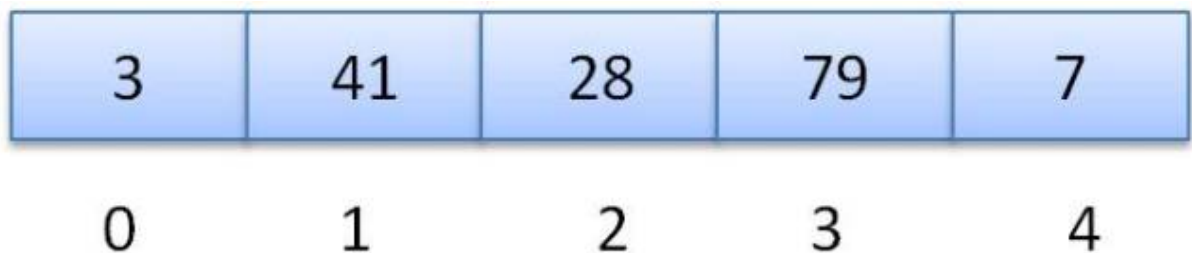


Figura 1: Um array de 5 elementos

Declarando Arrays

Na declaração de um array, cada elemento recebe um valor padrão, sendo 0 (**zero**) para números de tipo primitivo, falso (**false**) para elementos booleanos e nulo (**null**) para referências. O programa que está na Listagem 1, cria um array de inteiros, coloca alguns valores nela, e imprime cada valor à saída padrão.

Listagem 1: Declaração de arrays.

```
public class Declaracao_Array {  
    public static void main(String[] args) {  
        //[ ] - são inseridos em uma variável que referencia um array  
        int[] a = new int[4];  
        //OUTRA MANEIRA DE FAZER UMA DECLARAÇÃO DE ARRAY  
        int[] b;  
        b = new int[10];  
        //DECLARANDO VÁRIOS ARRAYS  
        int[] r = new int[44], k = new int[23];  
  
        //{ } - inicializar valores em um array sua declaração  
        int[] iniciaValores = {12,32,54,6,8,89,64,64,6};  
    }  
}
```

```
//DECLARA UM ARRAY DE INTEIROS
    int[] meuArray;

    //ALOCA MEMÓRIA PARA 10 INTEIROS
    meuArray = new int[10];

    //INICIALIZA O PRIMEIRO ELEMENTO
    meuArray [0] = 100;
    meuArray [1] = 85;
    meuArray [2] = 88;
    meuArray [3] = 93;
    meuArray [4] = 123;
    meuArray [5] = 952;
    meuArray [6] = 344;
    meuArray [7] = 233;
    meuArray [8] = 622;
    meuArray [9] = 8522;
    //meuArray [10] = 564; //ESTOURA A PILHA POIS NÃO EXISTE O ÍNDICE 10

    System.out.println(meuArray[9]);
    System.out.println(meuArray[2]);
}
}
```

Descobrimo o tamanho de um array

Por padrão, cada array sabe seu próprio tamanho, independente de quantos valores forem inseridos. O array armazena na variável de instância o método `length`, que retorna o tamanho do array especificado, conforme o exemplo da Listagem 2.

Saiba mais: [Cursos de Java](#)

Listagem 2: Verificando o tamanho de uma array.

```
public class TamanhoArray {

    public static void main(String[] args) {
        int[] arrayUm = {12,3,5,68,9,6,73,44,456,65,321};
```

```

int[] arrayDois = {43,42,4,8,55,21,2,45};

if(arrayDois.length > 8){
    System.out.println("Tamanho do ArrayDois - Maior que 8!");
}else{
    System.out.println("Tamanho do ArrayDois - Menor que 8!");
}
System.out.println("\nTamanho do ArrayUm = "+arrayUm.length);
}
}

```

Em Java, não é possível criar um array sem declarar seu tamanho, pois os arrays têm um tamanho fixo que deve ser definido no momento da criação. No entanto, você pode usar uma List (como ArrayList) que é uma estrutura de dados dinâmica e não requer a definição de um tamanho inicial. A ArrayList pode crescer automaticamente conforme você adiciona elementos.

Aqui está um exemplo de como usar ArrayList:

```

import java.util.ArrayList;

public class Exemplo {
    public static void main(String[] args) {
        // Criando uma ArrayList sem definir o tamanho inicial
        ArrayList<String> lista = new ArrayList<>();

        // Adicionando elementos à lista
        lista.add("Elemento 1");
        lista.add("Elemento 2");
        lista.add("Elemento 3");

        // Acessando elementos da lista
        for (String elemento : lista) {
            System.out.println(elemento);
        }

        System.out.println("Apenas a posição de 0:" + lista.get(0));

        // Tamanho da lista
        System.out.println("Tamanho da lista: " + lista.size());
    }
}

```