

Java

Vetores

Marcia Aguená - Março 2023

Array

Definição

- Um **array** ou **vetor** é uma **lista de itens similares**, acessíveis via um **índice** que representa a posição do item que queremos obter.
- Um array deve sempre ter:
 - Um **nome**
 - Um **tipo** de dado
 - Um **tamanho**

Array

Definição

- O tamanho de um **array** não pode ser alterado depois de sua criação, ou seja, sempre tem o **tamanho fixo**;
- Em Java, a primeira posição do vetor é sempre no índice de valor **0**. Assim, se criarmos um array de 5 posições, os respectivos índices de acesso são 0, 1, 2, 3 e 4.

Array

Inicializando

- Como **declarar** um array:

```
int myArray[];
```

```
int[] myArray; //também funciona
```

- Como **criar** um array:

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

- Como **declarar** e **criar** um array:

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

```
int[] myArray = new int[3]; //também funciona
```

Array

Inicializando

- Como **declarar, criar e inicializar** um array:

```
int myArray[] = new int[]{1,2,3};
```

```
int[] myArray = new int[]{1,2,3};
```

```
int[] myArray = {1,2,3};
```

- Como acessar uma **posição** em um array:

```
int myValue = myArray[2];
```

- Como obter o **tamanho** de um array:

```
myArray.length;
```

Array

Exemplo

```
1 v public class ArrayPessoas {  
2 v     public static void main (String[] args) {  
3         String[] pessoas = new String[]{"Luke", "Lea", "Hans", "Lando"};  
4 v         for(int i=0; i< pessoas.length ; i++){  
5             System.out.println(pessoas[i]);  
6         }  
7     }  
8 }
```

Array

Exercícios

- Para que serve o vetor de `String` chamado `args` passado ao método `public static void main` de toda classe principal em Java?
- Altere a classe **Soma** para que:
 - Some todos os números inteiros passados como parâmetro na linha de comando;
 - Deixe o tipo da variável `soma` como `short` e execute com valores que caibam dentro do limite da variável;
 - Teste agora valores que estourem o limite da variável `soma`;
- Dica: para converter uma `String` para um `short`, use `Short.parseShort("10");`

Ponteiros e Variáveis Dinâmicas

Definição

- Ponteiros são variáveis que guardam **referências** a uma **posição de memória**;
- Tipos em Java:
 - Primitivos (`short`, `int`, `long`, `char`, `float`...);
 - Referência (`class`, `interface`, `array`...).
- Tipos de Referência só podem ser instanciados através de **ponteiros implícitos**.

Ponteiros e Variáveis Dinâmicas

Definição

- Tipos de Referência só podem ser criados com o comando `new`;
- Todo gerenciamento de memória é feito **automaticamente** em Java, ou seja, não é possível:
 - Realizar aritmética de ponteiros;
 - Liberar memória.

Ponteiros e Variáveis Dinâmicas

Definição

- Tipos de Referência só podem ser criados com o comando `new`;
- Todo gerenciamento de memória é feito **automaticamente** em Java, ou seja, não é possível:
 - Realizar aritmética de ponteiros;
 - Liberar memória.

Ponteiros e Variáveis Dinâmicas

Definição

- A liberação de memória é feita pelo **Coletor de Lixo** (Garbage Collector):
 - A linguagem torna-se mais **segura**;
 - Consome **recursos**;
 - Porém o Coletor de Lixo é **não-determinístico**. É difícil prever ou forçar o momento de sua atuação.