PROGRAMA MONITORIA

Programação Java - Arrays

```
for (inicialização; condição; incremento/decremento) {
    // Bloco de código a ser executado
}
```

```
for (int i = 0; i < 10; i++) {
    System.out.println(i);
}
```

Java - Loop For

O que é e como usar?

O **loop for** é uma estrutura de controle que permite executar um bloco de código **repetidamente** em um determinado número de vezes.

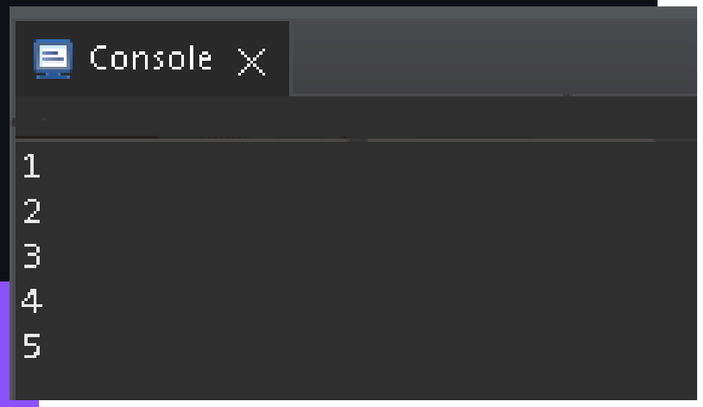
01 **Inicialização**: é uma expressão que é executada apenas uma vez no início do loop e é usada para inicializar a variável de controle do loop.

02 **Condição**: é uma expressão booleana que é verificada a cada loop. Se a condição for verdadeira, o bloco de código é executado. Caso contrário, o loop é encerrado.

01 **Incremento/decremento:** é uma expressão que é executada no final de cada iteração do loop e é usada para atualizar a variável de controle do loop.

```java
public class ExemploLoopFor {

    public static void main(String[] args) {
        // Usando o loop for para imprimir os números de 1 a 5
        for (int i = 1; i <= 5; i++) {
            System.out.println(i);
        }
    }
}
```



Exemplo Loop For

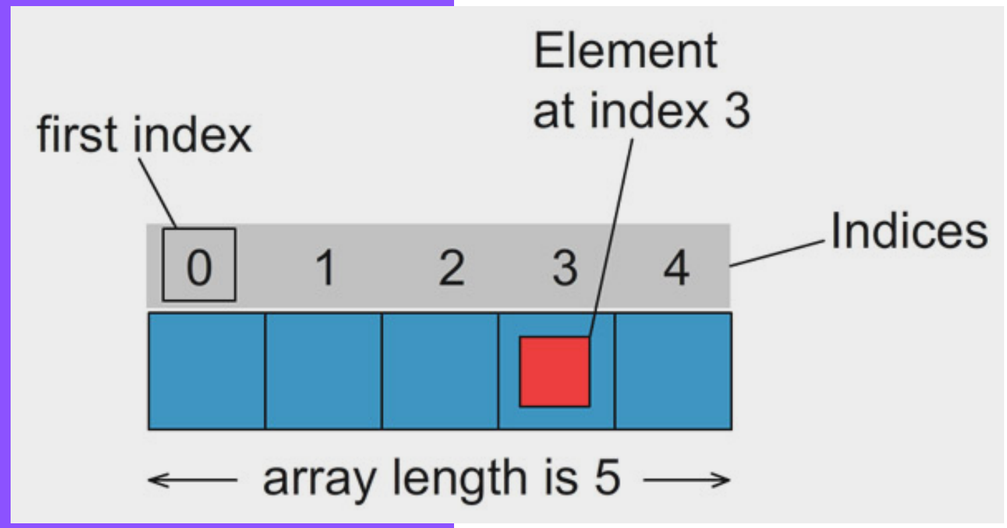
01 Nesse exemplo, o loop for é usado para executar o bloco de código que imprime o**s números de 1 a 5**.

02 A variável de controle i é inicializada com o **valor 1**, a condição é verificada **(i <= 5)** e, como é verdadeira, o bloco de código é executado.

03 Em seguida, a variável i é **incrementada em 1** (aumenta +1) e a condição é verificada novamente.

04 O loop é executado até que a condição seja falsa **(i = 6)** e, em seguida, é encerrado.

05 Como resultado, o programa **imprime** os números 1, 2, 3, 4 e 5 na tela.




```
int arr[] = new int[5];
```


```
int arr[] = {42, 51, 63, 90, 87};
```

www.tutorialkart.com

O que é Array?

01 Um **Array** é uma estrutura de dados que permite armazenar um conjunto de valores do mesmo tipo em uma única variável.

02 Os elementos em um array são armazenados em posições numeradas chamadas **índices**.

03 O primeiro elemento é armazenado na **posição 0**, o segundo na **posição 1** e assim por diante, como mostra na primeira imagem.

04 O **tamanho** de um array é definido na sua criação e **não pode ser alterado** posteriormente.

05 Os tipos de dados suportados em um array em Java incluem tipos primitivos (**int**, **double**, **boolean**, etc.) e **objetos**.

```java
public class ExemploArray {

    public static void main(String[] args) {
        // Criando um array de inteiros com tamanho 5
        int[] numeros = new int[5];

        // Atribuindo valores aos elementos do array
        numeros[0] = 10;
        numeros[1] = 20;
        numeros[2] = 30;
        numeros[3] = 40;
        numeros[4] = 50;

        // Imprimindo o valor de um elemento do array
        System.out.println("O valor do terceiro elemento do array é: " + numeros[2]);

        // Alterando o valor de um elemento do array
        numeros[1] = 25;

        // Imprimindo o valor de todos os elementos do array
        System.out.println("Os valores do array são: ");
        for (int i = 0; i < numeros.length; i++) {
            System.out.println(numeros[i]);
        }
    }
}
```

Exemplo uso de Array

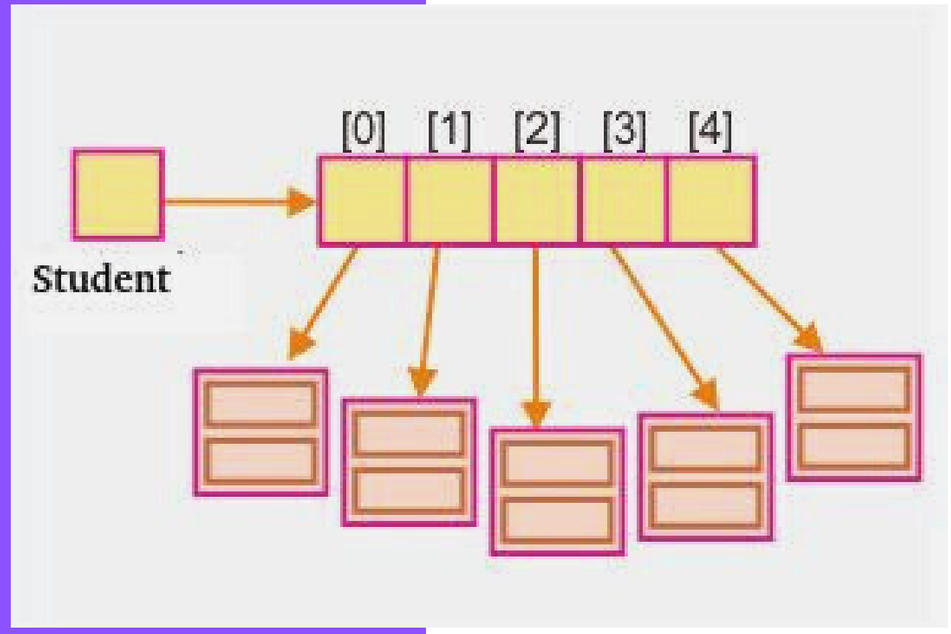
01 Nesse exemplo, criamos um array de **inteiros** com **tamanho 5**

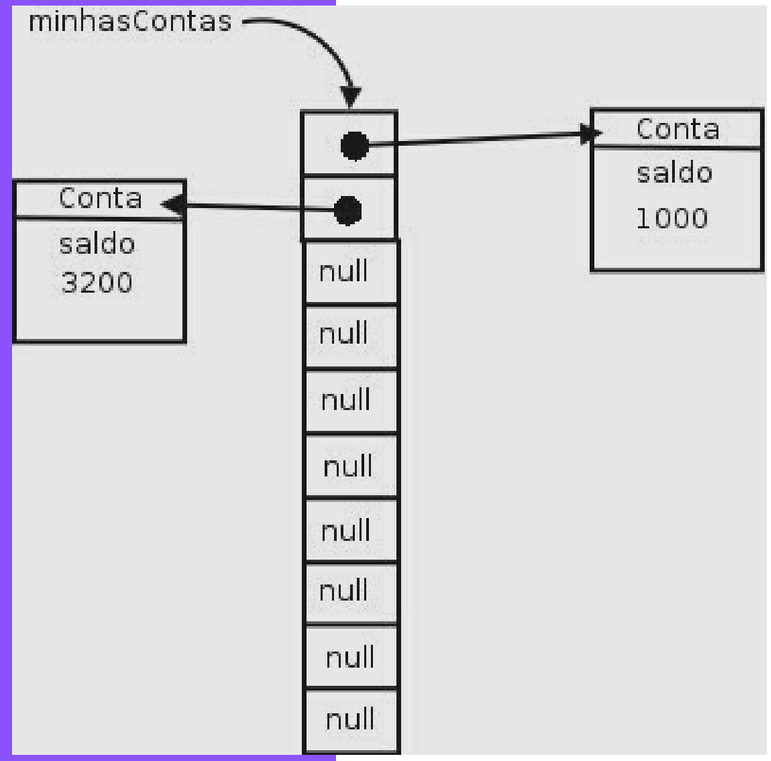
02 Atribuímos valores a cada um dos elementos do array

03 Depois, **imprimimos** o valor do terceiro elemento do array. Aparecerá o valor 30 na tela.

04 **Alteramos** o valor do segundo elemento, que antes era 20 e foi alterado para 25.

05 Por fim, imprimimos todos os valores do array usando um **loop for**.





Array de Objetos

01 **Array de objetos** em Java é uma **coleção de objetos** do mesmo tipo, armazenados em uma estrutura de array.

02 Para criar um array de objetos, é necessário criar um array com a palavra-chave ***new*** e especificar o **tamanho** do array.

03 Os elementos de um array de objetos podem ser acessados através do **índice** do array.

04 Cada elemento é uma referência a um objeto, que pode ser usado para chamar seus **métodos** e acessar seus **atributos**.

```java
class Pessoa {

    private String nome;
    private int idade;

    public Pessoa(String nome, int idade) {
        this.nome = nome;
        this.idade = idade;
    }

    public String getNome() {
        return nome;
    }

    public int getIdade() {
        return idade;
    }
}
```

```java
public class ExemploArrayObjetos {

    public static void main(String[] args) {

        // Cria um array de objetos do tipo "Pessoa"
        Pessoa[] pessoas = new Pessoa[3];

        // Inicializa cada elemento do array com uma nova instância de "Pessoa"
        pessoas[0] = new Pessoa("Isabela", 18);
        pessoas[1] = new Pessoa("Pedro", 21);
        pessoas[2] = new Pessoa("João", 24);

        // Percorre o array e imprime as informações de cada pessoa
        for (int i = 0; i < pessoas.length; i++) {
            System.out.println("Nome: " + pessoas[i].getNome()
                               + ", Idade: " + pessoas[i].getIdade());
        }
    }
}
```

Exemplo Array de Objetos

01 É criado um array de objetos do tipo **Pessoa**, que contém três elementos.

02 Cada elemento é inicializado com uma nova **instância**, que contém informações sobre uma pessoa específica (nome e idade).

03 Em seguida, o programa **percorre** o array utilizando o loop for

04 Para **cada elemento** do array, o loop for **imprime** as informações da pessoa (nome e idade) usando os métodos **acessores**

Resultado no console:

```
Nome: Isabela, Idade: 18
Nome: Pedro, Idade: 21
Nome: João, Idade: 24
```

Acesse os materiais no GitHub :)

Exemplos de atividades/códigos com tudo que envolve o novo assunto (Estrutura de Dados - Array)

https://github.com/IsabelaSilvaz/Monitoria_Java_POO