

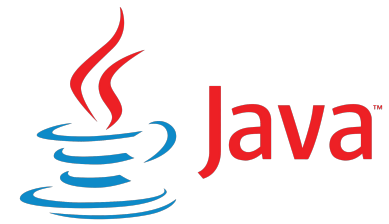
# Estruturas de controle

// Estruturas seletivas (condicionais) e de repetição

IT BOARDING

**BOOTCAMP**





# Índice



**01** Estruturas Seletivas  
(Condicionais)

**02** Estruturas de Repetição

**03** Exemplos

IT BOARDING

**BOOTCAMP**

# Estruturas Seletivas

IT BOARDING

**BOOTCAMP**





# Estrutura IF

A **estrutura de controle “if”** permite decidir entre duas opções resultantes da avaliação de uma declaração.

**Em Java**, é implementado usando a seguinte sintaxe:

Java

```
if (condicao){  
    . . .  
    . . .  
}
```



# Estrutura IF + ELSE

O **else** é a extensão da instrução IF. Significa “caso contrário” e permite a execução de um bloco de código **SOMENTE** se a condição da instrução IF for **falsa**.

**Em Java**, é implementado usando a seguinte sintaxe:

```
Java    if (condicao){  
        ...  
    }  
    else {  
        ...  
    }
```



## Exemplo IF + else

Java

```
int num;  
if (num>5){  
    System.out.println ("O número é maior que 5")  
}  
else {  
    System.out.println ("O número é menor que 5")  
}
```



# Estrutura IF + ELSE Múltiplos

As **estruturas if e else** permitem o **aninhamento**, ou seja, juntar várias condições if e else no caso de uma primeira condição avaliada não ser atendida.

**Em Java**, é implementado usando a seguinte sintaxe:

Java

```
if (condicao){  
    ...  
}  
else {  
    if (condicao){  
        ...  
    }  
    else {  
        ...  
    }  
}
```



# Estrutura Switch

A **estrutura do switch** permite vários caminhos possíveis a partir da avaliação de **uma única expressão / condição**.

Cada **início** do caminho é expresso pela palavra-chave **case** e cada **final** pela palavra de **break**.

Para atender às condições não contempladas nos casos, pode-se implementar um **default**.

Java

```
int tipoResul = 3;
String mensagem;

switch (tipoResul)
{
    case 1: mensagem = "Aprovado";
            break;
    case 2: mensagem = "Reprovado";
            break;
    case 3: mensagem = "Recuperação";
            break;
    default: mensagem = "Não fez a prova";
            break;
}

System.out.println(mensagem);
```



# Estruturas de Repetição

IT BOARDING

**BOOTCAMP**





# Estruturas de Repetição



Eles permitem que um bloco de código seja executado n várias vezes.



Cada repetição ou execução repetitiva de um bloco de código é conhecida como **loop**.



Existem **loops**:

- Controlado por um **contador**
- Controlado por uma **condição**



# do While

É uma **estrutura Java repetitiva** que permite a execução de um bloco de código quantas vezes forem necessárias dependendo de uma condição.

Possui as seguintes características:

- A **condição** é avaliada **no final** do bloco de código.
- Como a condição é avaliada por último, ela **sempre** é executada **pelo menos uma vez**.

Java

```
do{  
    ...  
    ...  
} while (condicao)
```



# While

É uma **estrutura Java repetitiva** que permite a execução de um bloco de código **desde que** uma determinada condição seja atendida.

Possui as seguintes características:

- A **condição** é avaliada **no início** do bloco de código.
- Como a condição é avaliada no início, ela é executada **apenas** se a condição **for atendida**.

Java

```
while (condicao){  
    ...  
    ...  
}
```



# For

O **for** é considerado uma estrutura controlada por contador em Java. Sua principal característica é que possui uma **variável de contador própria** que se encarrega de controlar o número de repetições que um bloco de código terá.

É composto por **3 partes**:

*Indica o valor no qual o contador é inicializado.  
Ele é executado apenas uma vez.*

*Indica quanto o contador aumentará cada vez que o loop for executado.*

```
Java for (inicializacao; condicao; incremento)
```

*Indica a condição que deve ser atendida para que o loop pare.*



# Exemplos

```
//do While
int cont = 0;
do {
    ...
    cont = cont + 1;
} while (cont <=10)
```

```
//while
int cont = 0;
while (cont <=10) {
    ...
    cont = cont + 1;
}
```

```
//for
for (int c=0; c<=10; c++) {
    ...
}
```





# Obrigado

IT BOARDING

**BOOTCAMP**

