
Programação Orientada a Objetos

— Comandos de Repetição —

Prof. Leandro Rodrigues Pinto
<leandrorodp@gmail.com>

Comandos de Repetição

Instruções de Repetição

- Permitem que um determinado trecho de código seja executado várias vezes.
- No java possuímos três instruções de repetição.

while

do-while

for

Comandos de Repetição

Instruções de Repetição

- Permitem que um determinado trecho de código seja executado várias vezes.
- No java possuímos três instruções de repetição.

while

do-while

for

Enquanto

Faça-Enquanto

Para

Comandos de Repetição

Instruções de Repetição

while

- A instrução de repetição **while** recebe como parâmetro uma condição.
- Enquanto essa condição for **verdadeira**, o bloco de comandos do corpo do laço será executado.
- A condição da instrução deve ser um valor do tipo boolean.
- **O teste da condição booleana vem antes de ser executado o laço.**

Comandos de Repetição

Instruções de Repetição

while

- Sintaxe do comando:

```
while(expressão) {  
    bloco de comandos;  
}
```

- Exemplo:

```
int num = 0;  
while(num <= 10) {  
    System.out.print(num + " ");  
    num += 1;  
}
```

Comandos de Repetição

Instruções de Repetição

do-while

- A instrução de repetição **do-while** recebe como parâmetro uma condição.
- Enquanto essa condição for **verdadeira**, o bloco de comandos do corpo do laço será executado.
- A condição da instrução deve ser um valor do tipo **boolean**.
- **O teste da condição booleana vem depois de ser executado o laço.**

Comandos de Repetição

Instruções de Repetição

do-while

- Sintaxe do comando:

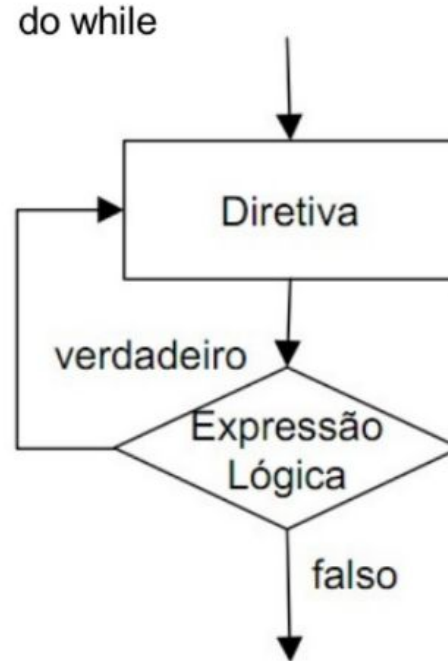
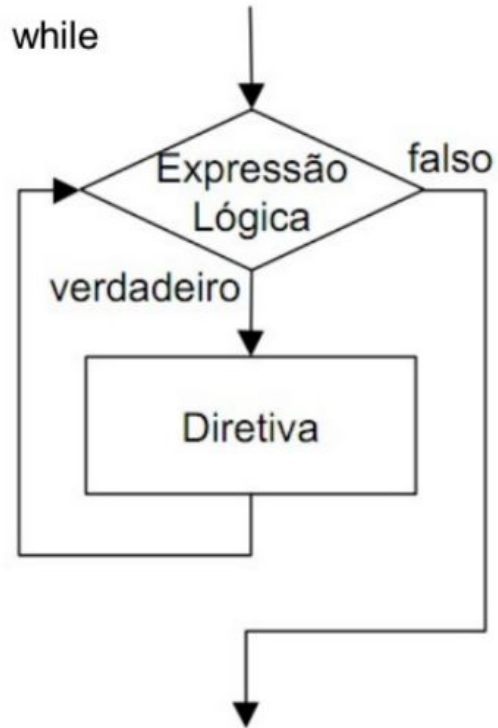
```
do {  
    bloco de comandos;  
} while(expressão);
```

- Exemplo:

```
int num = 0;  
do {  
    System.out.print(num + " ");  
    num = num + 1;  
} while(num <= 10);
```

Comandos de Repetição

while e do-while



Comandos de Repetição

Instruções de Repetição

for

- Outro comando de **loop** extremamente utilizado.
- A ideia é a mesma do **while** – fazer um trecho de código ser repetido enquanto uma condição continuar verdadeira.
- Possui também um espaço para inicialização e modificação das variáveis.
- Isso faz com que fiquem mais legíveis as variáveis relacionadas ao loop.

Comandos de Repetição

Instruções de Repetição

for

- Sintaxe do comando:

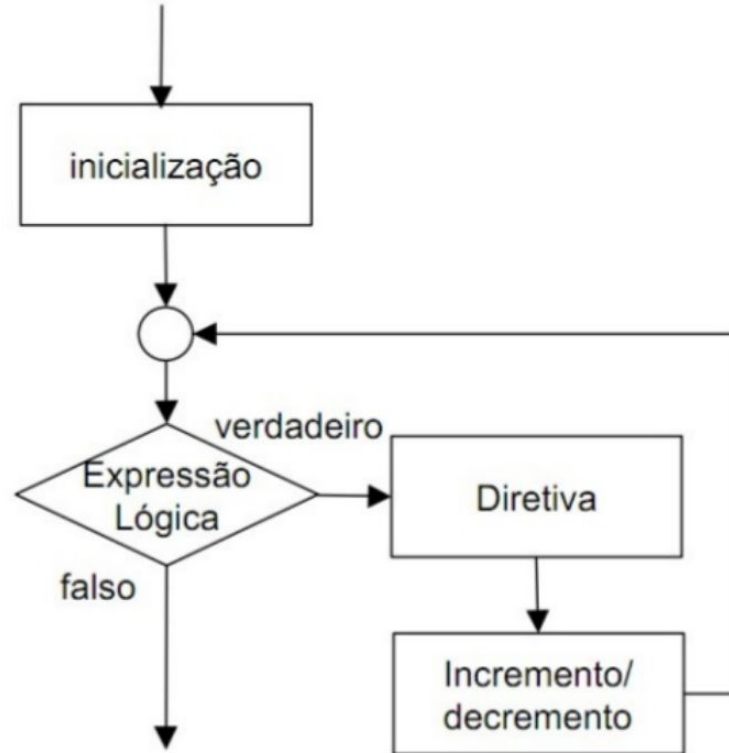
```
for (inicializacao; condicao; incremento) {  
    bloco de comandos;  
}
```

- Exemplo:

```
for (int i = 0; i <= 10; i = i + 1) {  
    System.out.print("Olá Mundo!!!");  
}
```

Comandos de Repetição

for



Comandos de Repetição

Instruções de Repetição - Dicas

- Pontos flutuantes: não recomendado
 - utilize inteiros para controlar loops de contagem
- Coloque linhas em branco antes e depois das instruções
- **contador** **<= 10** e não **contador** **< 11**
- No **for** cuidado com os dois 'ponto-e-vírgula' => `for(;;)`
- Novamente no **for** quando uma variável for declarada na inicialização ela não poderá ser utilizada fora do bloco de comandos

Comandos de Repetição

Instruções de Repetição - Dicas

- Incrementar **antes**

ou

depois?

```
for (int i = 0; i <= 10; ++i) {  
    System.out.print("Olá Mundo!!!");  
}
```

```
for (int i = 0; i <= 10; i++) {  
    System.out.print("Olá Mundo!!!");  
}
```

Comandos de Repetição

Controlando Loops

break

```
for (int i = 0; i <= 10; i++) {  
    if (i == 5) {  
        System.out.print("Nº 5");  
        break;  
    }  
    System.out.println( i );  
}
```

continue

```
for (int i = 0; i <= 10; i++) {  
    if (i == 5) {  
        System.out.print("Nº 5");  
        continue;  
    }  
    System.out.println( i );  
}
```

Atividade prática 1

Você deverá criar uma classe em Java, que imprima o seu nome 100 vezes.

Faça 3 versões deste programa:

1. Uma versão com uma declaração **while**
2. Uma versão com uma declaração **do-while**
3. Uma versão com uma declaração **for**

Atividade prática 2

Você deverá criar uma classe em Java, que peça para o usuário digitar 10 números e após os devidos testes seja informado qual foi o maior e o menor valor digitado.

1. Para isso crie um novo projeto Java.
2. Crie uma nova classe.
3. Importe as bibliotecas necessárias.
4. Crie as variáveis e objetos necessários.
5. Peça para o usuário digitar os valores.
6. Faça os devidos testes.
7. Exiba o resultado.

Atividade prática 3

Crie uma nova classe em Java, que solicite ao usuário um **valor inicial** e um **final** e a classe imprima todos os números entre esses dois valores.

Bibliografia

Cadenhead, Rogers; LEMAY, Laura, Aprenda em 21 dias Java 2. 4. ed. São Paulo: Campus, 2005.

DEITEL, Harvey H .; DEITEL, Paul J. Java: como Programar. 8. Ed. São Paulo: Pearson / Prentice-Hall, 2010.

Perguntas?



Obrigado!