

**ISUTC INSTITUTO SUPERIOR DE
TRANSPORTES E COMUNICAÇÕES**

Programação 1



sumário:

- Estrutura de Repetição

Objectivos:

- Conhecer os diferentes tipos de estruturas de repetição do Java,
- Ser capaz de aplicar as estruturas de repetição

Introdução:

Todos os programas computacionais em algum momento precisam de repetir acções.



Estruturas de repetição

Estruturas de repetição, são conhecidas como laços (loops) e são utilizadas para executar, repetidamente, uma instrução ou bloco de instruções enquanto determinada condição estiver sendo satisfeita.

Temos como estruturas:

- While
- Do..while
- for

Estruturas de repetição: elementos

Possui 4 elementos fundamentais:

- A **inicialização** compõe-se de todo código que determina a condição inicial da repetição.
- A **condição** é uma expressão booleana avaliada após cada leitura do corpo e determina se uma nova leitura deve ser feita ou se a estrutura de repetição deve ser encerrada.
- O **corpo** compõe-se de todas as instruções que são executadas repetidamente.
- A **iteração** é a instrução que deve ser executada depois do corpo e antes de uma nova repetição.

Estruturas de repetição: while

Estrutura de repetição, executada apenas quando a condição lógica for *true* e o número de iterações desejadas são desconhecidas.

Aplicabilidade:

- Aplicado em situações em que não se conhece o número de iterações.

```
<inicializar control>;  
while (<condicao com control>){  
    <instruções a repetir>;  
    <control actualiza>  
}
```

Exemplo while: Um programa que permite o usuário tentar adivinhar o numero que o programador colocou no programa directo é 10, que esteja no intervalo de 5 a 15.


```
import java.util.Scanner;
public class Exemplo1{
    public static void main(String [] args){
        Scanner ler = new Scanner(System.in);
        boolean encontrou = false;
        System.out.print("x: ");
        int x = ler.nextInt();
        while(encontrou != true){
            if(x == 10){
                encontrou = true;
            }else{
                System.out.print("tente novamente: ");
                x = ler.nextInt();
            }
        }
        System.out.print("acertou: ");
    }
}
```

Estruturas de repetição: do..while

Estrutura de repetição, utilizada no caso em que o teste de validação é o último recurso a ser executado e o número de iterações desejadas são desconhecidas.

Aplicabilidade:

- É usado geralmente para validação dos dados de entrada através do teclado

```
<inicializar control>;  
do{  
    <instruções a repetir>;  
    <control actualiza>;  
}while (<condicao com control>);
```

Exemplo do..while: Um programa que obriga o usuário a apenas inserir um numero positivo.

```
import java.util.Scanner;
public class Exemplo{
    public static void main(String [] args){
        Scanner ler = new Scanner(System.in);
        int x;
        do{
            System.out.print("insira um numero positivo: ");
            x = ler.nextInt();
        }while(x < 0);
    }
}
```

Estruturas de repetição: for

Estrutura de repetição automática, utilizada no caso em que se conhece o início e o fim do loop.

Aplicabilidade:

- É usada geralmente quando se tem a quantidade de vezes a repetir.

```
for (<inicializar control>; <condicao com control>; <control actualiza>) {  
    <instruções a repetir>;  
}
```

Exemplo for: imprimir os valores no intervalo de 1 a 10.

```
class Teste{  
    public static void main(String[] args) {  
        for(int i=1; i<11; i++){  
            System.out.println("Count is: " + i);  
        }  
    }  
}
```

Estruturas de repetição: for-each

For-each pode ser traduzido como “para cada”. É um laço *for* que busca uma correspondência pré-determinada para funcionar.

Em outras palavras, em vez de declarar uma variável para contar os loops que serão usados durante a operação, você vai **declarar o tipo da variável seguido do nome do array**.

```
for (<tipo> <variável>: <array>)  
{  
    <instruções a repetir usando variavel>;  
}
```

Estruturas de repetição: for-each Exemplo

```
int[] V = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
for (int item : V) {  
    System.out.println("Os valores sao: " + item);  
}
```

Exercícios:

1. Faça um programa que determina a soma dos primeiros 7 números ímpares no intervalo de $[0,20]$
2. Faça um programa em Java, que determina a soma dos números do intervalo de $[1,9]$

GARANTE O TEU FUTURO
COM UMA FORMAÇÃO SÓLIDA



Prolong. da Av. Kim Il Sung (IFT/TDM) Edifício
D1
Maputo, Moçambique

www.facebook.com/isutc

www.transcom.co.mz/isutc