

# LÓGICA DE PROGRAMAÇÃO - JAVA

Técnico de Desenvolvimento de Sistemas

# ESTRUTURAS DE REPETIÇÃO E DE SELEÇÃO

# ESTRUTURAS DE REPETIÇÃO

- Também são conhecidas como laços (loops) e são utilizados para executar, repetidamente, uma instrução ou bloco de instrução enquanto determinada condição estiver sendo satisfeita.
- Qualquer que seja a estrutura de repetição, ela contém quatro elementos fundamentais: inicialização, condição, corpo e iteração.

# ESTRUTURAS DE REPETIÇÃO



- 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.

# FOR

- A estrutura de repetição "for" é a mais utilizada quando precisamos executar diversas vezes um mesmo bloco de código.
- É muito comum a existência de repetições que fazem uso de variáveis contadoras, especialmente para contagens de 1 em 1.
- Conseguimos facilmente declarar, inicializar e incrementar valores no cabeçalho dessa estrutura.

# FOR

- Para melhor entendermos a estrutura de repetição "for", devemos dividi-la em 3 partes:

```
for(parte1, parte2, parte3)
```

- Inicialmente, nós temos que "for" é o nome da estrutura e os valores que estão contidos no parênteses são os valores do cabeçalho da estrutura.

# FOR

- Parte 1:
  - Região reservada para declaração e inicialização de variáveis.
  - Normalmente, declaramos uma única variável, porém, somos livres para declarar e inicializar quantas forem necessárias.

# FOR

- Parte2:
  - Onde colocamos a condição para que o laço seja repetido.
  - A condição a ser estabelecida pode ser qualquer uma, porém, é comum e recomendável colocar uma condição utilizando a variável declarada na <parte1>.
  - É a responsável por determinar quantas vezes o laço será repetido.



# FOR

- Parte3:
  - A última região é um local reservado utilizado para incrementar valores a variável que foi declarada na <parte1> e comparada na <parte2>.
  - A incrementação de valores a cada ciclo é o propósito da estrutura de repetição "for".

# FOR

- Equivalente ao comando PARA do VisuALG
- para <var> de <valor\_inicial> até <valor\_final> faça <comando>;

## VisuALG:

x: inteiro

**para x de 1 ate 10 faca**

...

**fimpara**

## Java:

**int** x;

**for**( x=1; x <= 10; x++)

{

...

}

# FOR

- Formato geral:

```
for (inicialização ; condiçãoDeParada; incremento){  
    instrução  
}
```

- Inicialização: Nomeia a variável de controle e inicializa seu valor.

# FOR

- Formato geral:

```
for (inicialização ; condiçãoDeParada; incremento){
    instrução
}
```

- Condição de parada: Condição que determina se o loop deve parar.
- Incremento: Modifica o valor do contador para que a condição de parada torne-se falsa ao final.

# FOR

- Se condição de parada for omitida, o compilador entende que a condição é sempre verdadeira e entra em Loop infinito.
- Exercício: Faça um programa que imprima todos os valores inteiros de 0 a 100.

# FOR

```
public class ExemploFor {  
    public static void main(String[] args) {  
        for (int i = 0; i <= 100; i++){  
            System.out.println( i );  
        }  
    }  
}
```

# BREAK

- Para parar qualquer estrutura de repetição a qualquer momento, usamos o comando:

`break;`

- Ex:

```
for(int x = 0; x <= 10; x++){  
    System.out.println(x);  
    if(x == 5)  
    {  
        break;  
    }  
}
```

# CONTINUE

- Para pular para a próxima iteração em qualquer estrutura de repetição, a qualquer momento, usamos o comando: **continue;**

- Ex:

```
for(int x = 0; x <= 10; x++){  
    System.out.print("Número: ");  
    if(x == 5){  
        continue;  
    }  
    System.out.print(x);  
}
```



- A estrutura switch verifica uma variável e age de acordo com seus cases.
- Os cases são as possibilidades de resultados que são obtidos por switch.
- Basicamente, o switch serve para controlar várias ações diferentes de acordo com o case definido dentro dele.

- A estrutura do Switch é:

```
switch (variável) {  
    case valor :
```

```
        Código a ser executado caso o valor de case seja o mesmo da  
        variável de switch  
}
```

# ESTRUTURAS DE SELEÇÃO



- A estrutura do Switch é:

```
switch (variável) {
```

```
    case valor :
```

```
        Código a ser executado caso o valor de case seja o mesmo da variável
```

```
}
```

- O switch recebe uma variável e abre um bloco de dados ( { } )
- Dentro desse bloco de dados há os cases.
- Cada case recebe um valor único, que não pode ser repetido no mesmo bloco de dados.
- Marcamos cada case com dois pontos ( : ).
- Após os dois pontos colocamos todo código que deverá ser executado que pode conter quantas linhas nós quisermos.

- É um bom costume sempre terminar um código após o case com um comando break.
- Assim, evitamos que o resto do código seja executado.
- Case não gera resultados booleanos, portanto, não há a possibilidade de fazer comparações (Ex. Isso está totalmente errado, `case var1 > var2:`).

- **Default**
  - Como switch pode receber várias possibilidades, pode ocorrer de algum caso estar fora do alcance ou não definido.
  - Nesse momento, default faz seu papel.
  - Default pega qualquer resultado que não esteja definido no case.
  - É o bloco de código padrão que deve ser executado quando nenhum case for satisfeito.

# SWITCH

```
switch( opção ){  
    case opção1:  
        comandos caso a opção 1 tenha sido escolhida  
        break;  
    case opção2:  
        comandos caso a opção 2 tenha sido escolhida  
        break;  
    case opção3:  
        comandos caso a opção 3 tenha sido escolhida  
        break;  
    default:  
        comandos caso nenhuma das opções anteriores tenha sido escolhida  
}
```

# SWITCH

- A variável 'opção' geralmente é uma variável que receberá um valor, que será utilizado para uma tomada de decisão.
- Se 'opção' receber 'opção1' como entrada, são os códigos contido na 'case opção1' que serão executados.

# SWITCH

- Se 'opção' receber 'opção2' como entrada, são os códigos contido na 'case opção2' que serão executados.
- Se 'opção' receber 'opção3' como entrada, são os códigos contido na 'case opção3' que serão executados.

# SWITCH

- Se 'opção' receber qualquer outra coisa que não seja 'opção1', 'opção2' ou 'opção3', são os códigos contidos em 'default' que serão executados.



# SWITCH

```
import java.util.Scanner;
class Selecao{
    public static void main(String args[]){
        String opcao;
        Scanner ler = new Scanner(System.in);
        System.out.println("Concorda ou não? (sim/não) ");
        opcao = ler.nextLine();
        switch {
            case "sim":
                System.out.println("Concorda");
                break;
            case "não":
                System.out.println("Não concorda");
                break;
            default:
                System.out.println("Resposta inválida");
                break;
        }
    }
}
```

# SWITCH

```
public class ExemploSwitch {  
    public static void main(String args[]) {  
        int diaDaSemana;  
        Scanner ler = new Scanner(System.in);  
        diaDaSemana = ler.nextInt();  
        switch (diaDaSemana) {  
            case 1:  
                System.out.println("Domingo");  
                break;  
            case 2:  
                System.out.println("Segunda-feira");  
                break;  
            case 3:  
                System.out.println("Terça-feira");  
                break;  
            case 4:  
                System.out.println("Quarta-feira");  
                break;  
            case 5:  
                System.out.println("Quinta-feira");  
                break;  
            case 6:  
                System.out.println("Sexta-feira");  
                break;  
            case 7:  
                System.out.println("Sábado");  
                break;  
            default:  
                System.out.println("Este não é um dia válido!");  
        }  
    }  
}
```

# VAMOS PRATICAR!!!