

## **Curso 4 – Lógica Condicional e Controle de Fluxos em Java**

### **Aula 1 – Operadores Relacionais**

#### **- O que são Operadores Relacionais ?**

##### **Conceituação**

“São símbolos especiais quais capazes de realizar comparações entre determinados operandos e, em seguida, retornar um resultado.”

##### **Tipos:**

- Similaridade: igual, diferente
- Tamanho: maior, maior igual, menor, menor igual

#### **- Utilizando Operadores Relacionais**

##### **Utilização**

##### **Similaridade**

- Igualdade: determina se um operando é igual ao outro
- Diferença: determina se um operando não é igual ao outro

##### **Simbologia**

- Igualdade: ==
- Diferença: !=

##### **Tamanho**

- Maior: determina se um operando é maior do que outro
- Maior Igual: determina se um operando é maior ou igual a outro

##### **Simbologia**

- Maior: >
- Maior Igual: >=

## Tamanho

- Menor: determina se um operando é menor do que outro
- Menor Igual: determina se um operando é menor ou igual a outro

## Simbologia

- Menor: <
- Menor Igual: <=

-Exercício de exemplo

## Exemplos

```
int i1 = 10; int i2 = 20; float f1 = 4.5f; float f2 = 3.5f;  
char c1 = 'x'; char c2 = 'y'; String s1 = "Fulano";  
String s2 = "Fulano"; boolean b1 = true; boolean = false;
```

```
i1 == i2, i1 != i2, i1 > i2, i1 <= i2
```

```
f1 == f2, f1 != f2, f1 >= f2, f1 < f2
```

```
c1 == c2, c1 != c2, c1 > c2, c1 <= c2
```

```
s1 == s2, s1 != s2, s1 >= s2, s1 < s2 -> EXPRESSÕES INVÁLIDAS
```

```
b1 == b2, b1 != b2, b1 > b2, b1 <= i1 -> EXPRESSÕES INVÁLIDAS
```

```
i2 > f1
```

## Aula 2 – Operadores Lógicos

### - O que são Operadores Lógicos ?

#### Conceituação

“São símbolos especiais quais são capazes de realizar comparações lógicas entre operandos lógicos ou expressões e, em seguida, retornar um resultado.”

#### Tipos:

- Conjunção
- Disjunção
- Disjunção exclusiva
- Negação

### - Como utilizar Operadores Lógicos

#### Conceituação

#### Tipos:

- Conjunção: operação lógica que só é verdadeira quando ambos os operandos ou expressões envolvidas são verdade.

#### Simbologia:

- &&

#### Terminologia:

- and(e)

O-E	O-E	R
V	V	V
V	F	F
F	V	F
F	F	F

O – Operando E – Expressão R - Resultado

## Conceituação

### Tipos:

- Disjunção: operação que só é falsa quando ambos os operandos ou expressões envolvidas são falsos.

### Simbologia:

- $\vee$

### Terminologia:

- or(ou)

O-E	O-E	R
V	V	V
V	F	V
F	V	V
F	F	F

O – Operando E – Expressão R - Resultado

## Conceituação

### Tipos:

- Disjunção exclusiva: operação que só é verdade quando ambos os operandos ou expressões são opostos

### Simbologia:

- $\wedge$

### Terminologia:

- xor

O-E	O-E	R
V	V	F
V	F	V
F	V	V
F	F	F

O – Operando E – Expressão R - Resultado

## Conceituação

### Tipos:

- Negação: operação que inverte o valor lógico de um operando ou expressão

### Simbologia:

- !

### Terminologia:

- inversão

O-E	R
V	F
F	V

O – Operando E – Expressão R – Resultado

## - Boas práticas e Exercitando

## Curiosidades

- Operadores bitwise: & e | -> Manipulação com bits
- Operadores shift: ~, >>, >>>, << -> inversão e deslocamento de bits

## Exemplos

```
boolean b1 = true; boolean b2 = false;
```

```
boolean b3 = true; boolean b4 = false;
```

```
b1 && b2, b1 && b3
```

```
b2 || b3, b2 || b4
```

```
b1 ^ b3, b4 ^ b1
```

```
!b1, !b2
```

```
(i1 > i2) || (f2 < f1)
```

```
((i1 + i2) < (f2 - f1)) && true
```

## **Boas práticas**

- Crie variáveis auxiliares para guardar resultados intermediários

```
(salarioMensal < mediaSalario) && (quantidadeDependentes >=
mediaDependentes);
```

Pode ser

```
(salarioBaixo) && (muitosDependentes)
```

```
boolean recebeAuxilio = (salarioBaixo) && (muitosDependentes);
```

## **Exercitando**

Criar um simples projeto e codificar os exemplos dos slides anteriores para compreender as operações lógicas. Utilize operandos e expressões.

## Aula 3 – Controle de Fluxo

### - O que são Controle de Fluxo ?

#### Conceituação

“São estruturas que tem a capacidade de direcionar o fluxo de execução do código.”

#### Tipos:

- Decisão: if, if-else, if-else-if, switch e operador ternário
- Repetição: for, while, do while
- Interrupção: break, continue e return

#### Criação

#### Tipos:

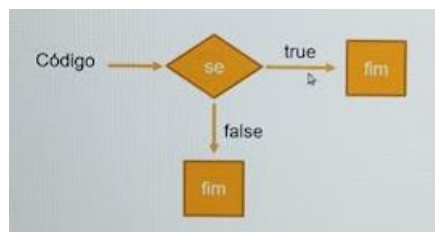
- Decisão: estrutura que avalia uma condição booleana ou variável para direcionar o fluxo de execução
- Opções: if(se), switch(escolha) e operador ternário

### - Criando estruturas de Controle de Fluxo

#### Criação

#### Tipos:

- Decisão: if, if-else, if-else-if, if,-else-if-else



## Criação

### Tipos:

- Decisão: if, if-else, if-else-if

```
if(condição) {  
}
```

```
if(condição){  
}else{  
}
```

```
if(condição){  
}else if(condição) {  
}else{  
}
```

## Criação

### Tipos:

- Decisão

```
if(idade > 18){  
}
```

```
if( aprovado) {  
}else{  
}
```

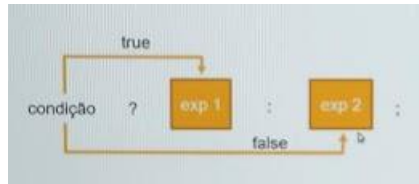
```
if(casado && temFilhos){  
}else if (casado && semFilhos){  
}else{  
}
```

## Criação



### Tipos:

- Decisão: operador ternário



### Criação

#### Tipos:

- Decisão: operador ternário

condição ? true : false;

condição ? true : null; “Não faz muito sentido”

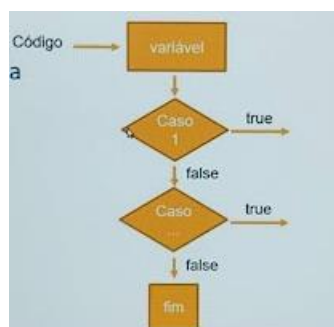
ligados ? desligar : ligar;

emMovimento ? freia : null;

### Criação

#### Tipos:

- Decisão: escolha



### Criação

#### Tipos:

- Decisão: switch

Variável:                      switch(variável){

byte                              case 1:

short	break;
char	case ...:
int	break;
Enum	default:
String	break;
	}

## Criação

### Tipos:

- Decisão

```
Switch(olhos){  
    case: "AZUIS":  
        break;  
    case: "VERDES":  
        break;  
    case: "CASTANHOS":  
        break;  
    default:  
        break;  
}
```

### - Boas práticas

- Switch é para valores exatos e if para expressões booleanas;
- Evitar usar o default do switch para "cases genéricos";
- Evitar o efeito "flecha" dos if 's;
- Evitar muitos if 's aninhados;
- Usar a boa prática da aula 2 para diminuir o tamanho do if.

## Exercitando

Criar um simples projeto e criar variáveis e expressões para usar nas estruturas **if** e **switch**.

Com if, exiba o nome do mês do ano de acordo com seu número. Evite o efeito “flecha”. Faça também outro if que verifique se o mês é julho, dezembro ou janeiro, para assim exibir o texto “Férias”.

Com switch use String para a partir do dia da semana, exibir seu número. Ainda no switch, faça outro exemplo onde, se uma variável inteira for entre 1 e 3 exibir o texto “Certo”. Se for 4 exibir “Errado”, se for 5 “Talvez”. Para demais valores exibir “Valor indefinido”

## Aula 4 – Blocos

### -Blocos, tipos e uso

#### Conceituação

“É um grupo de zero ou mais códigos que trabalham em conjunto para executar uma operação.”

#### Tipos:

- Locais: dentro de métodos
- Estático: dentro de classes
- Instância: dentro de classes

#### Criação

#### Tipos:

- Locais
  - {
  - ...
  - }

#### Exemplos

```
if(autorizado){  
    CarregarPerfil  
    DirecionarPáginaPrincipal  
}
```

```
if(menoridade)  
    DirecionarPáginaProibido
```

```
if(autorizado) {  
    ...  
}
```

```
} else {  
    ...  
}
```