**Lógica Condicional e Controle de Fluxos em Java**

**O que são operadores relacionais?**

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

**Tipos**

* Similaridade: igual e diferente
* Tamanho: maior, maior igual, menor, menor igual

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

**Simbologia**

* Maior: >
* Maior Igual: >=
* Menor: <
* Menor Igual: <=

**O que são operadores lógicos?**

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

**Tipos**:

* Conjunção;
* Disjunção;
* Disjunção exclusiva;
* Negação

**Como utilizar operadores lógicos?**

* Conjunção: operação lógica que só é verdadeira quando ambos os operadores ou expressões envolvidas são verdade

- Simbologia: &&

- Terminologia: and (e)



O – Operando E – Expressão R-Resultado

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

- Simbologia: ||;

- Terminologia: or (ou);



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

- Simbologia: ^;

- Terminologia: xor;



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

- Simbologia: !;

- Terminologia: inversão;



**Curiosidades**:

* Operadores bitwise: & e| // pode confundir, são operadores que irão mexer com bits
* Operadores shift: ~, >>, >>>, <<

**Boas práticas**

* Crie variáveis auxiliares para guardar resultados intermediários

Ex. (salarioMensal < mediaSalario) && (quantidadeDependentes >= mediaDependentes)

pode ser

(salarioBaixo) && (muitosDependentes)

boolean recebeAuxilio = (salarioBaixo)

&& (muitosDependentes);

**O que é controle de fluxo?**

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

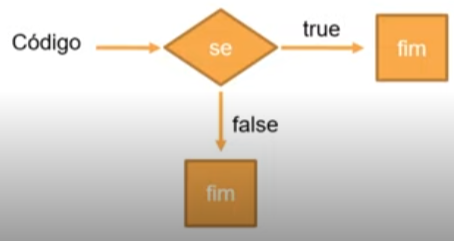
**Tipos**:

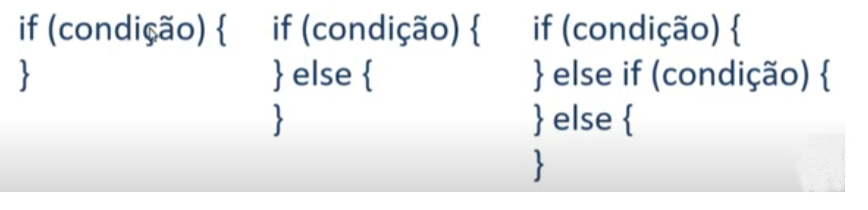
* Decisão: if, if – else, if – else – if, switch e operador ternário;
* Repetição: for, while, do while;
* Interrupção: break, continue, return.

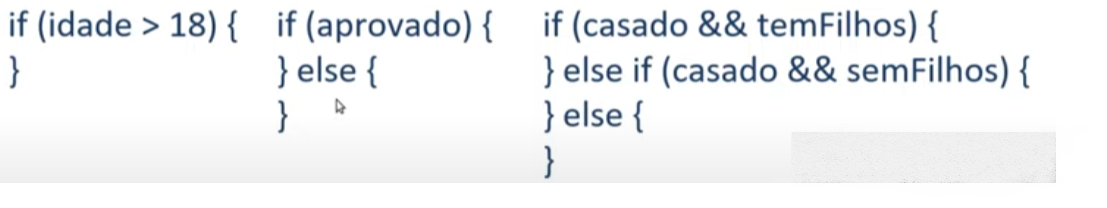
**Como utilizá-lo?**

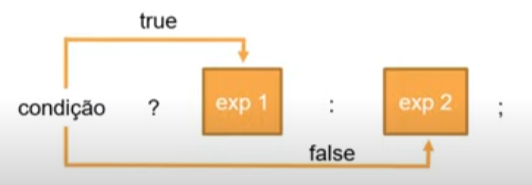
* Decisão: estrutura que avalia uma condição booleana ou variável para direcionar o fluxo de execução
  + Opções: if (se), switch (escolha) e operador ternário;

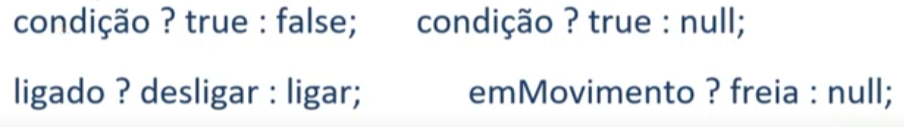
- if, if – else, if – else – if, if – else – if – else.

****

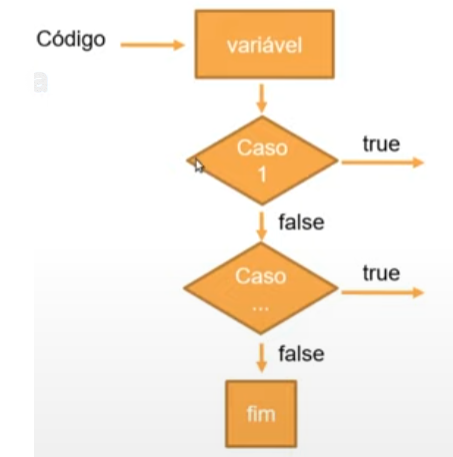


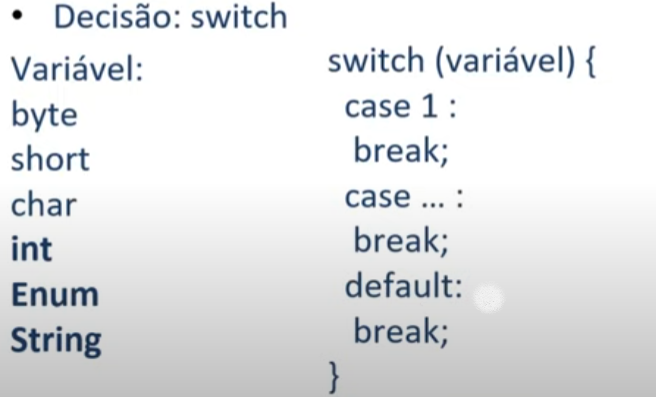


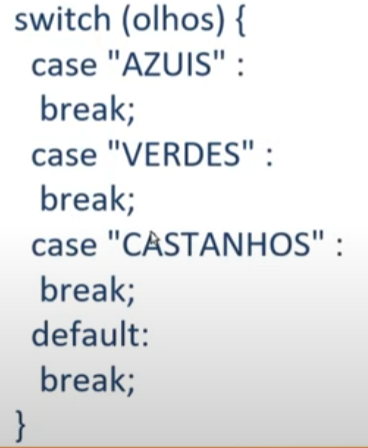
- Operador ternário: 



- Switch (escolha):







**Boas Práticas**

* Switch é para valores exatos e if para expressões booleanas;
* Evitar usar o default do switch para “cases genéricos” ;
* Evitar o efeito “flexa” dos if’s;
* Evitar muitos if’s aninhados;
* Usar boa prática da aula 2 para diminuir o tamanho do if.

**O que são blocos de execução?**

É um grupo de 0 ou mais códigos quais trabalham em conjunto para executar uma operação.

**Tipos**:

* Locais: dentro de métodos;
* Estáticos: dentro de classes;
* Instância: dentro de classe.

**Criação**:



