

AULA 8

OPERADOR TERNÁRIO

- Operador específico para fazer comparações

```
Se (n1 > n2) então  
    maior <- n1  
senao  
    maior <- n2  
FimSe
```

Isto era em algoritmos!

maior = n1 > n2 ? n1 : n2;

Em Java!

Isso quer dizer:

A variável > vai receber o que está antes dos : ou depois dos :

Ou seja, vai receber **n1 ou n2**

E isso vai depender do que está antes da interrogação

Se o n1 > n2, o **maior** vai receber **n1**

Senao, o **maior** vai receber **n2**

Os dois pontos
significam o **senao** de
algoritmos!

No NetBeans:

```
public static void main(String[] args) {  
    // TODO code application logic here  
    int n1, n2, r;  
    n1 = 4;  
    n2 = 8;  
    r = (n1 > n2) ? 0 : 1;  
}
```

Isso quer dizer: o r recebe 0 ou 1

Depende do que? Se o n1 for maior
do que o n2

Como n2 > n1, então o r vai receber 1

Outro exemplo:

```
public static void main(String[] args) {  
    // TODO code application logic here  
    int n1, n2, r;  
    n1 = 4;  
    n2 = 8;  
    r = (n1 > n2) ? n1 : n2;  
    System.out.println(r);  
}
```

Como n2 > n1, então o r vai receber
n2, que é 8

OBS: os **operadores ternários** também aceitam fórmulas! Como por exemplo:

```
public static void main(String[] args) {  
    // TODO code application logic here  
    int n1, n2, r;  
    n1 = 12;  
    n2 = 4;  
    r = (n1 > n2) ? n1 + n2 : n1 - n2;  
    System.out.println(r);  
}
```

Como n1 > n2, então r vai receber
n1 + n2, que é 16

Objetivo do **operador ternário**: atribuir um valor ou outro a uma mesma variável

OPERADORES RELACIONAIS

>	Maior que	5 > 2	true
<	Menor que	4 < 1	false
>=	Maior ou igual a	8 >= 3	true
<=	Menor ou igual a	6 <= 6	true
==	Igual a	9 == 8	false
!=	Diferente de	4 != 5	true

Comparando strings

```
String nome1 = "Gustavo";
String nome2 = "Gustavo";
String nome3 = new String("Gustavo");
```

- Existem maneiras diferentes de criar um mesmo **objeto**
- Lembrar: String com a letra maiúscula é uma **classe**, então automaticamente ele vai criar um **objeto**
- A diferença de instanciar ou não utilizando a cláusula new, vai fazer toda a diferença

No NetBeans:

```
public static void main(String[] args) {
    // TODO code application logic here
    String nome1 = "Gustavo";
    String nome2 = "Gustavo";
    String nome3 = new String("Gustavo");
    String res;
    res = (nome1==nome2)?"igual":"diferente";
    System.out.println(res);
}
```

A variável res vai receber a palavra "igual" se o nome1 for igual ao nome2, senao, vai receber a palavra "diferente"

```
public static void main(String[] args) {
    // TODO code application logic here
    String nome1 = "Gustavo";
    String nome2 = "Gustavo";
    String nome3 = new String("Gustavo");
    String res;
    res = (nome1==nome3)?"igual":"diferente";
    System.out.println(res);
}
```

O último objeto foi instanciado utilizando a cláusula new, então tecnicamente não tem a mesma estrutura dos outros, então não são iguais (no quesito conteúdo!).

Portanto, vai receber a palavra "diferente"

```
public static void main(String[] args) {
    // TODO code application logic here
    String nome1 = "Gustavo";
    String nome2 = "Gustavo";
    String nome3 = new String("Gustavo");
    String res;
    res = (nome1.equals(nome3))? "igual": "diferente";
    System.out.println(res);
}
```

O .equals é um método que verifica se o conteúdo de um objeto (no caso, nome1) ao conteúdo do outro (no caso, nome3).

Portanto, vai receber a palavra "igual"

OPERADORES LÓGICOS

&&	.E.	true && false	false
	.OU.	false true	true
^	.XOU.	true ^ true	false
!	.NAO.	! false	true

verdadeiro E falso

verdadeiro OU falso

verdadeiro OU EXCLUSIVAMENTE verdadeiro

NAO falso

Tabela verdade

- Lembrar sempre dos exemplos Paula (p) e Quézia (q)
- Verdadeiro (v) é feliz, falso (f) é triste

p	q	p && q
V	V	V
V	F	F
F	V	F
F	F	F

p	q	p q
V	V	V
V	F	V
F	V	V
F	F	F

p	q	p ^ q
V	V	F
V	F	V
F	V	V
F	F	F

p	! p
V	F
F	V

OBS: para o OU EXCLUSIVO:

- Eu quero que a Paula seja feliz ou exclusivamente a Quezia
- Ou seja, pode ser uma, pode ser outra, mas eu não quero que as duas sejam felizes

OBS: para o NAO:

- Uma pessoa que NAO está feliz, ela está triste

No NetBeans:

```
public static void main(String[] args) {  
    // TODO code application logic here  
    int x, y, z;  
    x = 4;  
    y = 7;  
    z = 12;  
    boolean r;  
    r = (x < y && y < z) ? true : false;  
    System.out.println(r);  
}
```

true

```
public static void main(String[] args) {  
    // TODO code application logic here  
    int x, y, z;  
    x = 4;  
    y = 7;  
    z = 12;  
    boolean r;  
    r = (x > y || y > z) ? true : false;  
    System.out.println(r);  
}
```

false

```
public static void main(String[] args) {  
    // TODO code application logic here  
    int x, y, z;  
    x = 4;  
    y = 7;  
    z = 12;  
    boolean r;  
    r = (x < y ^ y > z) ? true : false;  
    System.out.println(r);  
}
```

true

Projeto no Swing:

Variáveis:

txtAno

btnCalc

lblIdade

lblSituacao

Ano de Nascimento

Calcular Idade

Idade 0

Situação <vazio>

No código-fonte:

```
private void btnCalcActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int ano = Integer.parseInt(txtAno.getText());
    int idade = 2023 - ano;
    lblIdade.setText(Integer.toString(idade));
    String sit = (idade >= 18)?"MAIOR":"MENOR";
    lblSituacao.setText(sit);
}
```

Ano de Nascimento

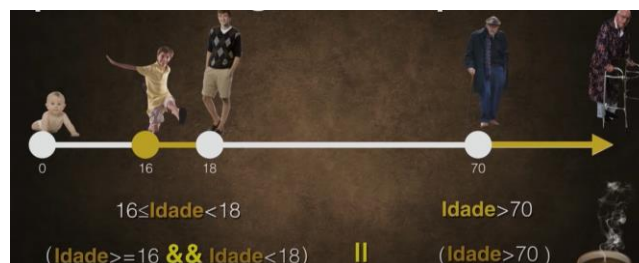
Calcular Idade

Idade 17

Situação MENOR

EXPRESSÕES LÓGICAS COMPOSTAS

Exemplo: voto de 16 e 17 anos é opcional, já de 18 a 70 é obrigatório.



Como eu leio isso:

- “A idade é maior ou igual a 16 e também é menor do que 18” OU ENTÃO “idade maior do que 70”

OBS: essa situação só abrange quem estiver entre esse intervalo. Ou seja, se eu tenho 14 anos, o voto é proibido, mas como eu faço isso?

- O operador ternário não suporta esse tipo de possibilidade adicional. Preciso ir para as estruturas de controle.

PRÓXIMA AULA!

Como fica o algoritmo:

```
private void btnCalcActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int ano = Integer.parseInt(txtAno.getText());
    int idade = 2023 - ano;
    lblIdade.setText(Integer.toString(idade));
    String sit = ((idade >= 16 && idade < 18) || (idade > 70))?"É OPCIONAL":"NÃO É OPCIONAL";
    lblSituacao.setText(sit);
}
```