

Lógica de Programação Orientada a Objetos

Aula 01

- Algoritmos
- Formas de Representação
- Português Estruturado
- Variáveis
- Operadores



Algoritmos

www.3way.com.br



Algoritmos

- Conjunto de instruções a serem seguidas para a solução de um determinado problema.
- Não se restringe ao universo computacional.
- Elaborado na fase de Concepção do sistema.
- “Projeto do Programa”.
- Fundamental para a formação de um bom desenvolvedor.



Algoritmos

Exemplo 1:

Algoritmo para um exercício físico:

Repetir 10 vezes os quatro passos abaixo:

- 1. Levantar e abaixar braço direito.*
- 2. Levantar e abaixar braço esquerdo.*
- 3. Levantar e abaixar perna direita.*
- 4. Levantar e abaixar perna esquerda.*



Algoritmos

Exemplo 2:

Algoritmo para verificar a aprovação de um aluno:

1. *Obter nota1.*
2. *Obter nota2.*
3. *Calcular $(nota1 + nota2) / 2$.*
4. *Se resultado for maior que 5 então escrever **Aprovado** senão escrever **Reprovado**.*



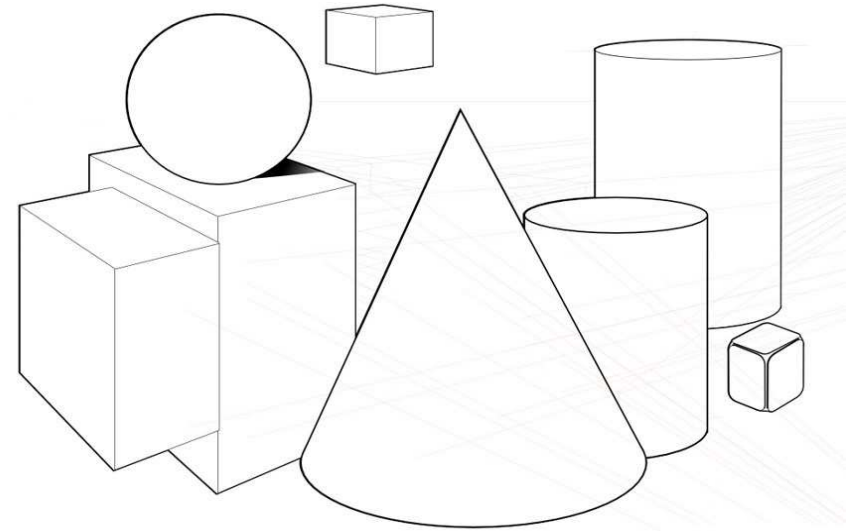
Formas de Representação

www.3way.com.br



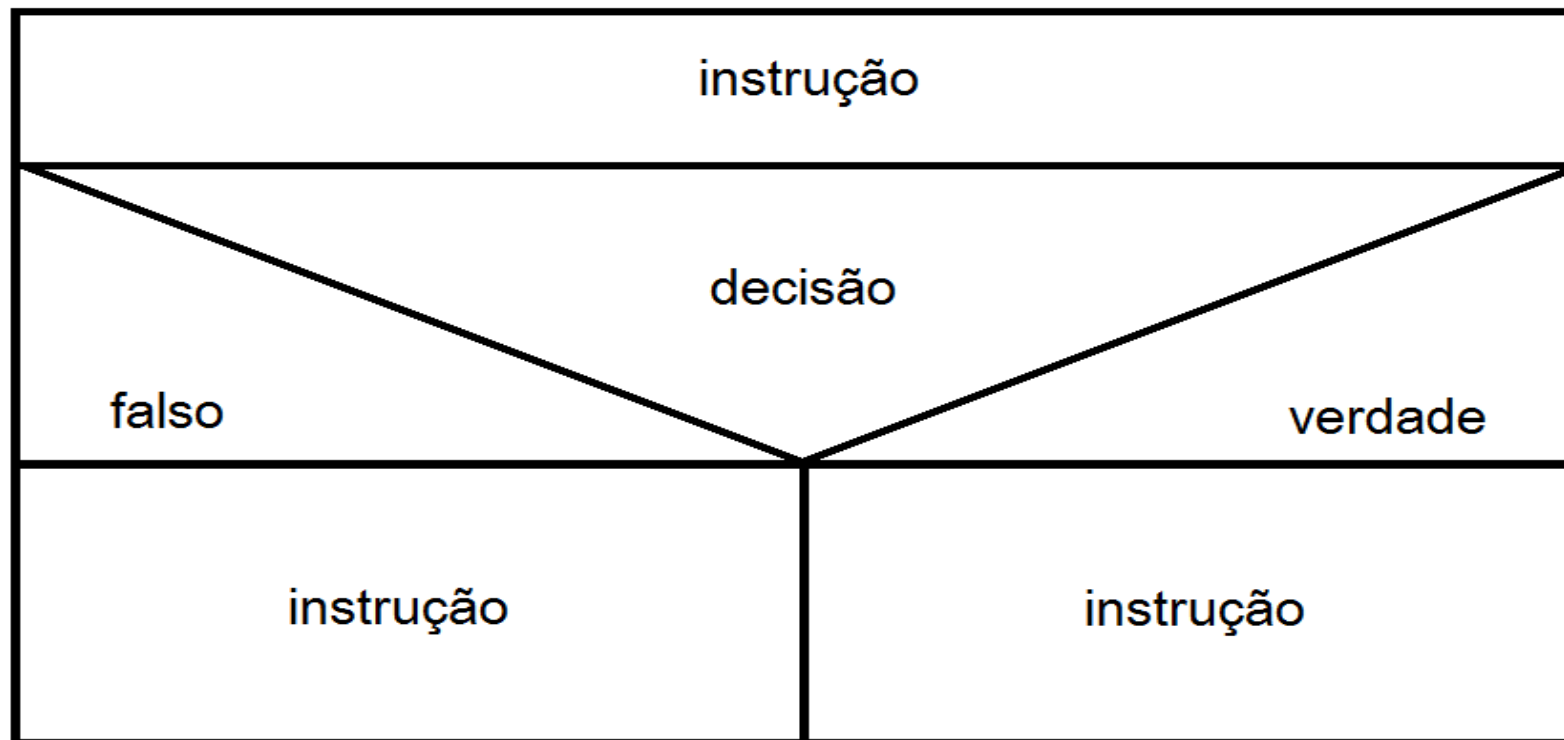
Formas de Representação

- Diagrama de Nassi-Shneiderman (Diagrama de Chapin)
- Fluxograma (Diagrama de Fluxo)
- Português Estruturado



Formas de Representação

Diagrama de Nassi-Shneiderman (Diagrama de Chapin)



Formas de Representação

Diagrama de Nassi-Shneiderman (Diagrama de Chapin)

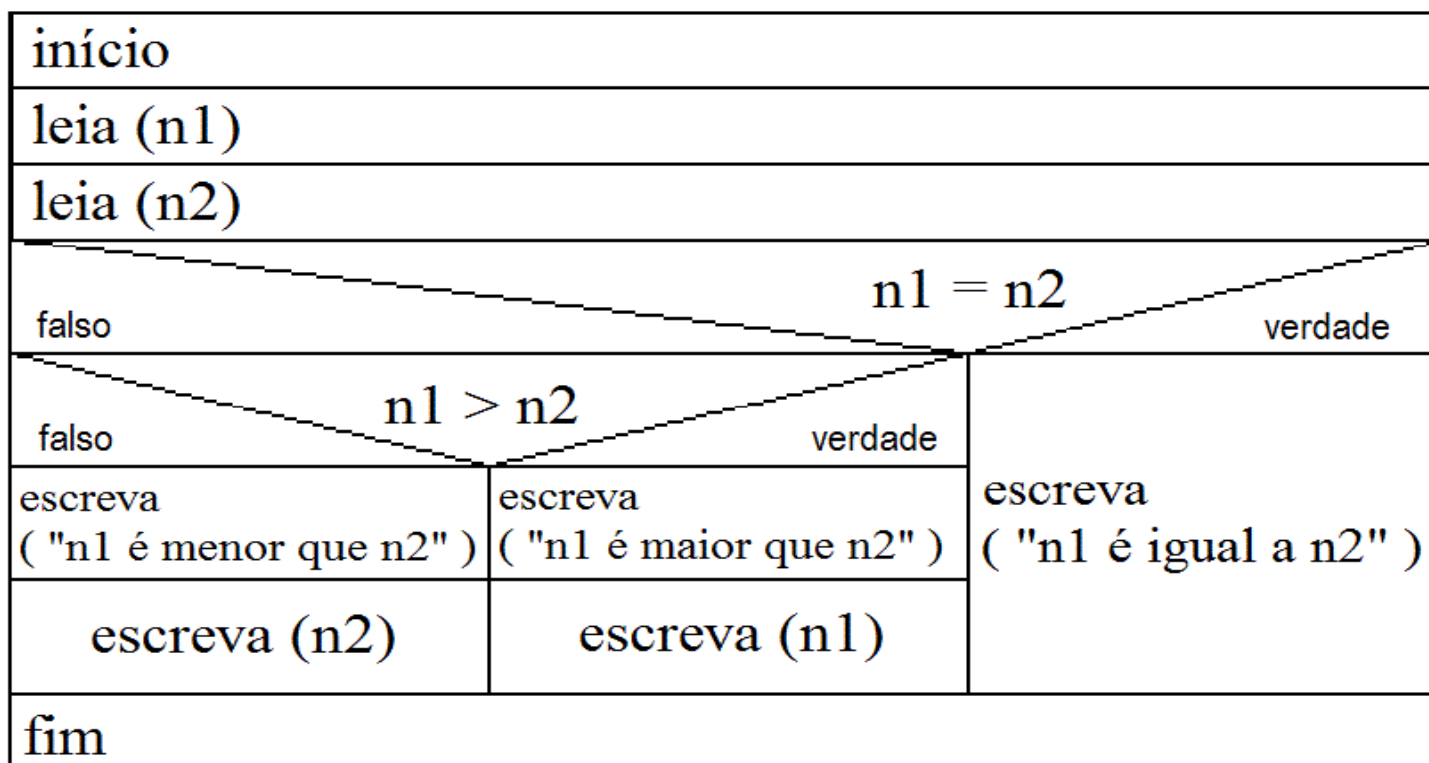
- Surgiram nos anos 70 para ajudar na representação das soluções de problemas.
- A idéia básica deste diagrama é representar as ações de um algoritmo dentro de um único retângulo, subdividindo-o em retângulos menores, que representam as seqüências de instruções do algoritmo.



Formas de Representação

Diagrama de Nassi-Shneiderman (Diagrama de Chapin)

Exemplo 1:



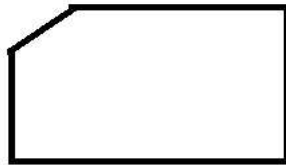
Formas de Representação

Fluxograma (Diagrama de Fluxo)

Formas geométricas utilizadas:



Início e fim do fluxograma



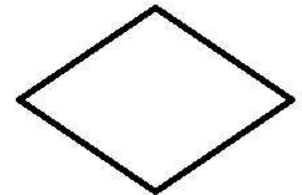
Operação de entrada de dados



Operação de saída de dados



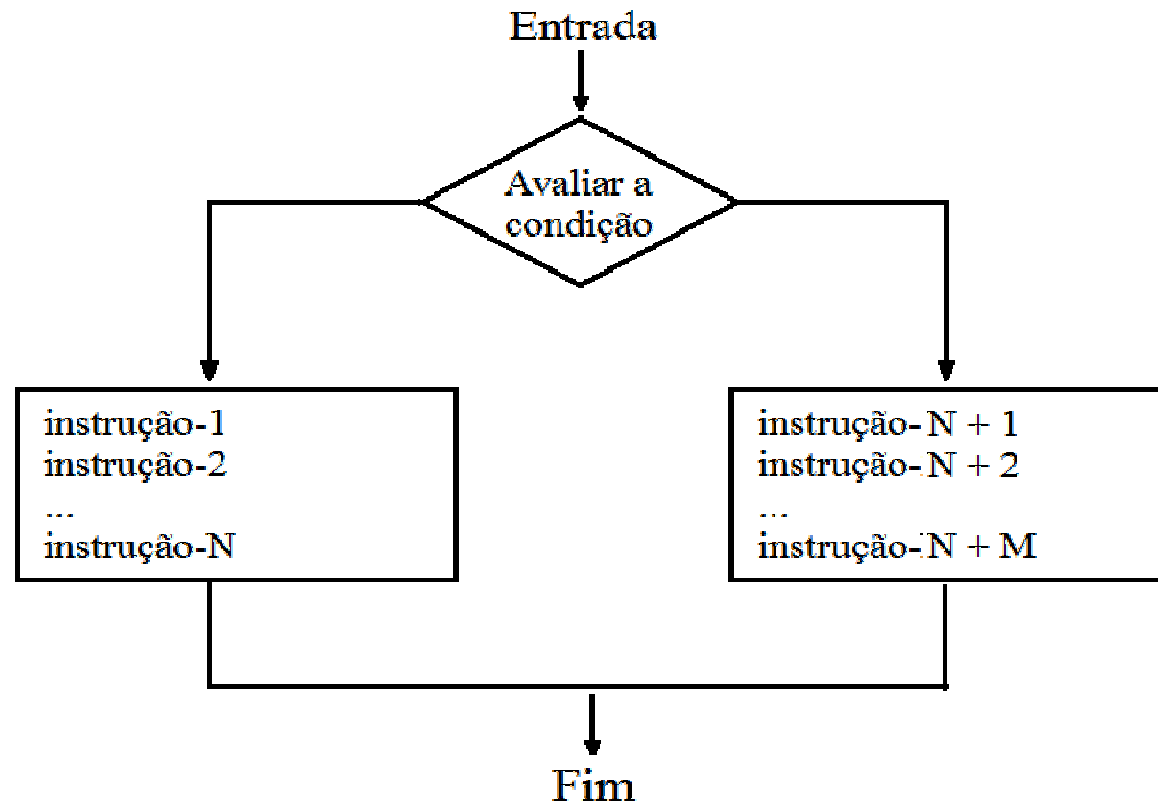
Operação de atribuição de valores
e cálculo de expressões



Tomada de decisões

Formas de Representação

Fluxograma (Diagrama de Fluxo)



Formas de Representação

Fluxograma (Diagrama de Fluxo)

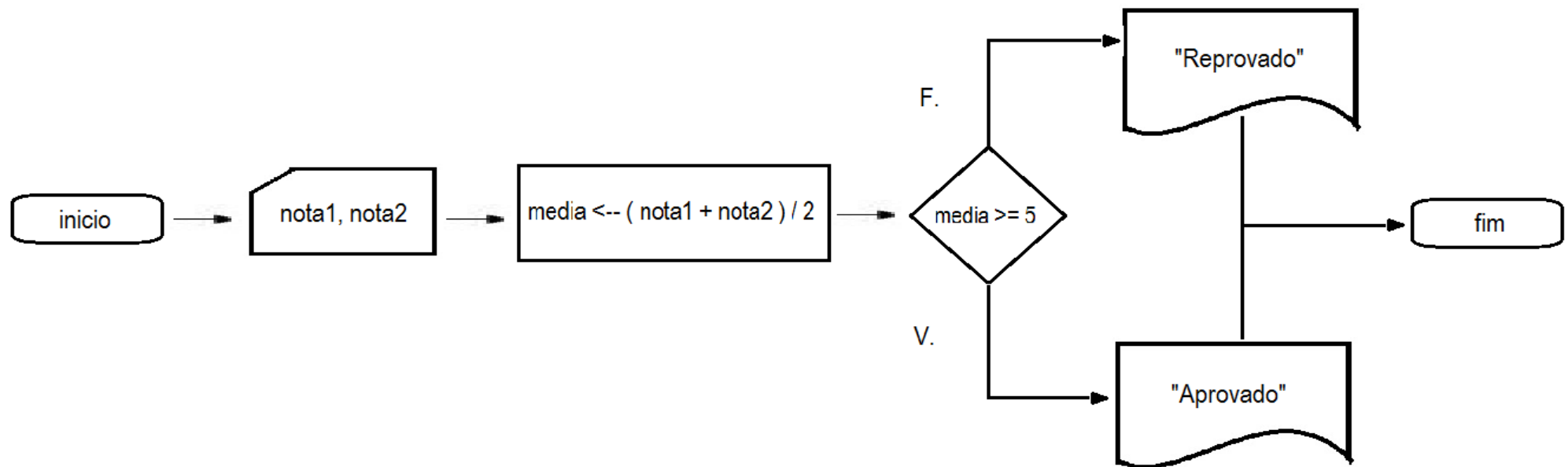
- São representações que utilizam formas geométricas bem definidas para indicar as ações e decisões tomadas pelo algoritmo.
- Permite a visualização dos vários caminhos (fluxos) que o algoritmo pode seguir.



Formas de Representação

Fluxograma (Diagrama de Fluxo)

Exemplo 1:



Formas de Representação

Português Estruturado

algoritmo { nome do algoritmo }

var

{ declaração de variáveis }

inicio

{ instruções do algoritmo }

fimalgoritmo



Formas de Representação

Português Estruturado

- Também conhecido como Portugol, Pseudocódigo ou Pseudolinguagem.
- Representa de forma bem definida todos os termos utilizados nas instruções de uma linguagem de programação.
- Não existe um padrão a ser seguido na sintaxe utilizada por esta forma de representação.



Formas de Representação

Português Estruturado

Exemplo 1:

algoritmo "CalculaMedia"

var

nota01, nota02 : inteiro

media : real

inicio

leia(*nota01*)



Formas de Representação

Português Estruturado

Exemplo 1:

leia (*nota02*)

media \leftarrow (*nota01* + *nota02*) / 2

se *media* \geq 5 **entao**

escreva (" Aprovado ")

senao



Formas de Representação

Português Estruturado

Exemplo 1:

escreva (" Reprovado ")

fimse

[fimalgoritmo](#)



Português Estruturado

www.3way.com.br



Português Estruturado

Instrução *LEIA*

- Uma das principais instruções de um algoritmo.
- Usada quando se deseja **obter** algum dado do usuário, ou seja, é uma instrução de **entrada de dados**.

Sintaxe: ***leia** (nome_variável)*

Exemplo 1: ***leia** (saldo)*



Português Estruturado

Instrução *ESCREVA*

- Uma das principais instruções de um algoritmo.
- Usada quando se deseja **exibir** algum dado para o usuário, ou seja, é uma instrução de **saída de dados**.

Sintaxe: ***escreva** (informação)*

Exemplo 1: ***escreva** (“ Hello World “)*

www.3way.com.br



Português Estruturado

Atribuição

- Expressão utilizada para atribuir um valor a uma variável.
- Normalmente é utilizada uma seta apontando para a esquerda para representar uma expressão de atribuição. (\leftarrow)
- Sempre teremos uma variável do lado esquerdo da expressão.



Português Estruturado

Atribuição



- expressão : uma variável, uma constante ou uma expressão matemática.
- '←' : expressão de atribuição.

Exemplos :

media ← 10.0

a ← b

x ← 10 - 2

Em todos os casos lê-se “ variável *recebe* valor “.

Ex.: media *recebe* 10.0

Ex.: a *recebe* b

Ex.: x *recebe* 8



Variáveis

www.3way.com.br



Variáveis

Variáveis

- Indispensável na construção de algoritmos.
- Uma variável nada mais é do que um **espaço na memória**.
- Pode armazenar valores diferentes no decorrer da execução do algoritmo.
- São referenciadas através de um identificador (nome).



Variáveis

Variáveis

- Os identificadores devem ser sugestivos.

identificador sugestivo: nomeCliente

Identificador não sugestivo: x

- Uma variável possui:
 - um **tipo**, que indica o tipo de valor que ela pode armazenar.
 - um **identificador** (nome) , utilizado para acessá-la.



Variáveis

Variáveis

Exemplo 1:

algoritmo “OrdenaCrescenteVariáveis”

var

va1, var2, var3, aux : **inteiro**

inicio

leia (var1)



Variáveis

Variáveis

Exemplo 1:

leia (var2)

leia (var3)

se var1 > var2 **entao**

aux ← var1

var1 ← var2

var2 ← aux

fimse



Variáveis

Variáveis

Exemplo 1:

```
se var1 > var3 entao  
  aux ← var1  
  var1 ← var3  
  var3 ← aux  
fimse
```

```
se var2 > var3 entao  
  aux ← var2  
  var2 ← var3  
  var3 ← aux  
fimse
```



Variáveis

Variáveis

Exemplo 1:

escreval (" Ordem crescente: ")

escreva (var1 , " – " , var2 , " – " , var3)

[fimalgoritmo](#)



Variáveis

Tipos de variáveis

- Quando se declara uma variável, é necessário informar seu tipo.
- O tipo da variável define o quanto de memória deve ser alocada para armazenar seu valor.
- Na construção de algoritmos, o tipo deve ser genérico de acordo com suas propriedades.



Variáveis

Tipos de variáveis

Na construção de algoritmos, podemos utilizar os seguintes tipos de variáveis:

- **inteiro:** *Números inteiros positivos ou negativos.*

Ex.: -150, 30, -1, 1...

- **real:** *Números fracionários positivos ou negativos.*

Ex.: 19.5, -1.5, 150.3, -7.50...



Variáveis

Tipos de variáveis

- **logico:** *Tipos lógicos ou booleanos que podem assumir **verdadeiro** ou **falso**.*
- **literal** ou **caractere:** *São formados por mais de um caractere, representando um texto, entre aspas duplas.*

Ex.: "Hello World", "a", "abcde", "3way Networks"...

www.3way.com.br



Variáveis

Declaração de variável

- Indicação das variáveis que iremos utilizar no algoritmo.
- As variáveis são declaradas no bloco var do algoritmo.

Sintaxe:

nome_da_variável : tipo_da_variavel



Exemplo 1:

nomeAluno : caractere



Variáveis

Declaração de variável

| Instrução | Memória | Descrição |
|-----------------|---|---------------------------|
| idade : inteiro | <div>XH22A</div> <div>idade </div> | Declarando a variável. |
| idade <-- 22 | <div>XH22A</div> <div>idade </div> | Inicializando a variável. |

Na instrução acima,

A **variável idade** é um espaço alocado na memória no **endereço XH22A** que armazena o **valor inteiro** igual a 22.

Matrizes



Run Jeliot - variáveis

www.3way.com.br



Variáveis

Constantes

- Informações (dados) que não variam com o tempo.
- Independente das instruções do algoritmo seu valor permanece sempre o mesmo.
- Possui um valor fixo.

π



Variáveis

Constantes

Exemplo 1:

algoritmo "NaoAlteraConstante"

var

idade, anoAtual, dataNascimento : **inteiro**

inicio

dataNascimento \leftarrow 1989



Variáveis

Constantes

Exemplo 1:

leia (*anoAtual*)

idade \leftarrow *anoAtual* – *anoNascimento*

anoNascimento \leftarrow 2011

X

escreva (“ Hoje estou com “ , *idade* , “ anos
de idade. “)

[fimalgoritmo](#)



Operadores

www.3way.com.br



Operadores

Operadores Aritméticos

| Operação Matemática | Símbolo utilizado |
|---------------------|-------------------|
| Multiplicação | * |
| Divisão | / |
| Adição | + |
| Subtração | - |
| Resto da divisão | % ou <i>mod</i> |
| Potenciação | ^ |



Operadores

Operadores Aritméticos

- As expressões escritas dentro de parênteses tem maior prioridade que as escritas de fora.
- Quando existem vários níveis de parênteses, ou seja, um dentro do outro, o mais interno tem maior prioridade que o mais externo.
- Quando duas ou mais expressões tiverem a mesma prioridade, as expressões serão executadas da esquerda para a direita.



Operadores

Operadores Aritméticos

Exemplos :

$$2 + (6 * (3 + 2)) = 32$$

$$2 + ((6 * 3) + 2) = 22$$

$$10 \% 3 = 1$$

$$10 \text{ mod } 4 = 2$$

$$5 \wedge 2 = 25$$



Operadores

Operadores de Caracteres

| Utilização | Símbolo utilizado |
|------------------|-------------------|
| texto + texto | + |
| texto + variável | , |



Operadores

Operadores de Caracteres

- texto + texto

Saída :

“ 3way networks”

escreva (“ 3way” + “networks”)

- texto + variável

idade : inteiro

Saída :

“ Tenho 22 anos de idade.”

idade ← 22

escreva (“Tenho “ , idade , “ anos de idade.”)

www.3way.com.br



Operadores

Operadores Relacionais

| Símbolo | Comparação |
|---------|------------------|
| > | maior que |
| < | menor que |
| = | igual a |
| >= | maior ou igual a |
| <= | menor ou igual a |
| <> | diferente de |



Operadores

Operadores Relacionais

Exemplos :

$10 > 5 = \text{VERDADEIRO}$

$10 < 8 = \text{FALSO}$

$1 <> 2 = \text{VERDADEIRO}$

$10.0 = 10 = \text{VERDADEIRO}$

$1.5 \geq 1 = \text{VERDADEIRO}$

$1 \leq 0 = \text{FALSO}$



Uma operação relacional **sempre** resultará
em um tipo lógico:
VERDADEIRO ou **FALSO**



Operadores



Operadores Lógicos

Uma operação lógica **sempre** resultará em um tipo lógico:
VERDADEIRO ou **FALSO**

| Operação | Operador |
|---------------------------|-------------------------------------|
| Negação | NAO |
| Conjunção | E |
| Disjunção (não-exclusiva) | OU |
| Disjunção exclusiva | XOU (lê-se <i>ou exclusivo</i>) |



Operadores

Operadores Lógicos

Operador *NAO*

| Operação | Resultado |
|------------------------------|------------|
| NAO <i>verdadeiro</i> | FALSO |
| NAO <i>falso</i> | VERDADEIRO |



Operadores

Operadores Lógicos

Operador *NAO*

Saída:

falso
verdadeiro

...

`var1 ← NAO (10 > 5)`

`var2 ← NAO (10 = 5)`

`escreva (var1)`

`escreva (var2)`



Operadores

Operadores Lógicos

Operador *E*

| Operação | Resultado |
|-------------|------------|
| $V \ E \ V$ | VERDADEIRO |
| $V \ E \ F$ | FALSO |
| $F \ E \ V$ | FALSO |
| $F \ E \ F$ | FALSO |



Operadores

Operadores Lógicos

Operador *E*

Saída:

verdadeiro
falso

...

`var1 ← (10 > 5) E (5 > 3)`

`var2 ← (10 = 5) E (1 >= 1.0) E (100 <> 100.0)`

`escreva (var1)`

`escreva (var2)`



Operadores

Operadores Lógicos

Operador *OU*

| Operação | Resultado |
|-------------------|------------|
| $V \text{ OU } V$ | VERDADEIRO |
| $V \text{ OU } F$ | VERDADEIRO |
| $F \text{ OU } V$ | VERDADEIRO |
| $F \text{ OU } F$ | FALSO |



Operadores

Operadores Lógicos

Operador *OU*

Saída:

verdadeiro
falso

...

```
var1 ← ( 10 > 5 ) OU ( 5 > 3 ) OU ( 2 = 4 )
```

```
var2 ← ( 10 = 5 ) OU ( 2 >= 5.0 ) OU FALSO
```

```
escreva ( var1 )
```

```
escreva ( var2 )
```



Operadores

Operadores Lógicos

Operador *XOU*

| Operação | Resultado |
|-----------------------------|------------|
| $V \text{ } XOU \text{ } V$ | FALSO |
| $V \text{ } XOU \text{ } F$ | VERDADEIRO |
| $F \text{ } XOU \text{ } V$ | VERDADEIRO |
| $F \text{ } XOU \text{ } F$ | FALSO |



Operadores

Operadores Lógicos

Operador *XOU*

Saída:

falso
verdadeiro

...

```
var1 ← ( 10 > 5 ) XOU ( 5 > 3 ) XOU ( 2 < 4 )
```

```
var2 ← ( 10 = 5 ) XOU ( 2 >= 5.0 ) XOU VERDADEIRO
```

```
escreva ( var1 )
```

```
escreva ( var2 )
```

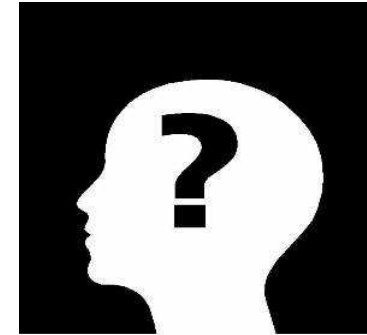


Operadores

Operadores Lógicos

Desafio

Saída:



...

```
var1 ← ( 10 < 100 ) E ( NAO ( NAO ( 10 < 4 ) ) OU ( falso ) )  
      XOu ( verdadeiro ) E ( ( NAO falso ) )
```

```
var2 ← verdadeiro OU ( falso E verdadeiro )
```

```
escreva ( var1 XOu var2 )
```



Operadores

Operadores Lógicos

Desafio

Saída:

verdadeiro

...

```
var1 ← ( 10 < 100 ) E ( NAO ( NAO ( 10 < 4 ) ) OU ( falso ) )  
      XOu ( verdadeiro ) E ( ( NAO falso ) )
```

```
var2 ← verdadeiro OU ( falso E verdadeiro )
```

```
escreva ( var1 XOu var2 )
```



Operadores

Precedência de Operadores

| Operador | Associatividade |
|-----------|-----------------------|
| () [] {} | esquerda-para-direita |
| ^ * / % | esquerda-para-direita |
| + - | esquerda-para-direita |
| < <= > >= | esquerda-para-direita |
| = <> | esquerda-para-direita |
| NÃO | esquerda-para-direita |
| XOU | esquerda-para-direita |
| E | esquerda-para-direita |
| OU | esquerda-para-direita |



Laboratório 01

www.3way.com.br

