

# **Algoritmos**

# **Elementos Básicos**

Prof. Wylliams Barbosa Santos  
wylliamss@gmail.com

Introdução à Programação

Crédito de conteúdo: Professora Ceça Moraes

Professor Leandro Galvão

# Elementos de um Algoritmo

- **Dados (variáveis e constantes)**
- **Tipos de dados**
- **Operadores**
- **Comandos**
- **Funções**
- **Comentários**

# Exemplo

```
algoritmo "perimetro circunferencia"  
// declaração de variáveis  
var  
perim, raio: real  
  
inicio  
  leia (raio)  
  perim <- 2 * PI * raio  
  escreval ("o perimetro é:", perim)  
finalgoritmo
```

comentário

variáveis

tipo de uma variável

operador

função

# Constantes e Variáveis

- Constantes
  - O valor de uma constante **não se altera** após sua definição.
  - **Exemplos:**
    - `const_pi ← 3,1415`
- Variáveis
  - Elemento que têm a função de **associar um nome a uma área da memória** onde um dado pode ser armazenado

# Tipos

- Definem:
  - a **natureza** do dado
  - as **operações** que podem ser realizadas com o dado
  - O **espaço** a ser ocupado na memória
- Declarações:
  - **a, b, c, maior: real**
  - **x,y: inteiro**
  - **achou: logico**
  - **texto: caractere**

- Tabela de tipos Visualg:

Tipo	Descrição
<b>inteiro</b>	Representa valores inteiros. Exemplos: 10, 5, -5, -10
<b>real ou numerico</b>	Representa valores reais (com ponto separador da parte decimal). Exemplos: 10, 15.5, -14.67
<b>literal ou caractere</b>	Representa texto (seqüência ou cadeia de caracteres) entre aspas duplas. Exemplo "Esta é uma cadeia de caracteres", "B", "1234"
<b>logico</b>	Representa valores lógicos (VERDADEIRO ou FALSO)

# Tipos

- **Exemplos:**

- Inteiro: 10 -5 -128
- Real (ponto flutuante): 1.34 13.4 -5.0
- String de caracteres: “quarta-feira” Abril”
- Lógico: VERDADEIRO (1)  
FALSO (0)

# Entrada e Saída

- Algoritmos objetivam transformar informações
  - Algoritmo = **Entrada** + **Processamento** + **Saída**
- **Entrada**: obtenção de dados provenientes do meio externo
  - Comando: **leia**
  - Exemplos:
    - **leia (X)**
    - **leia (A, NOTA)**

# Entrada e Saída

- **Saída:** entrega dos resultados ao meio externo
- Comandos: **escreva** ou **escreval**
- Exemplos:
  - **escreva** (X)
  - **escreva** (B, MEDIA, 2+2)
  - **escreval** (“cliente cadastrado com sucesso”)

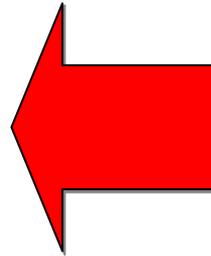
# Operadores

- **Atribuição**
- **Aritméticos**
- **Relacionais**
- **Lógicos**

# Operador de Atribuição

- Utilizado para atribuir um valor a uma variável (“=” ou “:=” ou “←”):

nome da variável



Valor

- Visualg: “:=” ou “←”
- Notação:

x1 ← 23;

temp ← x1;

nome ← “Carlos da Silva”;

# Operadores Aritméticos

- **Dados de entrada:** tipo numérico (inteiro ou real)
- **Resultado:** tipo numérico (inteiro ou real)
- Exemplos:
  - $x_2 \leftarrow 2 + 3;$
  - $x_3 \leftarrow 3 / 2;$
  - $\text{alfa} \leftarrow 1 \text{ div } 5;$
  - $\text{resto} \leftarrow 10 \% 3;$
  - $\text{resto} \leftarrow 1 \% 4;$
  - $\text{delta} \leftarrow 5 * 5 - 4 * 1 * 4;$

# Operadores Aritméticos

Operação	Operador
Adição	+
Subtração	-
Multiplicação	*
Divisão	/
Divisão Inteira	\ ou div
Exponenciação	<b>^ ou Exp (&lt;base&gt;, &lt;expoente&gt;)</b>
Resto da Divisão	%

# Operadores Relacionais

- **Dados de entrada:** tipo numérico (int ou float) ou caracteres
- **Resultado:** tipo lógico

Operador	Operação
$a < b$	a é menor que b
$a \leq b$	a é menor ou igual a b
$a > b$	a é maior que b
$a \geq b$	a é maior ou igual a b
$a = b$	a é igual a b
$a \neq b$	a não é igual a b

- Exemplos:

- cond1 ←  $2 = 3$  // (falso)
- cond2 ←  $1.6 <> 5.0$  // (verdadeiro)
- cond3 ←  $1 > 5$  // (falso)
- cond4 ←  $(1 + 2) < 5$  // (verdadeiro)
- cond5 ←  $10 >= 3$  // (verdadeiro)
- cond6 ←  $1 <= 4$  // (verdadeiro)
- cond7 ← “café” < “expresso” // (verdadeiro)
- cond8 ← “café” = “café” // (verdadeiro)
- cond9 ← “café” >= “mocha” // (falso)

- **Dados de entrada:** tipo lógico
- **Resultado:** tipo lógico
- **E (AND), OU (OR), NAO (NOT)**

Operação	Resultado
<b>a E b</b>	VERDADEIRO se ambas as partes ( <b>a</b> e <b>b</b> ) forem verdadeiras
<b>a OU b</b>	VERDADEIRO se apenas uma das partes ( <b>a</b> ou <b>b</b> ) é verdadeira.
<b>NAO a</b>	Nega uma afirmação, invertendo o seu valor lógico: se <b>a</b> for VERDADEIRO retorna FALSO, se <b>a</b> for FALSO retorna VERDADEIRO.

- Exemplos:
  - cond1 ← verdadeiro **E** falso // (falso)
  - cond2 ← falso **OU** falso //(falso)
  - cond3 ← **NAO** cond1 // (verdadeiro)
  - cond4 ← (verdadeiro **E** falso) **OU** (5 > 3)  
// (verdadeiro)

# Tabela Verdade

a	b	a E b	a OU b	NAO a	NAO b
V	V	V	V	F	F
V	F	F	V	F	V
F	V	F	V	V	F
F	F	F	F	V	V

# Prioridade dos Operadores

Operador Aritmético	Prioridade
Exponenciação	3 (maior)
Multiplicação	2
Divisão	2
Adição	1
Subtração	1 (menor)

Operador Lógico	Prioridade
E	3
OU	2
NAO	1

# Funções

- Podem ser pré-definidas da linguagem ou definidas pelo programador
- Exemplos:
  - seno(angulo)
  - exp(x,y)
  - ler(var1,var2,...)
  - escrever(resul1,result2,...)

# Funções Pré-definidas

Função	Descrição
Abs (valor : real) : real	Valor absoluto
Asc (s : caracter) : inteiro	Retorna o código ASCII
Compr (c : caracter) : inteiro	Retorna o tamanho da cadeia de caracteres
Cos (valor : real) : real	Cosseno
Exp (<base>,<expoente>)	Exponenciação
Int (valor : real) : inteiro	Converte o valor em inteiro
Log (valor : real) : real	Logaritmo de base 10
Pi : real	A constante PI
Sen (valor : real) : real	Seno
Raizq (valor : real) : real	Raiz quadrada
Quad (valor : real) : real	Elevado quadrado

Pressonando  
(CTRL+J) o  
visualg mostra  
uma Lista de  
funções  
predefinidas

# Comentários

- Utilizados para descrever texto esclarecendo trechos do código

– // (C++, Visualg) ou

– /\* (Java)

# COMANDOS



# Forma Geral do Algoritmo

```
Algoritmo "<nome do algoritmo>"  
  
var  
  
< declaração de variáveis >  
  
inicio  
  
< lista de comandos >  
  
finalgoritmo
```



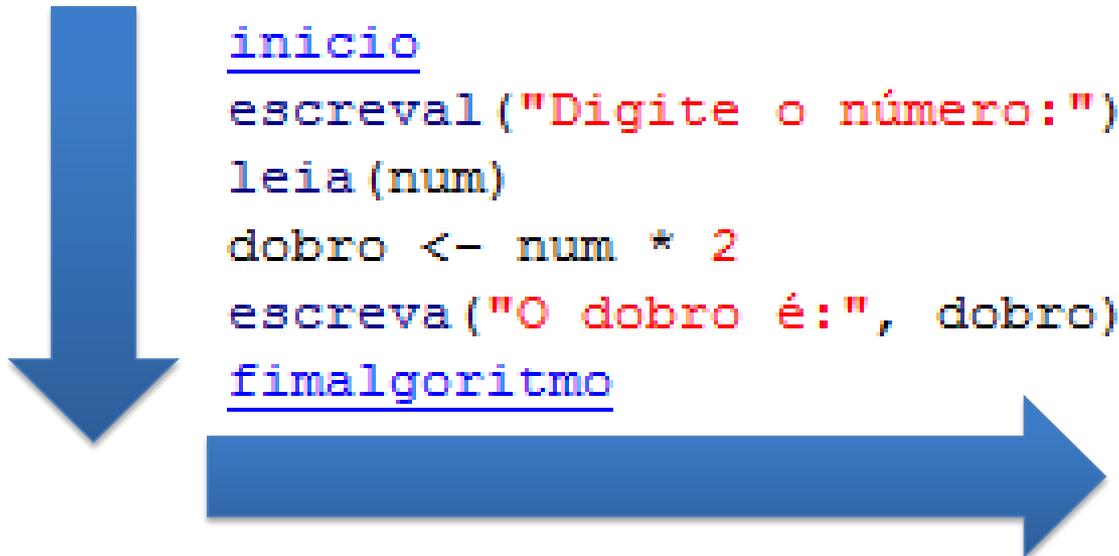
# Forma Geral

```
algoritmo "dobro"  
  
// Seção de Declarações  
var  
num, dobro :inteiro  
  
// comandos  
inicio  
escreval("Digite o número:")  
leia(num)  
dobro <- num * 2  
escreva("O dobro é:", dobro)  
fimalgoritmo
```

# Iniciando em algoritmos...

- De forma genérica, a construção de um algoritmo se resume às seguintes **etapas**:
  1. **entendimento do problema**
  2. **elaboração da solução algorítmica**
  3. **codificação da solução em pseudo-código ou linguagem de programação**
- Geralmente a etapa 2 é a mais complexa

- O fluxo de controle segue a mesma seqüência
- linear da nossa escrita, ou seja:
  - De cima para baixo;
  - Da esquerda para direita



```
inicio  
escreval("Digite o número:")  
leia(num)  
dobro <- num * 2  
escreva("O dobro é:", dobro)  
fimalgoritmo
```

# Exemplo

- Enunciado: Fazer um programa que leia dois valores reais, e calcule e exiba a média aritmética
- Uma boa prática seria decompor o problema em problemas menores!! “Dividir para conquistar”

# Exemplo

- **Passo 1**

- Qual a fórmula da média?
- A média aritmética de dois valores é calculada como  $(a+b)/2$

- **Passo 2**

- Os dados necessários serão os dois valores, que colocaremos em duas variáveis A e B do tipo real, e uma terceira variável, que chamaremos MEDIA, para armazenar a média aritmética calculada.

# Exemplo

- **Passo 3**

- A entrada dos dados neste programa é simples e direta.
- Pedir ao usuário que digite os dois valores

- **Passo 4**

- O processamento é o cálculo da média, usando a fórmula mencionada na etapa 1.
- O resultado do cálculo será armazenado na variável MEDIA.

Algoritmo "Cálculo de Média Aritmética"

var

A, B, Media : REAL

inicio

escreval ("Programa que calcula a média aritmética de dois valores.")

escreval ("Digite um valor : ")

leia (A)

escreval ("Digite outro valor : ")

leia (B)

media <- (A+B) / 2

escreval ("A média dos dois valores é : ", Media)

fimalgoritmo



**UFRPE**  
Universidade  
Federal Rural  
de Pernambuco

