

Expressões

▼ Sumário

[Dados](#)

[Informações](#)

[Dados de entrada](#)

[Dados de saída](#)

[Exemplos](#)

[Tipos de dados](#)

[Tipos básicos de dados \(primitivos\)](#)

[Inteiro](#)

[Real](#)

[Caractere ou Literal](#)

[Lógico](#)

[Constantes](#)

[Exemplo](#)

[Exemplo em pseudocódigo](#)

[Variáveis](#)

[Exemplo](#)

[Exemplo em pseudocódigo](#)

[Convenção para nomes das variáveis](#)

[Exemplos de nomes de variáveis conforme convenção](#)

[Operadores](#)

[Tipos de operadores](#)

[Operadores de atribuição \(=\)](#)

[Exemplo](#)

Operadores aritméticos
Exemplo
Mais operadores aritméticos
Exemplo
Incremento
Decremento
Exponenciação
Módulo

Operadores relacionais
Exemplo
Maior (>)
Maior ou igual (>=)
Menor (<)
Menor ou igual (<=)
Igual a (=)
Diferente de (<>)

Operadores lógicos
E (&&)
OU (||)
Não (!)

Tabela da verdade
Deduz-se:

Pseudocódigo - Nomenclatura
Exemplo de um pseudocódigo
Comandos de entrada e saída
Objetivo
Comando para entrada de dados
Comando para saída de dados
Ferramenta para treinar - VISUALG

Dados

- Utilizado para se referir ao que realmente está armazenado.
 - Exemplo: primeiro nome e sobrenome, logradouro, número, complemento.

Informações

- Utilizado para se referir ao significado dos dados para um determinado usuário.
 - Exemplo: nome e endereço.

INFORMAÇÕES x DADOS

Observamos que nossa função, basicamente, é transformar informações iniciais, que usualmente chamamos de dados, em informações finais, as respostas que serão usadas por quem utiliza um programa de computador.



Toda informação para ser útil deve ser: Fidedigna (Fidedigna, Temporal e Pertinente).

- **Fidedigna:** autêntico, genuíno, real, verdadeiro.
- **Temporal:** que passa com o tempo, transitório.
- **Pertinente:** apropriado, adequado.

Dados de entrada

Conjunto de informações que o programa precisa para o início de processamento.

Dados de saída

Conjunto de informações que o programa devolve como resposta após o processamento.

Exemplos

Para cada algoritmo pense qual a entrada e qual a saída:

- Algoritmo para calcular a idade de uma pessoa.
- Algoritmo que valida o login e senha de uma pessoa.
- Algoritmo para calcular a somatória de uma sequência de números.

Tipos de dados

Refere-se ao conjunto de valores que uma **constante**, **variável** ou **expressão** possam assumir.

No programa de computador devemos especificar os tipos de dados que serão utilizados como entrada e saída.

Exemplo: a linguagem Java é uma linguagem fortemente "tipada", ou seja, caso você tente fazer uma operação ilegal (multiplicar um caractere) o compilador dirá a você.

Tipos básicos de dados (primitivos)

Inteiro

Toda e qualquer formação numérica que pertença ao conjunto dos números inteiros relativos (negativo, nula ou positiva).

Real

Toda e qualquer informação numérica que pertença ao conjuntos dos números reais (negativa, nula ou positiva).

Caractere ou Literal

Toda e qualquer informação composta de um conjunto de caracteres alfanuméricos (0 a 9), alfabéticos (A ... Z, a ... z) e especiais (exemplos: , . \$ # ? ! @ ...).

Lógico

Toda e qualquer informação que pode assumir apenas duas situações: verdadeiro ou falso, sim ou não.

Constantes

É considerado constante um valor que **não** é modificado durante toda a execução de um algoritmo.

Exemplo

Valor do número PI = 3,1415

Exemplo em pseudocódigo

```
var PI: real  
PI <- 3,1415
```

Também são chamadas de variáveis somente de leitura.

Variáveis

São utilizadas para representar valores desconhecidos, porém necessários para a resolução de um problema. Armazenam dados temporariamente.

Contém valores que podem ser alterados durante a execução do algoritmo.

Exemplo

Idade de uma pessoa, cotação do dólar, índice de inflação.

Exemplo em pseudocódigo

```
var idadePessoa: inteiro  
var cotacaoDolar: real  
var indiceInflacao: real
```

Convenção para nomes das variáveis

1. Os nomes devem ser iniciados por um **caractere alfabético (letra) ou pelo caractere (_) sublinhado ou \$ cifrão (dólar)**.
2. Os nomes podem possuir números, desde que seja utilizado após o **primeiro caractere da palavra**.
3. Além de letras, números e (A ... Z, a ... z), cifrão (\$), o caractere sublinhado (_) e os números (0 a 9), **não é aceito nenhum outro símbolo**.
4. Para nome de variáveis devemos utilizar a **primeira letra minúscula**. As **palavras compostas internas devem iniciar com letra maiúscula**.
5. Letras fora do alfabeto ocidental, como letras gregas, **não são aceitas**.
6. Se fará diferenciação entre as **letras maiúsculas e minúsculas** nos nomes de variáveis. (Exemplo: o Java é "case sensitive", ou seja, faz diferenciação entre maiúsculas e minúsculas).
7. Não podem ser utilizadas **palavras reservadas** (palavras chaves) das linguagens.

| Podem existir variações de linguagem para linguagem.

Exemplos de nomes de variáveis conforme convenção

salario	→ CORRETO	
salHora	→ CORRETO	
ano1	→ CORRETO	
valorTotal1	→ CORRETO	
_ano	→ CORRETO	
ano_Nascimento	→ CORRETO	
SALARIO	→ CORRETO	
sal/hora	→ ERRADO	(contém o caractere "/")
sal Mensal	→ ERRADO	(contém o caractere espaço em branco)
1salmensal	→ ERRADO	(começa com número)

Operadores

São utilizados para representar expressões de cálculo, comparação, condição e atribuição.

Tipos de operadores

- Atribuição
- Aritméticos
- Relacionais
- Lógicos

Operadores de atribuição (=)

É a operação que permite armazenar um valor em uma variável e constantes.

Exemplo

Criação da variável em pseudocódigo

```
var area: inteiro
```

Alocando um valor na variável criada

```
area <- 10 // o símbolo "<-" é um operador de atribuição
```

Operadores aritméticos

Soma (+), subtração (-), multiplicação (*) e divisão (/)

São utilizados para realizar as operações matemáticas básicas.

Exemplo

Criando uma variável, neste momento o valor inicial é zero

```
var area: real
```

Alocando o resultado na variável criada anteriormente

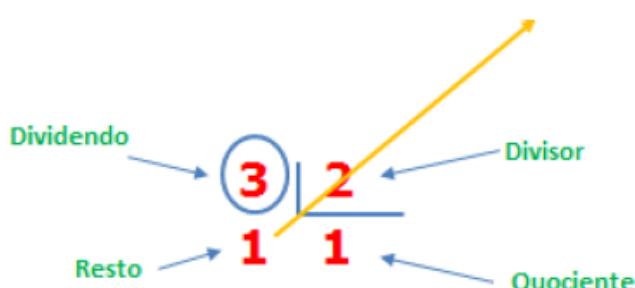
```
area <- base * altura // o símbolo "*" é um operador aritmético
```

Mais operadores aritméticos

Exemplo

Resto da divisão inteira de a por b.

```
resto <- 3 mod 2  
// resultado será igual a 1
```



Incremento

Utiliza-se uma expressão.

Por exemplo: $a + 1$

Decremento

Utiliza-se uma expressão.

Por exemplo: $a - 1$

Adiciona 1 ao valor de a.

Exponenciação

Usa-se o símbolo \wedge .

Por exemplo: 2^3 é $2\wedge 3$

Subtrai 1 do valor de a.

Módulo

Usa-se a palavra "mod".

Por exemplo: a mod b

Retorna o resto da divisão inteira de a por b.

Operadores relacionais

Permitem efetuar comparações entre duas variáveis. O resultado da comparação é sempre um valor lógico (booleano) verdadeiro ou falso.

- Igualdade ($=$)
- Menor ($<$)
- Maior ($>$)
- Menor igual ($<=$)
- Maior igual ($>=$)
- Diferente ($<>$)

Exemplo

A = 10

B = 20

A < B (Verdadeiro)

Maior ($>$)

$a > b$

Se o valor de a for maior do que o valor de b, retornará verdadeiro. Senão, retornará falso.

Menor ($<$)

$a < b$

Maior ou igual ($>=$)

$a >= b$

Se o valor de a for maior ou igual ao valor de b, retornará verdadeiro. Senão, retornará falso.

Menor ou igual ($<=$)

$a <= b$

Se o valor de a for menor do que o valor de b, retornará verdadeiro. Senão, retornará falso.

Igual a (=)

$a = b$

Se o valor de a for igual ao valor de b, retornará verdadeiro. Senão, retornará falso.

Se o valor de a for menor ou igual ao valor de b, retornará verdadeiro. Senão, retornará falso.

Diferente de (<>)

$a <> b$

Se o valor de a dor diferente do valor de b, retornará verdadeiro. Senão, retornará falso.

Operadores lógicos

São operações efetuadas com valores booleanos (true/false), resultando em valores booleanos.

- E $(\&\&)$
- Ou $(||)$
- Negação $(!)$

E ($\&\&$)

$a = 5 \&\& b <> 9$

Caso o valor de a seja igual a 5 e o valor de b seja diferente de 9, então retornará verdadeiro. Caso contrário, retornará falso.

OU ($||$)

$a = 5 || b <> 9$

Caso o valor de a seja igual a 5 ou o valor de b seja diferente de 9, então retornará verdadeiro. Se ambas as comparações retornarem falso, o resultado será falso.

Não (!)

$! a > 5$

Se o valor de a for maior do que 5, retornará falso. Caso contrário, retornará verdadeiro.

Tabela da verdade

Deduz-se:

- Para o operador `&&` (e) o resultado será verdadeiro apenas se as duas expressões assumirem o valor verdadeiro.
- Para o operador `||` (ou) o resultado será verdadeiro se pelo menos uma das expressões assumir o resultado verdadeiro.

			Operador		
			<code>&& (.e.)</code>	<code> (.ou.)</code>	<code>! (.não.)</code>
Expressão algoritmo	<code>A = 5</code>	<code>B <> 9</code>	<code>A = 5 .e. B <> 9</code>	<code>A = 5 .ou. B <> 9</code>	<code>.não. A = 5</code>
Expressão em Java	<code>A == 5</code>	<code>B != 9</code>	<code>A == 5 && B != 9</code>	<code>A == 5 B != 9</code>	<code>!A == 5</code>
Resultados possíveis	.v.	.v.	.v.	.v.	.f.
	.v.	.f.	.f.	.v.	.f.
	.f.	.v.	.f.	.v.	.v.
	.f.	.f.	.f.	.f.	.v.

Pseudocódigo - Nomenclatura

Nomenclatura	Função Realizada
Crie	Criação de variáveis, constantes, vetores e matrizes
Leia	Capturar um entrada via teclado
Calcule	Representação de expressões matemáticas
Exiba ou Escreva	Mostrar resultado no vídeo
Se	Instrução condicional
Enquanto	Instrução para indicar repetição
Para	Instrução para indicar repetição com variável de controle
Senão	Instrução para indicar caso contrário em uma estrutura condicional
Fim-Se	Instrução para encerrar uma estrutura condicional
Fim-Enquanto	Instrução para encerrar uma estrutura de repetição
Fim-Para	Instrução para encerrar uma estrutura de repetição com variável de controle
Vetor	Instrução para indicar a dimensão de um vetor
Matriz	Instrução para indicar a dimensão de uma matriz
Ir para	Instrução para indicar um desvio no fluxo do algoritmo
Imprima	Mostrar o resultado impresso

Exemplo de um pseudocódigo

Algoritmo para calcular a área de um triângulo

```
var
    base, altura, area: real

Inicio
    escreva("Informe o valor da base: ")
    leia(base)
    escreva("Informe o valor da altura: ")
    leia(altura)
    area <- (base * altura) / 2
    escreva(area)
Fim
```

Comandos de entrada e saída

Objetivo

Permitir a entrada e saída de dados (interação com usuário) durante a execução de um algoritmo.

Comando para entrada de dados

Permite capturar os dados informados através do teclado do computador.

```
// Exemplo
leia(nome)
```

Comando para saída de dados

Permite apresentar os dados resultantes do processamento realizado pelo algoritmo.

```
// Exemplo
escreva("Resultado da área: ", area)
// para saltar de linha após uma exibição utilize o comando: escreval
```

Ferramenta para treinar - VISUALG

VisuAlg 2.5 - Semnome.alg

Arquivo Editar Exibir Algoritmo Código Ferramentas Ajuda

0.5 s De: 0 Até: 100 Decimais: 0

```
algoritmo "semnome"
// Função :
// Autor :
// Data : 13/02/2014
// Seção de Declarações
var

início
// Seção de Comandos

fimalgoritmo
```

Escopo	Nome	Tipo	Valor

12:67 Modificado Use Ctrl+J para acessar a lista de comandos e funções do Visualg 2.5