

# **INSTRUÇÕES PRIMITIVAS**

# Instruções primitivas

- São os componentes básicos que efetuam tarefas essenciais para a operação dos computadores, como entrada e saída de dados e a movimentação dos mesmos na memória.

# Dispositivo de entrada

- É o meio pelo qual as informações são transferidas pelo usuário ou pelos níveis secundários de memória ao computador.
  - Exemplos: teclado, fitas, discos magnéticos, mouse, scanner.

# Dispositivo de saída

- É o meio pelo qual as informações (geralmente os resultados da execução de um programa) são transferidas pelo computador ao usuário ou aos níveis secundários de memória.
  - Exemplos: vídeo, impressora, fitas, discos magnéticos.

# Sintaxe

- É a forma como os comandos devem ser escritos, a fim de que possam ser entendidos pelo tradutor de programas.

# Semântica

- É o significado, ou seja, o conjunto de ações que serão exercidas pelo computador durante a execução do referido comando.

# Instrução de atribuição

- É a principal maneira de se armazenar uma informação numa variável.
  - Sintaxe: <nome\_da\_variável> ← <expressão>
  - Semântica:
    1. Avaliação da expressão
    2. Armazenamento do valor resultante na posição de memória correspondente à variável que aparece à esquerda do comando.

# Instrução de atribuição (continuação)

- Importante: na instrução de atribuição deve haver compatibilidade entre o tipo de dado resultante da avaliação da expressão e o tipo de dado da variável (a não ser, propositadamente, com tipos numéricos).

## — Exemplos:

area ← base \* altura

delta ← b \*\* 2 - 4.0 \* a \* c

contador ← contador + 1

nome ← “Caetano Veloso”

music ← “Dias de Luta” + “ e Flores em Você”

w ← verdadeiro

p ← x > y .E. y > z

q ← (a >= 0 .E. a <= 10) .OU. (a >= 100 .E. a <= 1000)



# Instrução de entrada de dados

- Sintaxe: `leia ( <lista_de_variáveis> )`
- Semântica: os dados são fornecidos ao computador por meio de um dispositivo de entrada e armazenados nas posições de memória das variáveis cujos nomes aparecem na lista.
  - Exemplo:
    - `leia ( x )`
    - `leia ( a, b, c )`

# Instrução de saída de dados

- Sintaxe: escreva ( <lista\_de\_expressões> )
- Semântica: os argumentos são enviados para o dispositivo de saída.
  - No caso de uma lista de variáveis, o conjunto de cada uma delas é pesquisado na posição de memória correspondente a variável.
  - No caso de argumento constante (número, literal ou lógico) este é enviado diretamente ao referido dispositivo.
  - E no caso de expressões, após sua avaliação, segue como uma constante.

– Exemplos:

# Interface com o usuário: fase de execução

- Regras básicas:
  1. Toda vez que um programa estiver esperando que o usuário forneça a ele um determinado dado (operação de leitura), ele deve antes enviar uma mensagem dizendo o que o usuário deve digitar, por meio de um instrução de saída.
  2. Antes de enviar qualquer resultado ao usuário, um programa deve escrever uma mensagem explicando o significado do mesmo.

# Linhas de comentários

- São observações que colocamos nos algoritmos, mas que não queremos que sejam confundidas com instruções. Para isto, devemos por no início de cada linha de comentário duas barras (//).
  - Exemplo:
    - //Início do cálculo da área do pentágono.
    - //Área do pentágono = semi-perímetro \* apótema

# Exemplos de algoritmos

- Dado o preço unitário e a quantidade de um produto, imprimir o valor da compra.

```
algoritmo "Valor da compra"
    real preco_unit , preco_total
    inteiro quantid
início
        escreva( "Programa que calcula o preco de uma certa
quantidade de um
                produto. ")
        escreva( "Digite o preco unitário e a quantidade: ")
        leia( preco_unit , quantid )
        preco_total ← preco_unit * quantid
        escreva( "Preco total = " , preco_total )
fim
finalgoritmo
```

# Exemplos de algoritmos

- Calcular a área e o perímetro de um retângulo, sendo dados as medidas dos lados.

algoritmo “retângulo”

real l1, l2, area, perimetro

início

escreva( “Digite as medidas dos lados do retângulo: “)

leia( l1 , l2 )

area  $\leftarrow$  l1 \* l2

perimetro  $\leftarrow$  2 \* ( l1 + l2 )

escreva( “O valor da área é : “ , area )

escreva( “O valor do perímetro é : “ , perimetro )

fim

fimalgoritmo

# Exemplos de algoritmos

- Algoritmo para adivinhar a soma de cinco números com, no máximo, 4 dígitos, com duas parcelas sendo geradas sem serem digitadas.

```
algoritmo "Advinha"  
  inteiro x  
  início  
    escreva ( "Digite um número com 4 algarismos: ")  
    leia ( x )  
    escreva ( " O resultado da nossa conta será: ", 19998+x )  
    escreva ( " Digite o segundo número com 4 dígitos: ")  
    leia ( x )  
    escreva ( " O meu número <terceiro>é : ", 9999 - x )  
    escreva ( " Digite o quarto número com 4 dígitos: " )  
    leia ( x )  
    escreva ( " Finalmente o quinto número é : ", 9999 - x )  
  fim  
fimalgoritmo
```

# Rastreamento de um algoritmo

- O rastreamento de um algoritmo consiste na **execução manual**, com **dados representativos** para **registrar os valores tomados** pelas **variáveis** em **cada passo do algoritmo**.
- Para facilitar o acompanhamento, colocamos todos os dados numa tabela de variáveis.
- **Devemos fazer tantos testes quantos forem necessários para nos convenceremos de que o algoritmo está perfeito.**



# Exemplo de rastreamento de um algoritmo

x  
3452  
1538  
5172

## Saída

Digite um número com 4 algarismos: 3452  
O resultado da nossa conta será: 23450  
Digite o segundo número com 4 dígitos: 1538  
O meu número <terceiro> é: 8461  
Digite o quarto número com 4 dígitos: 5172  
Finalmente o quinto número é: 4827