Universidade Federal do Rio Grande do Norte
Centro de Tecnologia
Departamento de Computação e Automação

ALGORITMOS

Professor Responsável:
Luiz Affonso Henderson Guedes de Oliveira
Prof. Do Estágio Docente:
Kliger Kissinger F. Rocha
Valnaide Gomes Bittencourt

Turma:
Engenharia Química – 2004.1

atal, RN, maio/2004

Primeira Aula – Teórica

- Introdução
  - Conceitos de algoritmos
- Formas de Representação de Algoritmos
  - Descrição Narrativa
  - Fluxograma Convencional
  - Pseudocódigo

Conceito de Algoritmo

- **“Algoritmo é um conjunto finito de regras, bem definidas, para a solução de um problema em um tempo finito e com um número finito de passos.”**
- "Serve como modelo para programas, pois sua linguagem é intermediária à linguagem humana e às linguagens de programação, sendo então, uma boa ferramenta na validação da lógica de tarefas a serem automatizadas.“
- “Os algoritmos, servem para representar a solução de qualquer problema, mas no caso do Processamento de Dados, eles devem seguir as regras básicas de programação para que sejam compatíveis com as linguagens de programação.”

Conceito de Algoritmo

- **Para se ter um algoritmo é necessário:**
  - **Que um número finito de passos;**
  - **Que cada passo esteja precisamente definido, sem possíveis ambigüidades;**
  - **Que existam zero ou mais entradas tomadas de conjuntos bem definidos;**
  - **Que existam uma ou mais saídas;**
  - **Que exista uma condição de fim sempre atingida para quaisquer entradas e num tempo finito.**

*Algoritmos **não** se aprendem:*
– Copiando algoritmos
– Estudando algoritmos

*Algoritmos **só** se aprendem:*
– Construindo algoritmos
– Testando algoritmos

Formas de Representação de Algoritmos

- Dentre as formas de representação de algoritmos mais conhecidas podemos citar:
  - **Descrição Narrativa**;
  - **Fluxograma Convencional**;
  - **Pseudocódigo**, também conhecido como Linguagem Estruturada ou Portugol.

Formas de Representação de Algoritmos

- Descrição Narrativa
  - Nesta forma de representação os algoritmos são expressos diretamente em **linguagem natural**.

***Receita de bolo**:*
Misture os ingredientes
Unte a forma com manteiga
Despeje a mistura na forma
Se houver coco ralado
então despeje sobre a mistura
Leve a forma ao forno
Enquanto não corar
deixe a forma no forno
Retire do forno
Deixe esfriar

Tomando um banho:
Entrar no banheiro e tirar a roupa
Abrir a torneira do chuveiro
Entrar na água
Ensaboar-se
Sair da água
Fechar a torneira
Enxugar-se
Vestir-se

Formas de Representação de Algoritmos

- Descrição Narrativa

Troca de um pneu furado

Afrouxar ligeiramente as porcas
Suspender o carro
Retirar as porcas e o pneu
Colocar o pneu reserva
Apertar as porcas
Abaixar o carro
Dar o aperto final nas porcas

Cálculo da média de um aluno

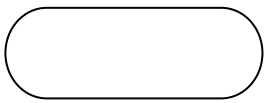
Obter as suas 2 notas de provas
Calcular a média aritmética
Se a média for maior que 7,
o aluno foi aprovado,
senão ele foi reprovado

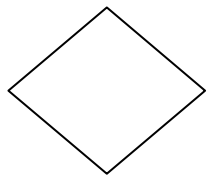
Formas de Representação de Algoritmos

- Fluxograma Convencional
  - É uma representação gráfica de algoritmos onde formas geométricas diferentes implicam ações (instruções, comandos) distintos.
  - Tal propriedade facilita o entendimento das idéias contidas nos algoritmos e justifica sua popularidade
  - Esta forma é aproximadamente intermediária à descrição narrativa e ao pseudocódigo (subitem seguinte), pois é menos imprecisa que a primeira e, no entanto, não se preocupa com detalhes de implementação do programa

Formas de Representação de Algoritmos

- Fluxograma Convencional


Inicio e Fim de Programa


Decisão

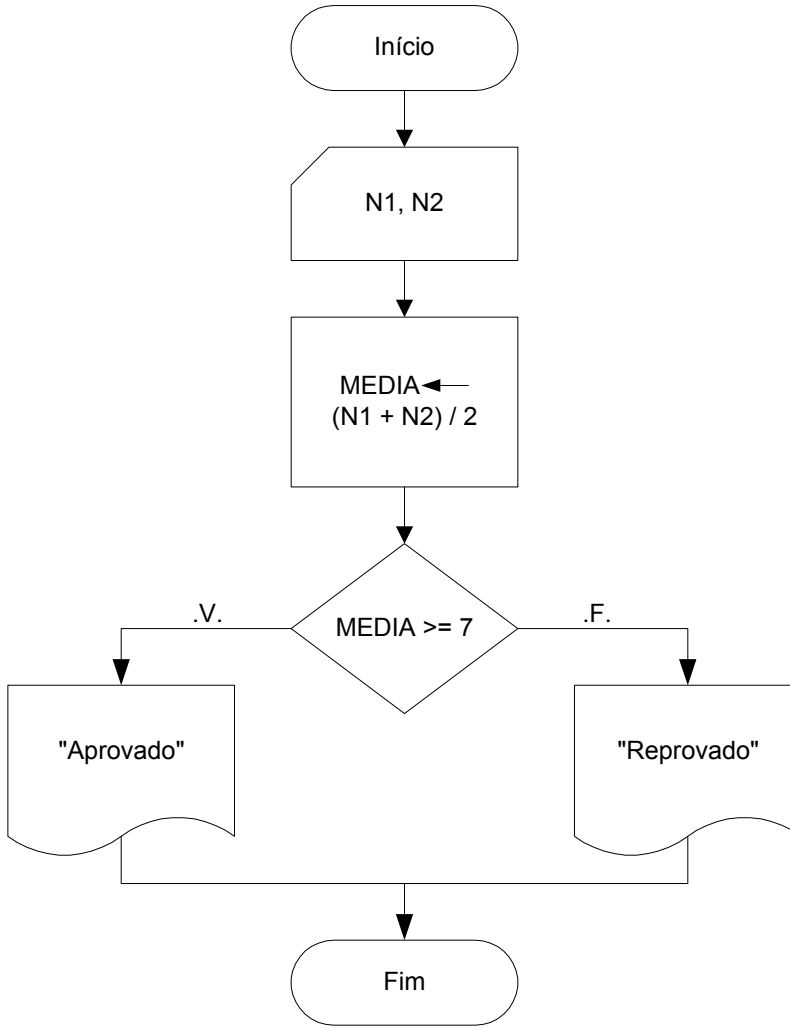

Operação de Atribuição


Operação de Saída


Operação de Entrada de Dados

Formas de Representação de Algoritmos

- Fluxograma Convencional
  - Exemplo: Cálculo da média de um aluno sob a forma de um fluxograma.
  - Partindo do símbolo inicial, há sempre um único caminho orientado a ser seguido, representando a existência de uma única seqüência de execução das instruções



Formas de Representação de Algoritmos

- **Pseudocódigo**
  - Esta forma de representação de algoritmos é rica em detalhes, como a definição dos tipos das variáveis usadas no algoritmo. Por assemelhar-se bastante à forma em que os programas são escritos, encontra muita aceitação.
  - Na verdade, esta representação é suficientemente geral para permitir a tradução de um algoritmo nela representado para uma linguagem de programação específica seja praticamente direta.

Formas de Representação de Algoritmos

- **Pseudocódigo**
  - A forma geral da representação de um algoritmo na forma de pseudocódigo

Algoritmo <nome_do_algoritmo>
<declaração_de_variáveis>
<subalgoritmos>
Início
<corpo do algoritmo>
Fim

- **Algoritmo** é uma palavra que indica o início da definição de um algoritmo em forma de pseudocódigo.
- **<nome_do_algoritmo>** é um nome simbólico dado ao algoritmo com a finalidade de distingui-los dos demais.
- **<declaração_de_variáveis>** consiste em uma porção opcional onde são declaradas as variáveis globais usadas no algoritmo principal e, eventualmente, nos subalgoritmos.
- **<subalgoritmos>** consiste de uma porção opcional do pseudocódigo onde são definidos os subalgoritmos.
- **Início** e **Fim** são respectivamente as palavras que delimitam o início e o término do conjunto de instruções do corpo do algoritmo.

Formas de Representação de Algoritmos

- **Pseudocódigo**
  - Representação do algoritmo do cálculo da média de um aluno, na forma de um pseudocódigo

```
Algoritmo Calculo_Media
    Var N1, N2, MEDIA: real
Início
      Leia N1, N2
      MEDIA ← (N1 + N2) / 2
      Se MEDIA >= 7 então
            Escreva "Aprovado"
      Senão
           Escreva "Reprovado"
      Fim_se
Fim
```

Formas de Representação de Algoritmos

- **Síntese**
  - Há diversas formas de representação de algoritmos que diferem entre si pela quantidade de detalhes de implementação que fornecem ou, inversamente, pelo grau de abstração que possibilitam com relação à implementação do algoritmo em termos de uma linguagem de programação específica.

  - Dentre as principais formas de representação de algoritmos destacam-se: a **descrição narrativa**, o **fluxograma convencional** e o **pseudocódigo** (ou linguagem estruturada).

Segunda Aula – Teórica

- Tipos de dados
- Variáveis
  - Armazenamento de dados na memória
  - Conceito e utilidade de variáveis
  - Definição de variáveis em algoritmos
  - Mapeamento de variáveis na memória
- Expressões
  - Conceitos
  - Operadores
  - Tipos de Expressões
  - Avaliação de Expressões

Tipos de Dados

- Dados Numéricos
  - Tornando ao aspecto computacional, os dados numéricos representáveis num computador são divididos em apenas duas classes: os **inteiros** e os **reais**
- Dados Numéricos Inteiros
  - Os números **inteiros** são aqueles que não possuem componentes decimais ou fracionários, podendo ser positivos ou negativos. **(Conj. N e Z)**

Ex.:24 - número inteiro positivo
0 - número inteiro
-12 - número inteiro negativo

Tipos de Dados

Dados Numéricos Reais

- Os dados de tipo **real** são aqueles que podem possuir componentes decimais ou fracionários, e podem também ser positivos ou negativos.

Exemplos de dados do tipo **real**:

24.01	- número real positivo com duas casas decimais
144.	- número real positivo com zero casas decimais
-13.3	- número real negativo com uma casa decimal
0.0	- número real com uma casa decimal
0.	- número real com zero casas decimais

Tipos de Dados

- Dados Literais
  - O tipo de dado **literal** é constituído por uma seqüência de caracteres contendo letras, dígitos e/ou símbolos especiais.
  - Este tipo de dados é também muitas vezes chamado de **alfanumérico**, **cadeia** (ou **cordão**) **de caracteres**, ainda, do inglês, **string**.
  - Usualmente, os dados literais são representados nos algoritmos pela coleção de caracteres, delimitada em seu início e término com o caractere aspas (**"**).
  - Diz-se que o dado do tipo literal possui um comprimento dado pelo número de caracteres nele contido

Tipos de Dados

- Exemplos de dados do tipo literal:
  - "QUAL ?" - literal de comprimento 6
  - " " - literal de comprimento 1
  - "qUaL ?!$" - literal de comprimento 8
  - " AbCdefGHi" - literal de comprimento 9
  - "1-2+3=" - literal de comprimento 6
  - “0” - literal de comprimento 1
  - Note que, por exemplo, "1.2" representa um dado do tipo **literal** de comprimento 3, constituído pelos caracteres "1", "." e "2", diferindo de 1.2 que é um dado do tipo **real**.

Tipos de Dados

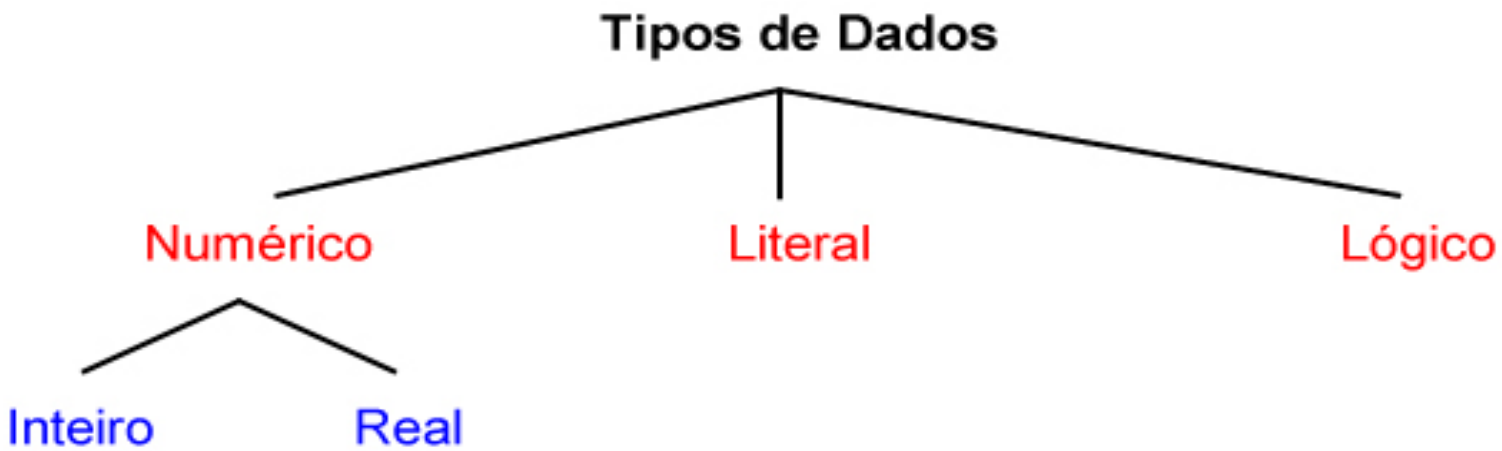
- Dados Lógicos (**booleanos**)
  - O tipo de dados **lógico** é usado para representar dois únicos valores lógicos possíveis: **verdadeiro** e **falso**. É comum encontrar-se em outras referências outros tipos de pares de valores lógicos como **sim/não**, **1/0**, **true/false**.
  - Nos algoritmos apresentados nesta apostila os valores lógicos serão delimitados pelo caractere ponto (**.**).
    - Exemplo: .V. - valor lógico verdadeiro
      .F. - valor lógico falso

Tipos de Dados

- Síntese
  - Os dados numéricos dividem-se em duas classes:
    - **inteiros**, que não possuem parte fracionária e podem ser positivos ou negativos;
    - **reais**, que podem possuir parte fracionária e podem ser positivos ou negativos.
    - Os dados do tipo **literal** podem conter seqüências de letras, dígitos ou símbolos especiais, delimitados por aspas (**"**). Seu comprimento é dado pelo número de caracteres em **string**.
    - Os dados do tipo **lógico** só possuem dois valores possíveis (**.V.** e **.F.**).

Tipos de Dados

- Síntese
  - A árvore abaixo resume a classificação dos dados com relação aos tipos de dados apresentados.



Variáveis

- Armazenamento de dados na memória
  - A todo momento durante a execução de qualquer tipo de programa os computadores estão manipulando informações representadas pelos diferentes tipos de dados descritos anteriormente.
  - Para que não se "esqueça" das informações, o computador precisa guardá-las em sua memória.

Variáveis

- Conceito e Utilidade de Variáveis
  - Basicamente, uma variável possui três atributos: um **nome**, um **tipo de dado** associado à mesma e a **informação** por ela guardada.
    - Um nome de variável deve necessariamente começar com uma letra;
    - Um nome de variável não deve conter nenhum símbolo especial exceto a sublinha (_).

SALARIO	***= correto***
1ANO	***= correto***
A CASA	***= errado (contém o caractere espaço em branco)***
SAL/HORA	***= errado (contém o caractere "/")***
SAL_HORA	***= correto***
_DESCONTO	***= errado (não começou com uma letra)***

Variáveis

- Definição de variáveis em algoritmos
  - Todas as variáveis utilizadas em algoritmos devem ser definidas antes de serem utilizadas.
  - Isto se faz necessário para permitir que o compilador reserve um espaço na memória para as mesmas.
  - Sintaxe:
    - *VAR <nome_da_variável> : <tipo_da_variável>*
    - *VAR <lista_de_variáveis> : <tipo_das_variáveis>*

      - *a palavra-chave **VAR** deverá estar presente sempre e será utilizada uma única vez na definição de um conjunto de uma ou mais variáveis;*

Variáveis

- Definição de variáveis em algoritmos

```
VAR   NOME        :    literal[10]
      IDADE       :    inteiro
      SALARIO     :    real
      TEM_FILHOS  :    lógico
```

Variáveis

- Síntese
  - A memória dos computadores é composta por células numeradas ordenadamente denominadas **bytes**. Cada byte é constituído por 8 **bits**.
  - Cada tipo de dado requer um número diferente de bytes para armazenar a informação representada por ele na memória. Esta quantidade também pode variar em função do tipo de computador considerado.
  - Uma **variável** é uma entidade dotada de um **nome** para diferenciá-la das demais e um **tipo de dado** que define o tipo de informação que ela é capaz de guardar. Uma vez definidos, o nome e o tipo de uma variável não podem ser alterados no decorrer de um programa. Por outro lado, a informação útil da variável é objeto de constante modificação durante o decorrer do programa, de acordo com o fluxo de execução do mesmo.

Exercício – 2ª Aula

Classifique os dados especificados abaixo de acordo com seu tipo, assinalando com **I** os dados do tipo inteiro, com **R** os reais, com **L** os literais, com **B** os lógicos (booleanos), e com **N** aqueles para os quais não é possível definir *a priori* um tipo de dado.

( ) 0.21	( ) 1	( ) V	( ) “0.”	( ) 1%	( ) “José”
( ) 0,35	( ) .F.	( ) -0.001	( ) .T.	( ) +3257	( ) “a”
( ) “+3257”	( ) +3257.	( ) “-0.0”	( ) “.F.”	( ) ± 3	( ) .V.
( ) .V		( ) “abc”	( ) F	( ) C	( ) Maria ( ) +36

Exercício – 2ª Aula

1. Explique o que está errado nos identificadores incorretos.

( ) valor	( ) _b248	( ) nota*do*aluno
( ) a1b2c3	( ) 3 x 4	( ) Maria
( ) km/h	( ) xyz	( ) nome empresa
( ) sala_215	( ) “nota”	( ) ah!

2. Supondo que as variáveis NB, NA, NMAT e SX sejam utilizadas para armazenar a nota do aluno, o nome do aluno, o número da matrícula e o sexo, declare-as corretamente, associando o tipo adequado ao dado que será armazenado.

Expressões

- Conceito
  - O conceito de **expressão** em termos computacionais está intimamente ligado ao conceito de expressão (ou fórmula) matemática, onde um conjunto de variáveis e constantes numéricas relacionam-se por meio de operadores aritméticos compondo uma fórmula que, uma vez avaliada, resulta num valor

AREA = BASE * ALTURA * 0,5

Expressões

- Operadores
  - **Operadores** são elementos funcionais que atuam sobre **operandos** e produzem um determinado resultado.
  - De acordo com o número de operandos sobre os quais os operadores atuam, os últimos podem ser classificados em:
    - **binários**
    - **unários**
    - relacionais

Expressões

Tipos de Expressões

- *Expressões Aritméticas*: são aquelas cujo resultado da avaliação é do tipo numérico, seja ele inteiro ou real. Somente o uso de operadores aritméticos e variáveis numéricas é permitido em expressões deste tipo.

Tabela 5.1 Operadores aritméticos e sua ordem de prioridade.

Operador	Tipo	Operação	Prioridade
+	Binário	Adição	4
-	Binário	Subtração	4
*	Binário	Multiplicação	3
/	Binário	Divisão	3
**	Binário	Exponenciação	2
+	Unário	Manutenção de sinal	1
-	Unário	Inversão de sinal	1

Expressões

- Tipos de Expressões
  - *Expressões Lógicas*: são aquelas cujo resultado da avaliação é um valor lógico (**.V.** ou **.F.**).

Tabela 5.2 Operadores lógicos e suas relações de prioridade.

Operador	Tipo	Operação	Prioridade
.OU.	Binário	Disjunção	3
.E.	Binário	Conjunção	2
.NÃO.	Unário	Negação	1

Expressões

- Tipos de Expressões
  - *Expressões Literais:* são aquelas cujo resultado da avaliação é um valor literal.
    - Os tipos de operadores existentes variam de uma linguagem de programação para outra, não havendo uma padronização.

"**REFRIGERA**" + "**DOR**" e o resultado de sua avaliação é "**REFRIGERADOR**

Expressões

- Tipos de Expressões
  - Regras são essenciais para a correta avaliação de expressões
    - Operadores de maior prioridade devem ser avaliados primeiro. Em caso de empate, a avaliação se faz da esquerda para a direita
    - O uso de parênteses em sub-expressões força a avaliação das mesmas com maior prioridade
    - Os diversos tipos de operadores devem ser avaliados na seguinte seqüência dentro de uma expressão complexa: primeiro os aritméticos e literais; a seguir, os relacionais e, por último, os lógicos

Expressões

- Síntese
  - Uma ***expressão*** é uma combinação de variáveis, constantes e operadores, que resulta num valor quando avaliada.
  - ***Operadores*** são elementos funcionais que atuam sobre ***operandos***. Segundo o número de operandos sobre os quais atua, um operador pode ser classificado em ***unário*** ou **binário**. Segundo os tipos de dados de seus operandos e do valor resultante de sua avaliação, os operadores podem ser classificados em aritméticos, lógicos ou literais.
  - Um tipo especial de operador é o ***relacional***, que é usado na comparação de operandos de um mesmo tipo de dado e cujo resultado da avaliação é sempre um valor lógico.

Expressões

- Síntese
  - As expressões são classificadas de acordo com o valor resultante de sua avaliação em:
    - **Aritméticas**, que resultam num valor numérico (real ou inteiro);
    - **lógicas**, que resultam num valor lógico;
    - **literais**, que resultam num valor literal

Expressões

- Síntese
  - As expressões são classificadas de acordo com o valor resultante de sua avaliação em:
    - **Aritméticas**, que resultam num valor numérico (real ou inteiro);
    - **lógicas**, que resultam num valor lógico;
    - **literais**, que resultam num valor literal

Exercício – 2ª Aula

1. Dada a declaração de variáveis:

```
VAR     A, B, C    :    inteiro
        X, Y, Z    :    real
        NOME, RUA  :    literal[20]
        L1, L2     :    lógico
```

Classifique as expressões seguintes de acordo com o tipo de dado do resultado de sua avaliação, em **I** (inteiro), **R** (real), **L** (literal), **B** (lógico) ou **N** (quando não for possível defini-lo):

( ) A + B + C ( ) A + B + Z ( ) NOME + RUA ( ) A B

( ) A Y ( ) NOME RUA ( ) L1 .OU. L2 ( ) RUA <> NOME

( ) A + B / C ( ) A + X / Z ( ) A + Z / A ( ) A B = L1

( ) (A = B) ( ) X + Y / Z ( ) X = Z / A ( ) L1 ** L2

( ) A + B / L2 ( ) X < L1 / RUA

Exercício – 2ª Aula

2. Para as mesmas variáveis declaradas no exercício 1, às quais são dados os valores seguintes:

A = 1 **B** = 2 **C** = 3 **X** = 2.0 **Y** = 10.0 **Z** = -1.0

L1 = .V. **L2** = .F. **NOME** = “PEDRO” **RUA** = “PEDRINHO”

A + C / B → _____

A + B + C → _____

C / B / A → _____

-X ** B → _____

-(X ** B) → _____

-NOME + RUA → _____

-L1 .OU. L2 → _____

-(L2 .E. (.NÃO. L1)) → _____

Terceira Aula – Teórica

- Instruções Primitivas
  - Instrução Primitiva de Atribuição
  - Instrução Primitiva de Saída de Dados
  - Instrução Primitiva de Entrada de Dados
- Controle de Fluxo de Execução
  - Comandos Compostos
  - Estrutura seqüencial
  - Estruturas de Decisão
  - Estruturas de Repetição
  - Aninhamentos

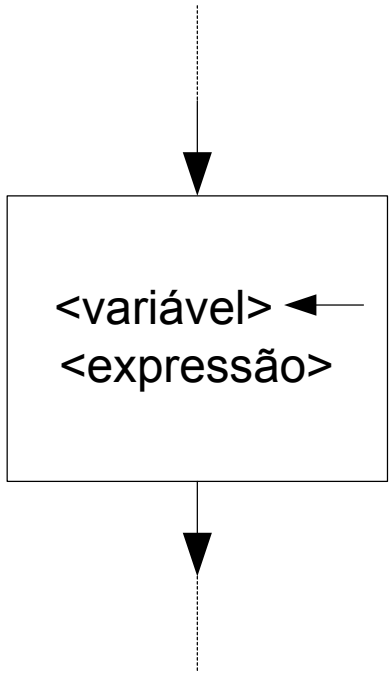
Instruções Primitivas

- Como o próprio nome diz, ***Instruções Primitivas*** são os comandos básicos que efetuam tarefas essenciais para a operação dos computadores, como entrada e saída de dados (comunicação com o usuário e com os dispositivos periféricos), e movimentação dos mesmos na memória.
  - Dispositivo de entrada

Instruções Primitivas

- Instrução Primitiva de Atribuição
  - A ***instrução primitiva de atribuição***, ou simplesmente ***atribuição***, é a principal maneira de se armazenar uma informação numa variável.
  - Sua sintaxe é:

<nome_de_variável> ← <expressão>

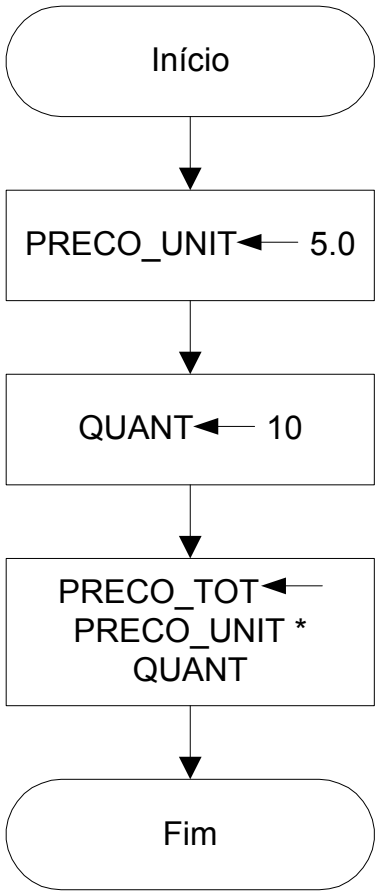


Instruções Primitivas

- Instrução Primitiva de Atribuição

Pseudocódigo

```
Algoritmo EXEMPLO
Var PRECO_UNIT, PRECO_TOT : real
    QUANT : inteiro
Início
    PRECO_UNIT ← 5.0
    QUANT ← 10
    PRECO_TOT ← PRECO_UNIT * QUANT
Fim.
```



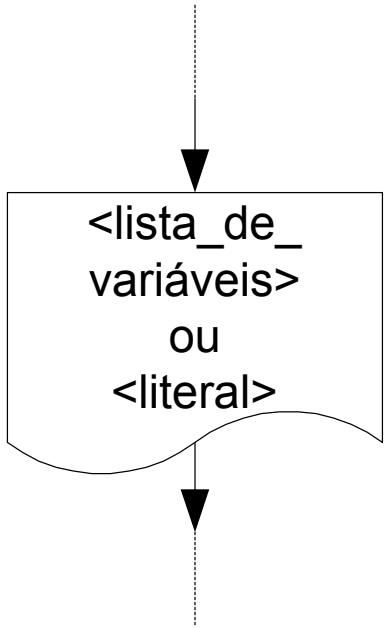
Instruções Primitivas

- Instrução Primitiva de Saída de Dados
  - As instruções primitivas de saída de dados são o meio pelo qual informações contidas na memória dos computadores são colocadas nos dispositivos de saída, para que o usuário possa visualizá-las.
  - Há duas sintaxes possíveis para esta instrução:

Escreva <lista de variáveis>

ou

Escreva <literal>

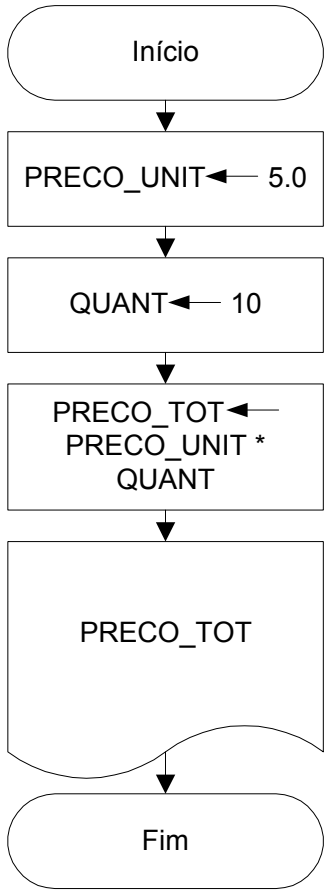


Instruções Primitivas

- Instrução Primitiva de Saída de Dados

Pseudocódigo

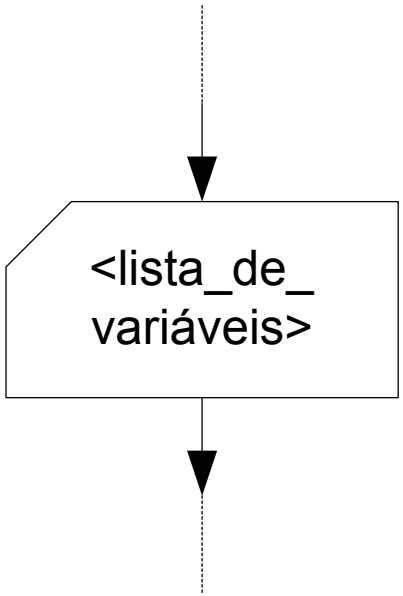
```
Algoritmo EXEMPLO
Var PRECO_UNIT, PRECO_TOT : real
    QUANT : inteiro
Início
    PRECO_UNIT ← 5.0
    QUANT ← 10
    PRECO_TOT ← PRECO_UNIT * QUANT
    Escreva PRECO_TOT
Fim.
```



Instruções Primitivas

- Instrução Primitiva de Entrada de Dados
  - As instruções primitivas de entrada de dados são o meio pelo qual informações são fornecidas ao computador para serem processadas.
  - Sua sintaxe é:

    Leia <lista_de_variáveis>

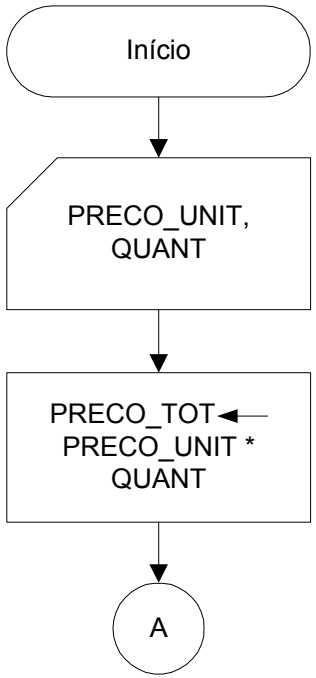


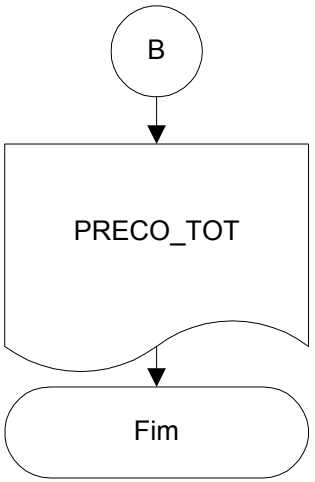
Instruções Primitivas

- Instrução Primitiva de Entrada de Dados

Pseudocódigo

```
Algoritmo EXEMPLO
Var PRECO_UNIT,
      PRECO_TOT : real
      QUANT : inteiro
Início
      Leia PRECO_UNIT, QUANT
      PRECO_TOT ← PRECO_UNIT * QUANT
      Escreva PRECO_TOT
Fim.
```





Instruções Primitivas

- Síntese

  - A ***instrução primitiva de atribuição*** avalia uma expressão e armazena o valor resultante numa variável. O valor resultante da expressão e a variável devem ter tipos compatíveis.

  - A ***instrução primitiva de saída de dados*** admite como argumentos uma lista de variáveis, um literal, ou uma mistura de ambos. No primeiro caso, o valor de cada uma das variáveis é buscado na memória e colocado no dispositivo de saída. No caso de literais, estes são copiados diretamente no dispositivo de saída.

  - A ***instrução primitiva de entrada de dados*** busca, no dispositivo de entrada, dados que são guardados nas posições de memória correspondentes às variáveis da lista que lhe são passadas como argumento.