



# Programação de Computadores I

## Estrutura de um Programa

### Procedimentos de Entrada e Saída

PROFESSORA CINTIA CAETANO

# Estrutura de um programa

---

- ▶ Todo programa é subdividido em três áreas distintas: cabeçalho do programa, área de declarações e corpo do programa.
  1. **Cabeçalho do Programa.**
  2. **Área de Definições e declarações** - área utilizada para validar o uso de qualquer tipo de identificador que não seja predefinido.
  3. **Programa principal** – área utilizada para a escrita do programa propriamente dito.

# Forma Geral de um ALGORITMO

---

- ▶ A estrutura geral de um algoritmo é:

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

# Cabeçalho do Programa

---

- ▶ Área utilizada para se fazer a identificação do programa com um nome, ou seja, serve apenas para nomear o programa.
- ▶ O cabeçalho deve ser iniciado com a palavra **Algoritmo..**
- ▶ Exemplo:
- ▶ Algoritmo “teste”

# Área de Definições e Declarações

---

## ▶ Declaração de Constantes

- ▶ Constantes são utilizadas para associar um valor fixo a uma variável.
- ▶ A área das constantes deve começar com a palavra reservada **CONST**.
  - ▶ Exemplo: `Const MAX<- 130`
- ▶ Uma convenção utilizada por programadores é colocar as constantes em letras maiúsculas.

# Área de Definições e Declarações

---

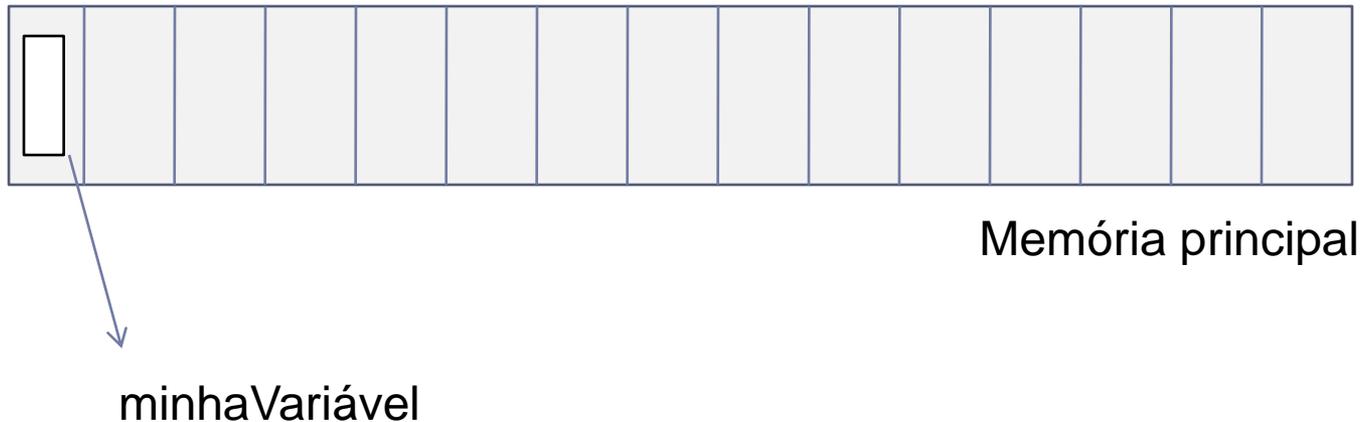
## ▶ Declaração de variáveis

- ▶ A declaração aloca um espaço na memória do computador, fornecendo um lugar para se colocar este dado específico.
- ▶ A declaração não guarda nenhum valor no espaço que foi reservado.
- ▶ O valor da variável fica indefinido até que a mesma receba um valor através da atribuição.
- ▶ Todas as variáveis devem ser declaradas na área de declaração de variáveis, que começa com a palavra reservada VAR.

# Declaração de Variáveis

---

- ▶ É um local (endereço) na memória principal utilizado pelo programa para armazenar um determinado conteúdo.



# Declaração de Variáveis

---

## ► Regras:

1. Começar com uma letra (a..z) ou por um sinal de sublinhado/underscore (\_).
2. Todos os outros caracteres devem ser uma letra (a..z), um número (0..9) ou um sinal de sublinhado (\_).
3. Não pode ter mais de 127 caracteres. DICA: é aconselhável utilizar no máximo 10 caracteres.
4. Não há diferença entre maiúsculas e minúsculas.
5. Os identificadores não podem ser palavras reservadas (nomes de comandos, procedimentos, etc.). Exemplo: se, então, enquanto, etc.

# Declaração de Variáveis: Sintaxe

---

## ▶ Sintaxe:

- ▶ nome\_da\_variável: tipo\_da\_variável

## ▶ Exemplos:

- ▶ idade: inteiro
- ▶ salario: real
- ▶ sexo: caracter
- ▶ nome: literal
- ▶ temCelular: lógico



# Declaração de Variáveis: Sintaxe

---

- ▶ Até que se atribua um valor a variável, a variável pode conter “lixo” da memória.
- ▶ Essa atribuição pode ser feita pelo operador de atribuição = ou pela seta para esquerda ←
- ▶ **Exemplo:**
  - ▶ Salario = 1000 (foi atribuído um valor diretamente)
  - ▶ read(salario) (observe que aqui o valor está sendo lido, ou seja, solicitado ao usuário)
  - ▶ inss ← salario \* taxa\_inss (foi atribuído o resultado de uma expressão)

# Área do Programa Principal

---

- ▶ Contém os comandos que definem a ação do programa: atribuição, controle de fluxo, procedimentos, funções, etc.
- ▶ Começa com a palavra reservada INICIO e termina com outra palavra reservada FIMALGORITMO.

# Procedimentos de Entrada e Saída

---

- ▶ Os procedimentos de entrada e saída permitem uma maior interação com o usuário, permitindo não só a visualização de informações na tela, bem como a entrada de dados via teclado.
- ▶ As entradas dos programas são armazenadas em variáveis.

# Procedimentos de Entrada

---

- ▶ Qual o dispositivo padrão de entrada?
  - ▶ Teclado!



# Procedimentos de Entrada

---

- ▶ Comando para entrada dos dados:
  - ▶ Leia (lista de variáveis)
- ▶ **NOTA:**A lista de variáveis pode variar de 1 a n, e devem ser separadas por vírgulas.

# Exemplo: Procedimentos de Entrada

---

## **algoritmo** “LeAno”

Var

ano: inteiro

inicio

Leia(ano)

fimalgoritmo



É bom observar que, em alguns casos, a digitação de um valor de um tipo diferente do tipo da variável provoca erro de execução.

## **algoritmo** “LeMesAno”

var

mes, ano: inteiro

inicio

Leia(mes, ano)

fimalgoritmo

# Procedimentos de Saída

---

- ▶ Comandos para entrada dos dados:
  - ▶ `Escreva`(mensagem e/ou lista de variáveis)
  - ▶ `Escreval`(mensagem e/ou lista de variáveis)
- ▶ A lista de variáveis pode variar de 1 a n, e devem ser separadas por vírgulas.
- ▶ Se a saída é uma mensagem, ela deve ser escrita entre aspas.
- ▶ O comando `ESCREVAL` sem parâmetros imprime uma linha em branco e salta para a linha seguinte.
- ▶ **NOTA: `Escreva` e `Escreval` desempenham a mesma função, a única diferença é que após utilizar o `Escreva` o cursor fica na mesma linha, e no caso do `Escreval` o curso vai para a linha de baixo.**

# Procedimentos de Saída

---

- ▶ Qual o dispositivo de saída padrão?
  - ▶ Monitor



# Procedimentos de Saída

---

## Algoritmo “BoasVindas”

### INICIO

Escreva (“Ola Mundo!”)

### FINALGORITMO

Escreve na tela a string  
Ola Mundo!



# Procedimentos de Saída

---

## Algoritmo “DeclaracaoImportante”

INICIO

Escreval(“Estou aprendendo a programar”)

FINALGORITMO

## Algoritmo

Var

X: Inteiro

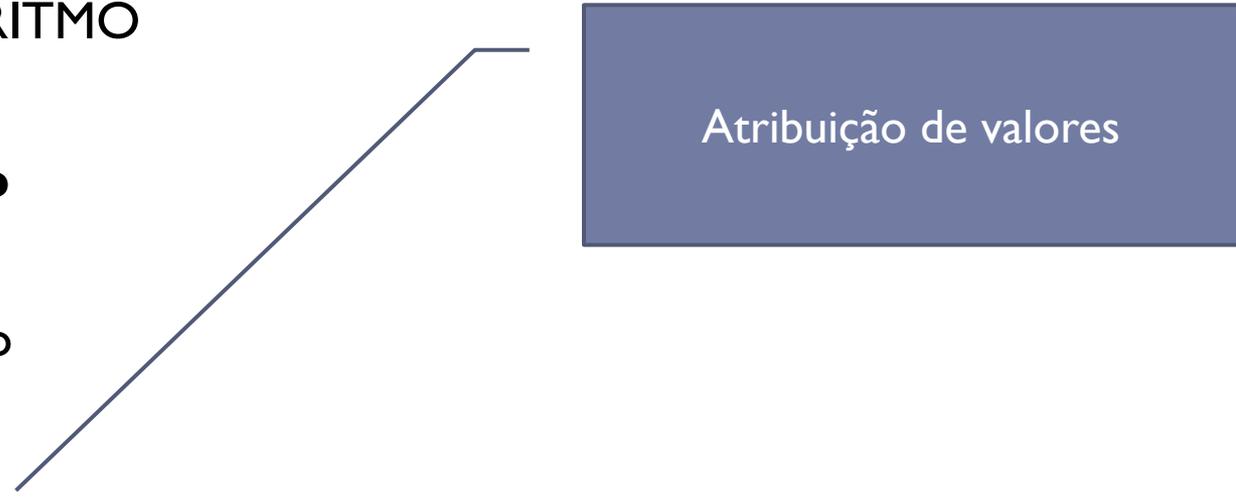
INICIO

X ← 12

Escreval (“Meu Programa”)

Escreval (“O Valor de X é “, X);

FINALGORITMO



Atribuição de valores

# Processamento de um Algoritmo

---

- ▶ Todos os algoritmos executam três passos:
  - ▶ Entrada, processamento e saída



# Exemplo

---

## Algoritmo “Quadrado”

var

num, result: inteiro

INICIO

Escreva (“Digite um numero inteiro: “)

Leia(num)

result  $\leftarrow$  num \* num

Escreval (“O resultado é:“, result)

FINALGORITMO



Atribuição de valores

# Exemplo

---

## **Algoritmo “AreaCirculo”**

```
var
    raio: real
INICIO
    Leia(raio)
    Escreva(3.1415 *raio*raio)
FINALGORITMO
```

## **Algoritmo “MEDIA”**

```
var
    a, b, c : real
INICIO
    Escreva (“Entre com três valores reais: “)
    Leia(a, b, c)
    Escreva((a+b+c)/3)
FINALGORITMO
```

# Linhas de Comentário

---

- ▶ Os comentários são declarações não compiladas que podem conter qualquer informação textual que você queira adicionar ao código-fonte para referência e documentação de seu programa.
- ▶ São representados por duas barras normais ( // ).

## Exemplo

```
// Este método calcula o fatorial de n...x <- y;  
// Inicializa a variável x com o valor de y
```

# Linhas de Comentário

---

## **Algoritmo “AreaCirculo”**

var

    raio: real //declara a variável raio

//programa

**INICIO**

    Leia(raio)

    Escreva(3.1415 \*raio\*raio) //imprime na tela

**FINALGORITMO**

# Construindo os Primeiros Algoritmos: Estruturas seqüenciais

---

- ▶ De forma genérica, a construção de um algoritmo se resume às seguintes etapas:
  - ▶ Entendimento do problema;
  - ▶ Elaboração da solução algorítmica; e
  - ▶ Codificação da solução no Português Estruturado;

# Exercício

---

1. Faça um programa que leia dois valores numéricos, e calcule e exiba a sua média aritmética.
2. Algoritmo que lê o nome de um aluno, as notas de suas três provas e calcule e exibe a média harmônica das provas.

$$\frac{3}{\frac{1}{a} + \frac{1}{b} + \frac{1}{c}}$$

3. Um algoritmo que lê o valor do raio e calcule a área do círculo correspondente. A área do círculo é  $\text{Pi} \cdot R^2$ .

O valor do pi já está definido internamente no VisualG (use CTRL + J). Ou seja, não precisa declarar variável e atribuir valor. Só multiplicar por pi.