

Algoritmo

algoritmo “nome”

declarações

início

comandos

fimalgoritmo

Algoritmo

- Uma declaração por linha
- Um comando por linha
- Acentos não são usados
- Não há distinção entre letras maiúsculas e minúsculas

Declarações

var nome : tipo

var nome, nome, ..., nome : tipo

Nomes

- Até 30 caracteres
- O primeiro caracter deve ser uma letra
- Os demais podem ser letras, dígitos ou “_”
- Não pode haver repetição de nomes
- Todas as variáveis devem ser declaradas

Tipos

- **inteiro** (número inteiro)
- **real** (número real)
- **logico** (verdadeiro, falso)
- **caracter** (seqüência de caracteres delimitada por aspas duplas “...”)

Comandos

Comandos básicos

- Leitura
- Escrita
- Atribuição

Comandos de estruturação

- Sequencial
- Condicional
- Iterativo

Leitura

`leia (nome)`

`leia (nome, nome, ..., nome)`

5. Lê os valores digitados pelo usuário no teclado do computador
6. Armazena nas variáveis designadas, na ordem em que foram listadas
7. Os valores anteriores são perdidos

Escrita

escreva (*expressão*)

escreva (*expressão, expressão, ..., expressão*)

- Avalia as expressões da esquerda para a direita
- Escreve os valores correspondentes na tela do computador
- **escreval** muda de linha no final da última saída de dados

Atribuição

nome <- expressão

3. Avalia a expressão
4. Armazena o valor calculado na variável designada pelo nome
5. O valor anterior da variável é perdido

Expressões

Operandos

- Constantes
- Variáveis

Operadores

- Aritméticos
- Caracteres
- Lógicos
- Relacionais

Constantes

- inteiro: 1, 2, 3
- real: 3.57, 290.1
- logico: verdadeiro, falso
- caracter: “mensagem”, “falso”, “3.57”

Operadores

Aritméticos:

- \wedge (exponenciação)
- $*$, $/$ (multiplicação, divisão)
- \backslash (divisão inteira)
- **mod** ou $\%$ (resto da divisão inteira)
- $+$, $-$ (adição e subtração)
- $+$, $-$ (unários)

Operadores

Caracteres:

- + (concatenação)

Operadores

Lógicos:

- **nao** (negação - unário)
- **ou** (ou)
- **e** (e)
- **xou** (ou exclusivo)

Operadores

Relacionais:

- = (igual)
- <> (diferente)
- > (maior)
- < (menor)
- >= (maior ou igual)
- <= (menor ou igual)

Operadores

- Precedência
- Associatividade
- Uso do parênteses

Comando condicional I e II

se expressão entao
comando

...

senao
comando

...

fimse

se expressão entao
comando

...

fimse

Comando condicional I e II

- Avaliar a expressão (precisa resultar valor lógico)
- Se V, executar os comandos seguintes ao **entao** e encerrar o comando **se**
- Se F, executar os comandos seguintes ao **entao** e encerrar o comando **se**
- Se F e não existir o **senao**, não fazer nada

Comando iterativo I

enquanto *expressão* faça
 comando

...

fimenquanto

Comando iterativo I

- Avaliar a expressão (precisa resultar valor lógico)
- Se V, executar os comandos seguintes ao **faca** e ir para 1.
- Se F, encerrar o comando **enquanto**

Comando iterativo II

repita

comando

...

ate *expressão*

Comando iterativo II

- Executar os comandos seguintes ao **repita**
- Avaliar a expressão (precisa resultar valor lógico)
- Se F, ir para 1.
- Se V, encerrar o comando **repita**

Comando iterativo III

para nome de expressão1 ate expressão2 faça
comando

...

fimpara

para nome de expressão1 ate expressão2 passo expressão3 faça
comando

...

fimpara

Comando iterativo III

- Avaliar a expressão₁ (precisa resultar valor aritmético)
- Armazenar o resultado na variável *nome*
- Avaliar a expressão₂
- Avaliar a expressão₃
- Verificar se o valor da variável *nome* é menor ou igual a expressão₂
- Se V, executar os comandos, adicionar o valor da expressão₃ à variável *nome* e ir para 5. Se não existir expressão₃, considerar o valor 1.
- Se F, encerrar o comando para