

Nome: _____ Data: ____/____/____

1. Introdução**1.1. Lógica**

A solução que criamos para um determinado problema necessita ser exteriorizada e expressa numa linguagem publicamente conhecida. Assim, utilizamos a **lógica** para desenvolver nossas soluções e os **algoritmos** para apresentar essas soluções ao mundo.

Segundo o dicionário Aurélio, lógica é a “coerência de raciocínio, de idéias”, ou ainda “a sequencia coerente, regular e necessária de acontecimentos, de coisas”.

As pessoas utilizam a lógica no cotidiano sem perceber, chegam mesmo a citá-la, sem entender direito seu significado.

1.2. Lógica de Programação

A lógica de programação nada mais é que contextualizar a lógica na programação de computadores, buscando uma melhor sequencia de ações para solucionar um determinado problema. É necessária para pessoas que desejam trabalhar com desenvolvimento de sistemas e programas.

Lógica de programação é a técnica de encadear pensamentos para atingir determinado objetivo.

1.3. Sequência Lógica

Estes pensamentos podem ser descritos como uma seqüência de instruções, que devem ser seguidas para se cumprir uma determinada tarefa.

Seqüência Lógica são passos executados até atingir um objetivo ou solução de um problema.

1.4. Algoritmo

De forma bem simples, um algoritmo pode ser definido como “um conjunto de passos lógicos, bem definidos, que descreve a solução de um problema”. Estas tarefas não podem ser redundantes nem subjetivas na sua definição, devem ser claras e precisas.

Ao pensarmos na solução de um problema, encontramos ações imperativas que são expressas por comandos. Os algoritmos não são aplicados apenas ao mundo da Informática; pelo contrário, usamos – até sem perceber – algoritmos em todos os momentos de nossa vida. Uma receita de cozinha é claramente um algoritmo.

Existem três formas para representação dos algoritmos:

- Pseudocódigo
- Fluxograma
- Diagrama de Chapin

2. Algoritmos**2.1. Constantes, Variáveis e Tipos de Dados**

Variáveis e constantes são os elementos básicos que um programa manipula. Uma variável é um espaço reservado na memória do computador para armazenar um tipo de dado determinado.

Variáveis devem receber nomes para poderem ser referenciadas e modificadas quando necessário. Um programa deve conter declarações que especificam de que tipo são as variáveis que

ele utilizará e as vezes um valor inicial. Tipos podem ser por exemplo: inteiros, reais, caracteres, etc. As expressões combinam variáveis e constantes para calcular novos valores.

2.2. Tipos de Variáveis

As variáveis e as constantes podem ser basicamente de três tipos: Numéricas, Literais ou lógicas.

- **Numéricas:** Específicas para armazenamento de números, que posteriormente poderão ser utilizados para cálculos. Podem ser ainda classificadas como Inteiras ou Reais. As variáveis do tipo inteiro são para armazenamento de números inteiros e as Reais são para o armazenamento de números que possuam casas decimais.
- **Literais:** Específicas para dados que contenham letras e/ou números. Pode em determinados momentos conter somente dados numéricos ou somente literais. Se usado somente para armazenamento de números, não poderá ser utilizada para operações matemáticas. (Tabela ASCII)
- **Lógicas** Armazenam somente dados lógicos que podem ser Verdadeiro ou Falso.

2.3. Declaração de Variáveis

As variáveis só podem armazenar valores de um mesmo tipo, de maneira que também são classificadas como sendo numéricas, lógicas e literais.

Sintaxe: Início
 A: literal
 N: numérico
 X: lógico
 fim

Observação: O nome de uma variável pode ser formado por um ou mais caracteres. O primeiro caractere deve ser, obrigatoriamente, uma letra e os caracteres seguintes podem ser letras ou números. Não é permitido o uso de símbolos, sinais gráficos e de pontuação, exceto o sinal “_” (underline).

2.4. Comentário

O comentário serve para orientar, explicar e ajudar o desenvolvedor no processo de programação. Em algumas linguagens, o comentário também serve para documentação.

Um documento pode ter comentários em qualquer lugar, pois estes, não influenciam no código do programa. É aconselhável que você procure documentar o máximo possível seus programas, pois irá facilitar no futuro, principalmente nas manutenções. Em pseudocódigo, os comentários serão colocados entre chaves.

Sintaxe: Início
 N: numérico {Declaração de uma variável numérica}
 fim

2.5. Atribuição

Ao criarmos uma variável, partimos do princípio que em algum momento ou vários momentos dentro do nosso algoritmo, ela receberá valores, ou seja, armazenaremos dados na memória através de nossas variáveis.

Para atribuírmos um valor ou uma expressão a uma variável, utilizamos o comando de atribuição ←.

Sintaxe: Início
 N: numérico
 N ← 10
 fim

2.6. Expressões

As expressões são meios pelo qual incrementamos, decrementamos, comparamos e avaliamos dados dentro do computador. Temos três tipos de expressões:

- Aritméticas
- Relacionais
- Lógicas

• Expressões Aritméticas

- Operadores Aritméticos

| Operador | Operação | Descrição |
|----------|---------------|--|
| + | Adição | Faz a soma de dois números |
| - | Subtração | Faz a subtração de dois números |
| * | Multiplicação | Faz a multiplicação de dois números |
| / | Divisão | Faz a divisão de dois números |
| ^ | Potenciação | Faz a elevação de um n ^o a uma potencia |

- Funções Aritméticas

| Função | Sintaxe | Descrição |
|--------------------------|-------------------|--|
| Divisão inteira | Div(x,y) | Retorna somente a parte inteira da divisão do n ^o x pelo n ^o y |
| Resto da divisão inteira | Resto(x,y) | Retorna o resto da divisão inteira do n ^o x pelo n ^o y |
| Valor absoluto | Abs(x) | Retorna o valor absoluto |
| Arredondamento | Arredonda(x) | Retorna o valor arredondado |
| Potenciação | Potenciação(x, y) | Retorna o resultado do n ^o x elevado ao n ^o y |
| Raiz quadrada | Rzqd(x) | Retorna o valor da raiz quadrada de x |

• Expressões Literais

- Operador Literal

| Operador | Operação | Descrição |
|----------|--------------|---------------------------------------|
| + | Concatenação | Faz a união de dois valores literais. |

- Funções Literais

| Função | Sintaxe | Descrição |
|-----------------------------|---------------|---|
| Eliminar brancos a esquerda | Ltrim(a) | Elimina os espaços em brancos a esquerda |
| Eliminar brancos a direita | Rtrim(a) | Elimina os espaços em brancos a direita |
| Quantidade de caracteres | Comp(a) | Retorna o n ^o de caracteres da palavra a Obs: retorna um valor numerico |
| Subpalavra a esquerda | Esquerda(a,x) | Retorna os x primeiros caracteres da palavra a |

| | | |
|-----------------------|--------------|--|
| Subpalavra a direita | Direita(a,x) | Retorna os x últimos caracteres da palavra a |
| Subpalavra | Sub(a,x,y) | Retorna os y caracteres, a partir da posição x, da palavra a |
| Posição de caracteres | Pos(a,b) | Retorna a posicao da expressão b na palavra a |

- **Expressões Relacionais**

São expressões compostas por outras expressões. Os operadores relacionais tem a finalidade de comparar dados. Sendo assim, retornam valores lógicos (verdadeiro ou falso).

- Operadores Relacionais

| Operador | Operação | Descrição |
|----------|------------------|------------------|
| = | Igualdade | Igualdade |
| <> | Diferença | Diferença |
| < | Menor que | Menor que |
| > | Maior que | Maior que |
| <= | Menor ou igual a | Menor ou igual a |
| >= | Maior ou igual a | Maior ou igual a |

- **Expressões Lógicas**

- Operadores Lógicos

| Operador | Operação | Descrição |
|------------|-------------|--|
| <u>e</u> | Intersecção | Retorna verdadeiro se e somente se todos os valores forem verdadeiros |
| <u>ou</u> | União | Se pelo menos um valor for verdadeiro, a resultado será verdadeiro |
| <u>não</u> | Negação | Inverte a resposta não falso = verdadeiro não verdadeiro = false |

2.7. Entrada e Saída dos Dados

- **Comando de Entrada**

Sintaxe: Inicio
 N: numérico
 leia N
 fim

- **Comando de Saída**

Sintaxe: Inicio
 N: numérico
 escreva N
 fim