

Aulas Introdutórias a LP

Lógica de Programação

Conceitos fundamentais

- **Lógica*** – É ciência autônoma formada por princípios, métodos e técnicas que objetiva organizar e estruturar o pensamento. Estuda as técnicas de formalização, dedução e análise que permitem verificar a validade dos argumentos.
- **Inferência** – é um processo pelo qual se chega a uma proposição.
- **Proposição** – designa o significado de uma sentença ou oração declarativa.

Lógica*

[Do gr. *logiké (téchne)*, pelo lat. tard. *logica*.] .Substantivo feminino.

1. **Filos.** Na tradição clássica, aristotélico-tomista, conjunto de estudos que visam a determinar os processos intelectuais que são condição geral do conhecimento verdadeiro. [Distinguem-se a *lógica formal* e a *lógica material*.]

2. **Filos.** Conjunto de estudos tendentes a expressar em linguagem matemática as estruturas e operações do pensamento, deduzindo-as de número reduzido de axiomas, com a intenção de criar uma linguagem rigorosa, adequada ao pensamento científico tal como o concebe a tradição empírico-positivista; lógica matemática, lógica simbólica.

3. **Filos.** Conjunto de estudos, originados no hegelianismo, que têm por fim determinar categorias racionais válidas para a apreensão da realidade concebida como uma totalidade em permanente transformação; lógica dialética. [São categorias dessa lógica a contradição, a totalidade, a ação recíproca, a síntese, etc.]

4. Tratado ou compêndio de lógica.

5. Exemplar de um desses tratados ou compêndios.

6. Coerência de raciocínio, de idéias.

7. Maneira de raciocinar particular a um indivíduo ou a um grupo:

a *lógica da criança*;

a *lógica do primitivo*;

a *lógica do louco*.

8.Fig. Seqüência coerente, regular e necessária de acontecimentos, de coisas. [Cf. *logica*, do v. *logicar*.]

9.Conjunto de regras e princípios que orientam, implícita ou explicitamente, o desenvolvimento de uma argumentação ou de um raciocínio, a resolução de um problema, etc.

10.Inform. Forma pela qual as assertivas, pressupostos e instruções são organizadas em um algoritmo para implementação de um programa de computador.

Lógica booliana. 1. Mat. Inform. V. *álgebra booliana*.

Lógica das proposições. 1. Filos. Parte da lógica formal (2) que trata do cálculo proposicional; lógica sentencial.

Lógica das relações. 1. Filos. Parte da lógica formal (2) que trata do cálculo funcional.

Lógica dialética. 1. Filos. Lógica (3).

Lógica difusa. 1. Extensão da lógica booliana, na qual as variáveis podem adquirir graus intermediários de veracidade ou falsidade, representados por valores fracionários entre 0 e 1; lógica *fuzzy*. [Foi introduzida na década de 1960 como meio de modelar graus de incerteza presentes na linguagem natural; é muito usada no desenvolvimento de sistemas especialistas.]

Lógica formal. Filos. 1. Na tradição clássica, o estudo das formas (conceitos, juízos e raciocínios) e leis do pensamento. 2. Na tradição empirista e positivista, o estudo da estrutura das proposições e das operações pelas quais, com base nessa estrutura, se deduzem conclusões válidas. [Distinguem-se a *lógica das proposições* e a *lógica das relações*.]

Lógica fuzzy. 1. Lógica difusa (q. v.).

Lógica matemática. 1. Filos. V. *lógica* (2).

Lógica material. 1. Filos. Estudo da relação entre as formas e leis do pensamento e a verdade, *i. e.*, estudo das operações do pensamento que conduzem a conhecimentos verdadeiros. [Cf. *lógica transcendental*.]

Lógica modal. 1. A que trata das noções de contingência, possibilidade, e necessidade, em oposição à simples afirmação ou negação.

Lógica polivalente. 1. Sistema lógico que admite mais de dois valores de verdade.

Lógica sentencial. 1. Filos. Lógica das proposições.

Lógica simbólica. 1. Filos. V. *lógica* (2).

Lógica transcendental. 1. Filos. Segundo Kant (v. *kantismo*), ciência do entendimento puro e do conhecimento racional, pela qual se determinam os conceitos que se relacionam aos objetos independentemente da experiência, e anteriormente a ela. [Cf. *lógica material*.]

- **Linguagem** - É um código de comunicação formado por sintaxe e semântica.
- **Linguagem de Programação** – É uma notação formal para descrever a execução de algoritmos em computador. Também é formada por uma *sintaxe* e uma *semântica*.
- **Programa** - É um conjunto de instruções (comandos) que o computador interpreta para poder executar uma tarefa. É a codificação de um algoritmo em uma determinada linguagem de programação

Sintaxe – é o conjunto de regras para composição de um texto na linguagem (programa) a partir do agrupamento de letras, dígitos e/ou outros caracteres (alfabeto da linguagem). Refere-se a forma de escrever.

Semântica – significado de um programa sintaticamente válido, escrito na linguagem considerada. Refere-se ao sentido, significado dos “termos” ou símbolos.

- **Algoritmo** – é seqüência lógica de procedimentos em face de objetivos pré-definidos.

É o conceito central da da programação, ou seja, programar é basicamente construir algoritmos.

- **Programação Estruturada** – é uma técnica que se baseia na utilização de 3 elementos básicos: a seqüência, a condição e a repetição.

Seqüência – implementação de etapas para execução linear.

Condição – facilita o processamento selecionado com base em alguma ocorrência lógica.

Repetição – permite os loopings (laços).

- **Linguagem (Portugol)** – é uma junção de Português com uma estrutura de Pascal. Pode-se dizer que é uma pseudolinguagem de programação para desenvolver algoritmo, expressando o raciocínio lógico formal de maneira mais simples.

Obs.: também possui uma sintaxe e uma semântica.

Evolução das linguagens de programação

- ✓ O objetivo de uma linguagem é a comunicação. A espécie humana usa a linguagem natural para se comunicar. As linguagens de programação são usadas para escrever programas que é o modo como um programador pode se comunicar com as máquinas ou fazer as máquinas se comunicarem entre si, quando são colocadas em rede. Programas, feitos na forma de linguagem escrita, estão evoluindo para acrescentar a comunicação verbal. Escrever um programa é algo parecido a escrever uma peça literária. Donald Knuth assim entendeu ao dar o título de “**The Art of Computer Programming**” aos seus livros sobre programação.

As primeiras linguagens formais e a evolução do alfabeto

A linguagem falada é tão antiga quanto o homem, porém extremamente complexa e difícil de ser formalizada. Todo o enfoque será dirigido para as linguagens escritas, que favorecem a análise formal. Entre as primeiras linguagens escritas estão o sumério, os hieróglifos egípcios, as escritas dos hindus, dos maias e dos incas. No ramo das linguas indo-européias, onde se encontram o português, o inglês, o russo e o hindu entre outras, o sânscrito é considerada a primeira linguagem escrita. O Veda, a mais antiga obra do hinduísmo foi escrita em sânscrito.

Os símbolos do alfabeto representavam idéias. Inicialmente os *pictogramas* associavam figuras de objetos aos respectivos conceitos daí evoluindo para *ideogramas* que eram símbolos para idéias, agora não mais na forma de figuras e sim de símbolos e ainda os *fonogramas* aproveitando a sensação sonora da linguagem oral. Como exemplo a idéia de sol pode ser representada no modo pictograma (figura representando o sol), ideograma figura trazendo a idéia de dia e fonograma (sun ≈son, em inglês).

Os fenícios, que povoaram o Mediterrâneo por volta de 1000 a.C. difundiram o alfabeto para os outros povos da área em função do comércio que praticavam. Numa etapa adiante foram criadas as CONSOANTES, reduzindo o número de símbolos.

Os gregos, que receberam o alfabeto dos fenícios, criaram as VOGAIS.

Algoritmos e notação matemática

As linguagens escritas propiciaram a representação de algoritmos, palavra originada do nome do matemático árabe Abnu Jáfar Mohammed ibn al-Khorezmi(780-850). Como exemplo, os babilônios usaram a aritmética base 60, os gregos o algoritmos de Euclides (m.d.c.), Pitágoras (triângulo retângulo). A introdução do zero permitiu a notação posicional na aritmética.

Até o século XIX a evolução da notação matemática foi lenta. Francois Viète(1540-1603) recomendou o uso de letras para representar quantidades desconhecidas. Uma formalização da linguagem matemática foi feita por Leibnitz (1646-1716), que também construiu a máquina analítica para executar as operações aritméticas. A idéia do computador atual se deve a Charles Babbage(1792-1871). **Uma mulher, Ada Augusta, condessa de Lovelace trabalhando com a máquina analítica idealizada por Babbage propôs regras de “como usar a máquina analítica para calcular” ou seja um programa. Surge assim a primeira programadora de computador.**

Contribuição da matemática moderna

Após 1800 a matemática começou a se expandir rapidamente. Vários matemáticos contribuíram com idéias importantes na formalização das linguagens.

Gottlob Frège (1848-1925) criou o cálculo proposicional, em substituição ao silogismo grego, ao usar funções e argumentos em substituição a sujeito e predicados. Criou também uma teoria sistemática para quantificação e derivação. A Lógica Proposicional e a Lógica Quantificacional são fundamentais para a compreensão das linguagens funcionais e linguagens lógicas como PROLOG.

Giuseppe Peano(1858-1932) criou a notação linear para símbolos lógicos.

Bertand Russel(1872-1970) orientou os estudos da aritmética e geometria para lógica e teoria dos conjuntos. Na sua obra “Principia Mathematica”, escrita com a cooperação de Alfred Whitehead(1861-1947) que num trecho gasta 300 páginas para provar que $1+1=2$. E’ conhecido o “paradoxo de Russel” que coloca o impasse do “barbeiro que não pode barbear a si próprio”.

O conceito de matemática construtiva, pelo qual a existência de um objeto matemático tem que ser provada pela sua construção é semelhante ao conceito de algoritmo que faz uso de uma linguagem de programação para a construção de objetos representáveis no computador.

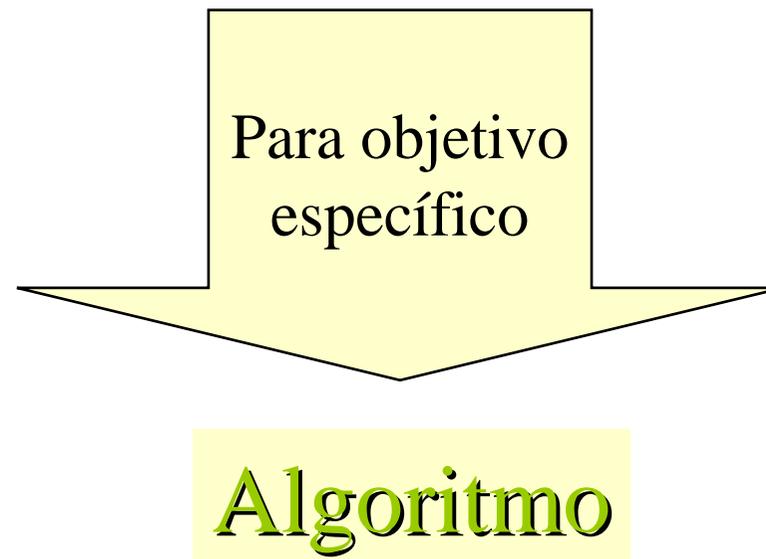
Alfred Tarski(1902-1983) formalizou a distinção entre símbolos e seu significados

Por meio de uma definição semântica do cálculo de predicados. Revelou a dicotomia entre sintaxe e semântica e *linguagem objeto e meta-linguagem*. A linguagem pode ser estudada matematicamente. O estudo de uma linguagem de programação se inicia fora da linguagem lógica e matemática. Como exemplo o português ou o inglês são usados para explicar a construção de uma linguagem de programação.

Revisão

Algoritmo

- Utilizar Lógica \implies colocar Ordem no Pensamento

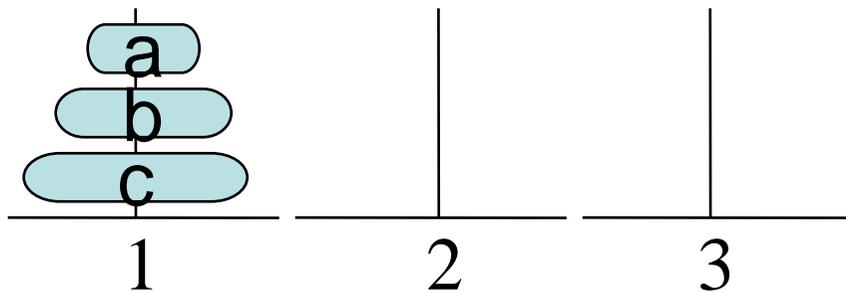


- Ex.: receita de um bolo, manual de instruções, etc.

Revisão

Torre de Hanói

- Mover os três discos de uma haste para outra
- Regras:
 - pode-se mover apenas um disco de cada vez
 - nunca pode ser colocado um disco maior sobre um menor



Solução

*move o disco a para a haste 2
move o disco b para a haste 3
move o disco a para a haste 3
move o disco c para a haste 2
move o disco a para a haste 1
move o disco b para a haste 2
move o disco a para a haste 2*

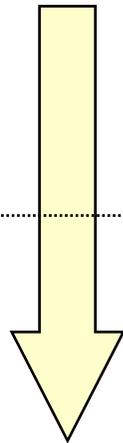
Revisão

Algoritmo X computador

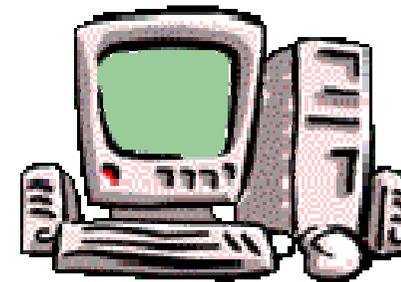
Algoritmo



Linguagem de
computação



Programa



Representação ou descrição do Algoritmo

- **Português Estruturado (Portugol)**
- **Diagrama de Chapin (Nassi e Schneider)**
- **Fluxograma**

Algoritmos computacionais

Algoritmo – Em um aspecto computacional, algoritmos possuem características específicas, tais como:

- Passos precisamente definidos;
- Quando codificado numa linguagem de programação corresponda aos critérios especificados;
- Seja preciso, seguro e eficiente;

Construir algoritmos consiste em estruturar um processo de forma lógica. Implica em definições precisas e otimizadas.

Continuação

- ✓ Um algoritmo só é entendido pela máquina quando for codificado na sintaxe de uma linguagem, isto é, atinge a condição de programa.
- ✓ Uma linguagem de programação traduz um algoritmo para uma forma de linguagem entendida pela máquina (linguagem de máquina), então, passa a assumir o *status* de programa.
- ✓ Um algoritmo, no entanto, é composto de instruções e dados que, por sua vez, precisam estar armazenados na memória principal do computador.

Memória do computador

- ✓ A memória principal de um computador é “semelhante” a de um ser humano, isto é, implica na faculdade de reter informações que serão utilizadas quando necessárias.
- ✓ Pode-se comparar a memória principal de um computador a uma seqüência de células numeradas, onde cada célula corresponde a uma espaço da memória, e constantemente informações são armazenadas no referido espaço.
- ✓ A referência a cada célula da memória é realizada através de seu *endereço numérico*.

Continuação

- ✓ Pode-se alterar o conteúdo da célula mas o endereço de uma posição de memória não pode ser alterado.

Graficamente, pode-se representar:

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

- ✓ **Imagine que se você queira armazenar o número 20 na posição de memória de endereço 5. Então, graficamente pode-se representar da seguinte forma:**

0	1	2	3	4	5	6	7
					20		

Tipos Primitivos

- ✓ Referem-se a maneira como o computador manipula as informações, sendo que essas informações classificam-se a grosso modo em dois tipos: dados e instruções.
- ✓ **Dados** - são representados pelas informações a serem tratadas (processadas) por um computador. Essas informações estão caracterizadas por três tipos de dados: dados numérico (inteiros e real), dados caracteres e dados lógicos.
- ✓ **Instruções** – Comandos, funções que determinam as operações com os dados – conteúdo das variáveis.

Ex:

Ao citarmos uma data como 21 de setembro, estamos apresentando um dado; ao dizermos que este é o dia da árvore, estamos agregando valor ao dado data, isto é, apresentando uma informação.

Tipos Primitivos

- **Inteiro:** toda informação numérica inteira (não fracionária) negativa, nula ou positiva. **Ex:** 100, 0, -3
- **Real:** toda informação numérica pertencente ao conjunto dos números reais (inteiras ou fracionárias), (negativa, nula ou positiva). **Ex:** 100, 0, -3, 1,7, 1000,50.
- **Caractere:** sequência contendo letras, números e símbolos especiais (caracteres alfanuméricos)
 - essa sequência deve ser indicada entre aspas (“ “)
 - **Ex.:** “Taguatinga - DF”, “356-9025”, “Desconto 10%”
 - também chamado de string ou cadeia
- **Lógico:** conjunto de valores *falso* ou *verdadeiro*.
 - esse tipo só apresenta um desses valores
 - também chamado de booleano

Variáveis

- ✓ Define-se como o nome de um local onde se pode colocar qualquer valor do conjunto de valores possíveis do tipo básico associado. Pode-se dizer que a variável é um identificador. Um dado é classificado como variável quando **tem a possibilidade de ser alterado** em algum instante no decorrer do tempo de execução do algoritmo.
- ✓ O nome de uma variável é utilizado para sua identificação e posterior uso dentro de um programa.

Regras de utilização de variáveis

- Nomes de uma variável poderão ser atribuídos com um ou mais caracteres;
- O primeiro caractere de uma variável **não pode ser número**;
- Não é possível espaços em branco entre os caracteres;
- Palavras reservadas não podem assumir nomes de variáveis;
- Não poderão ser utilizados caracteres diferentes de letras e números;

Ex.:

Válidos

Nome
Nome1
Fone_1
X, nota,media

Inválidos

n ome
1nome
#fone
@nota

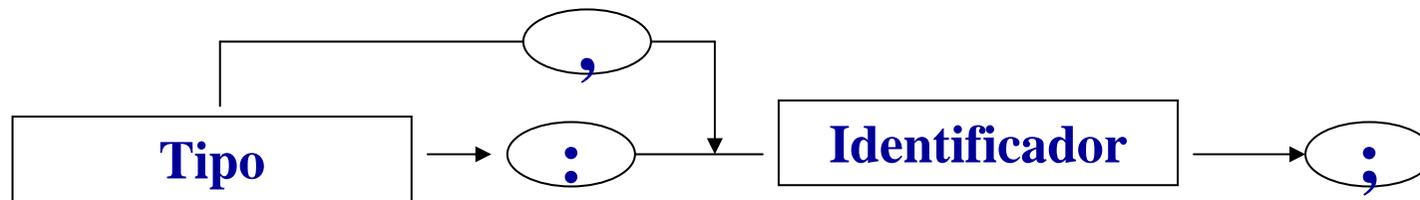
Exemplo de variáveis

- **Exemplos de declarações**

X,n1,N2:inteiro;
nome,Endereco,Data:caracter;
ABC,xpto,Peso,Dolar:real;
Resposta,h286:logico;
Sexo:Char:

Obs.: essa forma de representação diferencia-se dependendo da sintaxe do português ou da linguagem, quanto à separação do identificador e tipo por dois pontos ou ponto e vírgula ao final da instrução. Compiladores que executam português geralmente se diferenciam.

- **Gráfico**



Declaração de Variáveis

- As variáveis ao serem declaradas devem indicar tipo primitivo (inteiro, real, caracter ou lógico).
- Para declararmos as variáveis em um programa fazemos da seguinte forma:

Sintaxe:

<Tipo> <nome_da_variável>;

Exemplos:

- inteiro valor;
 - lógico gelado;
 - caracter mensagem;
 - real total;
- Também podemos definir várias variáveis de um mesmo tipo em uma só linha.

Exemplo:

```
real nota1, nota2, nota3, media;
```

Continuação ...

- ✓ Ao indicar o tipo da variável, informamos para o computador quantas posições de memória ele deve reservar para a variável.
- Um inteiro ocupa uma posição de memória = 1 byte;



- Cada caracter ocupa uma posição de memória = 1 byte por caracter;
- Um real ocupa duas posições de memória = 2 bytes;
- Um lógico ocupa uma posição de memória = 1 byte;
- **OBS.:** O **Bit** é a menor unidade de memória do computador (1 **Byte** = 8 **bits**).
- Logo, a palavra 'sapo' ocupara quatro posições na memória = 4 bytes.

Constantes

- As constantes são construídas para armazenar dados que não serão modificados durante a execução do programa. Para defini-las basta apenas:

Const

```
<nome_da_constante> <valor>;
```

Exemplos:

```
Const NUMERO = 5;
```

```
Const MAXIMO = 30; //Usar sempre maiúsculo.
```

Operações - Operadores Aritméticos

Operação	Símbolo	Prioridade
Adição	+	3
Subtração	-	3
Multiplicação	*	2
Divisão	/	2
divisão inteira	DIV	2
Resto da divisão	MOD	2
Potenciação	^	1
Radiciação	::	1

Continuação ...

- As prioridades indicam a ordem na qual as operações serão efetuadas. As prioridades de valor mais baixo são efetuadas antes. Se as prioridades são iguais, opera-se da esquerda para direita.

Analise os exemplos abaixo e verifique a ordem que as operações foram executadas:

- $5 + 6 * 4$ resultado = 29 $5 + 6 - 3$ resultado = 8
- $5 * 6 + 4$ resultado = 34 $12 / 6 * 4$ resultado = 8
- $4 * 12 / 6$ resultado = 8 $4 \wedge 2 + 1$
 resultado = 17
- $4 :: 2 * 3$ resultado = 6 $8 :: 3 * 5$
 resultado = 10
- $11 \text{ DIV } 2$ resultado = 5 $11 \text{ MOD } 2$
 resultado = 1

Continuação

- A utilização de parênteses permite alterar a prioridade das operações. Quando eles são usados em grupo, a prioridade maior é dos parênteses mais internos. Veja os exemplos:

$$(3 * (5 + 6)), \text{ resultado} = 33$$

$$3 ^ (1 + 2), \text{ resultado} = 27$$

$$3 - (5 + 6), \text{ resultado} = - 8$$

$$4 \text{ MOD } (5 - 3), \text{ resultado} = 0$$

Operadores Relacionais

- Os operadores relacionais permitem que uma comparação feita entre dois valores do mesmo tipo. Estes valores poderão ser identificadores, constantes ou expressões aritméticas.
- **Importante!** O resultado obtido será sempre um valor lógico, ou seja, F ou V. Os operadores utilizados na comparação são os seguintes:
 - = Igual
 - != Diferente
 - > Maior que
 - < Menor que
 - >= Maior que ou igual a
 - <= Menor que ou igual a

Exemplos:

a) $4 * 6 = 3 * 8$, é verdadeiro;

b) $5 + 2 > 7$, é falso;

c) $27 // 3 \Leftrightarrow 3$, é falso;

d) $3^2 = 9$, é verdadeiro;

e) $5 = 4 + 2$, é falso;

f) $4 * 6 = 3 * 8$, é verdadeiro;

g) $\text{int}(23 / 4) * 4 + 6 \bmod 5 \Leftrightarrow 9.6 + \text{frac}(4.2)$,

$5 * 4 + 1 \Leftrightarrow 9.6 + 2$;

$20 + 1 \Leftrightarrow 11.6$;

$21 \Leftrightarrow 11.6$;

F;

Operadores Lógicos

- A combinação de valores lógicos com os operadores lógicos tem como resultado outro valor lógico. Os operadores lógicos são:
- **E** - A sentença é verdadeira se TODAS as condições forem verdadeiras.
- **OU** A sentença é verdadeira se UMA das condições for verdadeira.
- **NÃO** Inverte o valor lógico da sentença (V à F) (F à V).
- **XOU** Se as duas condições forem diferentes, a sentença é verdadeira.
- A tabela verdade exprime todas as combinações possíveis entre os valores de diversas variáveis lógicas.

Continuação ...

Operador E	Operador OU	Operador XOU	Operador NÃO
$V e V = V$	$V ou V = V$	$V xou V = F$	$V = F$
$V e F = F$	$V ou F = V$	$V xou F = V$	$F = V$
$F e V = F$	$F ou V = V$	$F xou V = V$	
$F e F = F$	$F ou F = F$	$F xou F = F$	

Continuação ...

- **Prioridades entre os operadores lógicos:**
- 1) NÃO (NOT)
- 2) E (AND) OU (OR)
- 3) XOU (XOR)

Exemplos:

Suponha que $A=F$ ($F=falso$) e $B=V$ ($V=Verdadeiro$);

– $A \text{ OU } B \text{ XOU NOT } B$

– $F \text{ OU } V \text{ XOU } F$

– $V \text{ OU } F$

– V

2)

A E V O U F

F O U F

F

3)

A E B X O U B

F X O U F

F

Exercício

- Suponha que $w=F$, $x=V$ e $z=V$. Resolva as expressões:
 - a) $x \text{ OU } z \text{ E } w$;
 - b) $z \text{ XOU } x \text{ E } x$;
 - c) $z \text{ OU } z \text{ OU } w \text{ E } x$;
 - d) $x \text{ E } z \text{ E } w \text{ OU NAO } w$
 - b) $x \text{ OU NAO } z \text{ E } w \text{ OU } z \text{ E } x \text{ XOU NAO } w$;
 - c) $\text{NAO } w \text{ OU } z \text{ XOU } x \text{ E } x \text{ NAO } z \text{ E } w$;
 - d) $z \text{ E } z \text{ XOU } x \text{ OU } z \text{ E } w \text{ E } x \text{ OU } z \text{ OU } w \text{ XOU } z$;

Prioridade entre todos os operadores

- A maioria das linguagens de programação utilizam as seguintes prioridades de operadores:
 - 1º - Efetuar operações embutidas em parênteses "mais internos"
 - 2º - Efetuar funções;
 - 3º - Efetuar multiplicação e/ou divisão;
 - 4º - Efetuar adição e/ou subtração;
 - 5º - Operadores relacionais;
 - 6º - Operadores lógicos.

Continuação ...

Exemplos:

Suponha as variáveis e valores: fruta="maça" e aux=10.

1)

(aux>=10) OU (fruta="pera")

V O U F

V

2)

(30>aux) XOU (50!=100/3) E (2 MOD 2 > 0)

V X O U V E F

V X O U F

V

3)

(fruta="maça") E (27.1 DIV 9 = 3) XOU (aux>= 300/100)

V E V X O U V

V X O U V

F

Instruções Primitivas - Entrada e Saída

Para que o programa a ser construído interaja com o usuário, é necessário que seu algoritmo tenha instruções de entrada e saída.

- **Entrada:** São instruções que o computador recolhe do usuário, seja de um teclado, scanner, etc... A utilização de entrada mais comum é feita pelo teclado. Por exemplo, se o algoritmo tiver que solicitar ao usuário sua idade, é utilizada uma operação de entrada.

Sintaxe:

Leia (<variável_ou_lista>)

Exemplos:

Leia (idade);

Leia (nome, sobrenome);

Leia (indice);

Saída

- **Saída:** São instruções que o computador envia ao usuário, seja por um monitor, por uma impressora, etc... A saída mais comum é feita pelo monitor. Por exemplo se o algoritmo tiver que mostrar a idade do usuário expressa em dias.

Sintaxe:

Escreva <literal_ou_variável_ou_lista>;

Exemplos:

Escreva (“Digite sua Idade”);

Escreva (nome);

Escreva (idade * 365,25) ;

Escreva (“Você tem: ”, idade , “ de idade”);

✓ Exemplos:

$A \leftarrow 5;$ atribuição do valor cinco para a variável A

$B \leftarrow 4 + 2;$ atribuição da operação $4 + 2$ para a variável B

$C \leftarrow A + B;$ atribuição da operação $A + B$ para a variável C

- A variável só pode receber atribuições de valores do mesmo tipo que ela. Ou seja, se ela foi declarada como inteiro, somente poderá receber atribuições de valores inteiros.

A forma geral da representação de um algoritmo

Programa <nome_do_algoritmo>

Declarações

<declaração_de_variáveis>

<declaração_de_contantes>

<declaração_de_tipos>

Início

<corpo_do_algoritmo>

Fim.

Onde:

- **Programa ou algoritmo**- é a palavra que indica o início do algoritmo;
- **<nome_do_algoritmo>** - é um nome simbólico dado ao algoritmo;
- **<declaração_de_variáveis>** - consiste em uma porção opcional onde são declaradas as variáveis globais;
- **<declaração_de_contantes>** - consiste em uma porção opcional onde são declaradas as constantes;
- **<declaração_de_tipos>** - consiste em uma porção opcional onde são declarados os novos tipos;
- **Início e Fim** – delimitam respectivamente o início e o término do algoritmo.

Exemplos

Programa Teste;

Declaracoes

Inteiro A, B;

Inicio

$A \leftarrow 5;$

$B \leftarrow A / 2;$

Fim.

Programa exemplo1;

Declarações

inteiro val;

Inicio

Leia(val);

Escreva (val);

Fim.

Diretrizes para a Elaboração de Algoritmos

1. Identificação do problema: **determinar o que se quer resolver** ou **qual objetivo a ser atingido**;
2. Identificação das “entradas de dados”: informações fornecidas, a partir das quais se desenvolverão os cálculos.
3. Identificação das “saídas de dados”: as informações a serem geradas como resultado.
4. Identificar as regras e limitações do problema
5. Determinar o que deve ser feito para transformar as “entradas” em “saídas”.
6. Construir o Algoritmo
7. Testar a solução

Exemplo:

- ✓ Calcular a média final dos alunos da 6ª Série. Os alunos realizarão quatro provas: P1, P2, P3 e P4.

Para montar o algoritmo proposto, faremos três perguntas:

a) Quais são os dados de entrada?

R: Os dados de entrada são P1, P2, P3 e P4

b) Qual será o processamento a ser utilizado?

R: O procedimento será somar todos os dados de entrada e dividi-los por 4 (quatro)

$$- (P1 + P2 + P3 + P4)/4$$

c) Quais serão os dados de saída?

R: O dado de saída será a média final

Exemplo do calculo da média de 4 notas em psseudocódigo.

Programa *calculo_media*

Declarações

REAL N1, N2, N3, N4, MEDIA;

Inicio

Leia (N1, N2, N3, N4);

MEDIA := (N1 + N2 + N3 + N4)/4;

Escreva (MEDIA);

Fim.

Exercícios

- ✓ Resolver os exercícios abaixo, usando o português estruturado.
- 1) Leia um valor e exiba a sua metade;
- 2) Leia quatro valores e exiba a soma destes;
- 3) Leia quatro valores e exiba a media;
- 4) Dada a distância a ser percorrida em Km e o consumo de um carro (em Km por litro), calcule e exiba quantos litros são necessários para completar a distância.

Continuação ...

- 5) Ler um valor em graus Farenheit e exibi-lo em Celsius graus - $C = (5 * (F-32) / 9)$.
- 6) Ler duas variáveis A e B. Inverter seus valores.
- 7) Calcule o volume de um cubo. Será dado o valor de sua aresta. ($V = a^3$)
- 8) Calcule o volume de um cilindro, sendo que serão informados o raio da base e a altura do mesmo. ($V = A \cdot h$)
- 9) Calcule o salário final de um funcionário que obteve no primeiro mês um aumento de 10% e no segundo mês um aumento de 5%.

Exercícios

✓ Suponha $A=127$, $B=10$, $C=5$, $D=falso$ e $E=verdadeiro$. Qual é o valor produzido por cada uma das sentenças abaixo?

- a) não D
- b) D e E
- c) não $A>B$
- d) $A-B>C$
- e) $B-C<0$ ou $A>B$
- f) $(A>B)$ ou $(B<C)$
- g) não $(A<B)$
- h) $A/5>10$ e E
- i) D e E ou não $B<C$
- j) D ou E e $A/5-B>B$
- k) $(D e E)$ ou $(A=B)$

Instalação do compilador Visualg Compilador para PORTUGOL

1. Calcule e mostre o valor da área de uma circunferência, considerando que o valor do raio será informado pelo usuário e convencionado o valor para PI 3,14.
2. Calcule o salário líquido de um professor, devendo para isto se informado horas trabalhadas ao mês, o valor da hora aula, o percentual de desconto sobre a seguridade social de 8%.
3. Calcular o volume de uma lata de óleo com formato de cilindro utilizando PI 3,14 ($v = \pi * r^2 * H$).
4. Calcule a quantidade de litros de combustível gasta em uma viagem sabendo que um automóvel percorre 12 KM com um litro do combustível. Ao final o algoritmo deverá mostrar: a velocidade media, o tempo gasto, a distância percorrida e a quantidade de litros gasta na viagem.
5. Calcule e mostre o valor atual de uma prestação atrasada, considerando que a mesma tem 30 dias de atraso e sobre este incide uma multa de 10% ao mês. Mostre o valor atualizado, o valor original e os acréscimos por atraso.
6. Receber 4 valores inteiros, e aplique o conceito da propriedade distributiva para mostrar o valor final.
7. Calcular o volume de uma caixa com o formato retangular mostrando ao final as medidas informadas e o referido volume.
8. Receba duas variáveis, aplique a propriedade do quadrado da soma e do quadrado da diferença aplicada à propriedade dos produtos notáveis.
9. Desenvolva um algoritmo que recebendo um valor em dólar mostre o equivalente em real, bem como a operação encontrada e também fazer a situação inversa.
10. Considerando que uma situação do 2 grau possui 2 raízes reais e diferentes, calcule-as e mostre os co-eficientes da equação e as respectivas raízes