
Introdução à Lógica de Programação

Sistemas Numéricos

As informações inseridas em um computador são traduzidas em dados, ou seja, em sinais que podem ser manipulados pelo computador. O computador trabalha até que esses dados sejam processados para atingir o resultado desejado.

Os computadores trabalham com o sistema de numeração binário (ligado, desligado). Os programas de computadores e as informações são armazenadas em mídias (memórias, discos, etc) usando este sistema binário.

Base decimal

Neste sistema, utilizamos dez algarismos: 0 1 2 3 4 5 6 7 8 9 para representar os números.

O número dez, por exemplo, significa 9+1, ou seja, é o número seguinte ao 9 e será representado por 10. O número 1 neste exemplo, indica a quantidade de vezes que utilizamos a seqüência de algarismos do sistema, e o 0 indica que estamos no primeiro número desta seqüência.

Base binária

Neste sistema, utilizamos dois algarismos: 0 1. Como já vimos, o computador utiliza este sistema internamente.

Em computação, chama-se um dígito binário (0 ou 1) de bit, do termo em inglês *Binary Digit*. O byte equivale ao conjunto de 8 bits.

O sistema binário é base para a álgebra booleana, que permite realizar operações lógicas e aritméticas usando-se apenas dois dígitos ou estados (sim e não, falso e verdadeiro, aberto ou fechado, 1 ou 0, ligado e desligado).

Base hexadecimal

Neste sistema, utilizamos 16 algarismos: 0 1 2 3 4 5 6 7 8 9 A B C D E F. Os computadores costumam utilizar o byte (8 bits) como unidade básica da memória. Um byte pode ter $2^8 = 256$ valores diferentes. Que também pode ser representado: $2^8 = 2^4 \cdot 2^4 = 16 \cdot 16$.

Por exemplo, o valor 15 em decimal, equivale a 1111 em binário, o que equivale a F em hexadecimal. Sendo assim, um único dígito hexadecimal é capaz de representar qualquer combinação de 4 bits e, assim, dois dígitos hexadecimais podem representar um byte.

O código ASCII serve para representar números, letras, pontuação e outros caracteres. Ele é formado por todas combinações possíveis de 8 bits ($2^8 = 256$), sendo que ainda existem extensões que abrangem mais de 8 bits.

Tabela comparativa

decimal	binário	hexadecimal
0	0	0
1	1	1
2	10	2
3	11	3

4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Linguagem de Programação

A linguagem de programação é um método padronizado para passar instruções ao computador. Como qualquer linguagem, possui um conjunto de regras sintáticas e semânticas para definir um programa de computador. Desta forma, um programador pode criar programas para que o computador processe dados.

Linguagem de Máquina

O computador interpreta os sinais binários (ligado e desligado), é como ele lê e processa os programas. A partir dessa linguagem, foram surgindo outras linguagens mais próximas da linguagem humana, para que pessoas possam entender e programar.

Essas linguagens podem ser classificadas em Linguagem de baixo nível ou alto nível.

Linguagem de baixo nível

É uma linguagem de programação mais próxima à linguagem de máquina. Um exemplo é a linguagem Assembly que trabalha diretamente com os registradores – dispositivo que guarda informações – do processador para manipular dados.

Linguagem de alto nível

É uma linguagem mais próxima da linguagem humana. Os programas feitos em linguagem de alto nível são traduzidos para linguagem de máquina. Para programar em linguagem de alto nível, não é necessário conhecer uma linguagem de máquina ou de baixo nível. O texto do programa em linguagem de alto nível é chamado de código fonte, e o programa traduzido, chama-se código objeto ou código de máquina. Os tradutores são conhecidos como compiladores e interpretadores.

Compiladores: lêem e analisam todo o programa escrito em linguagem de alto nível, e depois traduzem o programa para linguagem de máquina. Exemplos: Pascal, C.

Interpretadores: traduzem o programa escrito conforme vai sendo executado. Geralmente são mais lentos que os compilados, mas também mais flexíveis e freqüentemente são chamados também de script. Exemplos: JavaScript, Python, Perl, PHP.

Existem linguagens que são compiladas e interpretadas por uma máquina virtual – um programa que emula essa máquina virtual – como por exemplo o Java. Um compilador traduz o código java para um código intermediário da JVM (Java Virtual

Machine), que por sua vez, está instalado no computador e interpreta o código intermediário.

Lógica de Programação

É a ordenação do raciocínio para realização de instruções a fim de atingir determinado objetivo.

Seqüência lógica

Para alcançar o objetivo, as instruções devem obedecer uma seqüência lógica.

Instrução

É a informação que indica ao computador a ação a ser executada.

Algoritmo

É a seqüência de passos definidos para executar determinada tarefa.

Funcionamento de um programa (processamento de dados)

O funcionamento básico de um programa constitui-se em três procedimentos: Entrada, Processamento e Saída de dados.

- **Entrada** (*input*) refere-se aos dados inseridos antes do processamento. Podem ser dados inseridos pelo usuário ou dados que serão manipulados a partir de um banco de dados.
- **Processamento** é a ação executada pelo computador, para processar os dados.
- **Saída** (*output*) é o resultado de todo o processamento. Pode ser mostrado em tela, em papel, armazenado em disco, ou ainda servir de entrar para outro processo.

Português estruturado ou pseudocódigo

É uma forma de escrever um algoritmo. Em forma de narrativa, sem a necessidade de uso de linguagem de programação.

Exemplo:

Passo a passo para somar dois números:

1. Escreva o primeiro número
2. Escreva o segundo número
3. Some os dois números
4. Mostre o resultado

Passo a passo do acordar até escovar os dentes

1. Abrir os olhos
2. Tirar a primeira perna da cama
3. Tirar a outra perna da cama
4. Levantar-se
5. Ir ao banheiro
6. Pegar a pasta de dente
7. Pegar a escova de dente

Fluxograma

É uma forma de descrever graficamente o fluxo de dados e seu processamento. Podemos entender como um diagrama que descreve os passos necessários para a executar determinadas tarefas.

Exemplo:

Fluxograma para somar dois números:

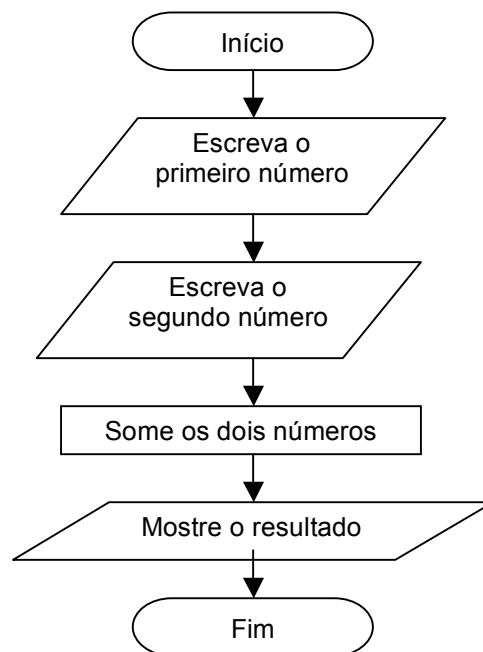


Tabela de símbolos:

Símbolo	Nome	O que representa
	Terminal	O início e o fim do diagrama ou de uma parte dele.
	Seta de fluxo de dados	O sentido do fluxo de dados. Serve para conectar os blocos existentes.
	Entrada / Saída	O recebimento de entrada de dados, ou então, saída de dados.
	Processamento	Executa um processamento interno.
	Decisão	Tomada de decisão ou desvio do fluxo do programa (estrutura de seleção ou de repetição).
	Conector	Representa a conexão do fluxo de dados do programa.
	Processamento predefinido	Representa um módulo, sub-rotina, procedimento ou função.

Exercícios:

1. Faça o fluxograma da seqüência para fazer um suco de laranja
2. Faça o fluxograma da seqüência para somar dois números e multiplicar o resultado da soma por um terceiro número.
3. Faça o fluxograma do procedimento para trocar uma lâmpada.
4. Faça um fluxograma que compare 2 números diferentes e mostre o maior dos 2.
5. Faça um fluxograma que compare 3 números diferentes e mostre o maior dos 3.

Variável

Uma variável é um espaço na memória do computador, utilizado para representar um valor ou expressão que pode ser alterado durante a execução de um programa.

A memória do computador guarda a informação em espaços como se fossem gavetas, e estes são etiquetados com um nome para facilitar o acesso a esta informação. As variáveis existem apenas durante a execução de um programa

Quando declaramos uma variável, reservamos um espaço na memória e identificamos este espaço. Durante a execução do programa, o conteúdo deste espaço pode ser alterado diversas vezes. Se um novo valor for atribuído a uma variável já armazenada, este novo valor irá sobrepor o valor antigo. E quando o programa termina de executar, as variáveis são descartadas.

Regras para criação de nomes de variáveis

- O primeiro caracter do nome da variável deve ser uma letra, nunca um símbolo ou número.
- Os nomes devem ser formados por caracteres alfanuméricos, ou seja, letras ou números.
- Não pode haver espaços em branco entre os caracteres do nome da variável. Se for necessário usar mais de uma palavra, pode-se separá-las por underline (_).
- Procurar escolher sempre nomes que expliquem o seu conteúdo.
- Usar nomes curtos. O uso de nomes longos dificultará a escrita do programa.
- Não usar acentos ou cê-cedilha.
- Não usar palavras reservadas da linguagem de programação.

Constantes

Ao contrário das variáveis, as constantes não sofrem alteração durante o processamento.

Tipos de dados básicos

Os tipos de dados variam de acordo com a linguagem de programação. Basicamente, pode-se dizer que são divididos em: numérico, caractere e lógico.

Numérico: uma variável declarada desse tipo, receberá apenas valores numéricos. Exemplo: 15, FF, 0, 10000, -234.

Caractere: pode armazenar valores alfanuméricos: números (0...9), alfabéticos (A...Z, a...z) e caracteres especiais (por exemplo, \$%^&*?!@). Uma seqüência de caracteres deve ser apresentada entre aspas. Exemplo: "fone: 1234-1567", "apelido@provedor.com.br", "esta é uma frase!".

Lógico: pode armazenar apenas os valores verdadeiro (true ou 1) e falso (false ou 0).

Declarar que uma variável é de um tipo, quer dizer que a variável vai receber só determinado tipo de dado.

Exercícios:

1. Assinale as variáveis que possuem nome inválido:

- | | | | |
|--------------|-------------|---------------|---------------|
| a) NOME | b) 5dia | c) iDAde | d) "contador" |
| e) 0_abc | f) vel1 | g) nota_final | h) a*c |
| i) data nasc | j) inicio_2 | h) dia/mes | k) código |
| l) var-2 | m) ano | n) volta"14" | p) !Media |

2. Qual a diferença entre variável e constante?

3. Quais são os tipos básicos das seguintes variáveis:

- a) "nome"
- b) "5+5"
- c) 10
- d) 15/100
- e) "12345"
- f) F0
- g) "FF"