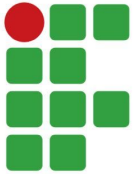


INSTITUTO FEDERAL
Bahia
Campus Santo Antônio de Jesus

http://www.portal.ifba.edu.br/santoantonio

LINGUAGEM DE PROGRAMAÇÃO

Prof. George Pacheco Pinto

Agenda

- Introdução à Linguagem de Programação

Algoritmos

- Um algoritmo é uma sequência lógica, finita e definida de instruções que devem ser seguidas para resolver um problema ou executar uma tarefa;

- Importante:
  - Um algoritmo é um “caminho” para a solução de um problema e, em geral, existem muitos caminhos que levam a uma solução satisfatória;

Estrutura Básica dos Algoritmos

Exemplo:
Algoritmo para calcular o dobro de um dado número.

Dados de Entrada:
Número

Processamento:
dobro = número x 2

Saída:
dobro do número

Formas de Representação

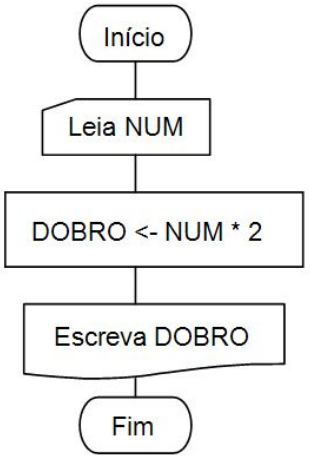
- Linguagem natural, através de uma lista de procedimentos bem definida, na qual as instruções são executadas passo a passo a partir do começo da lista;

- Fluxograma, uma ideia que pode ser facilmente visualizada através de uma notação gráfica;

- Linguagem de programação, tal como C, Java, Python, etc.

Fluxogramas

EXEMPLO



EXPLICAÇÃO

Início do algoritmo

Entrada do número

Cálculo do dobro do número

Apresentação do resultado

Fim do algoritmo

Fluxogramas

- VANTAGENS
  - Uma das ferramentas mais conhecidas
  - Figuras dizem muito mais que palavras
  - Padrão mundial

- DESVANTAGENS
  - Pouca atenção aos dados, não oferecendo recursos para descrevê-los ou representá-los
  - Complica-se à medida que o algoritmo cresce

Linguagens

```
Algoritmo CALCULA_DOBRO
Var
    NUM, DOBRO: INTEIRO
inicio
   Leia NUM
   DOBRO <- 2 * NUM
   Escreva DOBRO
fimalgoritmo
```

Linguagens

- VANTAGENS:
  - Linguagem que o computador entende;
  - Fortemente Padronizado;
  - Suporta qualquer tamanho de algoritmo;

- DESVANTAGENS:
  - Exige um conhecimento de lógica de programação;
  - É preciso conhecer a sintaxe da linguagem para extrair ideias básicas sobre o algoritmo.

Linguagens

- Característica das LP´S
  - número finito de instruções
  - rigidez Sintática
  - rigidez Semântica

- Classificação
  - Máquina
  - Baixo nível (Assembly)
  - Alto nível (C, JAVA, PASCAL)

Organização das linguagens

- Linguagens de Máquina
- Linguagens simbólicas ou de baixo nível
- Linguagens de alto nível

Linguagens de máquina

- A língua nativa que a máquina entende

- 1ª Geração
  - Definida pelo projeto de Hardware;
  - Dependente do Hardware;
  - Sequência de números ou bits;
  - Codificação Complexa;
  - Difícil entendimento.

Linguagem de baixo nível

- 2ª Geração de linguagens - Assembly

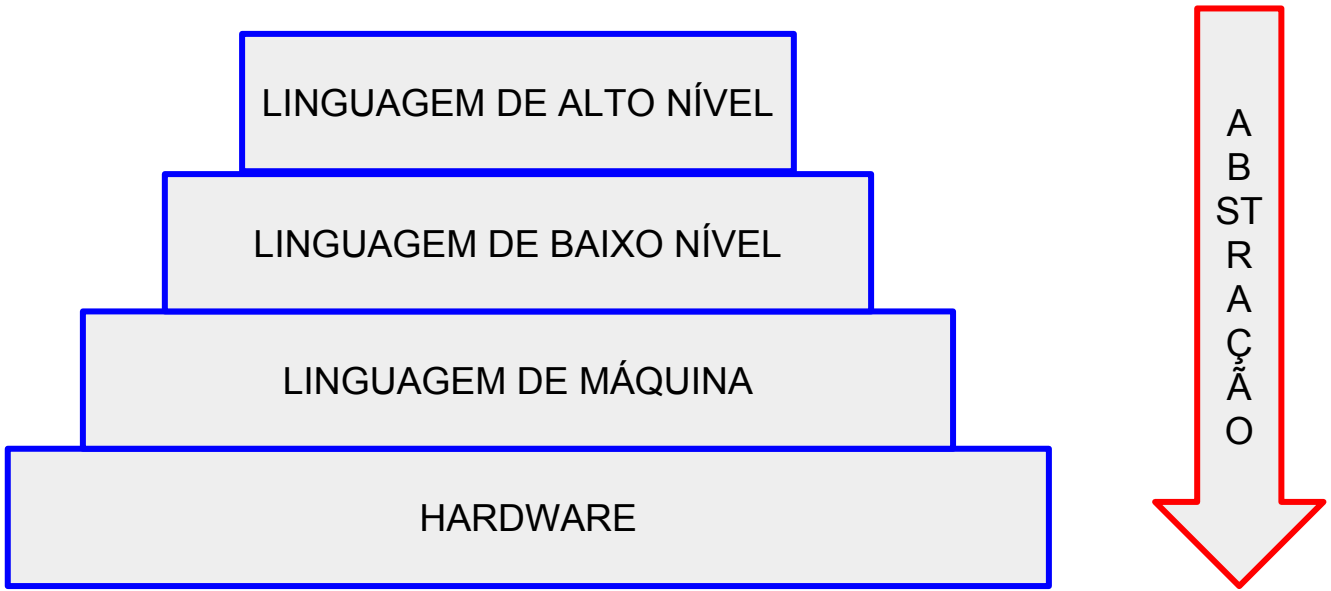
- Características:
  - ainda atrelado a arquitetura (Registradores e Instruções);
  - nomes e símbolos - Instruções elementares;
  - assemblers - Tradutores;
  - maior clareza da linguagem;
  - muitas instruções para poucas tarefas.
  - **Ex: ADD A,B**

LINGUAGEM DE ALTO NÍVEL

- 3ª Geração de linguagens – C/C++, JAVA, FORTRAN

- Características:
  - independente de arquitetura;
  - sintaxe próxima das linguagens naturais;
  - compiladores/interpretadores;
  - maior clareza da linguagem.

LINGUAGENS


LINGUAGEM DE ALTO NÍVEL
LINGUAGEM DE BAIXO NÍVEL
LINGUAGEM DE MÁQUINA
HARDWARE
ABSTRAÇÃO

MÉTODOS DE IMPLEMENTAÇÃO

- Compilação
  - lê todo o programa e o converte para código-objeto (0's e 1's) e pronto. Sempre quando tiver que ser executado é só chamá-lo, todas instruções já estão prontas para tal, não tem mais vínculo com seu código-fonte.

- Interpretação
  - lê linha a linha do fonte, o examina sintaticamente e o executa. Cada vez que o programa for executado esse processo tem de ser repetido e o interpretador é chamado.

Métodos de implementação

- Híbrido
  - traduzem os programas em linguagem de alto nível para uma linguagem intermediária projetada para facilitar a interpretação.

Referências

Consultar ementário.