

1 - Compreende-se por linguagens de baixo nível aquela cujas instruções estão mais próximas ou correspondem diretamente ao código de máquina. Estão diretamente relacionadas com a arquitetura do computador e com isso pode-se dizer que existem tantas linguagens de baixo nível quanto forem os processadores. Já as linguagens de alto nível possuem um nível de abstração maior, mais longe da linguagem da máquina e mais próxima à linguagem humana.

A compilação é uma etapa fundamental para que um programa possa ser executado, sendo necessário que seja transformado em um formato de máquina. Esse processo de transformação pode ser por compilação ou interpretação. A diferença entre os dois é que na compilação, a transformação ocorre apenas uma vez e o código final estará no formato de máquina. Já no caso de interpretação, a cada execução do programa ocorre essa transformação.

5 - A) Imperativas

É um modelo de programação que baseia-se na execução sequencial de comandos e no armazenamento de informações, dados, que são alteráveis ao longo do processamento do algoritmo. É definida por três conceitos: variáveis, atribuições e sequência.

No paradigma imperativo, o algoritmo é escrito de forma sequencial, ou seja, uma variável possui um valor diferente antes e depois do processo de atribuição. O uso de comandos de repetição são usados para acessar uma sequência de localizações na memória como um vetor, ou armazenar um valor em uma determinada variável.

Ex: C, Pascal e FORTRAN.

B) Funcionais

As linguagens funcionais estão voltadas à avaliação de expressões formadas com a utilização de funções que procuram combinar valores básicos e não simplesmente executar comandos, diferente das linguagens estruturadas.

Ex: LISP, Haskell e OCaml.

C) Lógicas

Na programação lógica, diferente da programação imperativa ou estruturada, descreve-se o resultado para assim obtê-lo. Faz o uso da lógica matemática em sua essência.

Ex: PROLOG, QLISP e Ether.

D) Marcação/Híbrida

As linguagem de marcação são aplicados a um texto ou a dados, com a finalidade de adicionar informações sobre esse texto ou dado, ou sobre trechos específicos.

Ex: XML, CSS e HTML.

6 - $\langle \text{expr} \rangle \rightarrow \langle \text{expo} \rangle * \langle \text{expr} \rangle \mid \langle \text{expo} \rangle$

$\langle \text{expo} \rangle \rightarrow \langle \text{base} \rangle ^{\langle \text{expo} \rangle} \mid \langle \text{base} \rangle$

$\langle \text{base} \rangle \rightarrow A \mid B \mid C$

7 - É a forma que o programador pensa ao abstrair e analisar informações para buscar a solução de um determinado problema. O paradigma permite ou proíbe o uso de algumas técnicas de programação. Em resumo, um paradigma pode ser visto como um padrão de resolução de problemas que se relaciona com um determinado gênero de linguagens.

Como por exemplo o paradigma estrutural, que permite a resolução de qualquer problema através do uso das estruturas de repetição, sequência ou condição. A ideia desse paradigma é que com o uso dessas estruturas, seja possível solucionar qualquer problema com o menor número possível de linhas de comandos.

Já o paradigma orientado à objeto é composto por objetos que possui suas propriedades (atributos) e operações (métodos) que podem ser executados por eles. Todas as funcionalidades e atributos de cada entidade do sistema são armazenados em classe que representam essa entidade.

8 - O analisador léxico é responsável por analisar todo o código fonte do programa e transformá-los em uma sequência de símbolos (ou tokens) para que possam ser manipulados mais facilmente por um parser.

$\langle \text{program} \rangle ::= \text{begin } \langle \text{while_loop} \rangle \text{ end}$

$\langle \text{while_loop} \rangle ::= \text{while } (\langle \text{conditions} \rangle) \langle \text{statements} \rangle$

$\langle \text{conditions} \rangle ::= \{ \langle \text{condition} \rangle (\&\& \mid ||) \} \mid \langle \text{condition} \rangle$

$\langle \text{condition} \rangle ::= \{ \langle \text{expression} \rangle (> | < | \geq | \leq | ==) \} | \langle \text{expression} \rangle$

$\langle \text{statements} \rangle ::= \{ \langle \text{statements} \rangle \langle \text{expression} \rangle \} | \langle \text{expression} \rangle$

$\langle \text{expression} \rangle ::= \{ \langle \text{integer} \rangle (+ | - | * | /) \langle \text{expression} \rangle \} | \langle \text{integer} \rangle$

$\langle \text{integer} \rangle ::= \{ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 \} | \langle \text{integer} \rangle$

10 - Análise semântica: Consiste na terceira etapa da compilação, e tem por finalidade garantir que as declarações e expressões de um programa estão semanticamente corretos. Ou seja, certificar que seus significados estão coerentes conforme a manipulação de dados é feita.

Semântica operacional: É uma categoria da programação formal em que certas propriedades desejadas do programa como exatidão e segurança são verificadas através do uso de declarações lógicas.

Semântica axiomática: Normalmente associada com o fato do programa estar correto, basicamente usando a análise imutável das instruções de códigos de um programa. Há também uma aproximação mais dinâmica que analisa um programa em que valores de variáveis mudam conforme o programa executa.

Semântica denotacional: Baseia-se no reconhecimento de que os programas e objetos que manipulam são modelados por símbolos matemáticos, geralmente por funções definidas que representam a execução de um programa.

11 - Programa A) $x = 122 * y - 144$

$$\{ x > 144 \}$$

$$122 * y - 144 > 144$$

$$y > 288/122$$

$$y > 2.3$$

Programa B) $y = 5 * x - 5$

$$x = y + 5$$

$$\{ x < 45 \}$$

$$y + 5 < 45$$

$$y < 40$$

$$5 * x - 5 < 40$$

$$x < 9$$

Programa C) $\{ y > 2 \}$

Axioma de atribuição para o IF:

$$y = y + 2$$

$$y + 2 > 2$$

$$y > 0$$

Axioma de atribuição para o Else:

$$y = y - 2$$

$$y - 2 > 2$$

$$y > 4$$

Sabendo que $\{ y > 4 \} \Rightarrow \{ y > 0 \}$, a regra de consequência nos permite usar $\{ y > 4 \}$ como pré-condição da instrução de seleção.

Programa D) $\{ i = N \}$

$$wp = (i = i + 1)$$

$$i + 1 = N$$

$$i = N - 1$$

A pré-condição para o loop while é o i ser menor ou igual a N .

Com isso, $\{ i \leq N \}$

Programa E) $\{ sn == n * a \}$

$$sn = sn + a$$

$$n * a = sn + a$$

$$n = (sn / a) + a / a$$

$$\{ n = (sn / a) + 1 \}$$