

FUNDAMENTOS

Plataformas

Configurando o ambiente

Escolha sua IDE

SINTAXE

Anatomia das classes

Tipos e Variáveis

Operadores

Métodos

Escopo

Palavras reservadas

Documentação

Terminal e Argumentos

CONTROLE DE FLUXO

Conceito

Estruturas condicionais

Estruturas de repetição

Estruturas excepcionais



Conceito de POO

Você já ouviu falar a expressão, linguagem de baixo e de alto nível?

À medida que a tecnologia vem evoluindo, as linguagens de programação também, e é esta transição natural, que determina, quando estamos nos referindo a linguagem de baixo e alto nível.

Baixo nível: São linguagens que estão mais próximas, da interpretação da máquina, diante do algoritmo desenvolvido. Exemplo: **Linguagem Assembly e C.**

Alto nível: São linguagens que disponibilizam, uma proposta de sintaxe (forma de escrever processos para serem executados pelo computador) mais próxima de interpretação humana. Exemplo: **Java, JavaScript, Python e C++**

Exemplo de um simples `Hello World` em **Assembly** versus **Python**:

Assembly

Python

```
section .text

    global _start

_start:

    mov     edx, len

    mov     ecx, msg

    mov     ebx, 1

    mov     eax, 4

    int     0x80

    mov     eax, 1

    int     0x80

section .data

msg db  'Hello, world!',0xa

len equ $ - msg
```

É bem notória a diferença, entre as duas perspectivas de linguagem.

Programação estruturada

A programação estruturada é um [paradigma de programação](#), que visa melhorar a clareza, a qualidade e o tempo de desenvolvimento de um [programa de computador](#), fazendo uso extensivo, das construções de fluxo de controle estruturado de seleção ([if / then / else](#)) e repetição (while e [for](#)), [estruturas de bloco](#) e [sub - rotinas](#) .

O que devemos ter em mente, é que na programação estruturada, implementamos algoritmos com estruturas sequenciais denominados de procedimentos lineares, podendo afetar o valor das variáveis de escopo local ou global em uma aplicação.

Programação orientada a objetos

POO é um [paradigma de programação](#), baseado no conceito de "[objetos](#)", que podem conter [dados](#) na forma de [campos](#), também conhecidos como *atributos*, e códigos, na forma de [procedimentos](#), também conhecidos como [métodos](#).

O que precisamos entender, é que cada vez mais as linguagens se adequam ao cenário real, proporcionando assim, que o programador desenvolva algoritmos mais próximo de fluxos comportamentais, logo, tudo ao nosso redor é representado como Objeto.

Enquanto a programação estruturada é voltada a procedimentos e funções, definidas pelo usuário, a programação orientada a objetos é voltada a conceitos, como o de classes e objetos.



Previous
Cases

Next
Classes



Last updated 1 year ago