Programação Orientada a Objetos

Leonardo Buta

Desenvolvedor .NET

 **@llbuta**

 **https://www.linkedin.com/in/leonardo-buta**

Objetivo Geral

Apresentar e explorar o paradigma de programação orientado a objeto, seus usos e como ele é aplicado no dia a dia da programação.

Percurso

Etapa 1 **Introdução POO, Abstração e Encapsulamento**

Etapa 2 Herança e Polimorfismo

Etapa 3 Classes Abstratas e Interfaces

Etapa 1

Introdução POO, Abstração e Encapsulamento

// Programação Orientada a Objetos

O que é a POO?

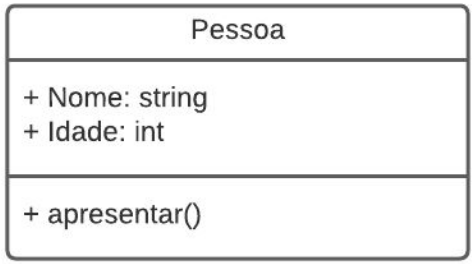
A POO é um paradigma de programação, ou seja, corresponde a uma técnica de programação para um fim específico.

Dentro desta técnica, existem quatro pilares:

- Abstração
- Encapsulamento
- Herança
- Polimorfismo

O que é a POO?

O principal conceito da POO são classes e objetos!



Classe

Objeto

Paradigmas de programação

Um paradigma nada mais é do que um modelo de técnicas, estruturas e formas de solucionar um problema.

Paradigma de programação é diferente de linguagem de programação.

Uma linguagem de programação implementa um ou mais paradigmas.

Paradigmas de programação

- Programação orientada a objetos (é o que estamos estudando!)
- Programação estruturada
- Programação imperativa
- Programação procedural
- Programação orientada a eventos
- Programação lógica

e por aí vai...

Tipos de paradigmas

Language overview [edit]

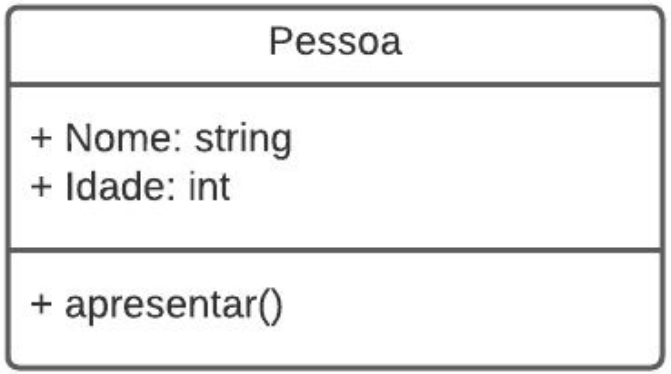
List of multi-paradigm programming languages

Language	Number of Paradigms	Concurrent	Constraints	Data-flow	Declarative	Distributed	Functional	Meta-programming	Generic	Imperative	Logic	Reflection	Objectoriented	Pipe-lines	Visual	Rule--based	Other paradigms
Ada[2][3][4][5][6]	5	Yes[a 1]	No	No	No	Yes	No	No	Yes	Yes	No	No	Yes[a 2]	No	No	No	No
ALF	2	No	No	No	No	No	Yes	No	No	No	Yes	No	No	No	No	No	No
AmigaE[citation needed]	2	No	No	No	No	No	No	No	No	Yes	No	No	Yes[a 2]	No	No	No	No
APL	3	No	No	No	No	No	Yes	No	No	Yes	No	No	No	No	No	No	Array (multi-dimensional)
BETA[citation needed]	3	No	No	No	No	No	Yes	No	No	Yes	No	No	Yes[a 2]	No	No	No	No
C++	7 (15)	Yes[7][8][9]	Library[10]	Library[11][12]	Library[13][14]	Library[15][16]	Yes	Yes[17]	Yes[a 3]	Yes	Library[18][19]	Library[20]	Yes[a 2]	Yes[21]	No	Library[22]	Array (multi-dimensional; using STL)
C#	6 (7)	Yes	No	Library[a 4]	No	No	Yes[a 5]	No	Yes	Yes	No	Yes	Yes[a 2]	No	No	No	reactive[a 6]
ChucK[citation needed]	3	Yes	No	No	No	No	No	No	No	Yes	No	No	Yes[a 2]	No	No	No	No
Claire	2	No	No	No	No	No	Yes	No	No	No	No	No	Yes[a 2]	No	No	No	No
Clojure	5	Yes[23][24]	No	No	Yes	No	Yes[25]	Yes[26]	No	No	Library[27]	No	No	Yes[28]	Editor[29]	No	Multiple dispatch,[30] Agents[31]
Common Lisp	7 (14)	Library[32]	Library[33]	Library[34]	Yes[35]	Library[36]	Yes	Yes	Yes[37]	Yes	Library[38]	Yes	Yes (multiple dispatch, method combinations)[39][a 2]	Library[40]	No	Library[41]	Multiple dispatch, meta-OOP system,[42] Language is extensible via metaprogramming.
Curl	5	No	No	No	No	No	Yes	No	Yes[a 3]	Yes	No	Yes	Yes[a 2]	No	No	No	No
Curry	4	Yes	Yes	No	No	No	Yes	No	No	No	Yes	No	No	No	No	No	No
D (version 2.0)[43][44]	6	Yes[a 7]	No	No	No	No	Yes	Yes[45][a 3]	Yes[a 3]	Yes	No	No	Yes[a 2]	No	No	No	No
Dylan[citation needed]	3	No	No	No	No	No	Yes	No	No	No	No	Yes	Yes[a 2]	No	No	No	No
E	3	Yes	No	No	No	Yes	No	No	No	No	No	No	Yes[a 2]	No	No	No	No
ECMAScript[46][47] (ActionScript, E4X, JavaScript, JScript)	4 (5)	partial (promises, native	No	No	Library[48][49]	No	Yes	No	No	Yes	No	Yes	Yes[a 9]	Library[50][51]	Editor[52]	No	reactive,[a 10][53] event driven[a 11][a 12]

Fonte: https://en.wikipedia.org/wiki/Comparison_of_multi-paradigm_programming_languages

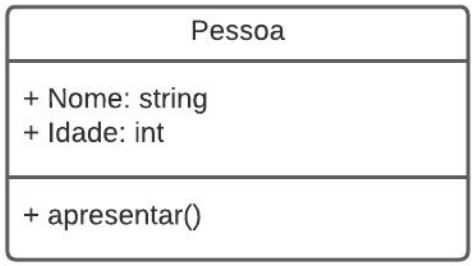
Abstração

Abstrair um objeto do mundo real para um contexto específico, considerando apenas os atributos importantes.



Classe

Abstração



Classe

Objeto

Encapsulamento

O encapsulamento serve para proteger uma classe e definir limites para alteração de suas propriedades.

Serve para ocultar seu comportamento e expor somente o necessário.

Encapsulamento