

## Comparando OOP com Programação Procedural - Parte I

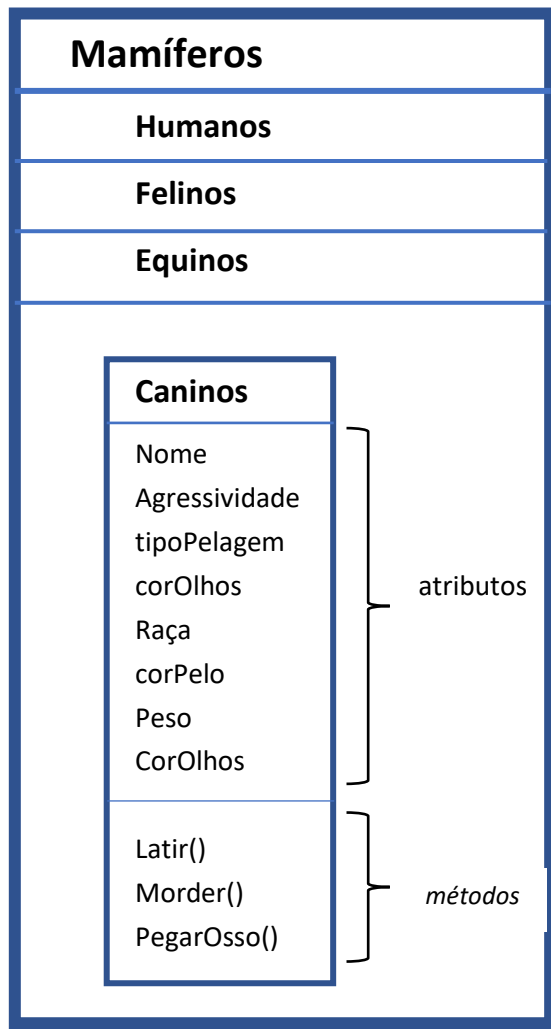
Mário Leite

...

A Programação Orientada a Objetos (OOP: **O**bject-**O**riented **P**rogramming em inglês) é uma das tecnologias de criação de sistemas computacionais mais divulgados atualmente. Na verdade, esta tecnologia é velha, desenvolvida no início dos anos 60 do Século XX, e de acordo com a literatura foi primeiramente apresentada em 1966, na Noruega, como uma extensão da linguagem ALGOL 60; esta considerada a mãe de todas as linguagens. Portanto, para a tristeza dos puristas desta tecnologia: a OOP veio da Programação Procedural e, embora seja uma tecnologia muito poderosa, ela tem partes procedurais pois, os *métodos* nada mais são do que funções (com ou sem retorno), escritas de maneira *procedural/estruturada* e modularizada com pequenas porções de código, que são os *objetos*; portanto, ela é uma extensão da Programação Modular que se utiliza sim, dos passos sequencialmente escritos pelo programador. Então, embora muito se fale no “paradigma de orientação a objetos”, e o termo esteja corretíssimo, a programação propriamente dita - a solução do problema de acordo com a regra do negócio do sistema - é baseada na execução do algoritmo, aproveitando dos recursos oferecidos por esta tecnologia: *atributos* (variáveis com valores pré-definidos para cada tipo de objeto) e *métodos* (funções pré-definidas para a classe a que o objeto pertence). A **figura 1** mostra um esquema clássico de apresentar a OOP: uma classe que serve como referência básica original chamada de superclasse (“Mamíferos”) com classes derivadas desta (subclasses) e, finalmente, os objetos (chamados de *instâncias*) criados de uma subclasse; destacando a classe “Caninos”. Estes objetos (cães/cadelas) possuem as características herdadas da sua subclasse, que por sua vez herdou as características da superclasse, mas que podem ter, adicionalmente, características próprias. Um exemplo deste objeto é apresentado em seguida à figura 1. Por exemplo: todo cão tem dentes caninos (uma de suas características básicas), mas alguns podem ter características adicionais próprias como, por exemplo, nome, diferente raça, mais pelagem, cor dos olhos, comportamento diferente (mais agressivo, mais inteligente, pegar osso ou não, etc). Os *atributos* são as características da classe do objeto, e os *métodos* as ações que eles podem executar. Assim, a compreensão do mundo real, no qual o paradigma da OOP se apoia é, por demais, simples e fácil de entender; entretanto, do ponto de vista prático nas aplicações em sistemas comerciais para atender exigências do USUÁRIO, a coisa não é bem assim. Para começar: em QUALQUER paradigma de programação o que se faz, rigorosamente nesta ordem é: *Entrar com os Dados – Processar os Dados – Exibir a Informação*: é sempre assim; não interessa o tipo de paradigma em que está sendo desenvolvido o sistema! Então, a semente original são os DADOS. e é fundamental saber onde estão esses dados; como estão armazenados, como serão lidos e como se relacionam.

---

**Continua na Parte II**



**Figura 1 - Esquema de classes na OOP**

Classe “**Canino**”

Subclasse: “**Cão**”

Nome: Katie

Agressividade: nenhuma

tipoPelagem: pelo longo e sedoso

corOlhos: Olhos pretos

Raça: Shi Tzu

corPelo: “branco e preto”

Peso: 3.5 kg

corOlhos: preto

Latir()

Abanar()

PegarOsso()