

DIFERENÇA PROGRAMAÇÃO ORIENTADA A OBJETOS E PROGRAMAÇÃO ESTRUTURADA:

A diferença principal é que na **programação estruturada**, um programa é tipicamente escrito em uma única rotina (ou função) podendo, é claro, ser quebrado em subrotinas. Mas o fluxo do programa continua o mesmo, como se pudéssemos copiar e colar o código das subrotinas diretamente nas rotinas que as chamam, de tal forma que, no final, só haja uma grande rotina que execute todo o programa.

A **programação orientada a objetos** surgiu como uma alternativa a essas características da programação estruturada. O intuito da sua criação também foi o de aproximar o manuseio das estruturas de um programa ao manuseio das coisas do mundo real, daí o nome "objeto" como uma algo genérico, que pode representar qualquer coisa tangível. Esse novo paradigma se baseia principalmente em dois conceitos chave: **classes** e **objetos**. Todos os outros conceitos, igualmente importantes, são construídos em cima desses dois.

Definição de classe: As **classes** de programação são receitas de um objeto, aonde têm características e comportamentos, permitindo assim armazenar propriedades e métodos dentro dela.

Definição de atributo: **Atributos** são as características de um objeto, essas características também são conhecidas como variáveis, utilizando o exemplo dos cães, temos alguns atributos, tais como: cor, peso, altura e nome

Definição de objeto: **Objetos** (computacionais) são caracterizados por atributos e métodos. Atributos são as propriedades de um objeto. Métodos são as ações que um objeto pode realizar. Os objetos são características definidas pelas classes. Neles é permitido instanciar objetos da classe para inicializar os atributos e invocar os métodos.

Definição de métodos: **Métodos** são as ações que os objetos podem exercer quando solicitados, onde podem interagir e se comunicarem com outros objetos.

Instanciação de objeto: A instanciação é um processo por meio do qual se realiza a cópia de um **objeto** (classe) existente. Uma classe, a qual tem a função de determinar um tipo de dado, deve ser **instanciada** para que possamos utilizá-la.

Modificadores de acesso:

- **public:** permite acesso a qualquer código externo a classe.
- **protected:** permite acesso às classes filhas, mas proíbe a qualquer outro acesso externo.
- **private:** proíbe qualquer acesso externo à própria classe, inclusive das classes filhas.

Quando não declaramos nenhum modificador de acesso nos métodos, eles assumem o modificador de acesso private.

Métodos statics:

Os métodos static ou métodos da classe são **funções que não dependem de nenhuma variável de instância**

Os métodos statics geralmente são usados para funções de entrada e saída de dados.