# Programação Orientada a Objeto

O que é Programação Orientada a Objeto (POO):

* Oriented Object Programming ou OOP;
* Um paradigma de desenvolvimento;
* Uma maneira de resolver um problema, um modo de pensar;
* Não está ligado somente à linguagem de programação, mas a um entendimento amplo e atemporal para criação de softwares.

Utilização no mercado:

* Na maioria dos sistemas atuais;
* Várias linguagens (JS, TS, Java, Python, ...);
* Utilizado por analistas de sistemas e não somente por programadores;
  + Na fase levantamento de requisitos de sistemas;
  + Via gráficos, desenhos e textos.

Onde e quando eu vou utilizar?

* Linguagens;
* Quando quiser aplicação Padrões de Projetos;
* Quando quiser organizar seu código;
* Para melhor entendimento do código, pensando nele como objetos;
* Quando precisar fazer reuso do código;
* Para separar a complexidade de código, abstrair código e expor de maneira mais simples;
* Para classificar as rotinas e trechos de código.

Conceitos:

* Precisamos entender os conceitos e paradigma e não somente códigos;
* Uma boa fundamentação;
* Pegue papel e caneta, e anote os pontos para que você possa revisar posteriormente.

## Objetos:

(Uma caneta é um objeto)

Todo objeto possui:

* Propriedades e Funcionalidades;
* Estado e Comportamentos;
* Atributos e métodos.

Abstratos Vs Mundo Real

Trazemos a representação de algo do mundo real para objetos.

* Pessoa;
* Aluno;
* Produto;
* Carrinho de compras.

Porém, alguns objetos na programação não são fácies de identificar, pois são abstratos como, por exemplo:

* Autenticação;
* Autorização;

Nem todo objeto do mundo real fará parte do seu sistema.

Classes:

As classes na orientação a objetos funcionam como um molde para os objetos. Os objetos são criados a partir de uma classe e muitos deles podem ser feitos da mesma classe.

* Exemplo: Máquina de biscoitos
  + Instâncias

Classes em JavaScript:

* Syntactical sugar;
* Prototype;

Encapsulamento:

(Dirigir carro Vs conhecer o motor do carro)

* Colocar numa cápsula;
* Agrupamento de funções e variáveis;
* Esconder detalhes de implementação;
* Camada de segurança para os atributos e método.

Confusão e Solução

Programação Estruturada x Orientação a Objetos

**Programação Estruturada**

* Processa a entrada e manipulação dos dados, até a saída;
* Uso de sequências, estruturas de repetição e condições;
* Uso de uma rotina maior, ou sub-rotinas;
* Não existem restrições às variáveis.

**Programação Orientada a Objetos**

* Surge para trazer um cuidado ao uso estruturado;
  + Não elimina por completo o uso estruturado;
* Conceitos como Objetos e Classes;
* Cuidados com variáveis e rotinas (Encapsulamento);
* Melhor reuso do código (Herança).

Herança:

* Pais e filhos;
* Objetos podem herdar, ou estender, características bases;
* Uma cópia de atributos e métodos de outra classe.

Saiba:

* Um objeto pode estender de outro objeto que pode estender de outro objeto;
* Fácil reutilização de código.

Polimorfismo

Quando um objeto estende de outro (herança) talvez haja a necessidade de reescrever uma ou mais características (atributos e métodos) nesse povo objeto.

Recriamos então um método (ou mais) da classe herdada.

Polimorfismo significa muitas formas.

Abstração

Template ou identidade de uma classe que será construída no futuro.

* Atributos e métodos podem ser criados na classe de Abstração (Superclasse) MAS;
* A implementação dos métodos e atributos, só poderá ser feita na classe que irá herdar essa Abstração.