

# Python - Conceitos OOP

Python tem sido uma linguagem orientada a objetos desde a época em que existiu. Devido a isso, criar e usar classes e objetos é absolutamente fácil. Este capítulo ajuda você a se tornar um especialista no uso do suporte de programação orientada a objetos do Python.

Se você não tem nenhuma experiência anterior com programação orientada a objetos (OO), você pode querer consultar um curso introdutório sobre ela ou pelo menos algum tipo de tutorial para ter uma compreensão dos conceitos básicos. No entanto, aqui está uma pequena introdução à Programação Orientada a Objetos (OOP) para ajudá-lo.

## Abordagem Orientada a Procedimentos

As primeiras linguagens de programação desenvolvidas nas décadas de 50 e 60 são reconhecidas como linguagens procedurais (ou orientadas a procedimentos).

Um programa de computador descreve o procedimento para realizar determinada tarefa escrevendo uma série de instruções em uma ordem lógica. A lógica de um programa mais complexo é dividida em blocos de instruções menores, mas independentes e reutilizáveis, chamados funções.

Cada função é escrita de forma que possa interagir com outras funções do programa. Os dados pertencentes a uma função podem ser facilmente compartilhados com outros na forma de argumentos, e a função chamada pode retornar seu resultado para a função chamadora.

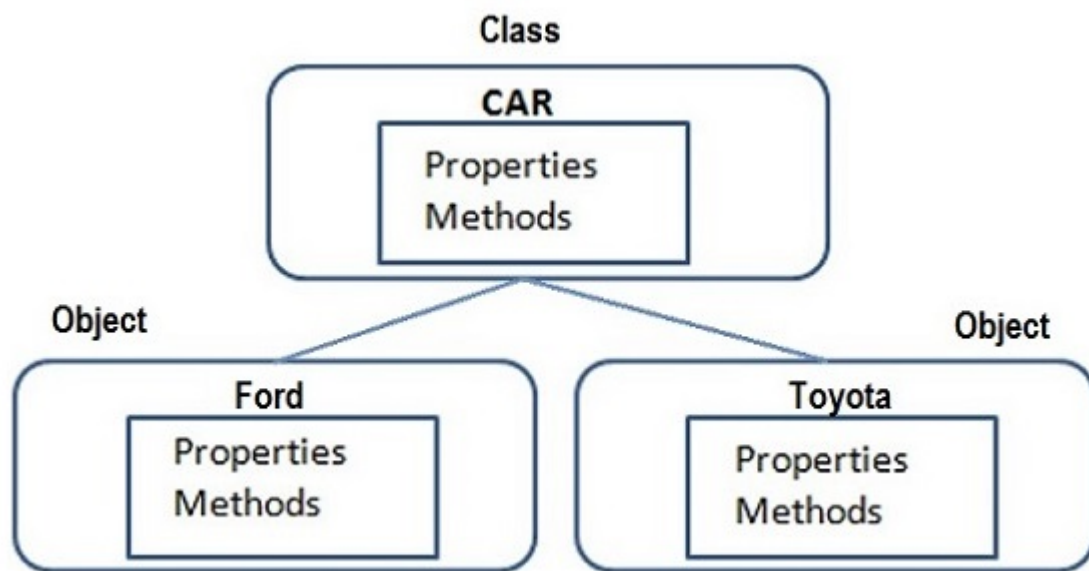
Problemas proeminentes relacionados à abordagem processual são os seguintes -

- A sua abordagem de cima para baixo torna o programa difícil de manter.
- Ele usa muitos itens de dados globais, o que é indesejado. Muitos itens de dados globais aumentariam a sobrecarga de memória.
- Dá mais importância ao processo e não considera dados da mesma importância e os toma como garantidos, movimentando-se livremente pelo programa.
- A movimentação de dados entre funções é irrestrita. No cenário da vida real, onde existe uma associação inequívoca de uma função com os dados que se espera que ela processe.

## Python - Conceitos OOP

No mundo real, lidamos e processamos objetos, como aluno, funcionário, fatura, carro, etc. Objetos não são apenas dados e nem apenas funções, mas uma combinação de ambos. Um objeto do mundo real possui atributos e comportamento associados a ele.





## Atributos

- Nome, turma, disciplinas, notas, etc., do aluno
- Nome, designação, departamento, salário, etc., do funcionário
- Número da fatura, cliente, código e nome do produto, preço e quantidade, etc., em uma fatura
- Número de matrícula, proprietário, empresa, marca, potência, velocidade, etc., do carro

Cada atributo terá um valor associado a ele. Atributo é equivalente a dados.

## Comportamento

Processando atributos associados a um objeto.

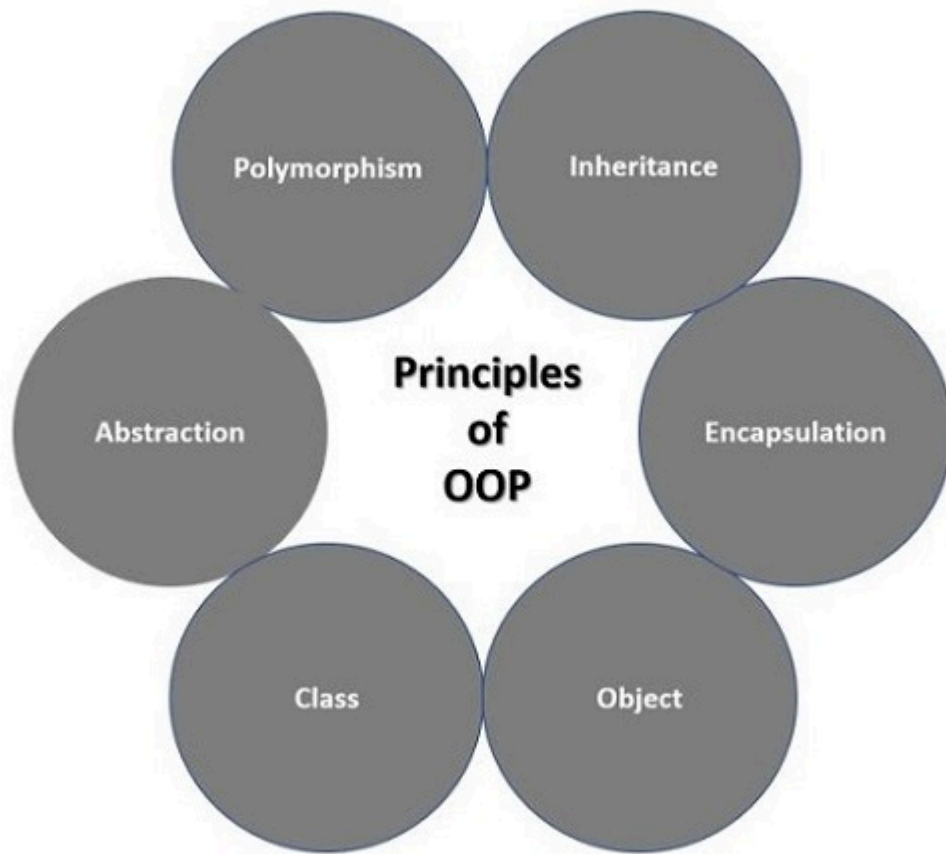
- Calcular a porcentagem das notas do aluno
- Calcular incentivos a pagar ao funcionário
- Aplicar GST ao valor da fatura
- Medir a velocidade do carro

O comportamento é equivalente à função. Na vida real, atributos e comportamento não são independentes um do outro, mas sim coexistem.

A característica mais importante da abordagem orientada a objetos é definir atributos e suas funcionalidades como uma única unidade chamada classe. Ele serve como um modelo para todos os objetos com atributos e comportamento semelhantes.

Na POO, a classe define quais são os atributos que seu objeto possui e como é seu comportamento. Object, por outro lado, é uma instância da classe.

O paradigma de programação orientada a objetos é caracterizado pelos seguintes princípios -



## Aula

Um protótipo definido pelo usuário para um objeto que define um conjunto de atributos que caracterizam qualquer objeto da classe. Os atributos são membros de dados (variáveis de classe e variáveis de instância) e métodos, acessados via notação de ponto.

## Objeto

Um objeto individual de uma determinada classe. Um objeto obj que pertence a uma classe Circle, por exemplo, é uma instância da classe Circle. Uma instância exclusiva de uma estrutura de dados definida por sua classe. Um objeto compreende membros de dados (variáveis de classe e variáveis de instância) e métodos.

## Encapsulamento

Os membros de dados da classe estão disponíveis para processamento apenas em funções definidas dentro da classe. As funções de classe, por outro lado, são acessíveis fora do contexto de classe. Portanto, os dados do objeto ficam ocultos do ambiente externo à classe. A função de classe (também chamada de método) encapsula os dados do objeto para que o acesso injustificado a eles seja evitado.

## Herança

Uma abordagem de modelagem de software de OOP permite estender a capacidade de uma classe existente para construir uma nova classe em vez de construir do zero. Na terminologia OOP, a classe existente é chamada de classe base ou pai, enquanto a nova classe é chamada de classe filha ou subclasse.

A classe filha herda definições de dados e métodos da classe pai. Isso facilita a reutilização de recursos já disponíveis. A classe filha pode adicionar mais algumas definições ou redefinir uma função da classe base.

## Polimorfismo

Polimorfismo é uma palavra grega que significa ter múltiplas formas. Na POO, o polimorfismo ocorre quando cada subclasse fornece sua própria implementação de um método abstrato na classe base.