

# Objetos

Carlos Arruda Baltazar  
UNIP – Cidade Universitária

Programas desenvolvidos utilizando programação estruturada, mesmo particionado em funções, acabam ficando muito extensos, o que acaba gerando uma dificuldade na manutenção. A Programação Orientada a Objetos (POO) surgiu com a finalidade de facilitar o processo de desenvolvimento, tornando-o mais intuitivo e otimizado.

## Vantagens:

- Reutilização de código;
- Projetos bem definidos;
- Trechos de código com atividade bem determinada;
- Fácil manutenção;

## Desvantagens:

- Dificuldade de aprendizado;
- Projeto mal definido;

Na literatura, um objeto pode ser definido como um artefato material ou abstrato que pode ser percebido pelos sentidos e descrito através de suas características, comportamentos e estados atuais. Logo, em programação, o objeto é uma abstração de software que pode ser definido por características e ações:

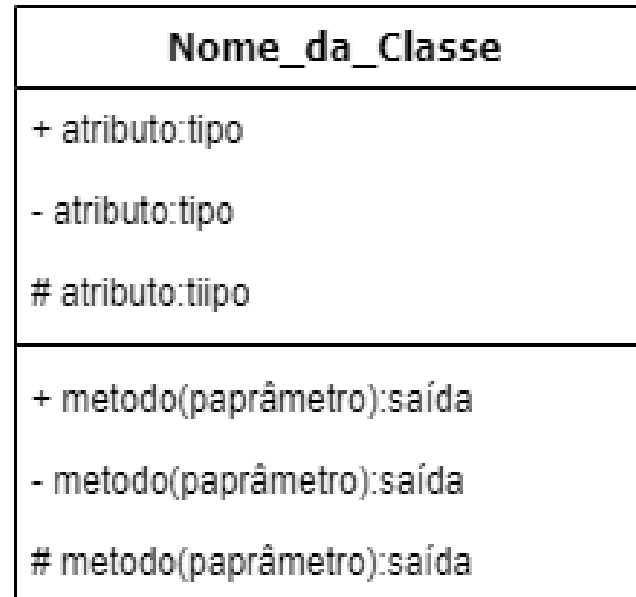
- **Atributos:** são variáveis responsáveis por armazenar as características do objeto;
- **Métodos:** são rotinas que, quando executadas realizam algumas tarefas como alterar o conteúdo dos próprios atributos, bem como realizar interações entre objetos;
- **Estado:** emerge quando valores são definidos para os atributos de um objeto.

Uma classe pode ser considerada um modelo para a criação de objetos. Portanto são as classes que dão “vida” ao objeto, pois nela são definidas as características do objeto através dos atributos e também são criadas as suas funcionalidades através dos métodos. Assim, uma classe tem o papel de definir e estabelecer o comportamento do objeto. Também, é necessário lembrar que vários objetos podem ser criados a partir de uma única classe, o que diferencia os objetos são suas características.

- Público (*public*): a classe, atributo ou método são vistos pelas demais classes se estiverem dentro do mesmo pacote;
- Protegida (*protected*): o acesso a uma classe, atributo ou método se restringem apenas a própria classe e suas subclasses;
- Privado (*private*) – o acesso o acesso a uma classe, atributo ou método só é permitido somente pela própria classe.



Em programação, um diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos. É uma modelagem muito útil para o desenvolvimento de sistemas, pois define todas as classes que o sistema necessita possuir.



Nome_da_Classe
+ atributo:tipo
- atributo:tipo
# atributo:tipo
+ metodo(papramento):saída
- metodo(papramento):saída
# metodo(papramento):saída



Pessoa
- nome:String
- endereco:String
- telefone:String
+ SetNome(String nome):void
+ SetEndereco(String:Endereco):void
+ SetTelefone(Striing:Telefone):void
+ GetTelefone():String
+ GetEndereco():String
+ GetNome():String

# OBRIGADO