



Programação Orientada a Objetos – 2025.2

Prof. João Luiz

joao.laoliveira@gmail.com



Aula 1 – Classes e Objetos

O que é um Objeto?

- Coisa material (tangível)
- Percebido pelos sentidos
- Sentir, manipular, interagir (diretamente ou com outros objetos)
- Características
- Realizar determinadas ações



Objeto em Programação

- Características

Conjunto de dados, **atributos**



- Realizar determinadas ações

Comportamentos associados: **métodos**



Orientação a Objetos

Softwares:

- Análise (entender, especificações e requisitos)
 - Projeto (esqueleto, estrutura, diagramas)
 - Programação (escrita, especificações da linguagem) 
-
- Análise orientada a objetos
 - Projeto orientado a objetos
 - Programação orientada a objetos 



O que são Classes?

- Estrutura (molde)
- Define o objeto

 Analogia:

Classe é a **receita** de um bolo.

Objeto é o **bolo pronto**, feito a partir da receita.

Obs: No Python, tudo é objeto. 



Atributos e Métodos

- **Atributos** (propriedades)
 - Características relevantes
 - O que diferencia de outros objetos
 - Análogo às **variáveis**
- **Métodos**
 - Ações que podem ser tomadas naquela objeto.
 - Análogo às **funções**



Exemplo prático

- Diagrama de Classes:

<https://lucid.app>

<https://www.cybermedian.com/pt/a-comprehensive-guide-to-uml-class-diagram/>

- Exemplos:

Numa casa, tem um cachorro chamado Toby e um gato chamado Tom.

Numa fazenda, temos maçãs que são colhidas em barris e laranjas que são colhidas em cestas.





Relações Entre Classes



Relação Entre Classes

- **Associação**

Relação mais simples

- **Composição e Agregação**

Uma classe é um componente de outra

- **Herança**

Relação de especialização/abstração entre classes





Cronograma do Curso

Semana	Data	Conteúdo
01		Apresentação do curso; Conceitos básicos: Objeto, classe, abstração; composição e herança.
02 e 03		Classes em Python; Atributos e Métodos; Criação e instanciação de objetos; Escopo.
04		Boas práticas de programação; Documentação de código; Uso de docstrings; Tipagem em Python; Princípio DRY; Modularidade; Google Python Style Guide; PEP 8.
05		Módulos e Pacotes; Organização de módulos; Imports absolutos e relativos; Importação e reutilização de código.
06		Herança e polimorfismo; Relacionamento entre classes: herança simples e múltipla; Sobrescrita de métodos e herança de atributos.
07		Encapsulamento e visibilidade de membros; Propriedades.
08		Composição e agregação; Implementação de relacionamentos em Python.
09		1ª Avaliação

Semana	Data	Conteúdo
10		Tratamento de exceções em Python; Uso de blocos try-except-finally; Lançamento e captura de exceções personalizadas.
11 e 12		Testes em Python; Comando assert; Uso do módulo de testes unitários (unittest).
13		Manipulação de arquivos em Python; Leitura e escrita de arquivos de texto; Serialização de objetos em arquivos; Gerenciamento de memória e coleta de lixo em Python; Uso de referências e contagem de referências; Ciclo de vida dos objetos.
14		Comprehensions; Expressões Lambda.
15		Introdução a Padrões de Projeto; Padrão decorador; Padrão Observador.
16		Padrão de Estratégia; Padrão de Estado.
17		Padrão Singleton; Padrão Template; Outros padrões.
18		2ª Avaliação



Obrigado!

Prof. João Luiz

joao.laoliveira@gmail.com