

**INSTITUTO
FEDERAL**

Paraíba

Campus
Cajazeiras

PROGRAMAÇÃO ORIENTADA À OBJETO

PROFESSOR MICHEL
CAJAZEIRAS / IFPB

Herança

OBJETIVOS

- ▶ Compreender o conceito de Herança;
- ▶ Compreender o conceito de super classe;
- ▶ Desenvolver algoritmos utilizando herança em python;
- ▶

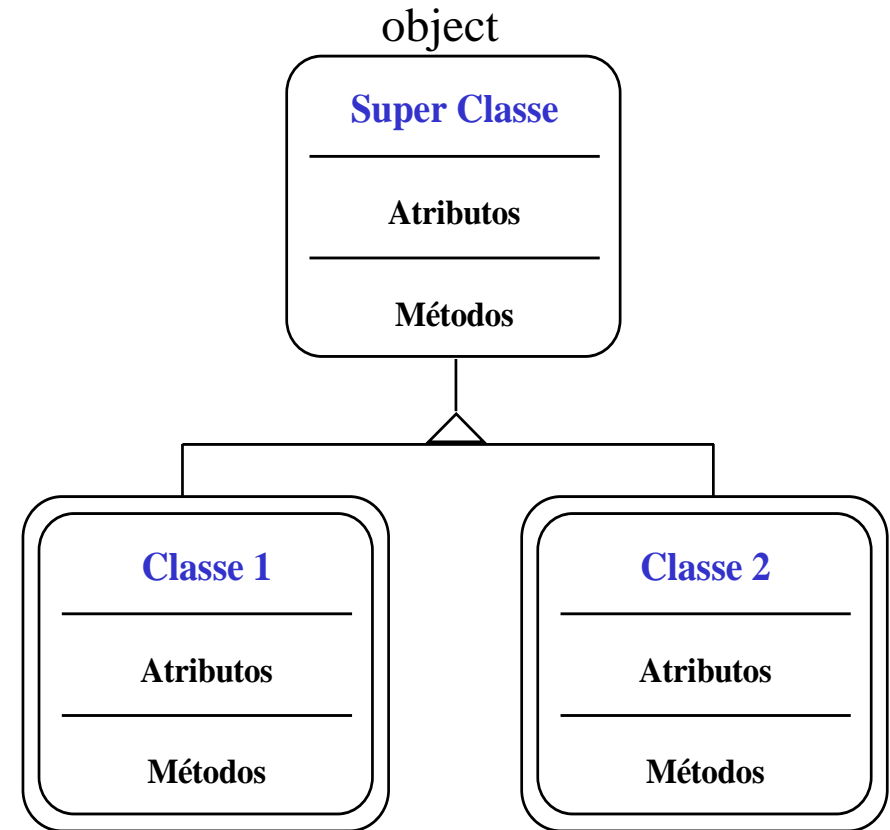
ROTEIRO

- ▶ Herança
 - ▶ Em Python
 - ▶ Super classe
 - ▶ Exemplo



Herança

- ▶ É a capacidade de criar objetos e manter as propriedades e comportamento de seus ancestrais.
- ▶ Este conceito permite que você crie uma hierarquia de objetos.
- ▶ Primeiro cria-se um objeto genérico, depois descendente com funcionalidades específicas.
- ▶ A grande vantagem de herança é o compartilhamento do código entre objetos distintos.



Herança

- ▶ Classe base no Python: **object**
- ▶ Object – **super classe**
- ▶ MinhaClass1 e MinhaClass2 – **sub classe**

test.py

```
1 class MinhaClass1():  
2     | pass  
3  
4 class MinhaClass2(object):  
5     | pass  
6
```

Herança

▶ É comum que algumas classes em teoria possuam exatamente as mesmas características que outras.

▶ Isso não é nada eficiente no que diz respeito a quantidade de linhas de código uma vez que esses blocos, já que são idênticos.

▶ Exemplo 01

heranca01.py

```
1  #
2  class Jipe:
3      def __init__(self, modelo, ano):
4          self.modelo = modelo
5          self.ano = ano
6
7  class Caminhao:
8      def __init__(self, modelo, ano):
9          self.modelo = modelo
10         self.ano = ano
11
```

Herança

▶ Reformulando o código, criando uma classe de nome Veiculo que servirá de molde para as outras, dentro de si ela tem um método “construtor” assim como recebe como atributos de classe um modelo e um ano.

▶ Exemplo 02

```
#  
class Veiculo:  
    def __init__(self, modelo, ano):  
        self.modelo = modelo  
        self.ano = ano
```

```
1  #  
2  from veiculo import Veiculo as V  
3  
4  class Jipe(V):  
5  |  pass
```

```
#  
from veiculo import Veiculo as V  
  
class Caminhao(V):  
|  pass
```

Exercícios

Crie uma classe chamada Aluno a partir da classe Pessoa com os seguintes atributos:

- Nome
- Nota 1
- Nota 2
- Crie um inicializador (construtor) para a classe (`__init__`)

Crie as seguintes funções (métodos):

- Calcula média, retornando a média aritmética entre as notas
- Mostra dados, que somente imprime o valor de todos os atributos
- Resultado, que verifica se o aluno está aprovado ou reprovado (se a média for maior ou igual a 7.0, o aluno está aprovado)

Crie dois objetos (aluno1 e aluno2) e teste os métodos

Obs: o acesso aos atributos da classe Pessoa devem ser feito usando técnicas de encapsulamento.