

FUNDAMENTOS DE POO
MODELO DE OBJETOS
IMPLEMENTANDO UMA CLASSE EM PYTHON

INE5402 A
INE 5603 B

UFSC / CTC / INE
Professora: Luciana de Oliveira Rech

- A **Orientação a Objetos** é uma tecnologia que enxerga os sistemas como sendo uma coleção de objetos integrantes.
- Ela permite melhorar a reusabilidade (reaproveitamento de código) e extensibilidade dos softwares.

O Que é a Orientação a Objetos?

- É um paradigma para o desenvolvimento de software que baseia-se na **utilização de componentes** (objetos) que colaboram para construir sistemas mais complexos.
- Um paradigma é um conjunto de regras que estabelecem fronteiras e descrevem como resolver problemas dentro desta fronteira.
- Um paradigma nos ajuda a organizar e coordenar a maneira como iremos implementar um problema (sistema).

• Os 4 pilares da Programação Orientada a Objetos:

- 1- Abstração
- 2- Encapsulamento
- 3 – Herança
- 4 - Polimorfismo

Paradigma Orientação a Objetos

- Este paradigma procura abordar a resolução de um problema através de uma construção que apresente, da melhor forma possível, como as coisas acontecem no mundo real.
- O mundo real é constituído por entidades as quais interagem entre si.
- Uma **entidade** pode ser definida como algo que possua sua própria existência, **características** e que apresente alguma **função** dentro do mundo real.
- Uma entidade é também denominada de **Objeto**.

• Como reconhecer um objeto?

- Precisamos entender o mundo a nossa volta...
- O ser humano utiliza seus sentidos (audição, visão, olfato, tato) para interagir com o mundo...
- Assim, estamos a todo o tempo reconhecendo padrões.
 - Ex.: cadeiras.



Como fazer o mesmo utilizando máquinas/computadores?

- Ensinar as **características** deste objeto para a máquina/computador...
- Ensinar também quais são as **ações** possíveis para este objeto...
- Definir, por fim, os possíveis **estados** deste objeto.



Cadeira:

Construção: madeira.
 Cor: marrom pardo.
 Dimensões do acento: 30 x 40 x 2 cm.
 Altura dos "pés": 45 cm.
 Altura do Encosto: 50 cm.

Atributos

- Os objetos do mundo real possuem propriedades que possuem valores.
 - As **propriedades** recebem o nome de atributos em OO.
 - Os **valores** definem o estado do objeto.
- O **estado** de um objeto é o conjunto de valores de seus atributos em um determinado instante.



Métodos

- São procedimentos ou funções que realizam as ações próprias do objeto.
- Os métodos são responsáveis por acessar ou alterar os valores dos atributos de um objeto.
- Um objeto exibe algum comportamento (executa uma ação) quando recebe um estímulo de outro objeto (através de mensagens).

Definindo um objeto.

Exemplo2 :

O que é um **objeto**?

- É algo que possui características, comportamento e estado em um determinado momento.
- **Características**: o que é o objeto?
 - Cor, peso, tamanho, identidade...
- **Comportamento**: o que o objeto faz?
 - Correr, andar, ligar...
- **Estado**: como está o objeto?
 - Lâmpada: ora ligada, ora desligada.



Definindo um objeto

Exemplo 3:

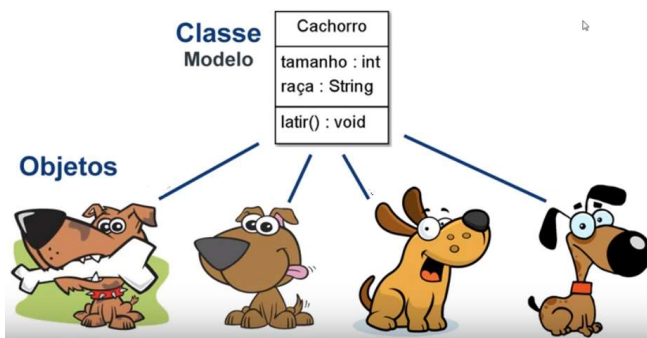
- O que é? [**Características**]
- Smartphone...
 - Cor? Dimensões? Capacidade?
- O que faz? [**Comportamento**]
- Ligar, Navegar, Despertar, Fotografar...
- Como está? [**Estado**]
- Tela?
 - Wi-Fi?
 - Bateria?
 - Despertador?



Os atributos de um objeto somente mudam de valor através de estímulos externos ou internos. A única forma de modificar os atributos de um objeto é disparando eventos que provocam a transição desses estados no objeto.

Classe

- Representa um conjunto de objetos que possuem características e comportamentos comuns.
- Um objeto é uma instância de uma classe.



A partir deste Modelo(Classe) é possível criar diversos cachorros com as mesmas propriedades, porém com valores diferentes.

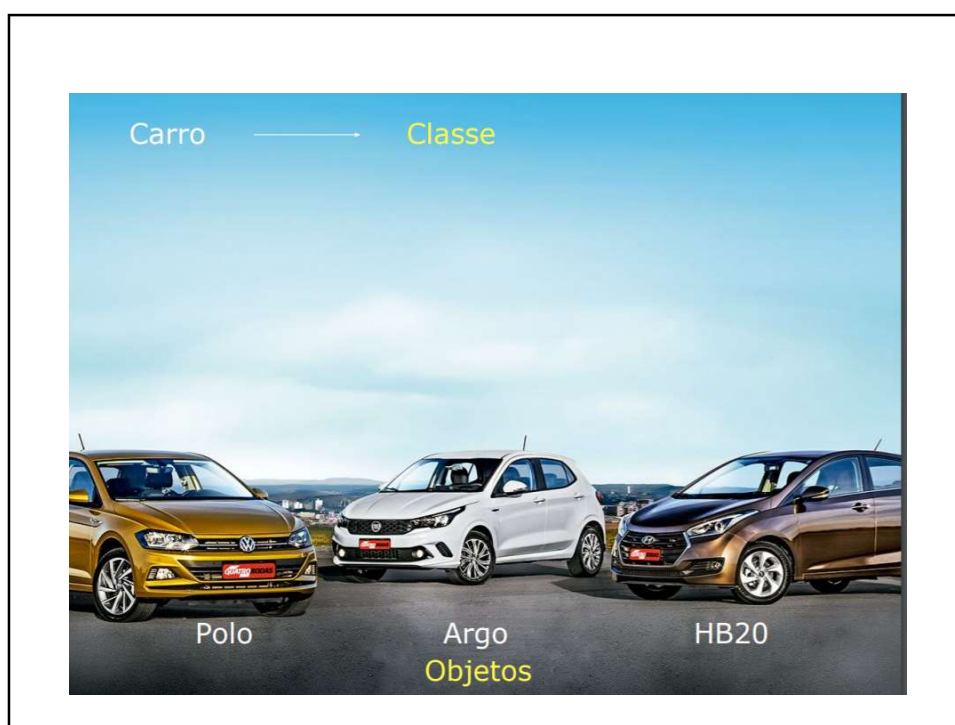
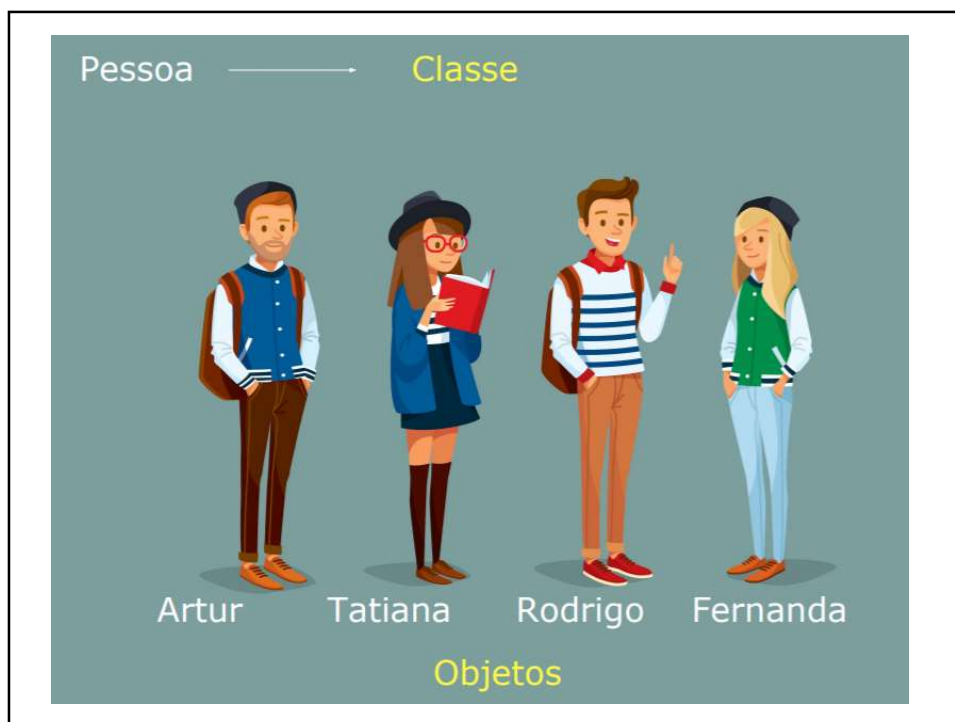
- Faz parte da natureza humana classificar objetos parecidos, ou seja, com as mesmas características, comportamentos e possíveis estados.
- Tais objetos são classificados através de um nome genérico que os representa.
- Nosso cérebro tem uma "forma" para cada objeto do mundo real, sendo capaz de classificá-los de acordo com o padrão já conhecido
- Por exemplo, quando conseguimos diferenciar (apenas olhando) um gato de um cachorro.



Classe x Objeto

- Classe:
 - É a "forma" (ou a ideia) do que seria o objeto;
 - Um formalismo que identifica as **características** e **comportamentos** dos objetos.
 - De forma geral, a classe **abstrai o conceito** do objeto.
 - Ex.: carro.
- Objeto:
 - É o item "real", que existe, que segue a "forma" definida para aquele tipo de objeto.





Classe

- É a forma (ou a ideia) do que seria o objeto;
- É o projeto do seu objeto, uma classe representa o tipo de objeto que você constrói a partir dela.
- Pode ser definida como um formalismo que identifica as **características** e **comportamentos** dos objetos.
- Objetos com as mesmas características e mesmos comportamentos podem ser considerados de uma mesma classe.
- A partir desta classificação, pode-se identificar os objetos com um nome próprio.
 - Ex: Maria (classe Pessoa)
- Convenções:
 - Nome de classe inicia com letra maiúscula.
 - Ex: **P**essoa, **C**achorro

Implementando uma classe em Python

- Exemplo:

Criar um programa que cadastre informações de uma pessoa.

- Informações: nome, idade e peso.

Seu sistema deve ser capaz de cadastrar, atualizar e imprimir as informações da pessoa cadastrada.

```
#ver os códigos fonte (disponível no moodle)
CriandoClasseExemplo00_CadastroPessoa.py
CriandoClasseExemplo01_CadastroPessoa(get,set).py
```

Exercícios:

1) Utilizando os conceitos de POO vistos em sala, implemente um programa que calcule e imprima o cubo de um número lido do teclado.

2) Implementar um programa que calcule o aumento de salário dos funcionários de uma empresa.

O aumento está condicionado ao atual salário de cada funcionário.

Regras:

- para abaixo ou igual a R\$2.000,00 : aumento de 15%;
 - para $2.000,00 < \text{salarioAtual} \leq 3.000,00$: aumento de 10%;
 - para acima de 3.000,00: aumento igual a 5%.
- Calcular o aumento de 3 funcionários;
 - A cada cálculo efetuado imprimir as seguintes informações: nome do funcionário, salário atual e salário com reajuste.
 - Devem ser implementadas 2 arquivos (.py):
 - Um arquivo contendo uma classe Funcionário que conterá os métodos para calcular o aumento de salário e o construtor.
 - Um arquivo contendo o controle do Funcionário (main), onde serão feitas chamadas para criar instâncias da classe Funcionário e também acessar os métodos que pertencem a classe Funcionário.
 - Ver código fonte (disponível no moodle):
 - funcionario.py
 - funcionario_principal.py