

# Programação Orientada a Objetos

OBJETOS, ATRIBUTOS, MÉTODOS E  
ENCAPSULAMENTO

PROF. EDUARDO HABIB BECHELANE MAIA  
HABIB@CEFETMG.BR

# Objetos

---

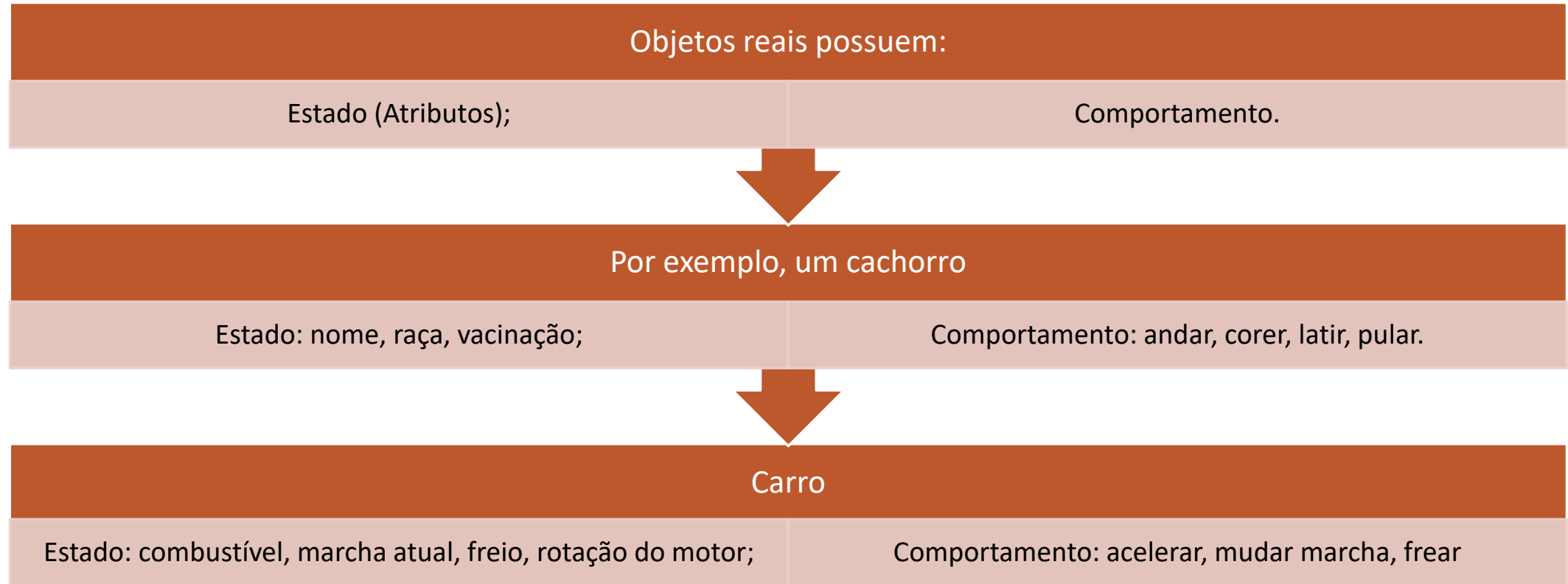
**Chave para entender a OO;**

**No mundo real podemos encontrar exemplos de objetos reais:**

- Porta;
- Smartphone;
- Mesa;
- Janela;
- TV;
- *Etc.*

# Objetos

---



# Objetos

---

Quais são as características de uma lâmpada?

Quais são as características de um projetor?

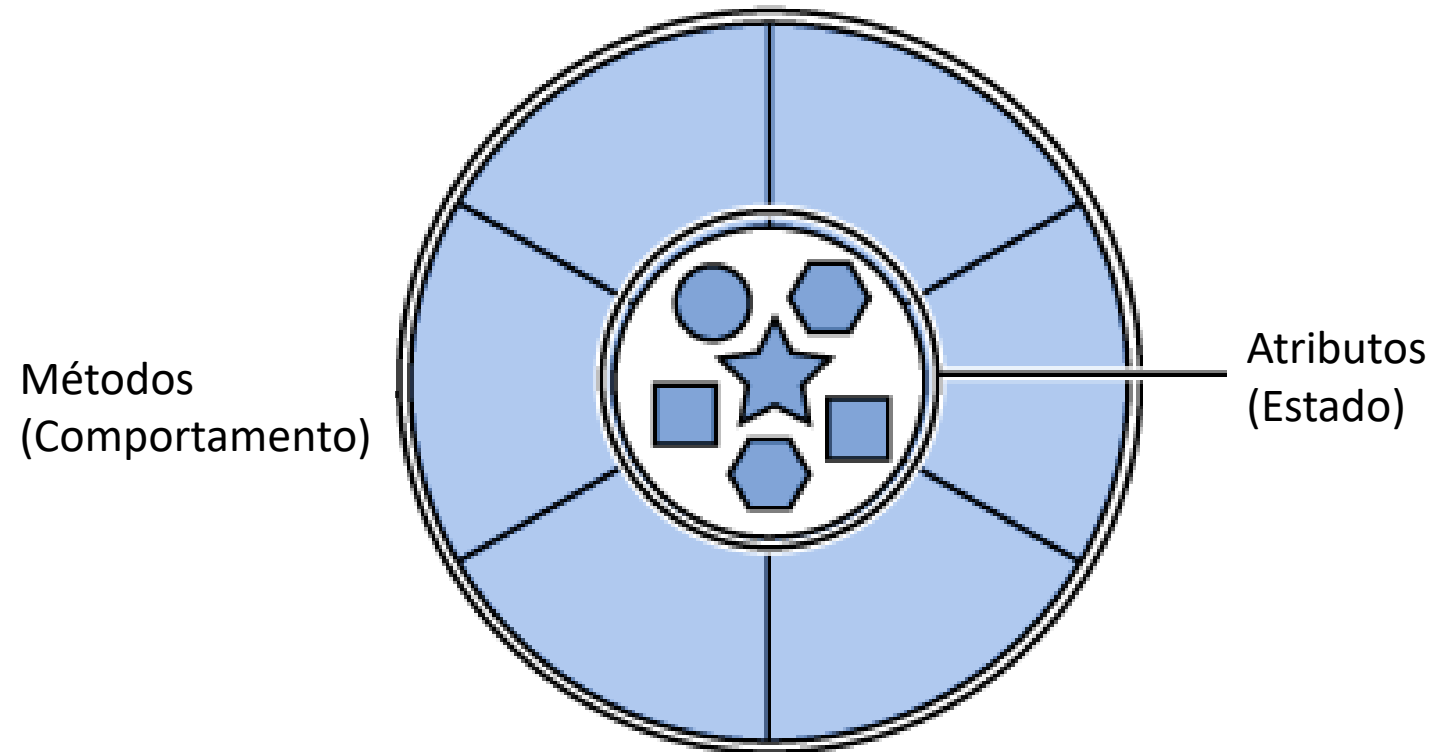
- E a lâmpada do projetor?

Objetos variam em complexidade

- Detalhes dependem do contexto;
- Esta análise de características é traduzível em orientação a objetos.

# Objetos

---



# Atributos e Métodos

---

Objetos armazenam seu estado em **atributos**

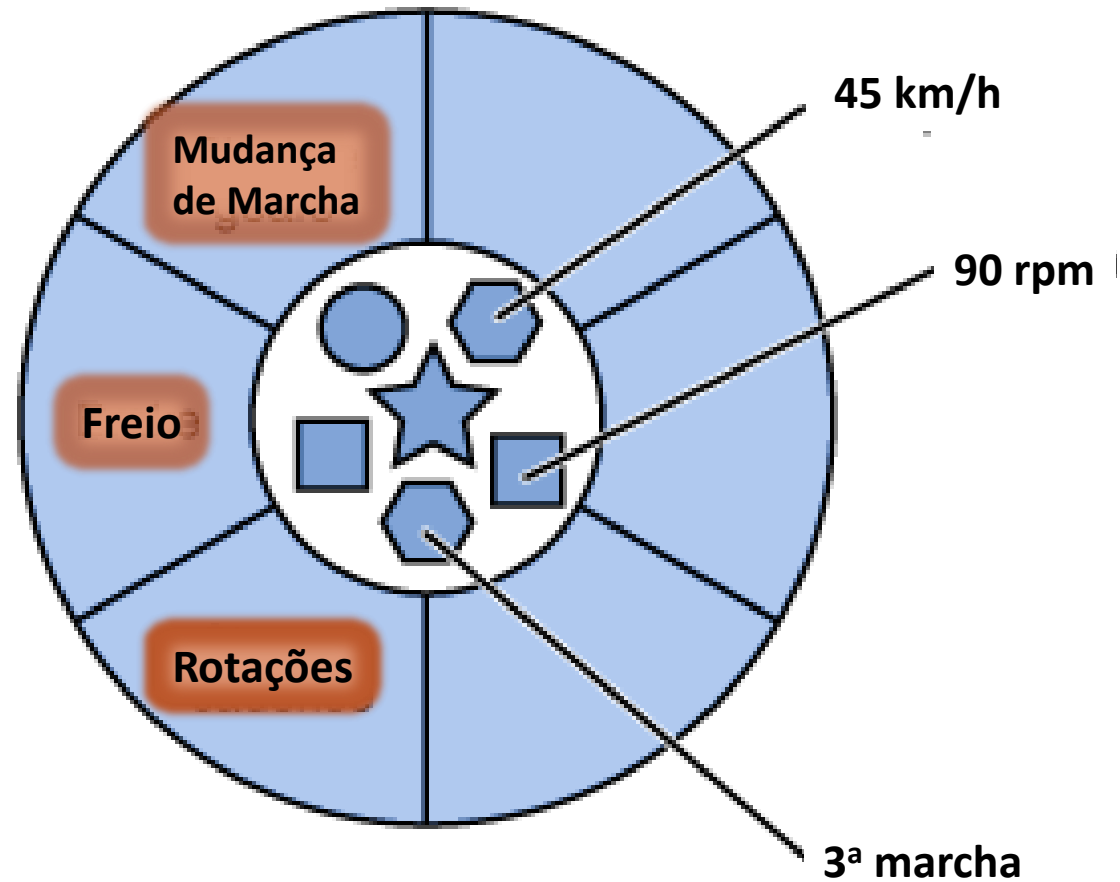
- Correspondentes às variáveis em programação estruturada.

Objetos expõem seu comportamento através de **métodos**

- Correspondentes às funções em programação estruturada.

# Objetos

---



# Objetos

---

Empacotar o código em objetos individuais fornece:

## Modularidade

- Objetos são independentes.

## Ocultação de informação

- Os detalhes da implementação de um objeto permanecem ocultos.

## Reuso

- Objetos podem ser reutilizados em diferentes programas.

## Plugabilidade

- Objetos podem ser substituídos em um programa, como peças.



# Encapsulamento

---

# Encapsulamento de Dados

---

Métodos definem o estado interno de um objeto

## Encapsulamento de dados

Impede o acesso  
direto aos atributos da  
classe

Cria métodos tanto  
para **obter** os valores  
dos atributos, quanto  
para **atribuir** valores a  
eles.

Getters e Setters (do  
inglês get e set)

# Encapsulamento de Dados

Esconder o estado interno e requerer que toda interação seja feita através de métodos é chamado de **encapsulamento de dados**.

Um princípio fundamental de OO.

Encapsulamento de dados, evita alterações acidentais nos atributos de um objeto

Não permite, por exemplo, passar a 6ª marcha se o objeto só possuir 5 marchas;

# Classes

---

# Classes

---

No mundo real, encontra-se vários objetos de um mesmo tipo:

Muitos carros, do mesmo fabricante e modelo;

Bicicleta pode ser produzida:

a partir do mesmo conjunto de projetos e;

conter as mesmas peças.

# Classes

Em orientação a objetos, dizemos que um objeto é uma **instância** de uma **classe de objetos**;

Uma classe é o projeto a partir do qual objetos individuais são criados

- Ela define os atributos e os métodos correspondentes aos seus objetos.

# Classes

---

Para definir uma classe é necessário *abstrair* um conjunto de objetos com características similares;

Outros possíveis membros de uma classe são:

**Construtores**

Define as operações a serem realizadas quando um objeto é criado.

**Destrutores**

Define as operações a serem realizadas quando um objeto é destruído.

# Exercício

---

1. Tendo como base os exemplos visto em sala, implemente uma classe pessoa que possui os atributos Peso, idade, Altura, sexo, nome, cpf e lista de irmãos. Em seguida, crie os métodos get e set e implemente os comportamentos abaixo:

- a) Calcular IMC ( $\text{Peso}/(\text{altura}*\text{altura})$ );
- b) Verificar se é maior de idade;
- c) Calcular se a pessoa é alta (Considere alto acima de 1,90m).
- d) Calcular o número de irmãos.
- e) Imprimir lista de irmãos
- f) Adicionar irmã(o)
- g) Remover irmã(o)