

# CONCEITOS PRELIMINARES

Prof<sup>a</sup> Renata Morais

## ENFOQUE DA OO

O enfoque da programação orientada a objetos procura enxergar o mundo como um conjunto de objetos que interagem entre si e apresentam características e comportamento próprios representados por seus atributos (dados) e suas operações (procedimentos executados por um objeto).

# PRINCIPAIS VANTAGENS DE POO

1. Reutilização de Código
2. Leitura e Manutenção de Código
3. Criação de bibliotecas

# REUTILIZAÇÃO DE CÓDIGO

1. Um dos principais requisitos no desenvolvimento de software atual.
2. Com a complexidade dos sistemas cada vez maior, o tempo de desenvolvimento iria aumentar exponencialmente caso não fosse possível a reutilização.
3. Isso é possível devido ao fato de que as linguagens de programação orientada a objetos trazem representações muito claras de cada um dos elementos, e esses elementos normalmente não são interdependentes. Essa independência entre as partes do software é o que permite que esse código seja reutilizado em outros sistemas no futuro.

# LEITURA E MANUTENÇÃO DE CÓDIGO

1. Como a representação do sistema se aproxima muito do que vemos na vida real, o entendimento do sistema como um todo e de cada parte individualmente fica muito mais simples.
2. Permite que a equipe de desenvolvimento não fique dependente de uma pessoa apenas, como acontecia com frequência em linguagens estruturadas como o C, por exemplo.

# CRIAÇÃO DE BIBLIOTECA

1. A criação de bibliotecas é outro ponto que é muito mais simples com a OO.
2. No caso das linguagens estruturadas, como o C, temos que as bibliotecas são coleções de procedimentos (ou funções) que podem ser reutilizadas.
3. No caso da POO, as bibliotecas trazem representações de classes, que são muito mais claras para permitirem a reutilização.

# NEM TUDO SÃO FLORES

1. A execução de uma aplicação orientada a objetos é mais lenta do que o que vemos na programação estruturada, por exemplo.
2. Isso acontece devido à complexidade do modelo, que traz representações na forma de classes. Essas representações irão fazer com que a execução do programa tenha muitos desvios, diferente da execução sequencial da programação estruturada.
3. Esse é o grande motivo por trás da preferência pela linguagem C em hardware limitado, como sistemas embarcados. Também é o motivo pelo qual a programação para sistemas móveis como o Google Android, embora em Java (linguagem orientada a objetos), seja feita o menos orientada a objetos possível.

# PILARES DA POO





# ABSTRAÇÃO

A abstração consiste em um dos pontos mais importantes, pois estamos lidando com uma representação de um objeto real, temos que imaginar o que esse objeto irá realizar dentro de nosso sistema.



# Abstração - pontos de observação

## Identidade

**A identidade deve ser única dentro do sistema para que não haja conflito.**

**1º Ponto**

## Características

No mundo real qualquer objeto possui elementos que o definem. Dentro da POO, essas características são nomeadas propriedades. Por exemplo, as propriedades de um objeto "Gato" poderiam ser "Tamanho", "Raça" e "Idade".

**2º Ponto**

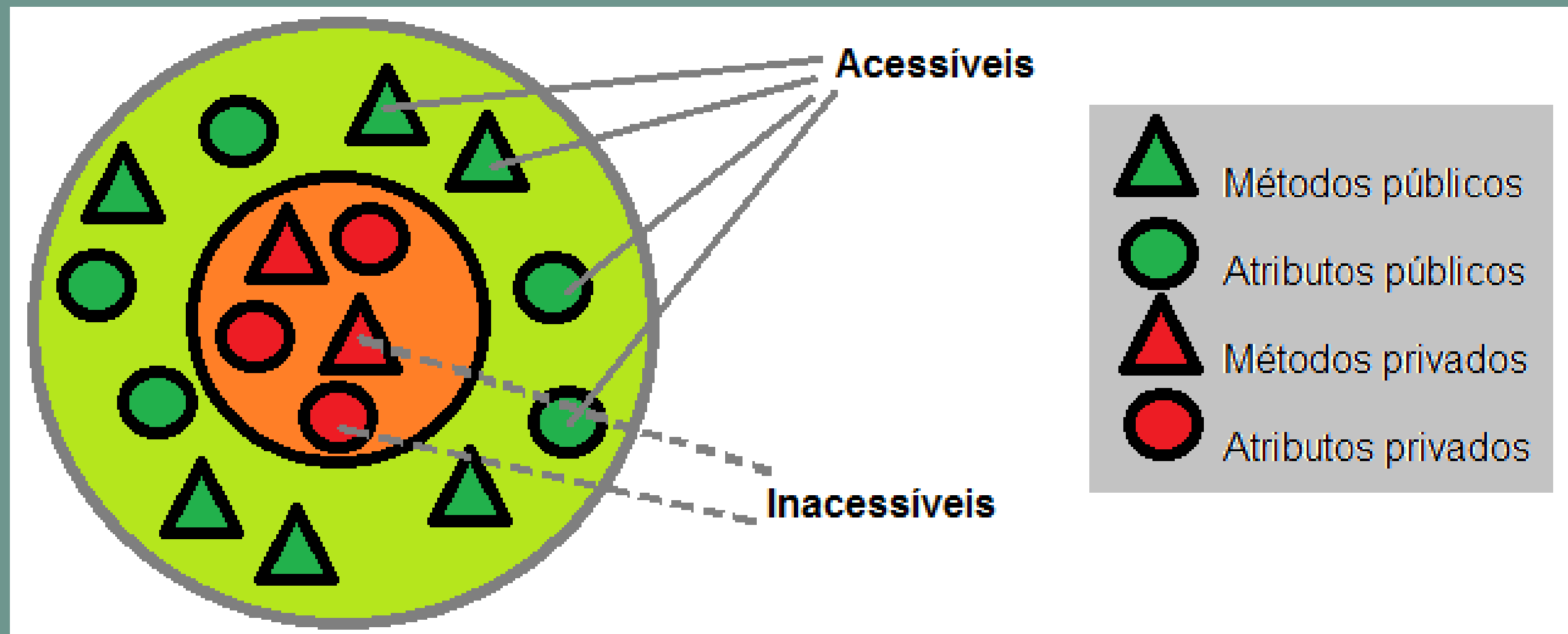
## Métodos

Na linguagem orientada objeto, as ações e eventos dos objetivos, são chamados de Métodos. Esses métodos podem ser extremamente variáveis, desde "Acender()" em um objeto lâmpada até "Latir()" em um objeto cachorro

**3º Ponto**

# ENCAPSULAMENTO

Se trata de um dos elementos que adicionam segurança à aplicação em uma programação orientada a objetos pelo fato de esconder as propriedades, criando uma espécie de caixa preta.

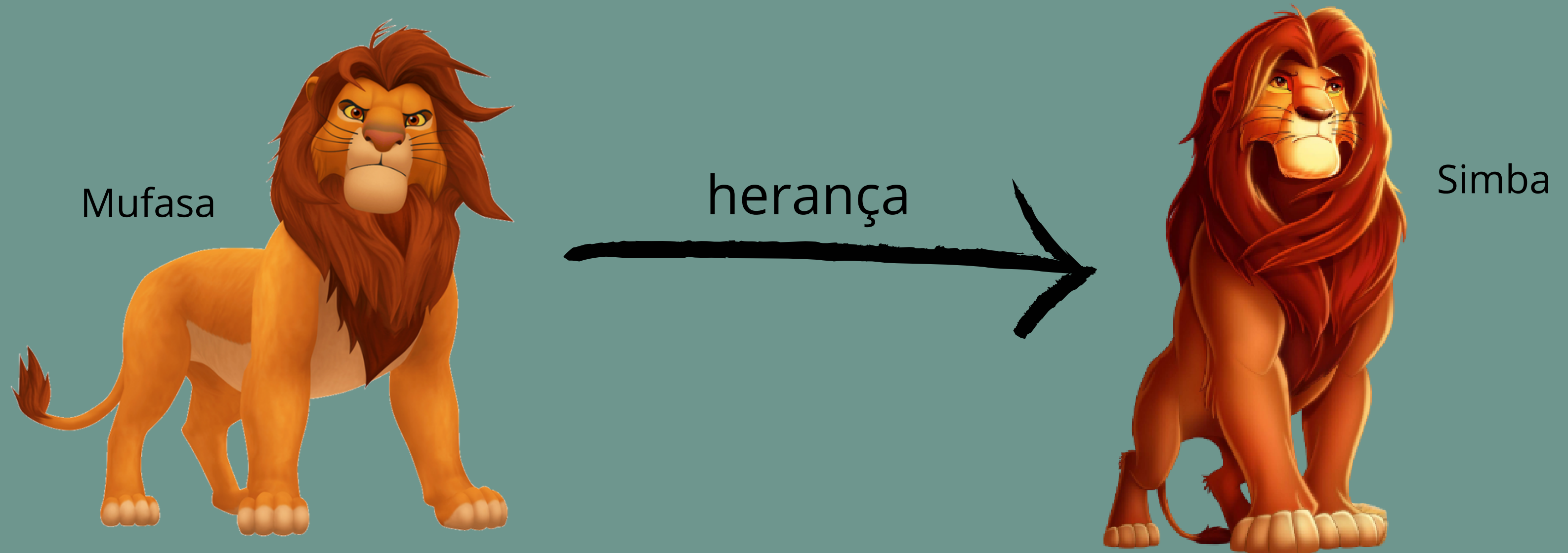


# Encapsulamento

- A maior parte das linguagens orientadas a objetos implementam o encapsulamento baseado em propriedades privadas, ligadas a métodos especiais chamados getters e setters, que irão retornar e setar o valor da propriedade, respectivamente.
- Essa atitude evita o acesso direto a propriedade do objeto, adicionando uma outra camada de segurança à aplicação.
- Fazendo um paralelo com o mundo real, o botão ligar da televisão por exemplo, não sabemos o que está acontecendo internamente. Podemos então dizer que os métodos que ligam a televisão estão encapsulados.

# HERANÇA

Herança é basicamente o reuso de códigos (uma das grandes vantagens da programação orientada a objetos). Essa característica otimiza a produção da aplicação em tempo e linhas de código.



# HERANÇA

- Numa família por exemplo, a criança está herdando características de seus pais. Os pais, por sua vez, herdam algo dos avós, o que faz com que a criança também o faça, e assim sucessivamente. Na orientação a objetos, a questão é exatamente assim.
- Por exemplo, na família, a criança herda diretamente do pai e indiretamente do avô e do bisavô.
- A questão da herança varia bastante de linguagem para linguagem.
- Em algumas delas, há a questão da herança múltipla. Isso, essencialmente, significa que o objeto pode herdar características de vários “ancestrais” ao mesmo tempo diretamente

# POLIMORFISMO

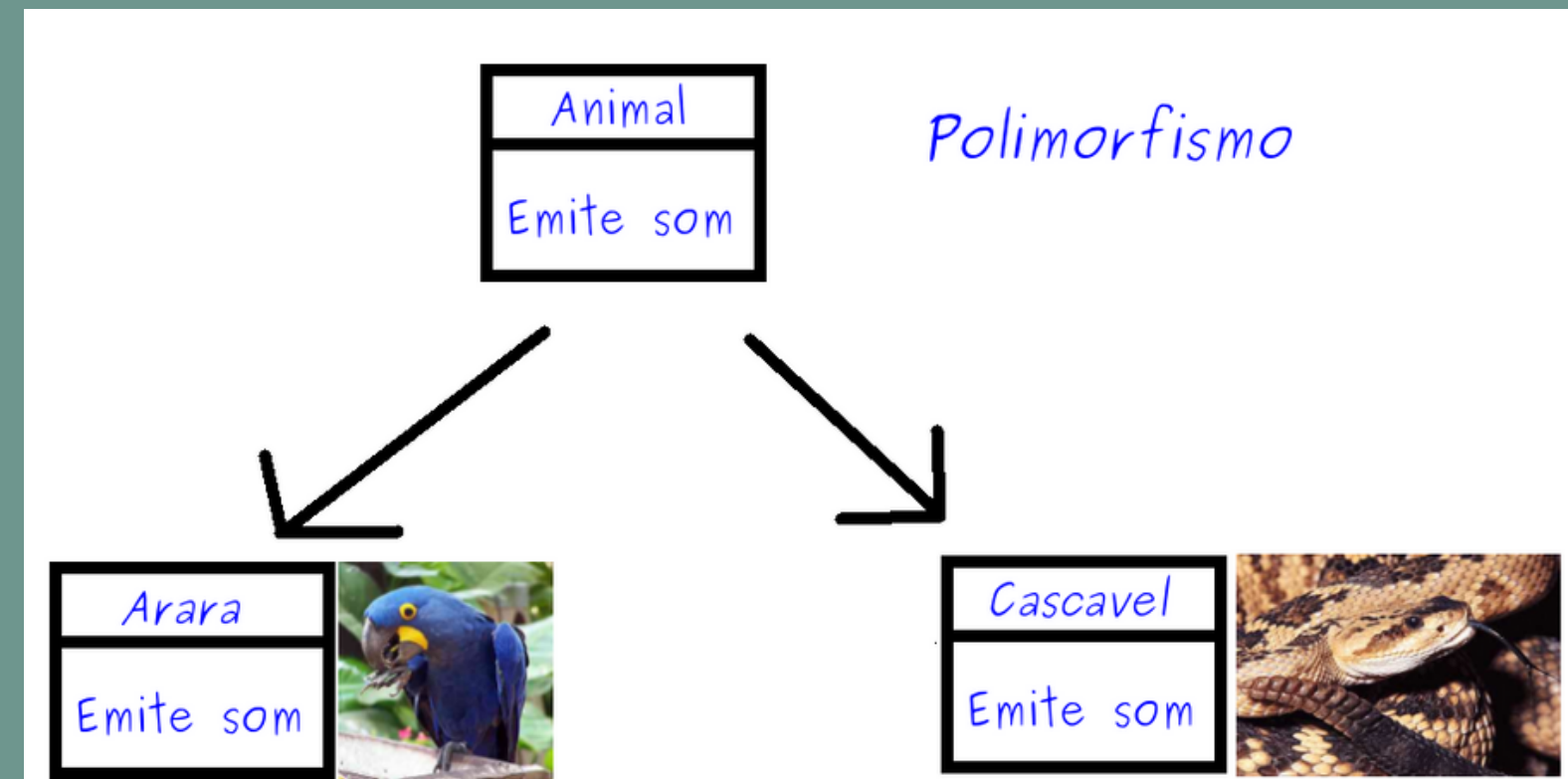
- Na natureza, vemos animais que são capazes de alterar sua forma conforme a necessidade, e é dessa ideia que vem o polimorfismo na orientação a objetos.
- Como sabemos, os objetos filhos herdam as características e ações de seus “ancestrais”.
- Em alguns casos, é necessário que as ações para um mesmo método seja diferente.
- Em outras palavras, o polimorfismo consiste na alteração do funcionamento interno de um método herdado de um objeto pai.

# POLIMORFISMO

## Exemplo 1

Tendo um eletrodoméstico como exemplo, esse objeto possui um método, ou ação, “Ligar()”. Temos dois objetos, “Televisão” e “Geladeira”, que não irão ser ligados da mesma forma. Assim, precisamos, para cada uma das classes filhas, reescrever o método “Ligar()”.

## Exemplo 2





# OBJETO

- Um objeto é uma entidade do mundo real que tem uma identidade.
- Objetos são instâncias de classes, que determinam qual informação um objeto contém e como ele pode manipulá-la.
- É uma entidade capaz de reter um estado (informação) e que oferece uma série de informações (comportamento) ou para examinar ou para afetar este estado.
- A estrutura de um objeto é representada em termos de atributos (propriedades dos objetos).

# OBJETO

- Cada objeto, tem sua identidade, significa que dois objetos são distintos mesmo que eles apresentem exatamente as mesmas propriedades.
- Objetos com a mesma estrutura e o mesmo comportamento são agrupados em classes.
- É através deles que praticamente todo o processamento ocorre em sistemas implementados com linguagens de programação orientada a objetos.

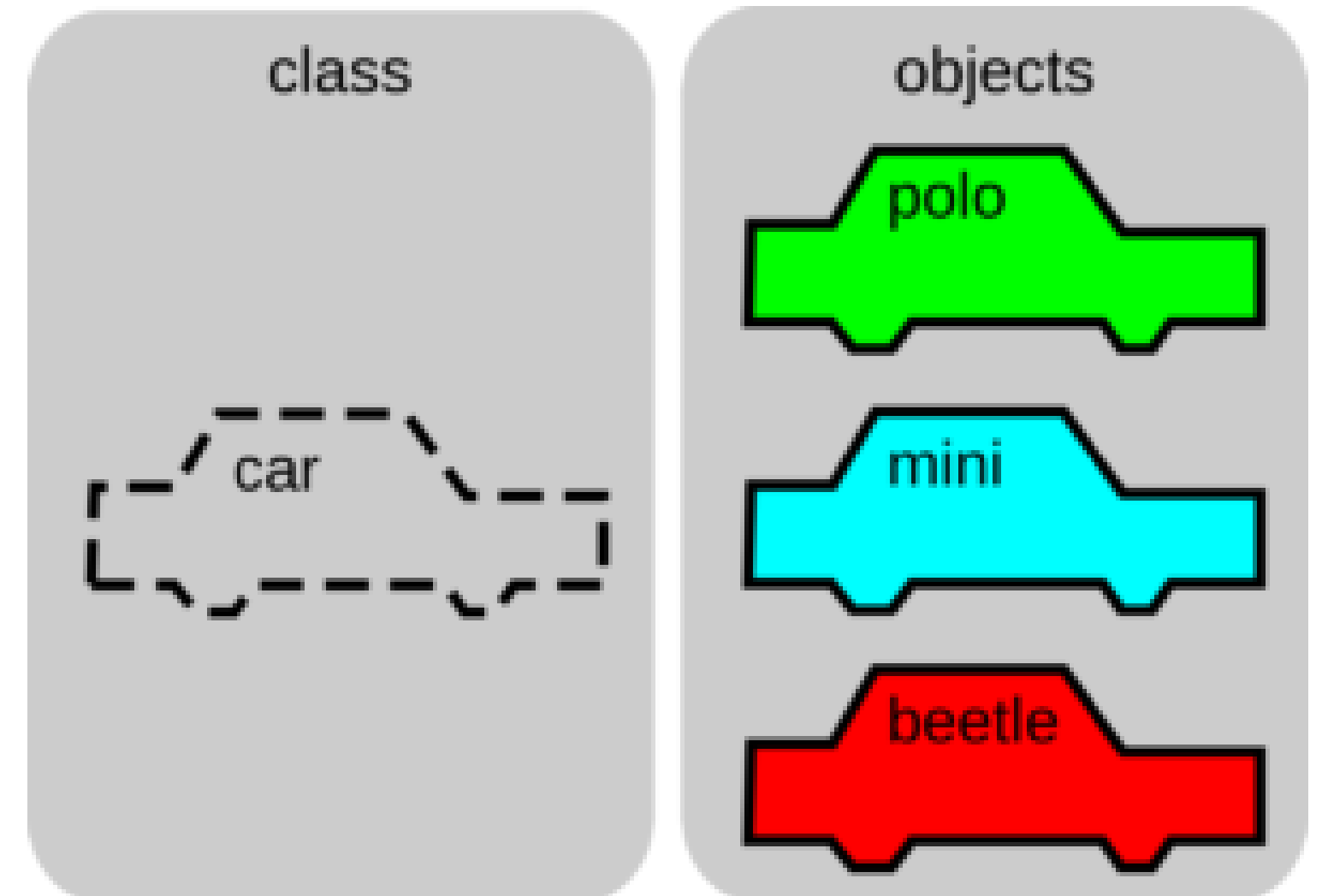
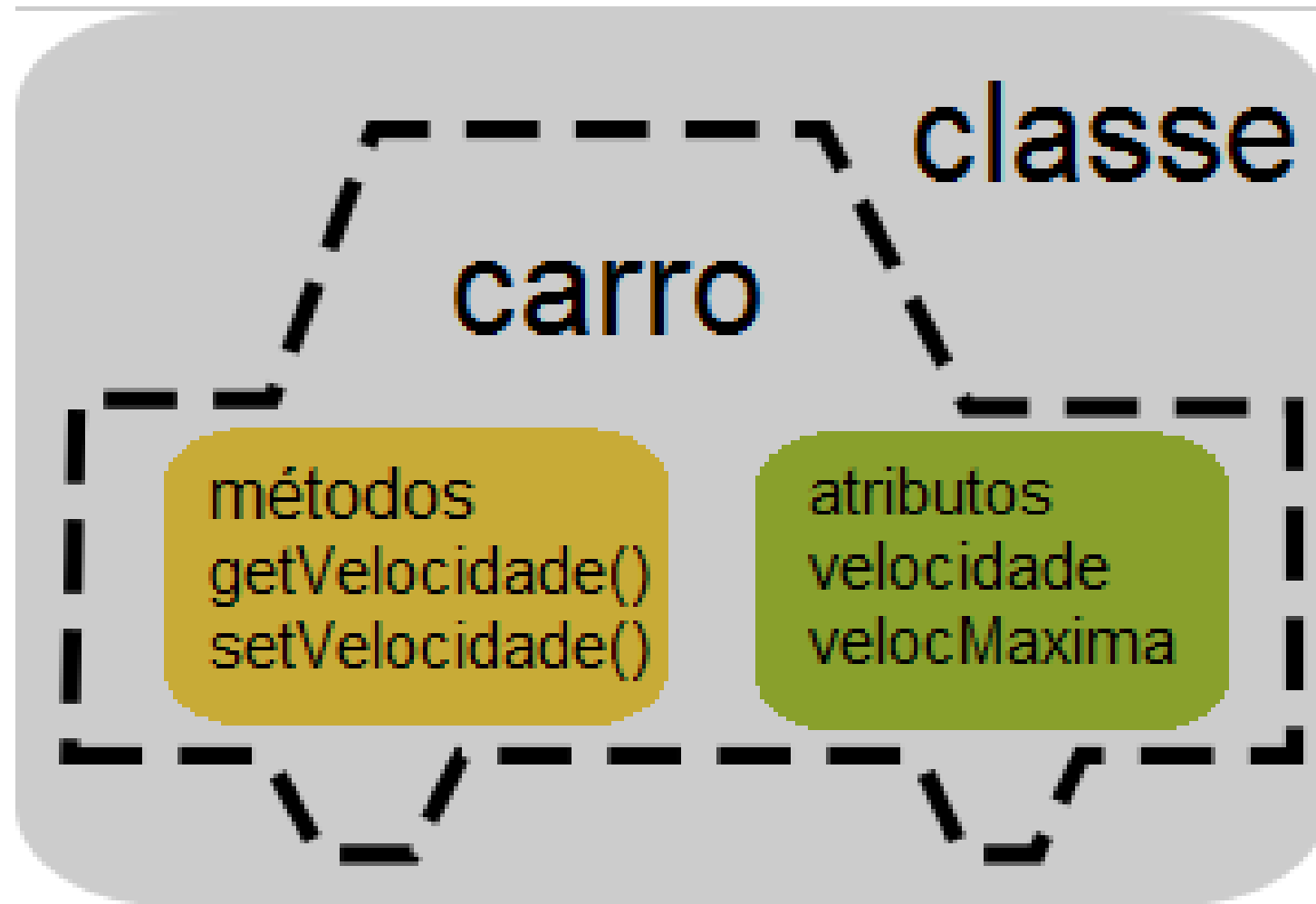
# CLASSE

- Uma Classe é uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica.
- Cada classe descreve um conjunto (possivelmente infinito) de objetos individuais.
- Cada objeto é dito ser uma instância de uma classe.
- Implicitamente, cada objeto contém uma referência para sua própria classe -- em outras palavras, ele sabe o que ele é.

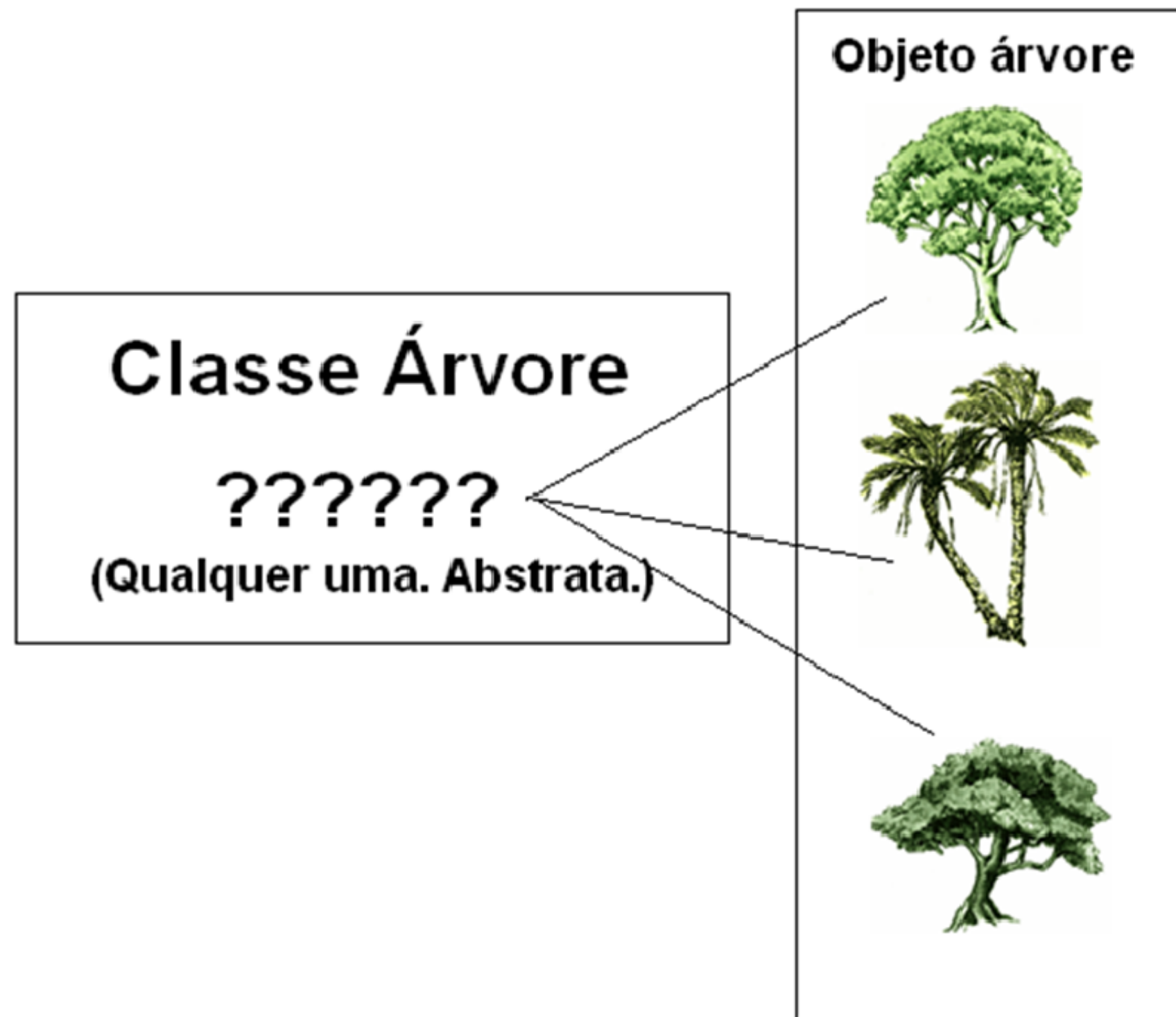
# CLASSE

- Assim, cada instância de uma classe tem seus próprios valores para cada atributo, mas dividem os nomes dos atributos e métodos com as outras instâncias da classe.
- Características:
  - **Nome** - a diferencia das outras classes.
  - **Atributos** - propriedades que descrevem um intervalo de Valores que as instâncias da classe podem apresentar.
  - **Métodos** - implementam serviços que podem Ser solicitados por algum objeto da classe para modificar o Comportamento.

# Classe e Objeto



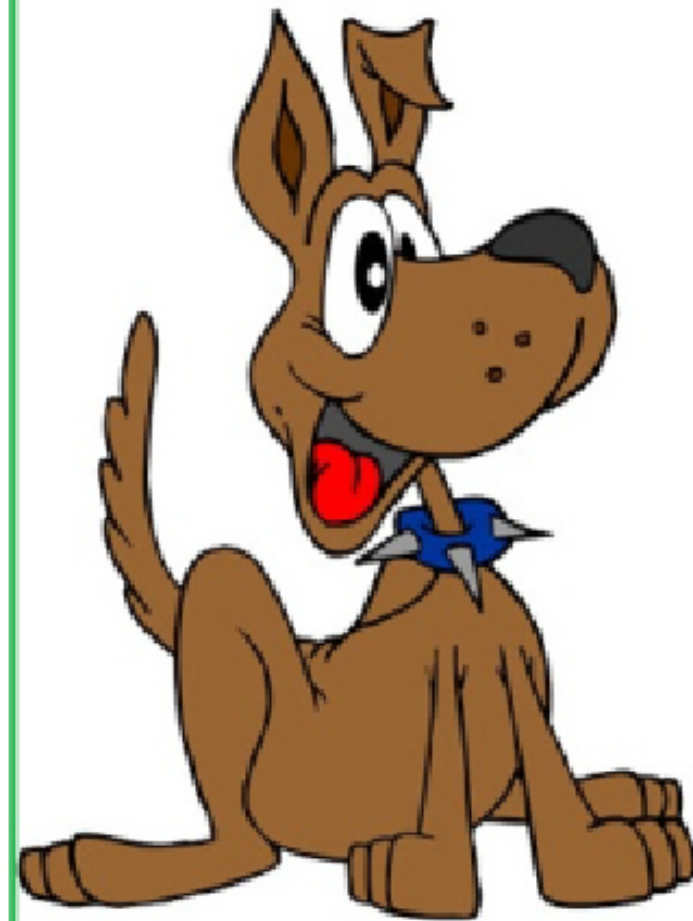
# Classe e Objeto



# Classe e Objeto

## Classe Cães

### Objetos cachorros



BOB

Anda  
Fala  
Come  
Dorme  
PegaOsso



SCOOBY

Anda  
Fala  
Come  
Dorme  
PegaOsso



REX

Anda  
Fala  
Come  
Dorme  
PegaOsso

# Um objeto no mundo do software

- Objetos duas características principais:
  - um **estado** - conjunto de propriedades do objeto (**atributos**)
  - um **comportamento** - ações possíveis do objeto (**métodos**)



**Cadastro de Clientes**

Nome:

CPF:  RG:

Endereço:

Bairro:  CEP:

Cidade:  Estado:

Login:

Senha:

Telefone:  Celular:

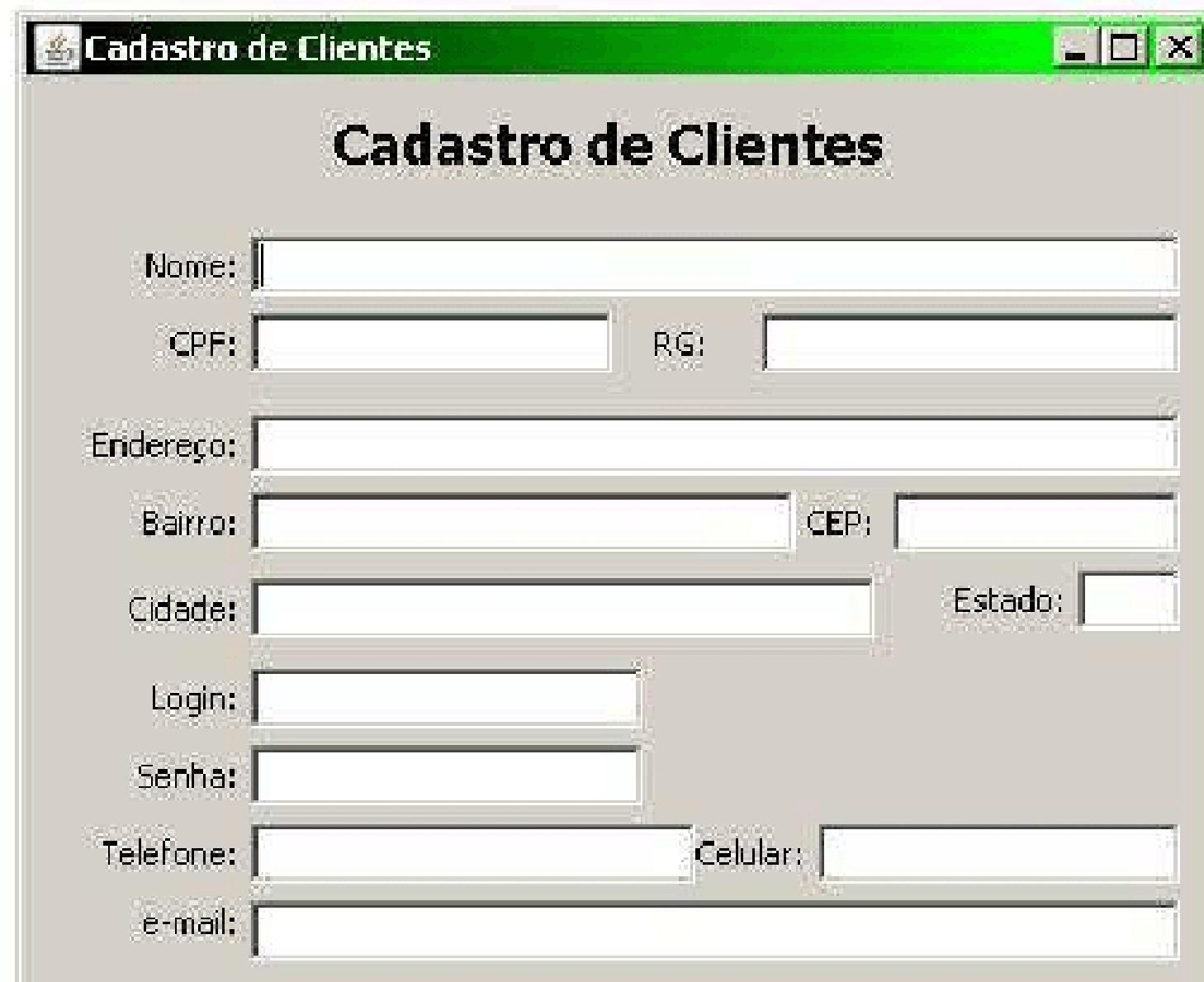
e-mail:

Inserir Alterar Excluir Pesquisar Fechar



# Um objeto no mundo do software

- **estado** - conjunto de propriedades do objeto (**atributos**)



- um **comportamento** - ações possíveis do objeto (**métodos**)



# Representação gráfica de Classe na POO pela UML

## Classe



## Objetos



A  
T  
R  
I  
B  
U  
T  
O  
S  
  
M  
É  
T  
O  
D  
O  
S

# REGRAS PARA NOMEAÇÃO DE CLASSES

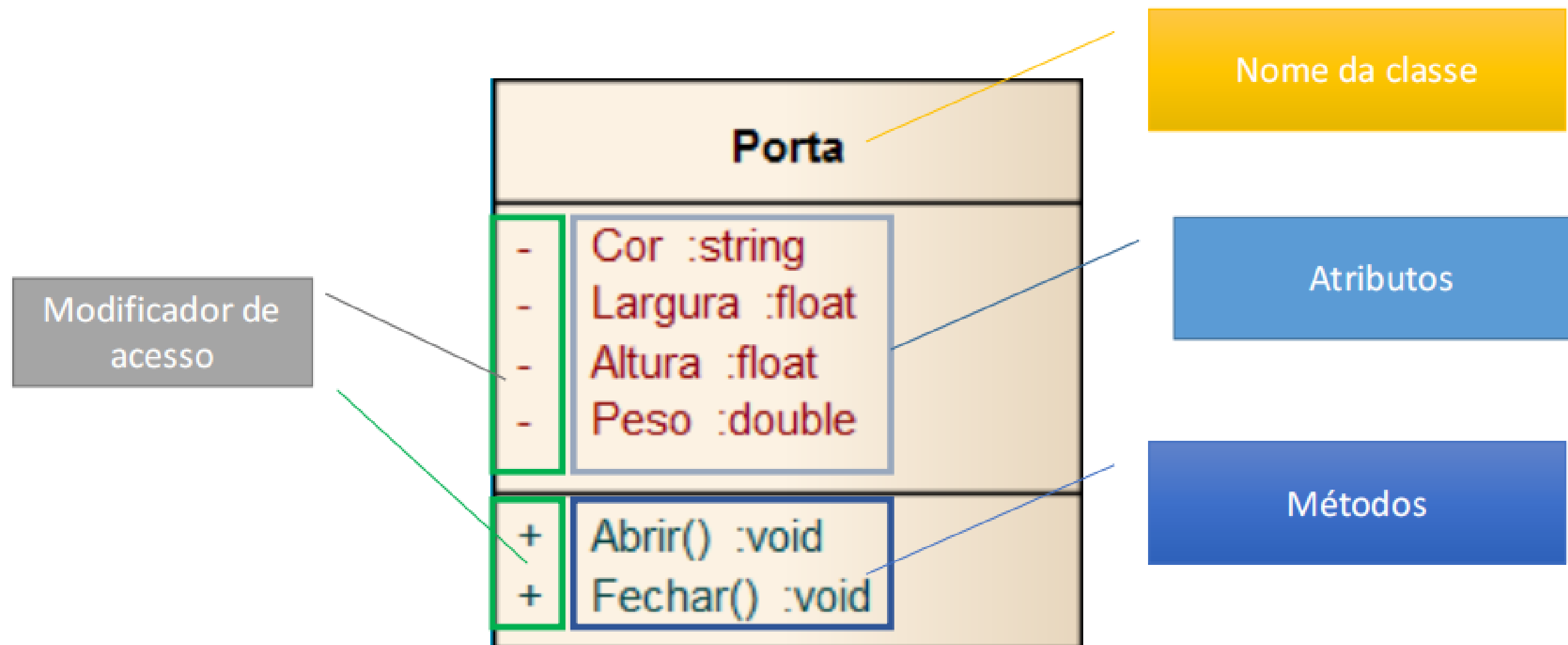
- Ao nomear uma classe algumas convenções devem ser seguidas:
  - Manter o nome simples e descritivo;
  - Usar palavras inteiras, isto é, sem siglas e abreviações;
  - A primeira letra de cada palavra deve estar em caixa alta.

# SINTAXE BÁSICA DE CRIAÇÃO DE UMA CLASSE

```
<modificador de acesso> class NomeDaClasse{  
    //declaração dos atributos  
    //declaração dos métodos  
}
```

Palavra reservada que indica a criação da classe

# ENTENDENDO A REPRESENTAÇÃO DA CLASSE PELA NOTAÇÃO DA UML



## Atividade Prática

- Disponível no Google sala de aula
- Na próxima aula serão escolhidos alguns alunos para falar/apresentar resolução das questões.