

# Polimorfismo

**Prof. Fábio Procópio**

**Prof. João Nascimento**



2

## Relembrando...

- Na [aula passada](#), finalizamos os estudos sobre o conteúdo de Herança;
- Nesta aula, vamos começar a falar sobre Polimorfismo, que é uma estratégia de implementar um método com o mesmo nome de formas diferentes. Vamos nessa?!





# Introdução – 1 de 3

- **Polimorfismo** é uma palavra de origem grega:
  - As palavras **Poli** e **Morphos**, quando combinadas, significam múltiplas formas
- No contexto de Orientação a Objetos, o polimorfismo viabiliza a implementação de um mesmo método de formas diferentes;
- Utilizando polimorfismo, as subclasses são capazes de invocar métodos com a mesma assinatura das superclasses, porém de maneiras diferentes;
- Uma superclasse pode definir “**métodos virtuais**” e as subclasses podem “**substituí-los**”, isto é, elas fornecem suas próprias implementações.

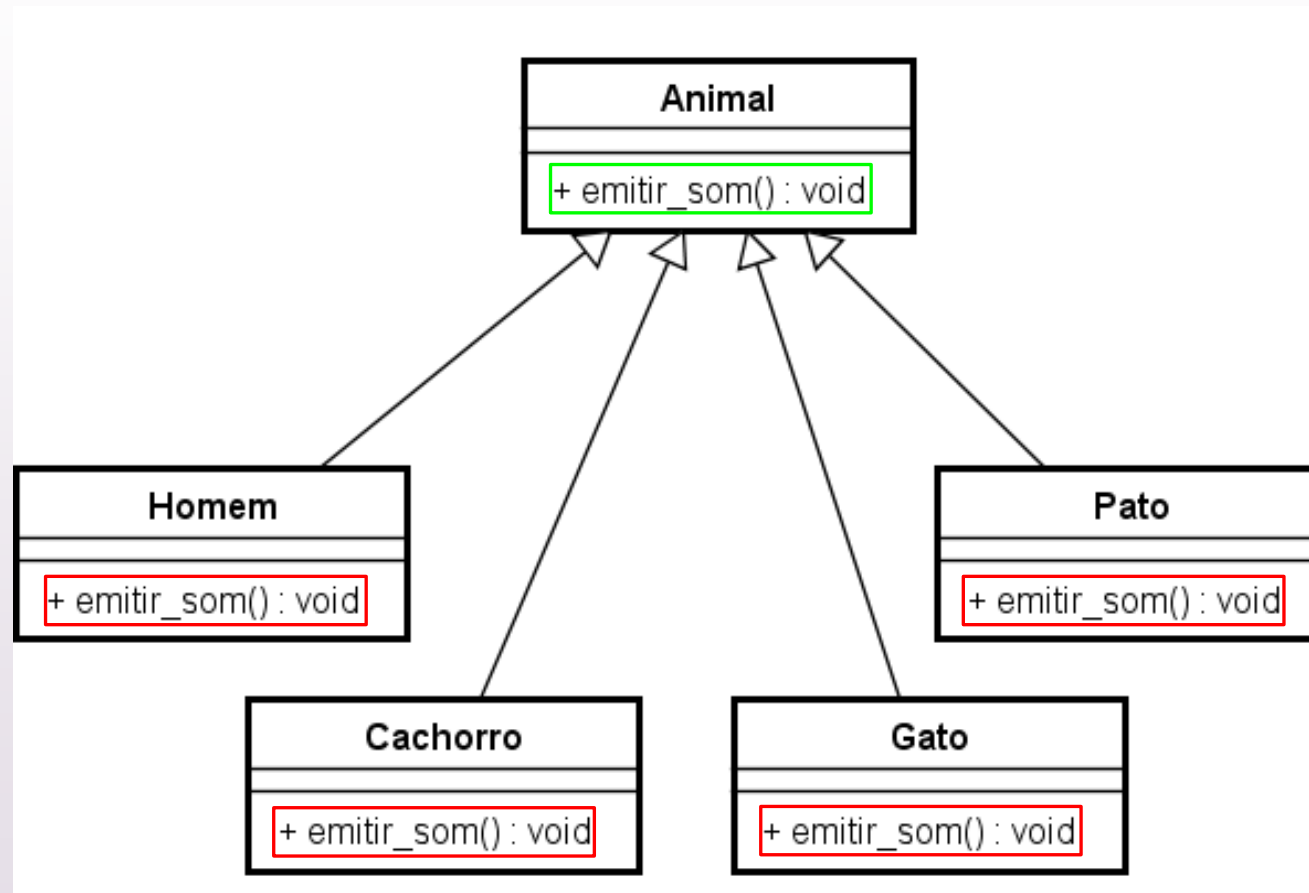


## Introdução – 2 de 3





# Introdução – 3 de 3

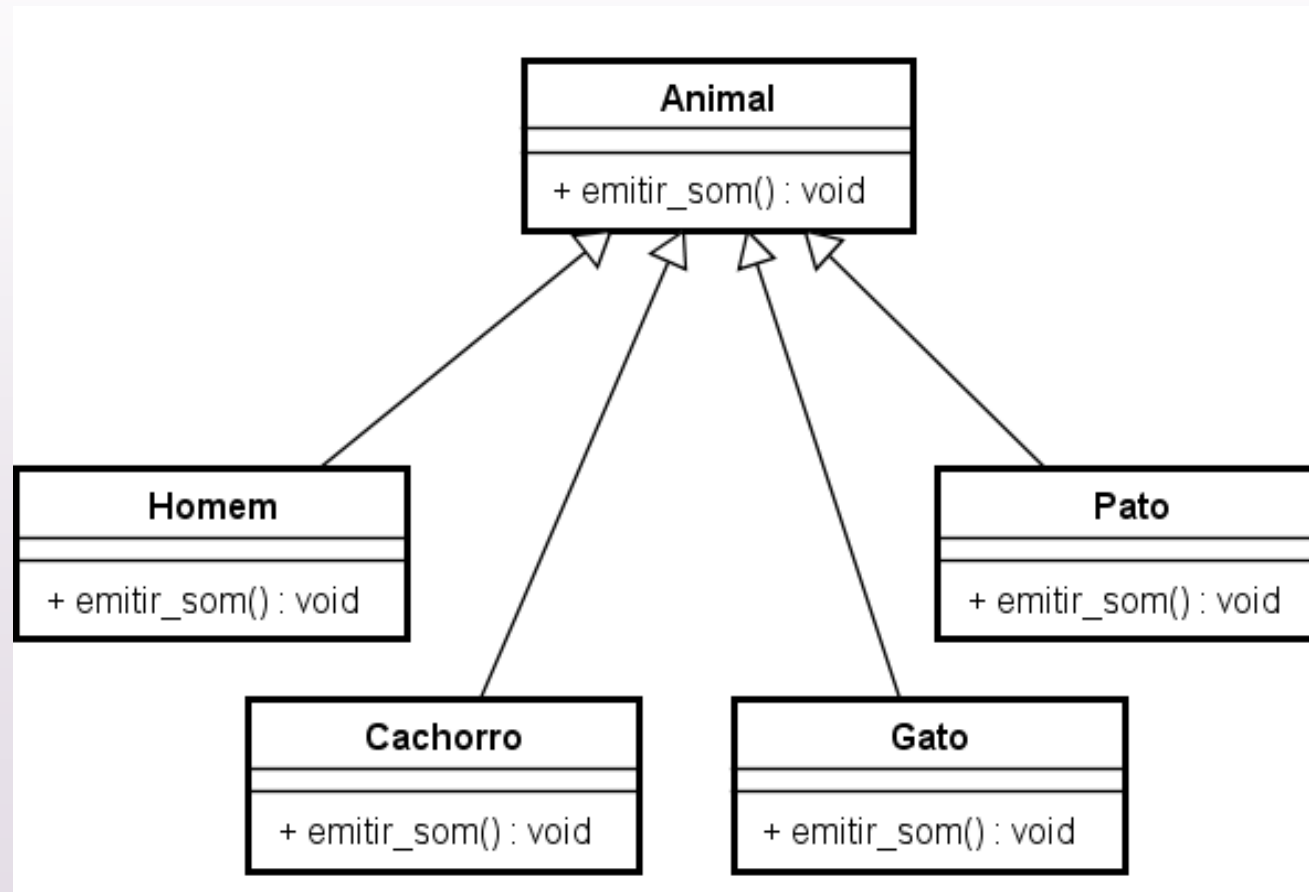




6

# Exercício Resolvido 01 – 1 de 3

➤ Vamos implementar o modelo abaixo.





7

## Exercício Resolvido 01 – 2 de 3

```
class Animal:  
    def __init__(self):  
        pass  
  
    def emitir_som(self):  
        print("Os animais produzem som.")
```

```
from Animal import *  
  
class Cachorro(Animal):  
    def __init__(self):  
        Animal.__init__(self)  
  
    def emitir_som(self):  
        print("Auauau")
```

```
from Animal import *  
  
class Homem(Animal):  
    def __init__(self):  
        Animal.__init__(self)  
  
    def emitir_som(self):  
        print("Olá")
```

```
from Animal import *  
  
class Gato(Animal):  
    def __init__(self):  
        Animal.__init__(self)  
  
    def emitir_som(self):  
        print("Miau")
```



## Exercício Resolvido 01 – 3 de 3

```
from Animal import *

class Pato(Animal):
    def __init__(self):
        Animal.__init__(self)

    def emitir_som(self):
        print("Quack Quack!")
```

```
from Homem import *
from Cachorro import *
from Gato import *

c = Cachorro()
c.emitir_som()

g = Gato()
g.emitir_som()

h = Homem()
h.emitir_som()

p = Pato()
p.emitir_som()

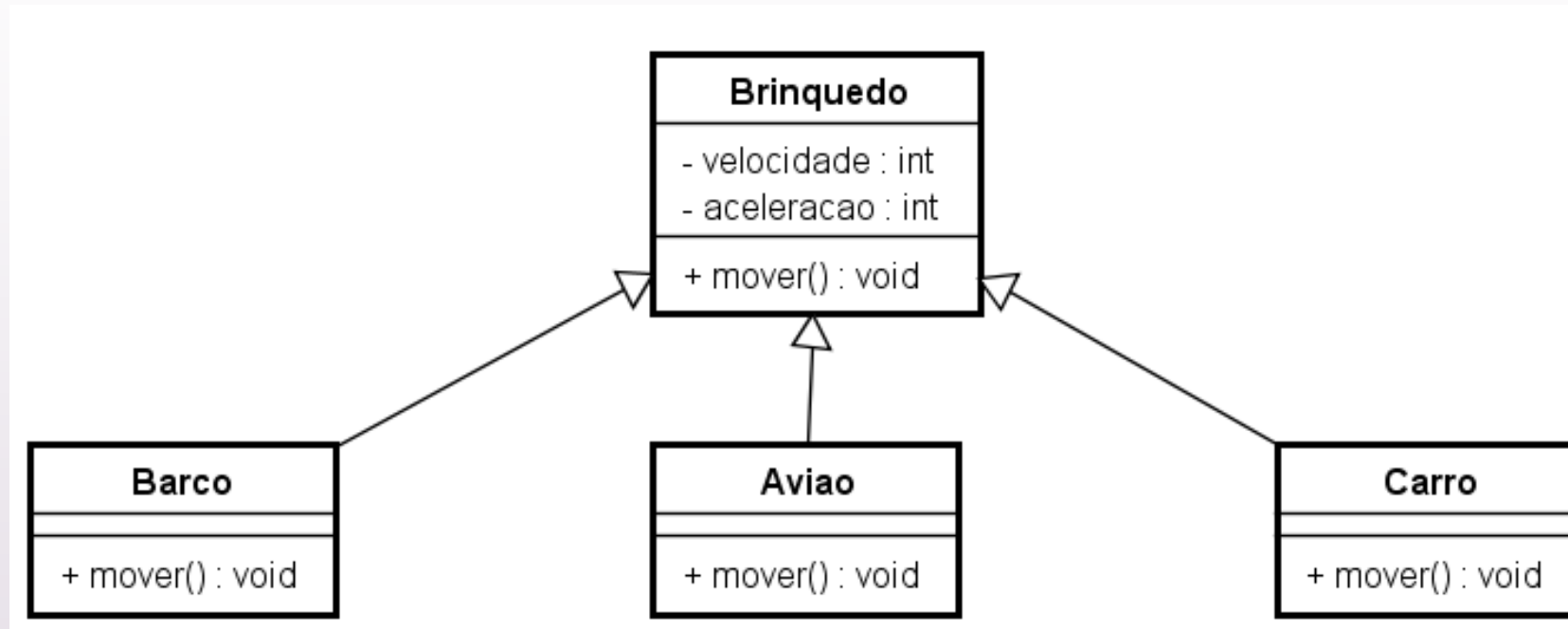
animal = Animal()
animal.emitir_som()
```





# Exercício de Fixação 01

- Implemente e teste o modelo a seguir:



O método **mover()** deve fazer o seguinte

- em **Brinquedo**: nada
- em **Barco**, imprimir "Navegando"
- em **Aviao**, imprimir "Voando"
- em **Carro**, imprimir "Correndo"



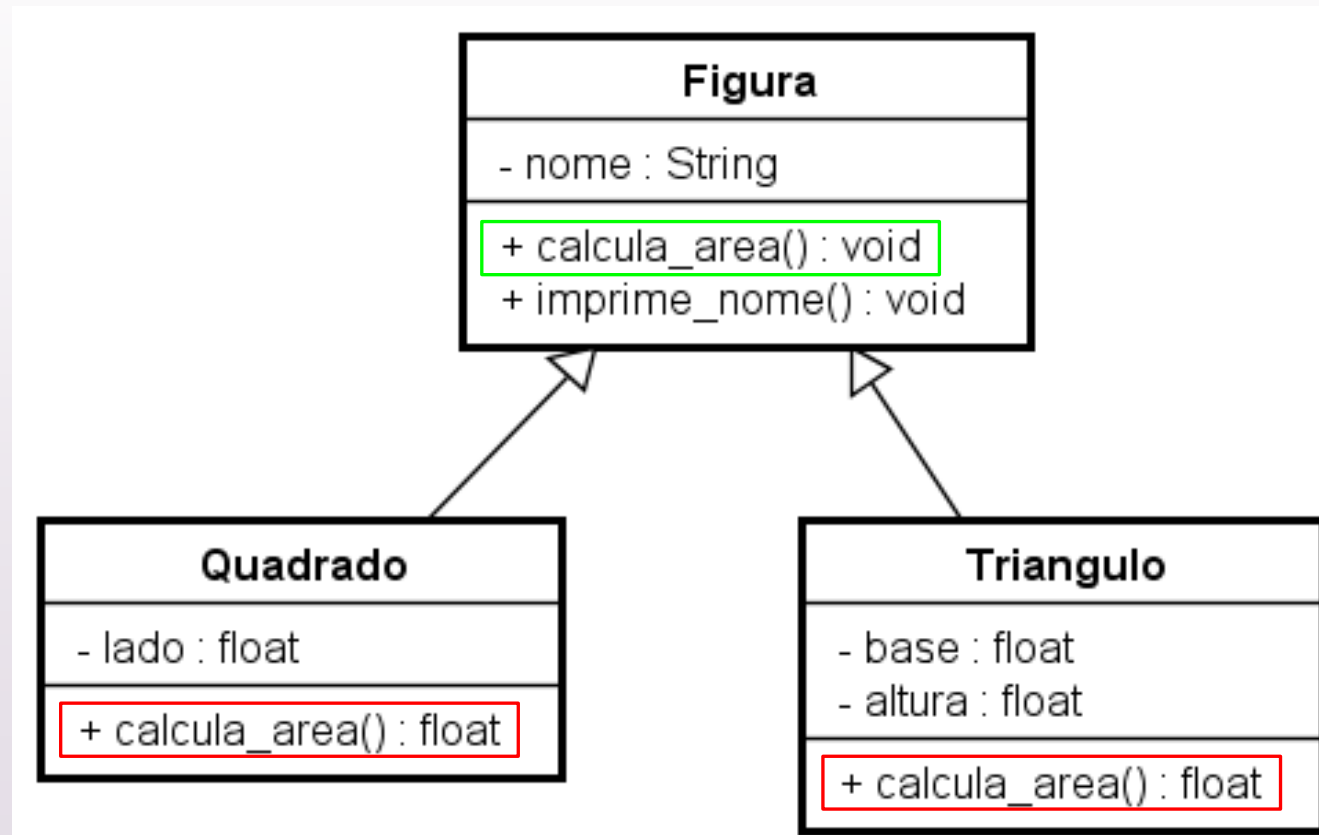
## Exercício de Fixação 2

- Implemente uma subclasse de Homem, chamada Veterinario que
  - não possui atributos
  - possui um método chamado **examinar()**, cujo argumento de entrada é um objeto do tipo Animal
- Quando o animal for examinado, ele deve emitir o seu som;
- Em TestaClasses, invoque o método `examinar()` passando objetos do tipo Cachorro, Gato e de Pato.



## Exercício de Fixação 03

- Implemente e teste o modelo a seguir:





# Canal no Youtube

Em nosso canal no Youtube, você pode acessar uma playlist que preparamos com várias vídeo-aulas falando sobre o assunto. Veja:

➡ [Polimorfismo](#)