



# Orientação a objetos

---

Capacitação em lógica de Programação  
Momento Pedagógico 2023/1

SENAI

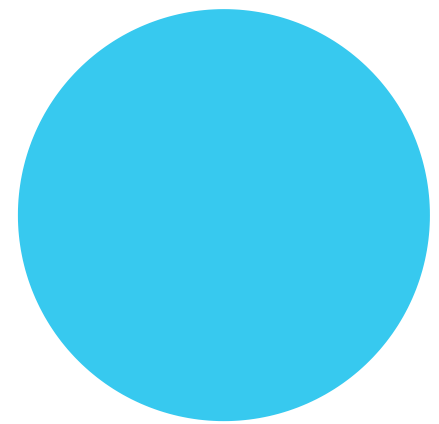
Marisangila Alves

# Cronograma



# Linguagem de programação e Ambiente





# Problema: Cadastro de funcionários

## Ações:

- O usuário deve cadastrar dados de um funcionário.
- O usuário deve visualizar os dados inseridos.
- O usuário deve ter a opção de definir um ajuste de salário.
- O usuário deve ter a opção de calcular o tempo de empresa.

## Dados:

- nome.
- CPF.
- ano de admissão.
- salário.

# Estrutural(Procedural) Orientação a Objetos

## Tipo Primitivo

Inteiro -> int

Caractere -> char, string

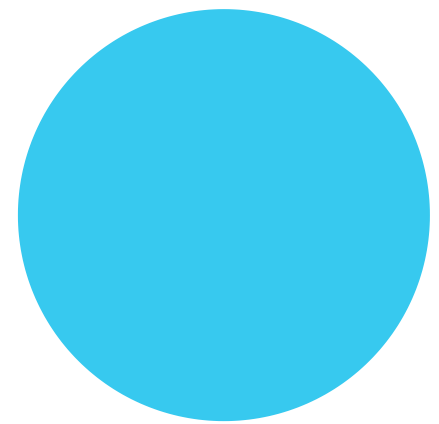
Real -> float, double

Binário -> boolean, bit

## Tipo Abstrato

Pessoa (nome, e-mail, CPF, idade, salário) -> **(string, string, string, int, float)**

Podemos construir um novo tipo de dado (Classes).



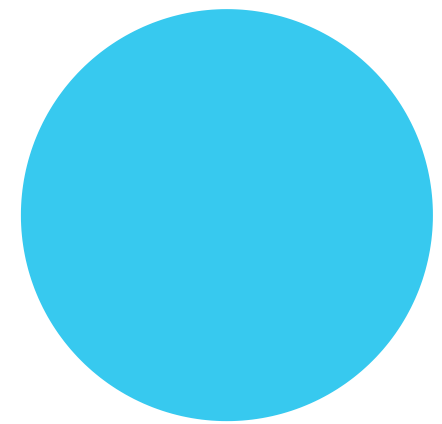
# **Vantagens Orientação a Objetos**

- Estrutura Modular. (É como juntar peças)
- Reutilização dessas estruturas em outras aplicações
- Menor dificuldade de Manutenção
- Desenvolvimento mais rápido devido ao reuso de código.
- Não é necessário conhecer o funcionamento interno de uma estrutura para utilizá-la.



# Classe

- Representa uma ideia.
- Responsabilidades bem definidas.



# Classes possuem atributos

- Atributo é uma característica ou propriedade da classe.

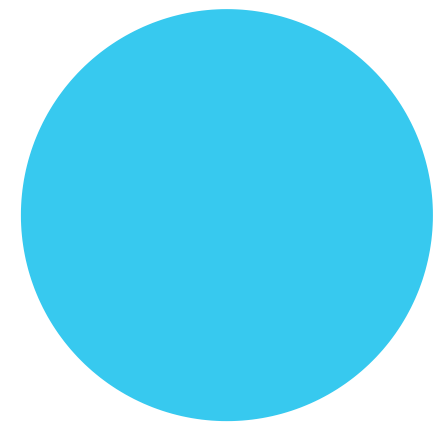
**VARIÁVEL -> ATRIBUTO**

## **Cachorro**

Nome  
Raça  
Cor  
Pelagem  
Porte  
Peso  
Idade







# Classes possuem **métodos**

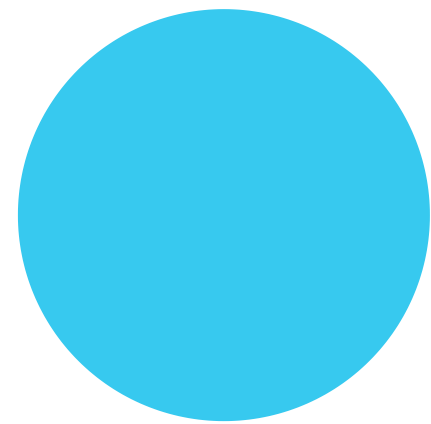
- Métodos são ações de uma classe.

**FUNÇÃO-> MÉTODO**

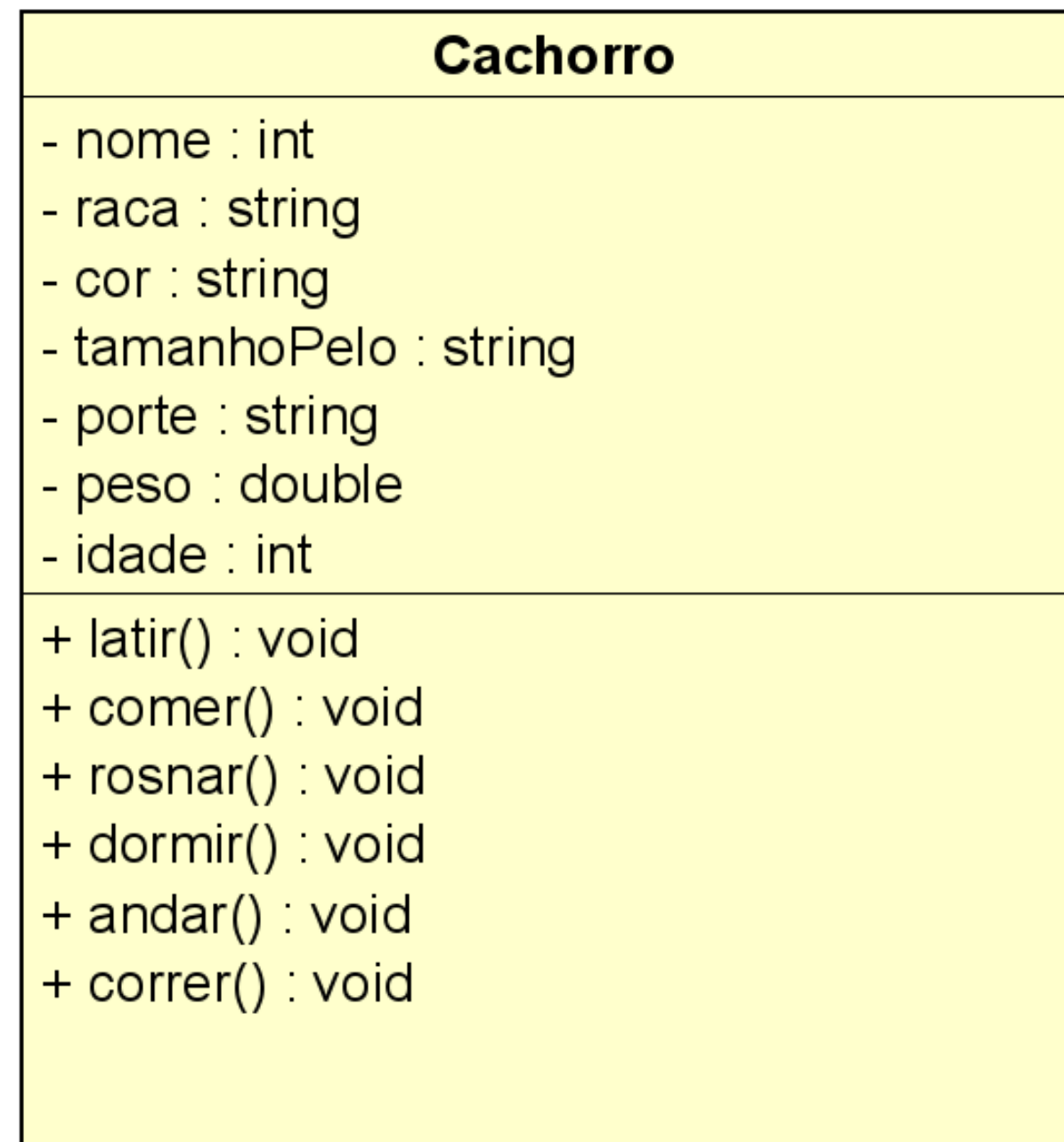
## Cachorro

Latir  
Comer  
Rosnar  
Dormir  
Andar  
Correr





# Diagrama de Classe UML



# Objeto

## Cachorro

Maya, SRD, Caramelo, Curta,  
Médio,7.6,4

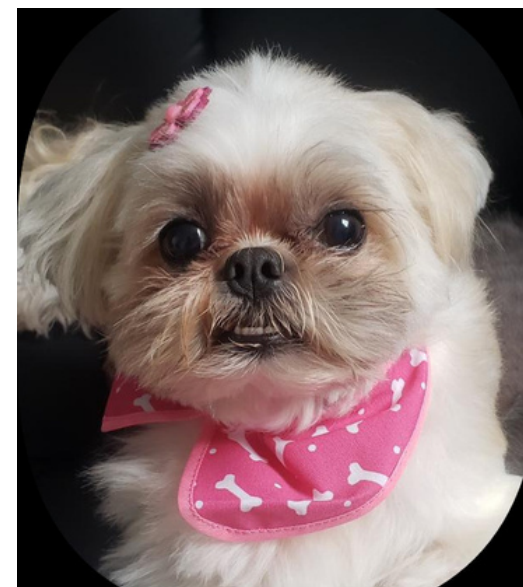
Latir, Comer, Rosnar, Dormir,  
Andar, Correr



## Cachorro

Lucy, Shih Tzu, Malhada, Longa,  
Pequeno,4,2,8

Latir, Comer, Rosnar, Dormir,  
Andar, Correr



## Cachorro

Scooby, SRD, Caramelo, Curta,  
Médio,20,11

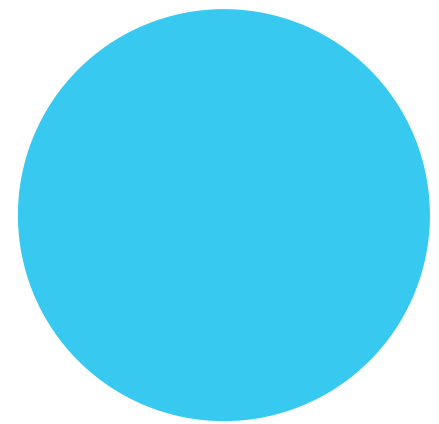
Latir, Comer, Rosnar, Dormir,  
Andar, Correr





# Classe e Objeto

- Representa uma ideia.
- Responsabilidades bem definidas.
- Classe é um molde o objeto é o item em si.
- Objeto é uma classe instanciada.



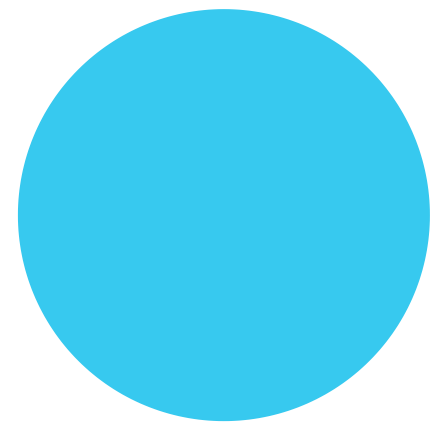
# Problema: Cadastro de funcionários

## Ações:

- O usuário deve cadastrar dados de um funcionário.
- O usuário deve visualizar os dados inseridos.
- O usuário deve ter a opção de definir um ajuste de salário.
- O usuário deve ter a opção de calcular o tempo de empresa.

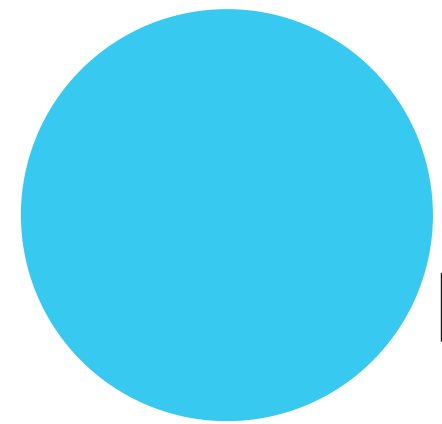
## Dados:

- nome.
- CPF.
- ano de admissão.
- salário.



## Problema: Cadastro de funcionários

Funcionario
<ul style="list-style-type: none"><li>- nome : string</li><li>- cpf : string</li><li>- anoAdmissao : string</li><li>- salario : float</li></ul>
<ul style="list-style-type: none"><li>+ cadastrarFuncionario() : void</li><li>+ calcularAjusteSalario() : void</li><li>+ calcularTempoEmpresa() : void</li><li>+ visualizarFuncionario() : void</li></ul>



# Desafio: Criar uma classe que represente um carro

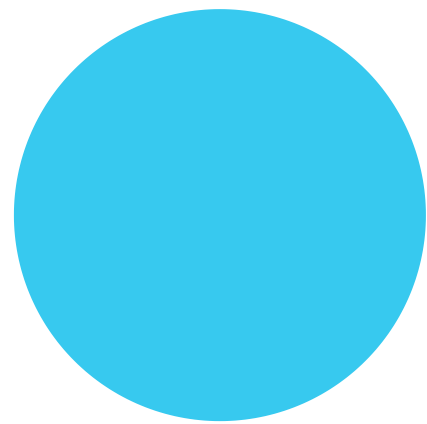
Atenção: implementar apenas os métodos `ligar()` e `desligar()` por completo, para os demais implemente apenas a assinatura do método.

Atributos:

- Cor.
- Modelo.
- Placa.
- Estado.

Ações:

- `acelerar()`.
- `frear()`.
- `ligar()`.
- `desligar()`.



# Desafio: Criar uma classe que represente um controle de TV

Atenção: implementar **todos** os métodos.

Atributos:

- Marca.
- Tamanho.
- Estado.
- Volume.
- Canal

Ações:

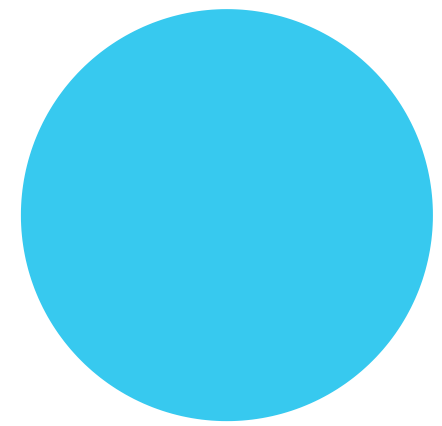
- Ligar().
- Desligar().
- AumentarVolume().
- ReduzirVolume().
- TrocarCanal().





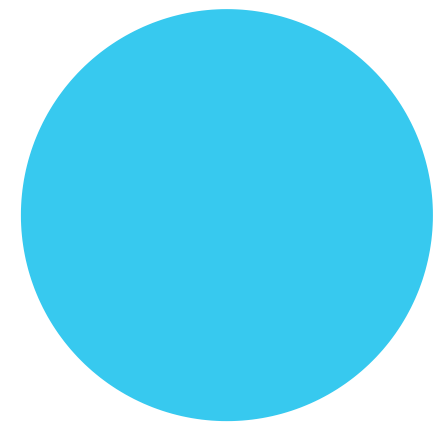
# **Método construtor**

- Ao instância a classe é necessário atribuir valor a atributos ou executar métodos.



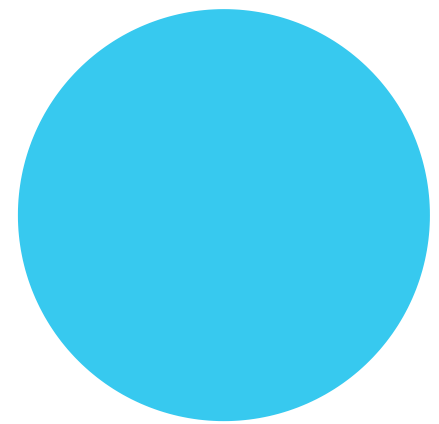
## Problema: Cadastro de funcionários

- Agora vamos criar um método **construtor** em vez de criar um método para cadastrar o funcionário.



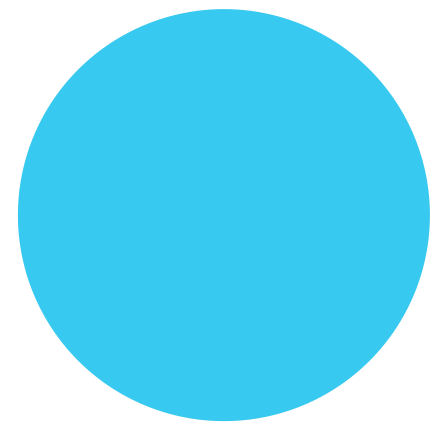
## **Sobrecarga de Métodos. (Overloading)**

- Ações similares.
- Consiste no mesmo método com parâmetros diferentes (tipo, ordem, quantidade),
- Podem ter retorno diferente ou comportamento diferente.



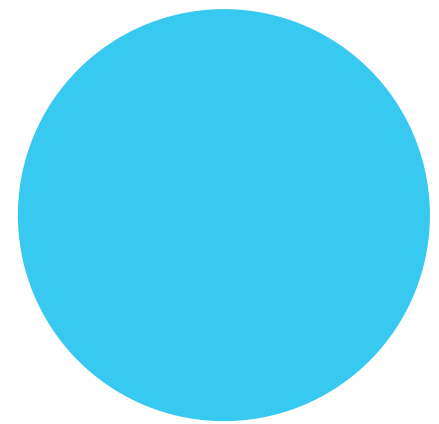
## **Problema: Cadastro de funcionários**

- Agora podemos cadastrar também funcionário que possuem contrato de pessoa jurídica, para isso ao cadastrar o funcionário é necessário incluir também o CNPJ.
- Teremos dois métodos para cadastrar um funcionário (um para pessoa física e outro para pessoa jurídica).



## **Desafio: Crie uma Classe chamada forma que represente formas geométricas.**

- Insira os atributos e métodos para calcular a área da forma.
- Note que a área é calculada de diferentes maneiras para um retângulo e para um círculo.



## Desafio: Crie uma Classe chamada Potência.

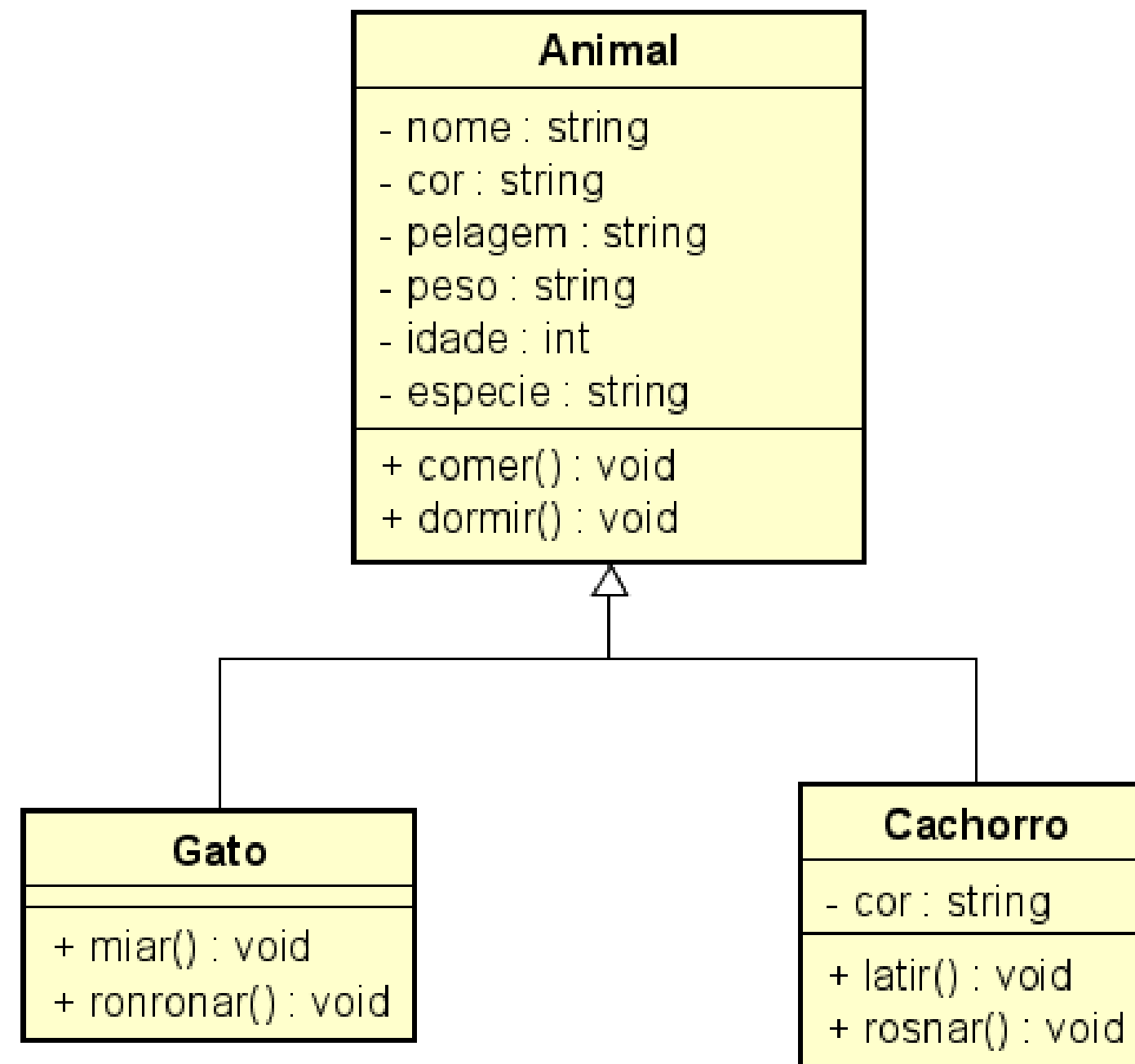
- Crie um método para calcular quando um número é elevado ao quadrado (recebe apenas base).
- Crie um método para calcular quando um número *i* é elevado a um número *j* (recebe base e expoente).



# Herança

- Uma classe pode possuir características semelhantes a uma outra classe.
- Pode existir níveis de hierarquia entre as classes.
- A partir de característica iguais podemos reutilizar o código.

# Herança





# Objeto



## Gato extends Animal

Dory, Cinza , Curta,  
,0.9,0,Felino

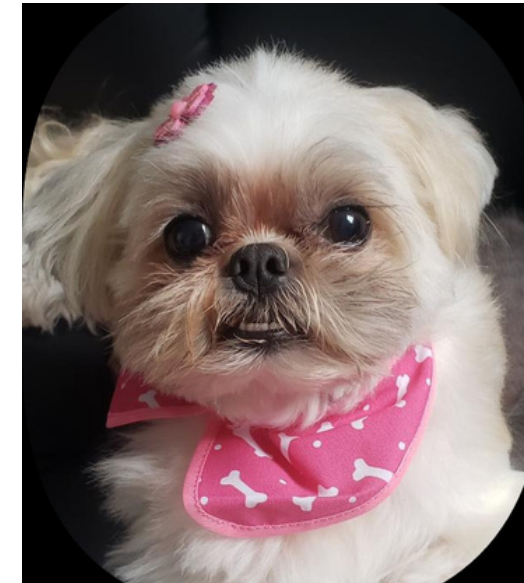
Comer,Dormir, Miar,  
Ronronar



## Cachorro extends Animal

Maya, Caramelo, Curta,  
Médio,7.6,4,Canina

Comer,Dormir, Latir, Rosnar



## Cachorro extends Animal

Lucy, Malhada, Longa,  
Pequeno,4,2,8, Canina

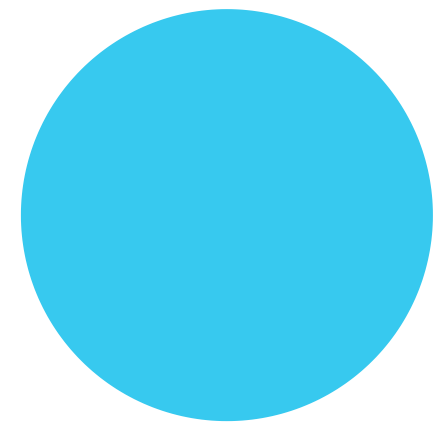
Comer,Dormir, Latir, Rosnar



## Cachorro extends Animal

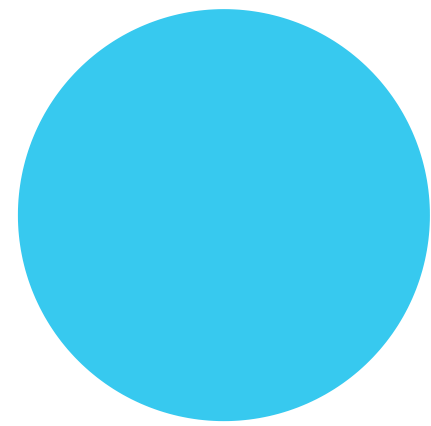
Scooby, Caramelo, Curta,  
Médio,20,11,Canina

Comer,Dormir, Latir, Rosnar

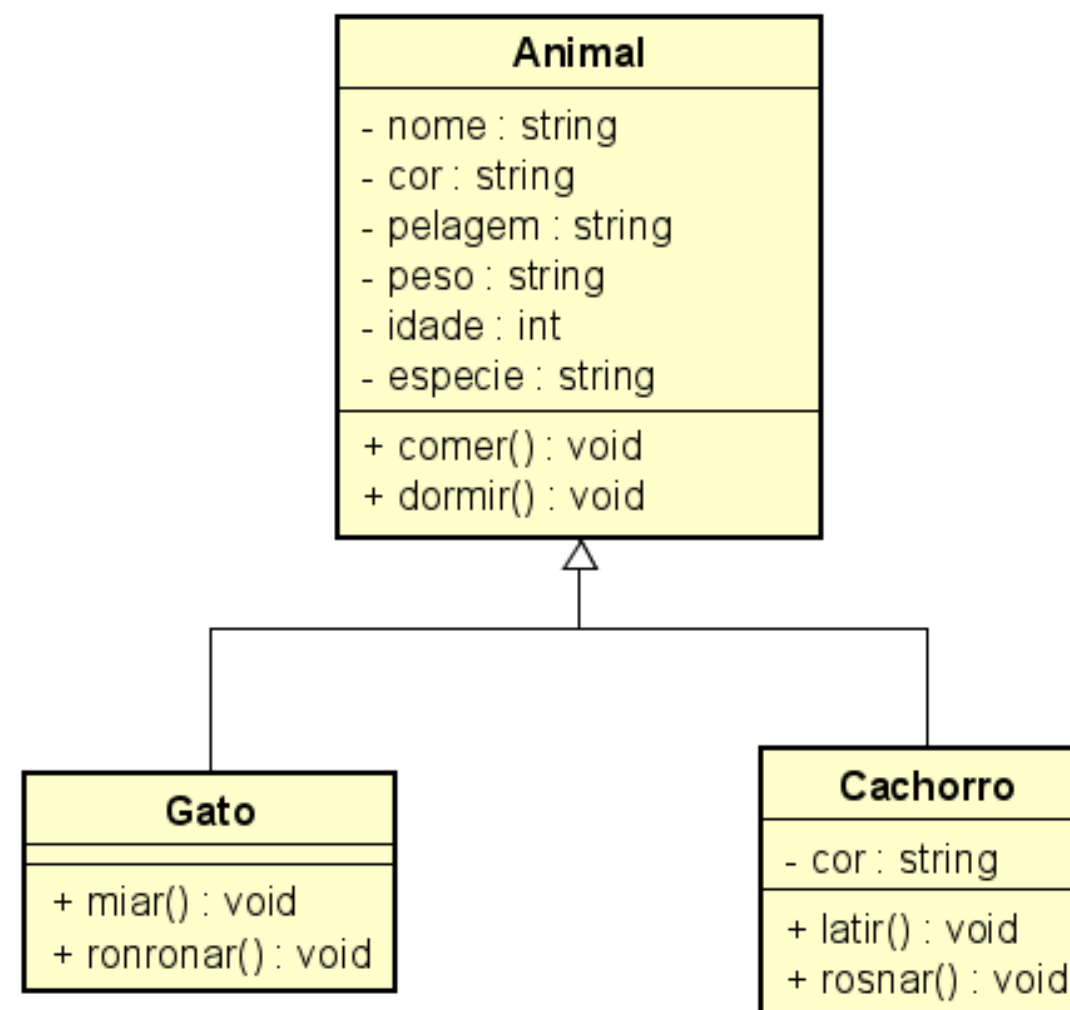


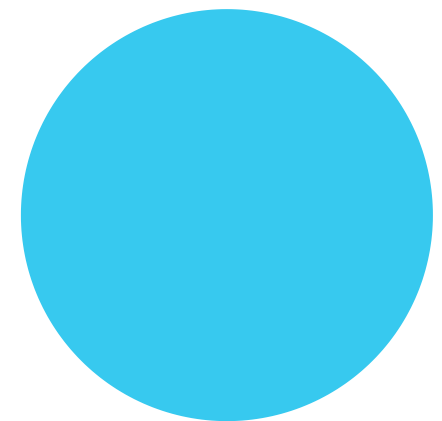
## **Problema: Cadastro de funcionários**

- Agora vamos criar uma Classe Professor que herda características da classe Funcionário.



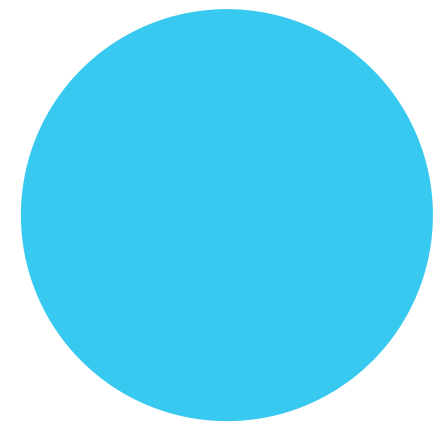
**Desafio: implemente as classes para Animal, Cachorro e Gato conforme o diagrama.**





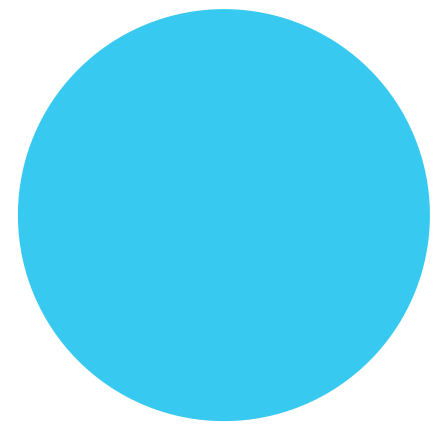
# Encapsulamento

- Uma camada de segurança.
- Impedir acesso direto ao funcionamento da classe.
- **Getters e Setters**
  - Métodos criado para acessar os atributos da classe.
  - E alterar atributos com modificadores private.



## **Problema: Cadastro de funcionários**

- Agora vamos adicionar os métodos getters e setters.



**Desafio: implemente os métodos  
getters e setters para as  
classes Animal, Cachorro e  
Gato**