

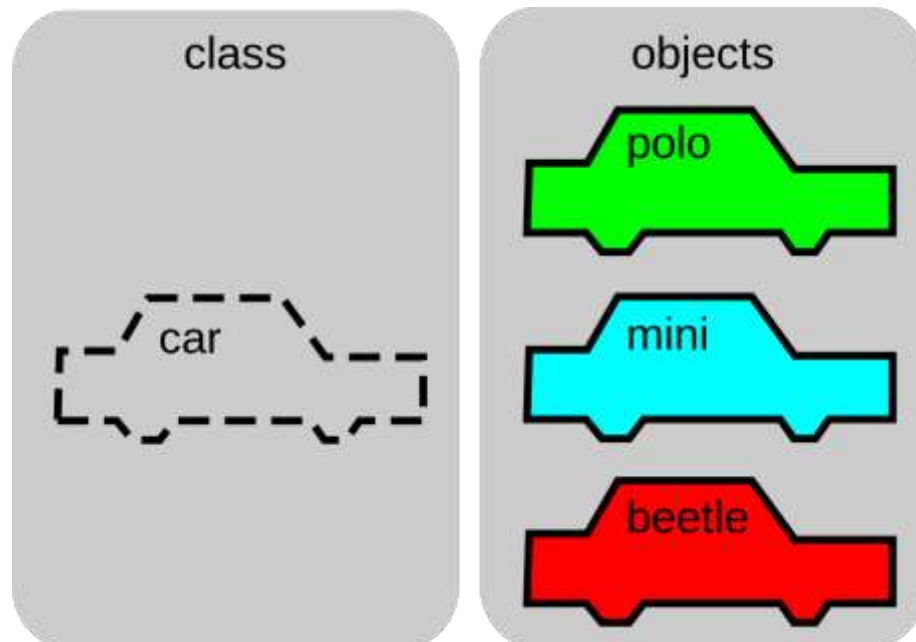
# Programação Orientada a Objetos



**Profº Ms Gustavo Molina**

# Agenda

- Classes.
- Objetos.
- Diferença entre Classe e Objeto.
- Atributos.
- Métodos.



# Conceito

## O que é uma Classe?

*“Uma **classe** é uma **entidade** descreve um conjunto de objetos com propriedades e comportamentos semelhantes e com relacionamentos comuns com outros objetos”*

**As classes são as partes mais importantes de qualquer sistema orientada a objetos.**

Usamos as **classes para capturar o vocabulário do** sistema que está em desenvolvimento. Essas classes podem incluir abstrações que são parte do *domínio* do problema, assim como as classes que fazem uma implementação. Podemos usar ainda as classes para representar itens de software, de hardware e até itens que sejam somente conceituais.

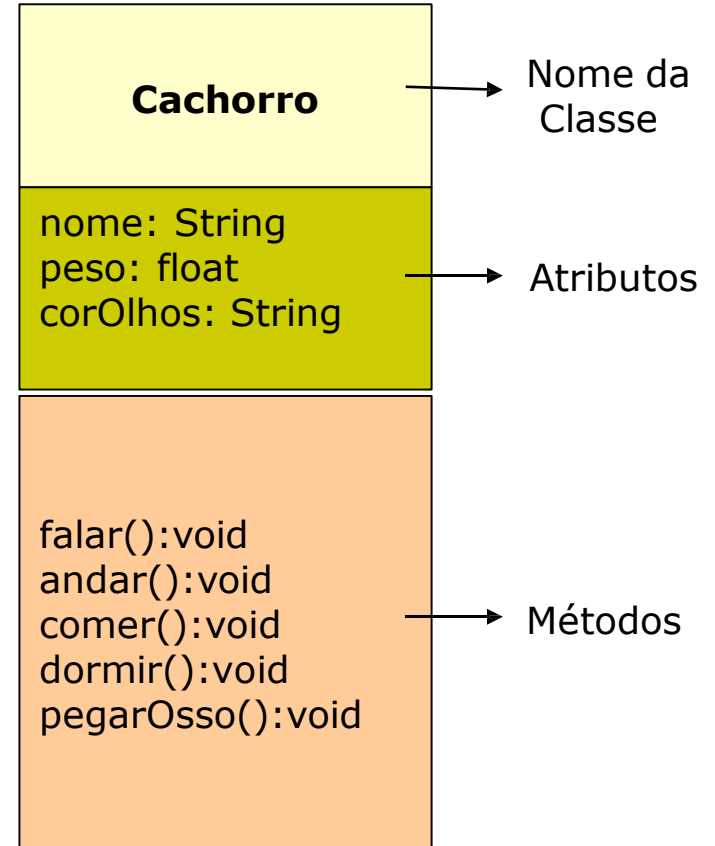
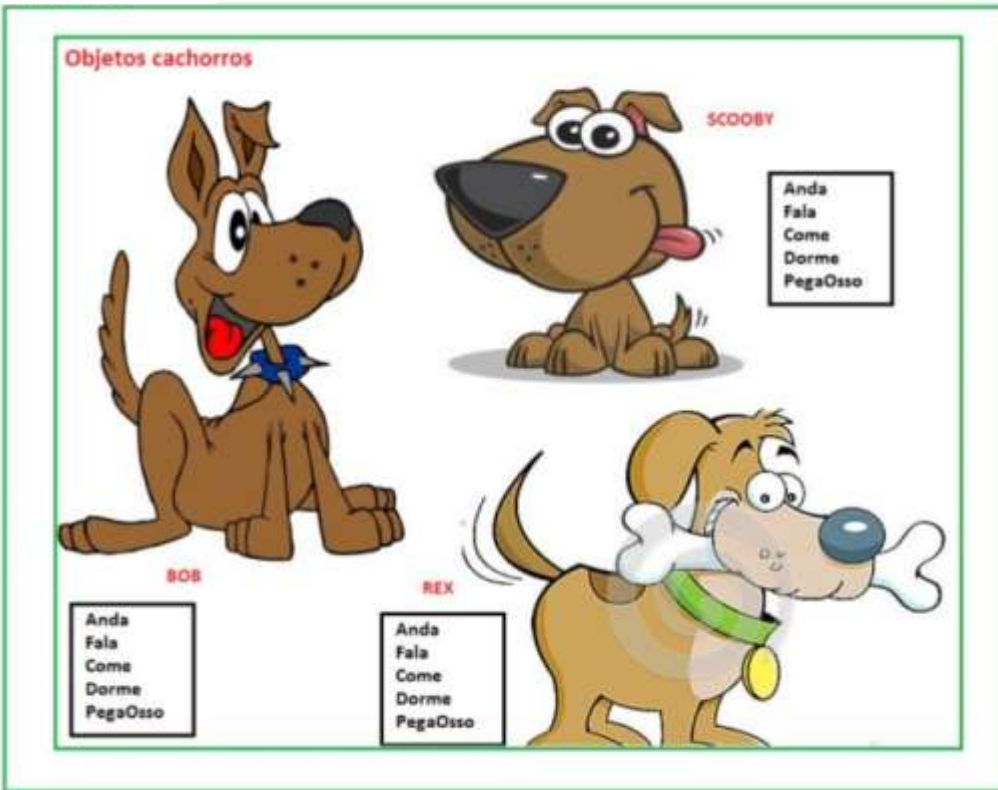
- Toda classe possui um nome;
- Possuem visibilidade, exemplo: public, private, protected;
- Possuem membros como: Características e Ações;

# Exemplo de declaração de uma classe

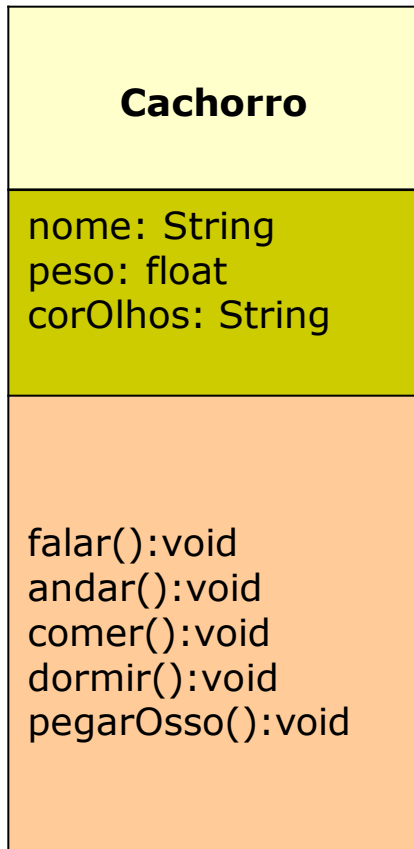
```
public class MinhaClasse{  
    //ATRIBUTOS OU PROPRIEDADES  
    //MÉTODOS  
}
```

# Exemplo

Classe Cachorro



# Exemplo



```
public class Cachorro{  
  
    public String nome;  
    public float peso;  
    public String corOlhos;  
  
    public void falar(){  
        //MÉTODO FALAR  
    }  
  
    public void andar(){  
        //MÉTODO ANDAR  
    }  
  
    public void comer(){  
        //MÉTODO COMER  
    }  
  
    public void dormir(){  
        //MÉTODO DORMIR  
    }  
  
    public void pegarOsso(){  
        //MÉTODO PEGAROSSO  
    }  
  
}
```

# Atributos

- Os atributos são as **propriedades de um objeto**, também são conhecidos como variáveis ou campos.
- Essas propriedades definem o **estado de um objeto**, fazendo com que esses valores possam sofrer alterações.

```
public class Cachorro{  
  
    public String nome;  
    public int peso;  
    public String corOlhos;  
  
}
```

# Métodos

- Os métodos são ações ou procedimentos, onde podem interagir e se comunicarem com outros objetos.
- A execução dessas ações se dá através de mensagens, tendo como função o envio de uma solicitação ao objeto para que seja efetuada a rotina desejada.
- Como boas práticas, é indicado sempre usar o nome dos métodos declarados como verbos, para que quando for efetuada alguma manutenção seja de fácil entendimento. Exemplos de nomes de métodos:
  - voltar
  - avançar
  - correr
  - resgatarValor
  - pesquisarNomes



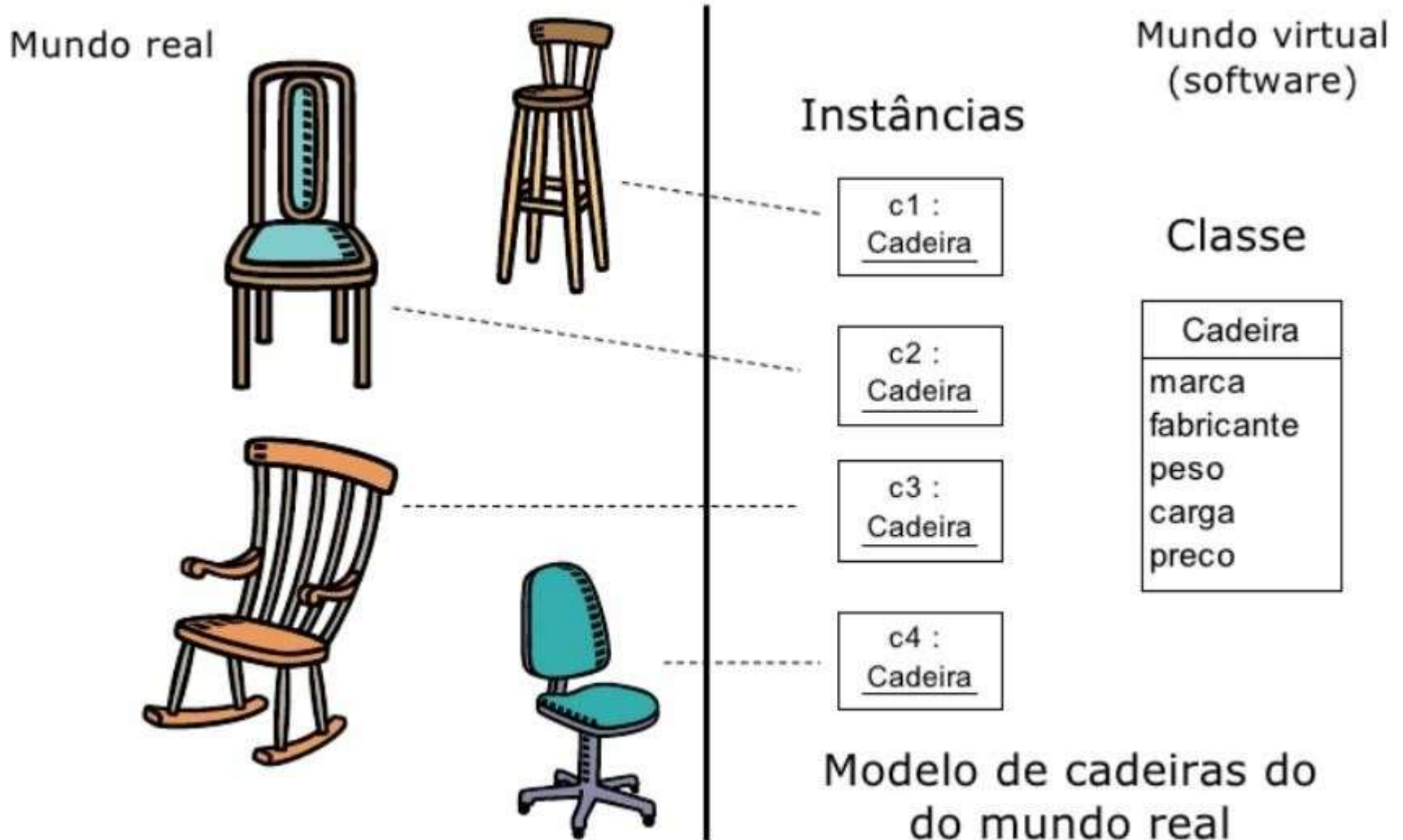
# Exemplo de implementação de um método da Classe Cachorro

```
class Cachorro{
    int tamanho;
    String nome;

    void falar(){
        if(tamanho > 60)
            System.out.println("Woof, Woof!");
        else if(tamanho > 14)
            System.out.println("Ruff!, Ruff!");
        else
            System.out.println("Yip!, Yip!");
    }
}
```

# Diferença entre Classe e Objeto

- Objetos são características definidas pela classe.
- Uma classe é um modelo, um projeto algo que não se pode tocar.



# Instanciando a classe Cachorro

```
public class TestaCachorro {  
  
    public static void main(String[] args) {  
        Cachorro cachorro1 = new Cachorro();  
        cachorro1.nome = "Pluto";  
        cachorro1.corOlhos = "azuis";  
        cachorro1.peso = 53f;  
  
        Cachorro cachorro2 = new Cachorro();  
        cachorro2.nome = "Rex";  
        cachorro2.corOlhos = "amarelo";  
        cachorro2.peso = 22f;  
  
        Cachorro cachorro3 = new Cachorro();  
        cachorro3.nome = "Bob";  
        cachorro3.corOlhos = "marrom";  
        cachorro3.peso = 13f;  
  
    }  
}
```

# Classe de teste TestaCachorro: invocando método

```
public class Testa_Cachorro {  
  
    public static void main(String[] args) {  
  
        Cachorro bob = new Cachorro();  
        bob.tamanho = 70;  
        Cachorro rex = new Cachorro();  
        rex.tamanho = 8;  
        Cachorro scooby = new Cachorro();  
        scooby.tamanho = 35;  
  
        bob.falar();  
        rex.falar();  
        scooby.falar();  
  
    }  
}
```