

# Programação Orientada a Objetos

## Semana 07 Herança e Polimorfismo

## Reflexão

“É no problema da educação que assenta o grande segredo do aperfeiçoamento da humanidade.”

Immanuel Kant (1724-1804)

# Herança

- ✓ Um mecanismo para permitir que as classes compartilhem atributos e operações baseados em um relacionamento.
- ✓ Uma subclasse herda atributos e métodos da superclasse.

## Conceitos da Herança

**Classe mãe:** superclasse, classe base;

**Classe filha/filho:** subclasse, classe derivada;

A classe filha (mais específica) herda atributos e métodos da classe mãe (mais geral);

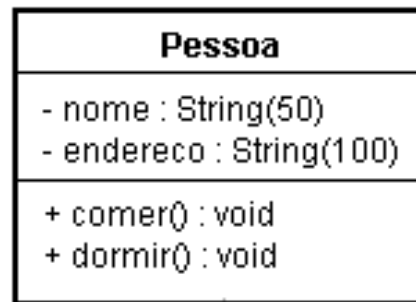
A classe filha possui atributos e métodos próprios.

## Vantagens de Herança

- ✓ Permite a modificação de uma classe (inserção de novos métodos e variáveis) sem mudanças na classe original.
- ✓ Permite a reutilização do código.
- ✓ Permite a alteração do comportamento de uma classe.

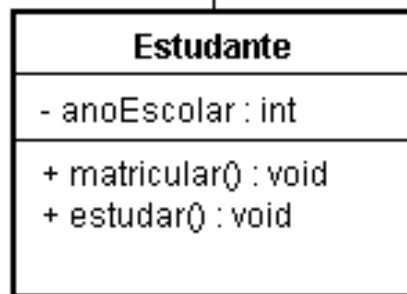
## Exemplo

- 2 atributos
- 2 operações

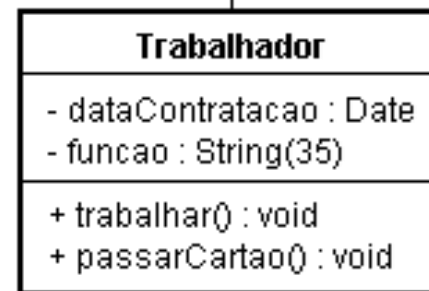


superclasse  
(mais genérica)

- 3 atributos
- 4 operações



subclasse  
(mais específica)

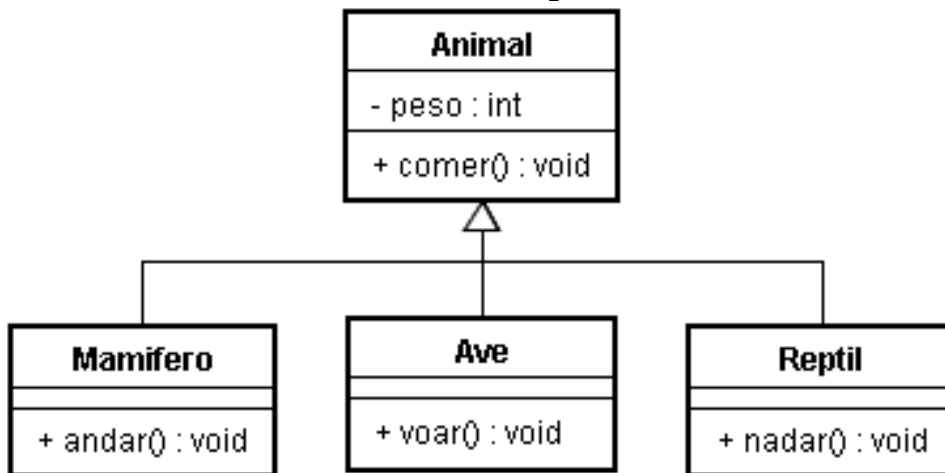


subclasse  
(mais específica)

- 4 atributos
- 4 operações

# Hierarquia de Classes

O atributo **peso** e o método **comer()** estão presentes em todos os objetos criados para as classes Animal, Mamífero, Ave e Réptil.



```
Ave pardal = new Ave();
pardal.setPeso(700);
```

```
Mamifero boi = new Mamifero();
boi.setPeso(3000);
boi.comer();
boi.andar();
boi.voar(); //Erro! não existe
```

*O método voar() pertence apenas a subclasse Ave*

## Definir Subclasse

Para se definir uma subclasse, aplicando o conceito de hierarquia, é preciso inserir a palavra reservada ***extends***.

### Sintaxe:

```
class <nome-da-classe> extends SuperClass
```

✓ Em Java, é permitido apenas uma superclasse.



# Polimorfismo

- ✓ É a capacidade de um objeto em decidir que método aplicar a si mesmo
- ✓ Termo originário do grego: "muitas formas"  
(**poli** = muitas, **morphos** = formas)
- ✓ Mecanismo que permite a um objeto assumir múltiplas formas (uma de cada vez)
- ✓ A propriedade segundo a qual uma mesma operação (método) pode comportar-se de maneira diferente em um mesmo objeto

## Polimorfismo

- ✓ A subclasse redefine a implementação de um método herdado da superclasse
- ✓ O mesmo nome e assinatura de um método podem ser utilizados para fazer com que diferentes ações ocorram, dependendo do tipo de objeto em que o método é invocado.
- ✓ Facilita a adição de novas classes a um sistema com o mínimo de modificações no código.

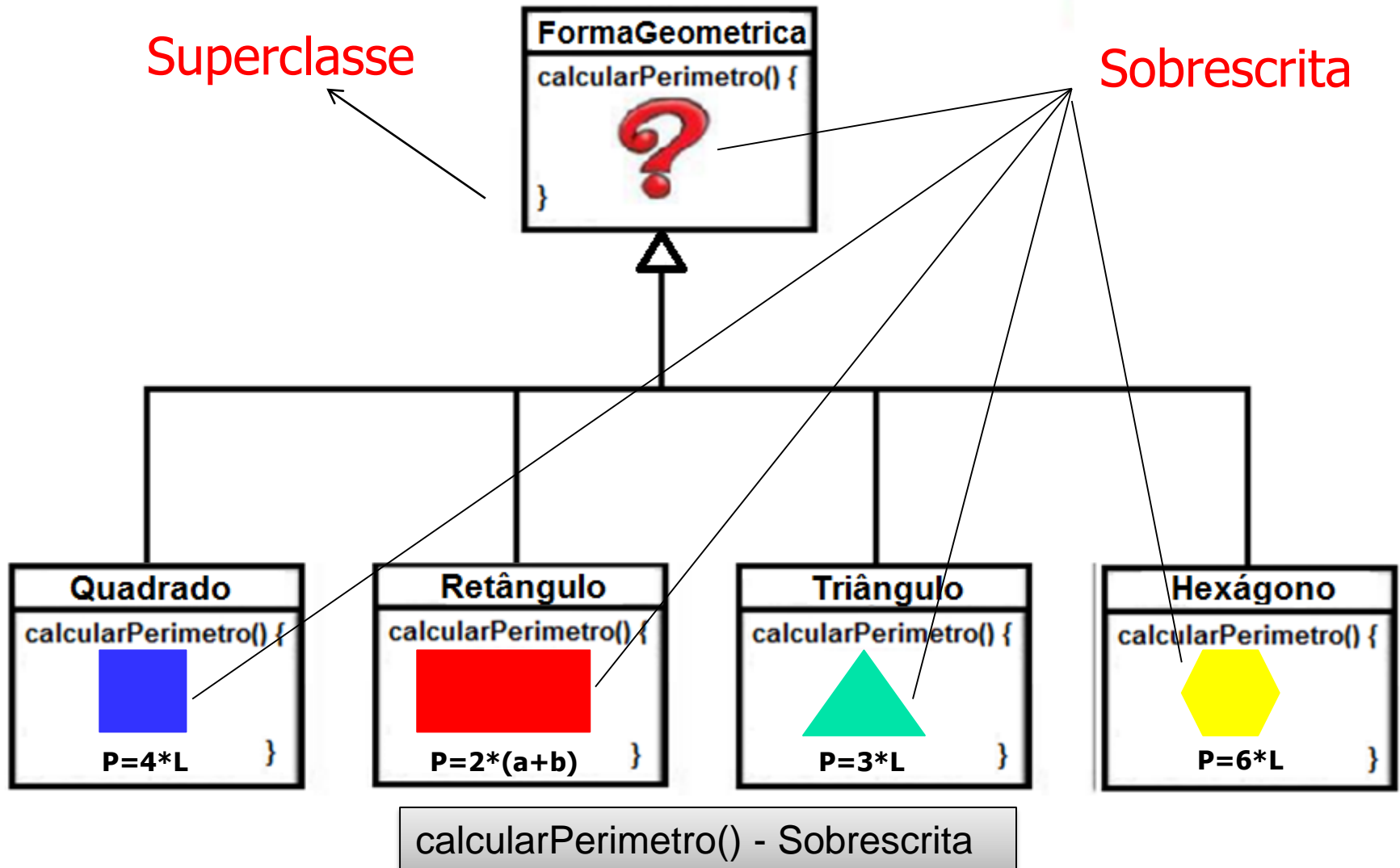
## Pré-requisitos do polimorfismo

- ✓ Deve existir uma hierarquia de classes implementando a herança;
- ✓ Deve existir a sobrescrita de métodos (override) entre as classes
- ✓ Todas as classes devem possuir métodos com a mesma assinatura (nome e parâmetros), porém com funcionalidades diferentes.

## Vantagens de Polimorfismo

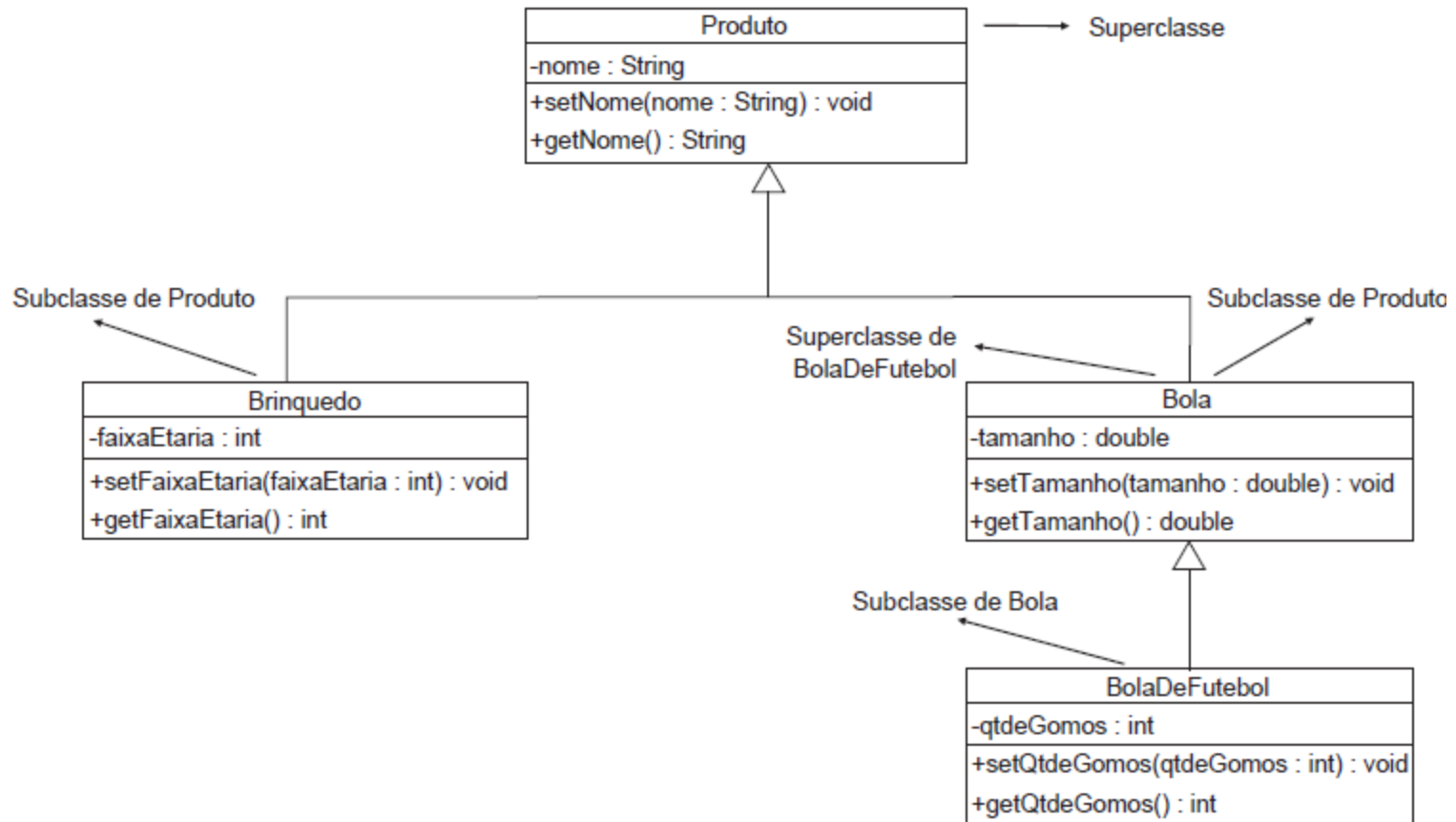
- ✓ Economiza recursos, já que um mesmo objeto pode ser usado em diferentes circunstâncias.
- ✓ Promove a reutilização de código.
- ✓ Sistema mais flexível e resistente a modificações.

## Exemplo



## Exercícios

1. Elabore a estrutura de classes em Java de acordo com o diagrama UML abaixo.



## Exercícios

2. Faça classes em Java que contenham os atributos privados que aparecem entre parênteses e implemente os métodos get e set necessários:
- a) Pneus (preço, tipo, fabricante)
  - b) Motor (potência, preço, fabricante)
  - c) Bancos (preço, modelo, fabricante)
  - d) Carro (Composta por Pneus, Motor e Bancos)

Instancie 3 objetos da classe Carro. Após instanciar um carro, imprima o seu preço (preço do pneu + preço motor + preço do bancos).

3. Faça uma aplicação que implemente uma classe Imovel, que possui um endereço e um preço. Crie as subclasses Casa, Apartamento e Chacara que herdam da classe imóvel. Para cada subclasse, defina os atributos seguintes: Casa (possuiQuintal), apartamento (possuiSacada) e Chacara (tamanhoCampoFutebol, possuiPiscina, metragemDaCasa). Elabore o diagrama da UML correspondente a sua solução.

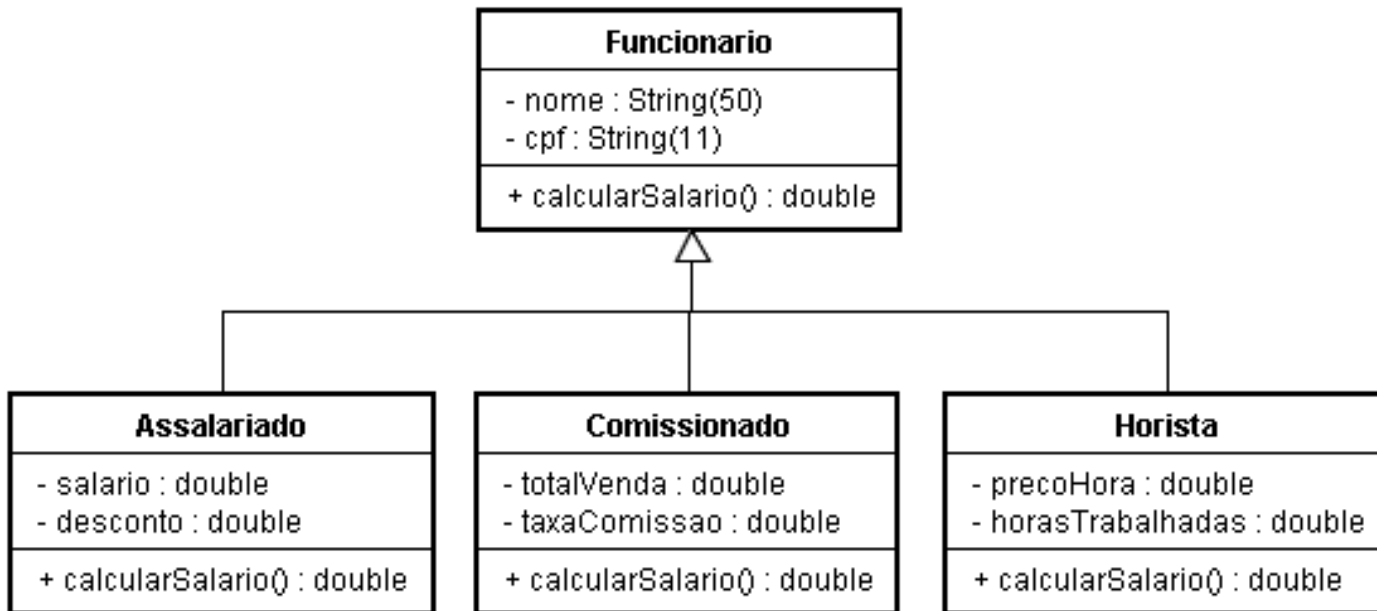
## Exercícios

4. Considere uma superclasse chamada Figura contendo o método calcularArea. Considere também duas subclasses de Figura chamadas Quadrado e Triangulo. A primeira possui o método calcularArea, e a segunda, o método calcular. Em função disso, em qual classe não será possível implementar o polimorfismo? Por quê?
5. Usando o mecanismo de herança, faça um diagrama considerando as seguintes classes: Pessoa, PessoaFisica, PessoaJuridica e Motorista. Considere que esse conjunto de classes será usado em um site de uma empresa de transporte. Defina pelo menos dois atributos para cada uma das classes, de acordo com o contexto apresentado.
6. Crie uma aplicação que implemente o código das classes do exercício anterior. Elabore também uma classe de teste que utilize a classe Motorista e imprima em tela o conteúdo de seus atributos.



## Exercícios

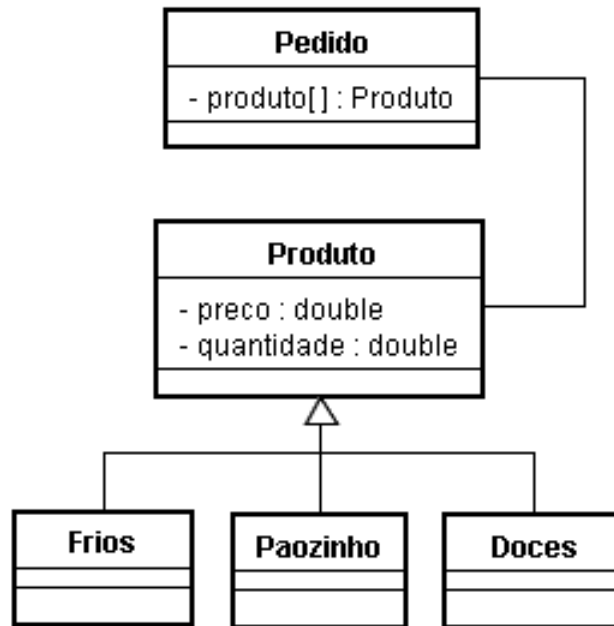
7. Observe o diagrama abaixo:



Crie o código das classes considerando o conceito do polimorfismo.

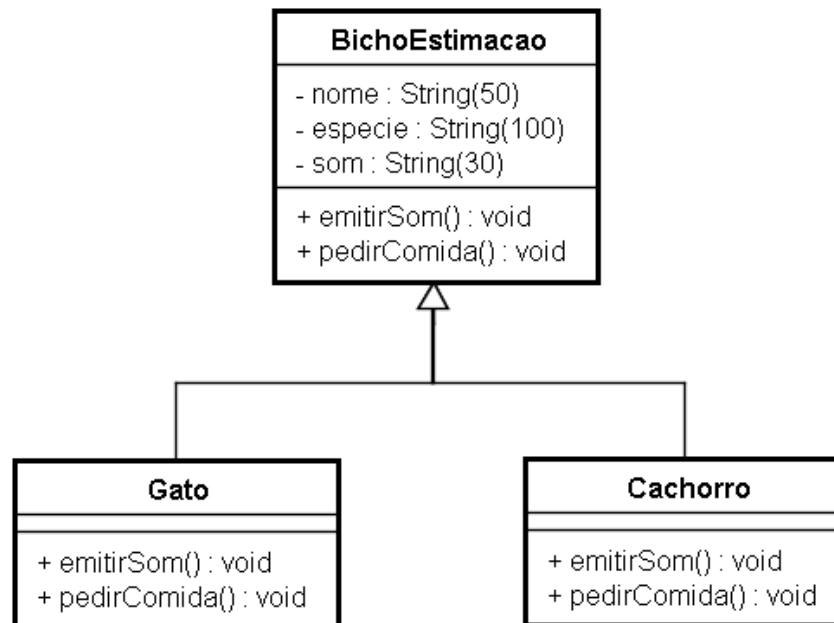
## Exercícios

8. Crie uma aplicação que implemente a classe `UsaFuncionario`, que solicita ao usuário a escolha do tipo de `Funcionario` e mostra em tela o valor do salário por meio dos métodos criados no exercício 7.
9. Crie uma aplicação que implemente as classes abaixo e sua hierarquia.



## Exercícios

10. Crie uma aplicação que implemente a hierarquia conforme o diagrama abaixo, seguindo o conteúdo de cada método conforme a tabela.



	<b>void emitirSom()</b>	<b>void pedirComida()</b>
<b>Gato</b>	Imprimir "Miau Miau"	Chamar o método <b>void emitirSom()</b> e depois imprimir "Gato faz olhar triste".
<b>Cachorro</b>	Imprimir "Au Au"	Chamar o método <b>void emitirSom()</b> e depois imprimir "Cachorro abana o rabo".