



HERANÇA E POLIMORFISMO

Wesley Reis da Silva



Curiosidade

Início de Orientação a Objetos surgiu nos anos 60 onde foram introduzidos os conceitos de classe e herança.



Pilares Orientação Objeto

Orientação a Objetos foi criada de forma a refletir o que acontece no mundo real (mistura biologia com computação).



Herança é um dos pilares da OO

O que é Herança?



Mais do que apenas bens materiais.



O que é Herança?



“Ela puxou ... da mãe”

No nosso dia a dia, herança significa descendência, herdar características e/ou comportamentos.



Herança P00

O mesmo conceito é utilizado em P00.

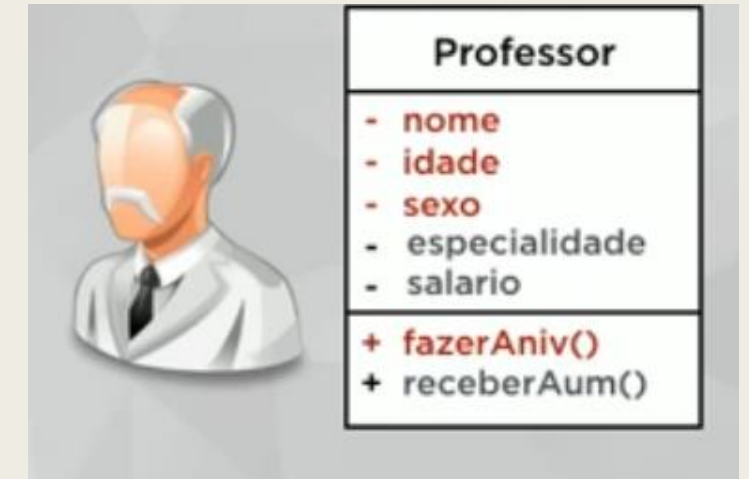
As classes podem ser mães de outras classes, assim suas filhas preservam, herdam, os atributos(características) e métodos (comportamento) da classe mãe.

É também conhecido como generalização, a classe mãe funciona como uma classe padrão para criar outras. Evitando repetição de código.



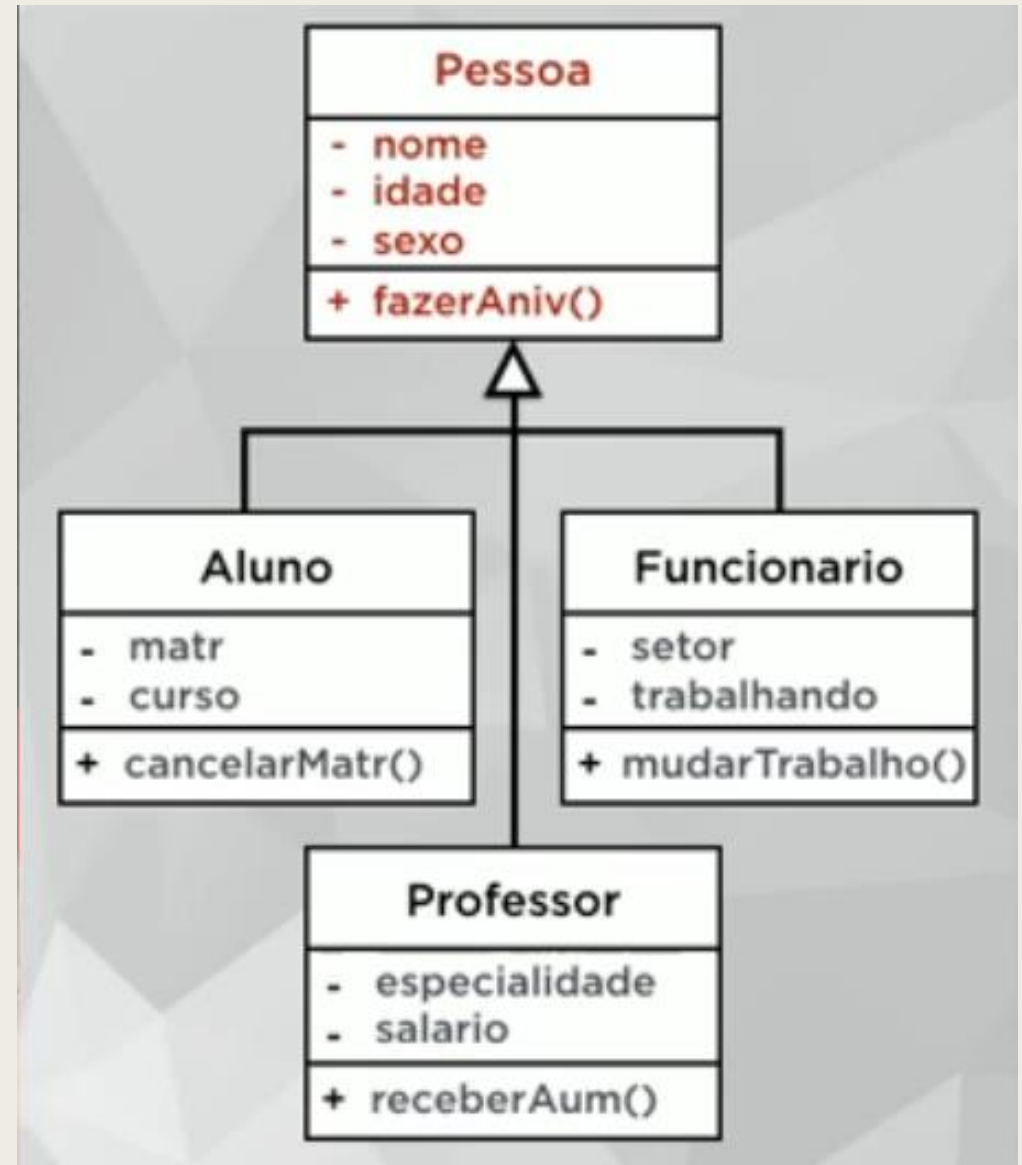
Exemplo

É possível perceber que existem características e ações em comum para todas as classes.



Exemplo

Assim criamos uma classe que continha as dados comum, uma generalização. Com a seta informamos que as classes Aluno, Funcionário e Professor são filhas da classe pessoa, assim além das características ações contidas na classe mãe. Elas se estendem e agregam mais algumas características e/ou ações próprias.

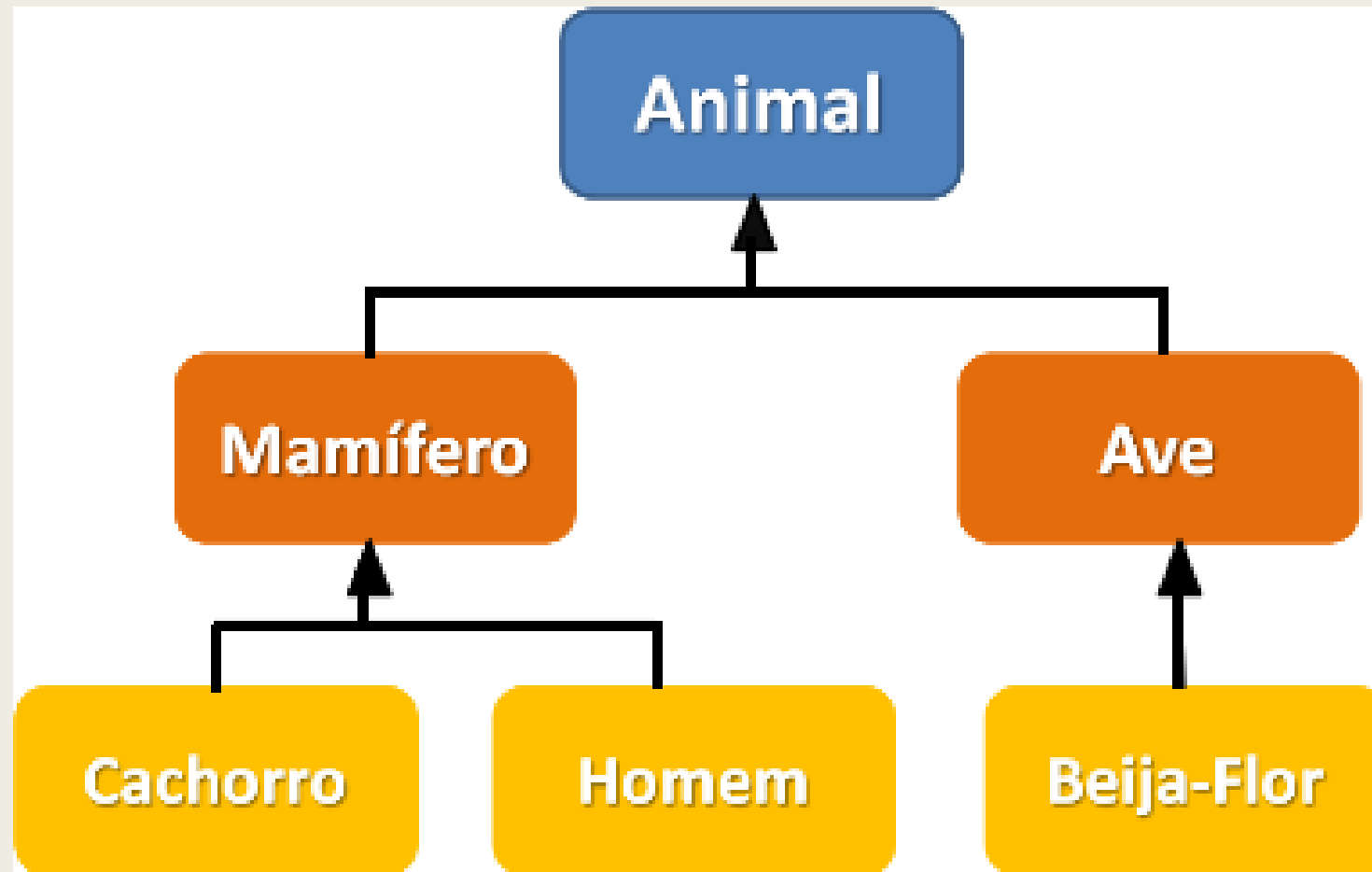


Herança Java

Para dizer que uma classes é filha de outra, no Java utilizasse da palavra reservada **extends**.

```
public class Gerente extends Funcionario{  
  
    private int senhaEspecial;  
    private int numFuncionarios;  
    // ...restante da classe  
}
```

Exemplo



Reescrita de Métodos

Uma subclasse pode precisar de uma implementação diferente da que sua classe mãe realizou, assim é possível que a subclasse reescreva esse método fazendo sua própria implementação, utilizando do conceito de sobreposição.

```
public class Gerente extends Funcionario{

    private int senhaEspecial;
    private int numFuncionarios;

    @Override
    public float calculaBonificacao()
    {
        return super.calculaBonificacao() + 1000;
    }
    // ... restante da classe
}
```

Polimorfismo

Poli = Mais de um

Morfo = Forma

Polimorfismo permite que um mesmo nome represente vários comportamentos ou objetos diferentes.



Bolacha



Polimorfismo

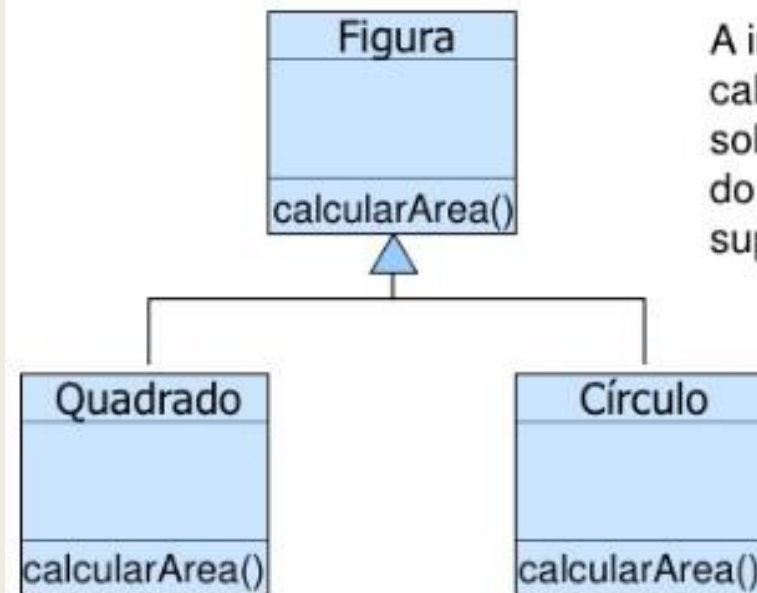
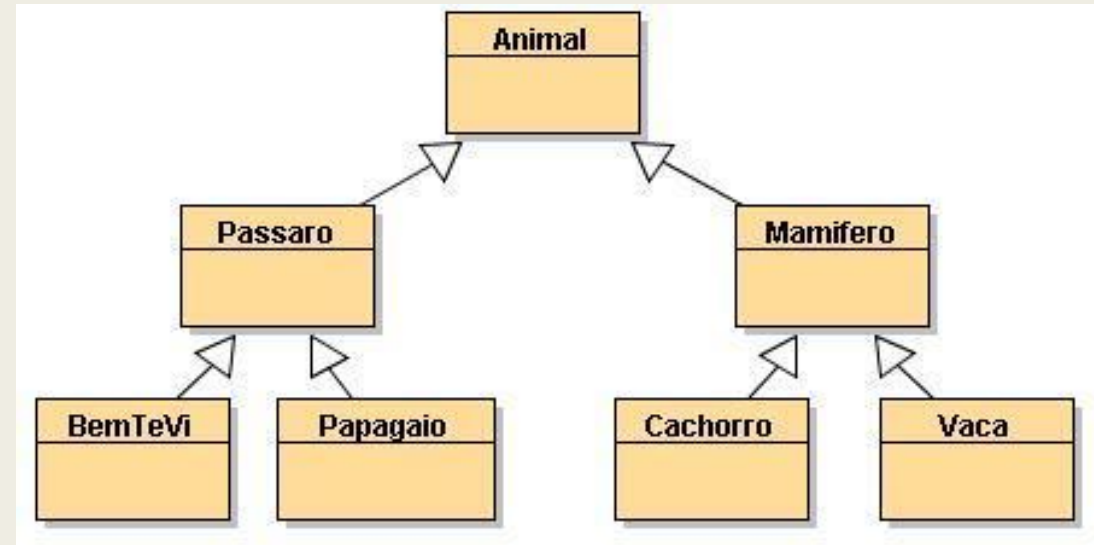
Existem vários tipos de polimorfismo mas iremos focar em 2 tipos:

- Sobreposição: permite referenciar um objeto de uma subclasse como um objeto da superclasse. Sobrescrever métodos herdados.

```
List lista = new LinkedList();
```

- Sobrecarga: ter vários métodos com o mesmo nome, o diferenciando entre ele por sua assinatura, quantidade e os tipos dos parâmetros.

Sobreposição



A implementação do método `calcularArea()` das subclasses sobrepõe a implementação do mesmo método na superclasse.

Classes e métodos finais

Classes Finais: Não podem ser herdadas por outra classe. São obrigatoriamente folhas.

Métodos Finais: Não pode ser sobrescrito pelas suas subclasses. Obrigatoriamente herdado.

Material auxiliar

- <https://www.devmedia.com.br/metodos-atributos-e-classes-no-java/25404>

Projeto

- Máximo 3 pessoas;
- Deve ser um sistema de cadastro;
- Deve possuir no mínimo 3 CRUDs (cadastro, listagem, atualização e remoção);
- Deve persistir os dados (arquivo .txt ou BD);
- Temas devem ser decididos até dia 02/10;
- Deve ser apresentado junto com projeto um documento, contendo descrição do projeto, classes utilizadas e ação que cada classe realiza;
- Entrega do projeto por E-mail até dia 26/11 as 23:59.