



Programação Orientada a Objetos 2022

Prof. Dr. José Erinaldo da Fonsêca

São Paulo - SP, Abril de 2022.

Pilares de POO



- ▶ Encapsulamento
- ▶ Herança
- ▶ Polimorfismo

Herança

- ▶ Mãe e filha
- ▶ Normalmente os filhos trazem características e comportamentos da mãe.
- ▶ Este é conceito de Herança.
- ▶ As Classes são consideradas mães que podem gerar filhas.
- ▶ As filhas podem herdar características e comportamentos das mães.



Herança

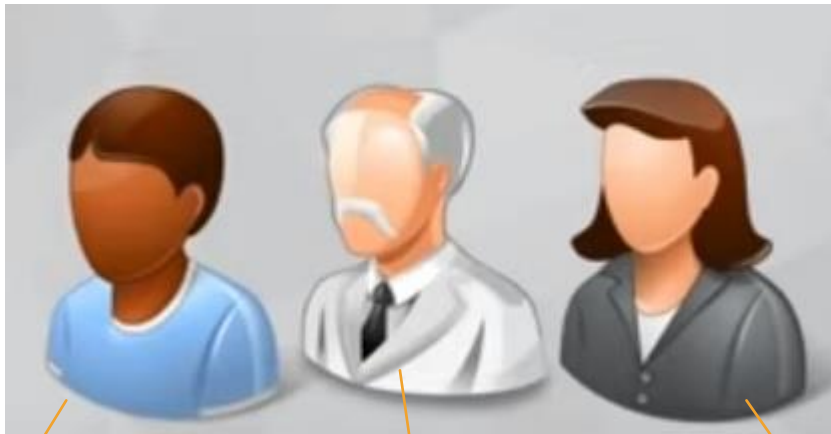
- ▶ Permite basear uma nova classe na definição de uma outra classe previamente existente.

Herança

- ▶ A herança será aplicada tanto para as características quanto para os comportamentos

Herança

- Suponha essas três pessoas serem um aluno, um professor e uma funcionária.



Aluno

Professor

Funcionária

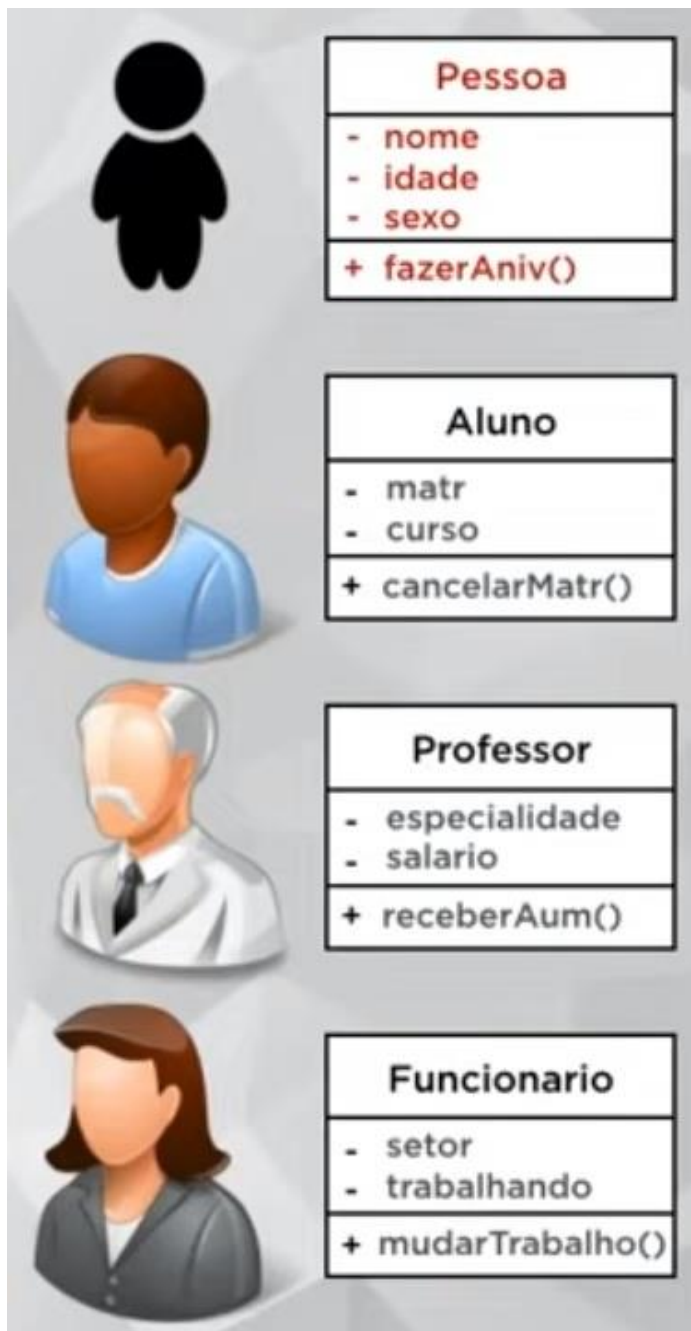


- ▶ Todas as pessoas (Aluno, Professor e Funcionário) apresentam:

Atributos

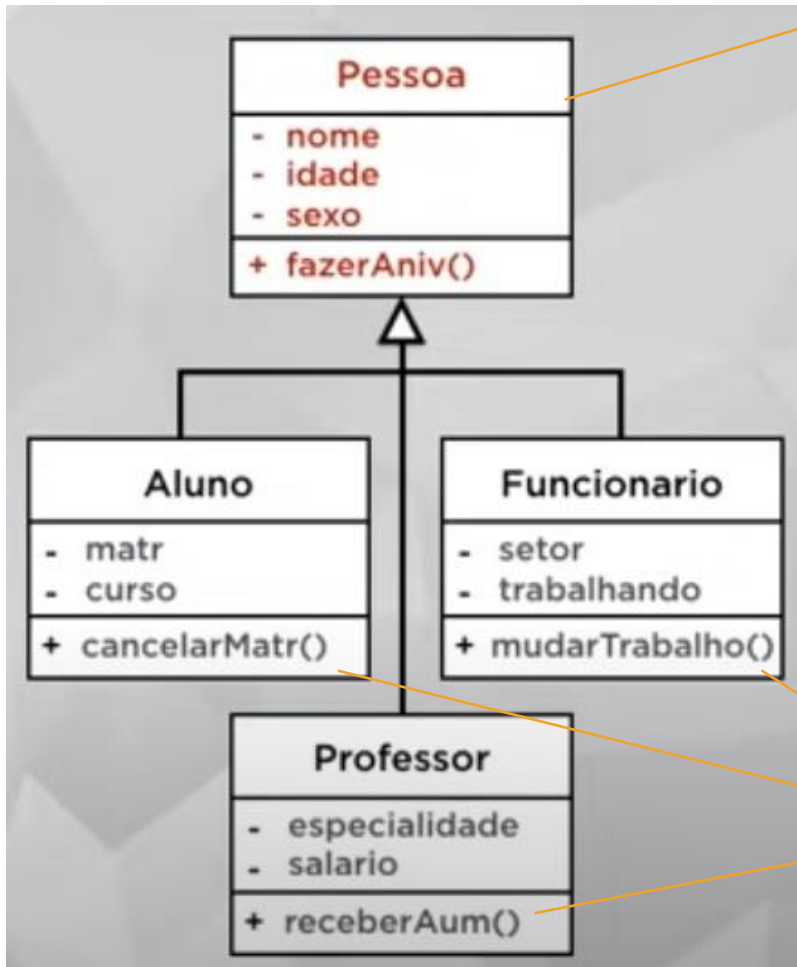
Métodos

- ▶ Em vermelho está destacado os atributos e métodos que ambos compartilham.
- ▶ O que isso sugere?



- ▶ Os atributos e métodos comuns, sugerem que uma classe mãe pode ser construída.
- ▶ As demais classes podem herdar tais características e comportamentos.
- ▶ Todos (Aluno, Professor e Funcionário) herdam características de Pessoa.

Herança



Mãe
SuperClasse

- O diagrama mostra que as classes Aluno, Professor e Funcionário, herdam características e comportamentos da Classe mãe (Pessoa).

Filha
SubClasse

```

classe Pessoa
    // Atributos
    privado nome: Caractere
    privado idade: Inteiro
    privado sexo: Caractere
    // Métodos
    publico metodo fazerAniv()
        (...)
    fimMetodo
    // Métodos Especiais
    publico metodo getNome()
        (...)
    fimMetodo
    publico metodo setNome()
        (...)
    fimMetodo
    publico metodo getIdade()
        (...)
    fimMetodo
    publico metodo setIdade()
        (...)
    fimMetodo
    publico metodo getSexo()
        (...)
    fimMetodo
    publico metodo setSexo()
        (...)
    fimMetodo
FimClasse

```

Pessoa
- nome
- idade
- sexo
+ fazerAniv()

- ▶ Inicialmente construímos a Classe mãe, Pessoa.
- ▶ Inserimos também os Gets e Sets.

Aluno
- matr
- curso
+ cancelarMatr()

```
classe Aluno estende Pessoa
// Atributos
privado matr: Inteiro
privado curso: Caractere
// Métodos
publico metodo cancelarMatr()
    (...)
fimMetodo
// Métodos Especiais
publico metodo getMatr()
    (...)
fimMetodo
publico metodo setMatr()
    (...)
fimMetodo
publico metodo getCurso()
    (...)
fimMetodo
publico metodo setCurso()
    (...)
fimMetodo
FimClasse
```

- ▶ Criamos a classe Aluno com os seus devidos atributos e métodos.
- ▶ Veja que a classe Aluno se estende a classe Pessoa.

Professor
- especialidade
- salario
+ receberAum()

```
classe Professor estende Pessoa
// Atributos
privado especialidade: Caractere
privado salario: Real
// Métodos
publico metodo ReceberAum()
    (...)
fimMetodo
// Métodos Especiais
publico metodo getEspecialidade()
    (...)
fimMetodo
publico metodo setEspecialidade()
    (...)
fimMetodo
publico metodo getSalario()
    (...)
fimMetodo
publico metodo setSalario()
    (...)
fimMetodo
FimClasse
```

- ▶ Criamos a classe Professor com os seus devidos atributos e métodos.
- ▶ Veja que a classe Professor se estende a classe Pessoa.

Funcionario
- setor
- trabalhando
+ mudarTrabalho()

```
classe Funcionario estende Pessoa
// Atributos
privado setor: Caractere
privado trabalhando: Logico
// Métodos
publico metodo mudarTrabalho()
    (...)
fimMetodo
// Métodos Especiais
publico metodo getSetor()
    (...)
fimMetodo
publico metodo setSetor()
    (...)
fimMetodo
publico metodo getTrabalhando()
    (...)
fimMetodo
publico metodo setTrabalhando()
    (...)
fimMetodo
FimClasse
```

- ▶ Criamos a classe Funcionário com os seus devidos atributos e métodos.
- ▶ Veja que a classe Funcionário se estende a classe Pessoa.

Desenvolvido
todas as Classes,
podemos instanciar os objetos.



```
// Programa Principal
p1 = new Pessoa()
p2 = new Aluno()
p3 = new Professor()
p4 = new Funcionario()

p1.setNome("Pedro")
p2.setNome("Maria")
p3.setNome("Cláudio")
p4.setNome("Fabiana")

p2.setCurso("Informática")
p3.setSalario(2500.75)
p4.setSetor("Estoque")

p1.receberAum(550.20)
p2.mudarTrabalho()
p4.cancelarMatr()
```

- ▶ Inicialmente instanciamos os objetos.
- ▶ Podemos inserir os nomes.
- ▶ Podemos inserir o curso ou salário, quando pertinente.
- ▶ Veja que os últimos tópicos não podem acontecer, pois P1(que é pessoa) não recebe aumento. Quem recebe aumento é o professor.
- ▶ Veja que P2 (que é Aluno) não pode mudar de trabalho.
- ▶ P4 (que é Funcionário) não pode cancelar matrícula.

Vamos praticar?
Chegou seu momento de
colocar em java todo o código.

Referências

- ▶ DEITEL, P.; DEITEL, H.; **Java: como programar**. 10ª ed. São Paulo: Pearson Education do Brasil, 2017. E-book
- ▶ SCHILDT, H.; **Java para iniciantes: crie, compile e execute programas Java rapidamente** [recurso eletrônico]. 6. ed. Porto Alegre: Bookman, 2015. E-book
- ▶ SEBESTA, R.W; **Conceitos de linguagem de programação** [recurso eletrônico]. 11ª ed. Porto Alegre: Bookman, 2018. E-book.
- ▶ Acessado em 10/02/2022. <https://www.cursoemvideo.com/curso/java-poo/>



► FIM!!