



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

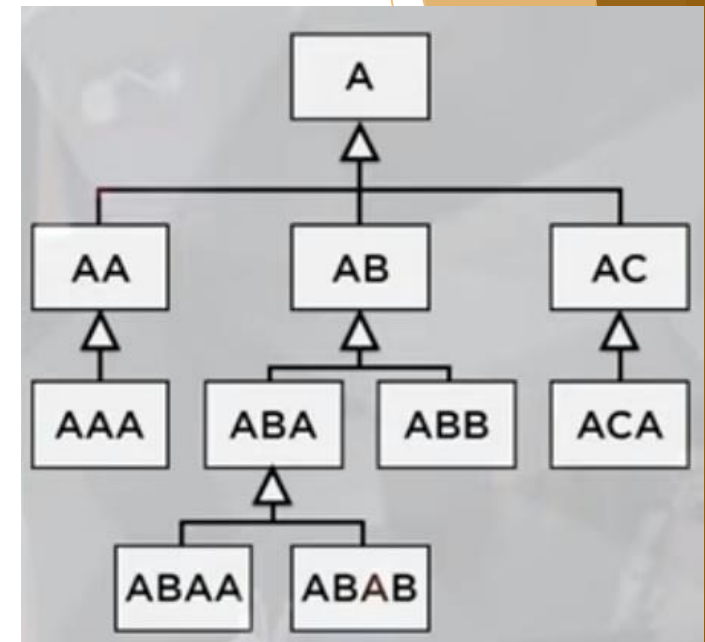
- ▶ Permite basear uma nova classe na definição de uma outra classe previamente existente.

Herança

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

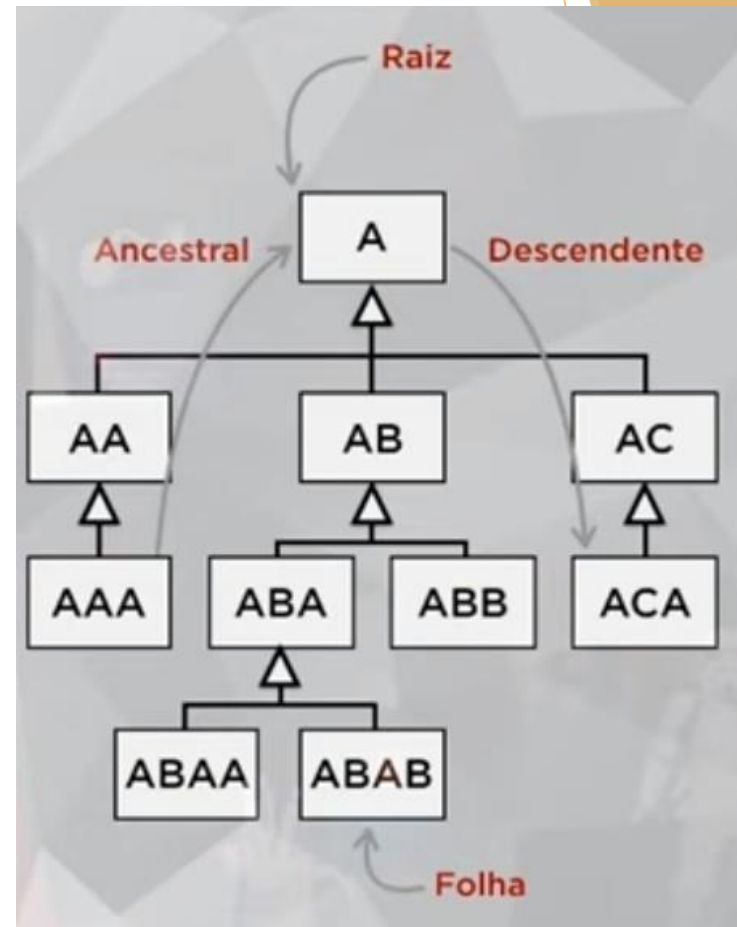
Herança

- ▶ Podemos criar uma árvore de heranças.
- ▶ Portanto, é possível navegar pela herança.
- ▶ Dizemos que “A” é uma superClasse ou Progenitora.
- ▶ Dizemos que “ABAA” é uma subClasse.
- ▶ Mas também dizemos que “AB” é uma superClasse e uma subclasse.
- ▶ Portanto, dizer o que é superclasse ou subClasse depende do referencial.



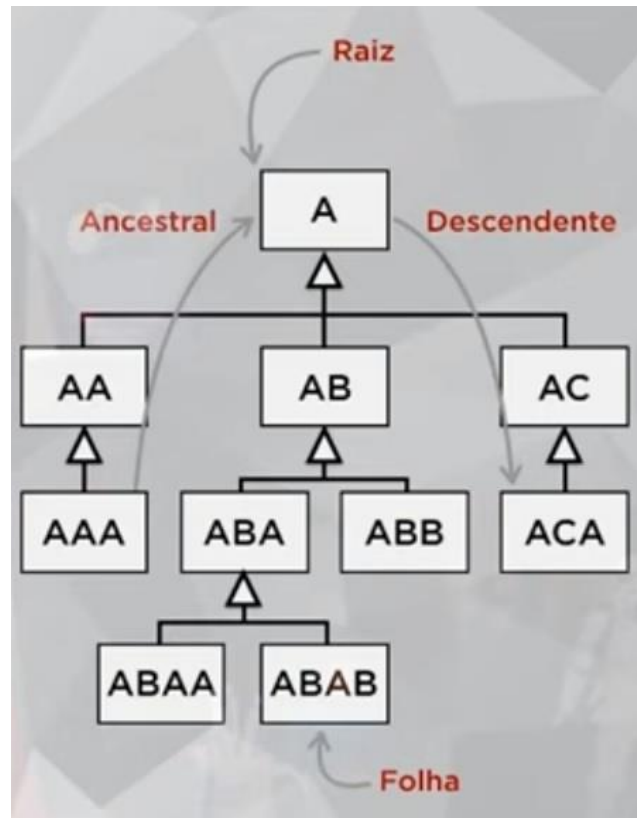
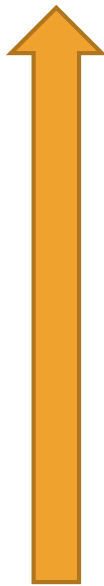
Árvore Hierárquica

- ▶ Raiz: a Classe origem de tudo.
- ▶ Folha: a ultima classe da árvore. Não existe outras classes após ela.
- ▶ Ancestral: Classes a partir da hierarquia “netos”. Portanto, filhas não são ancestrais.
- ▶ Descendentes: Classes a partir da hierarquia “netos”. Portanto, filhas não são descendentes.



Árvore Hierárquica

Generalização



Especialização



Tipos de Herança

► Herança de Implementação.

É o tipo de herança mais simples que existe, podemos também dizer que é a herança mais pobre que existe.

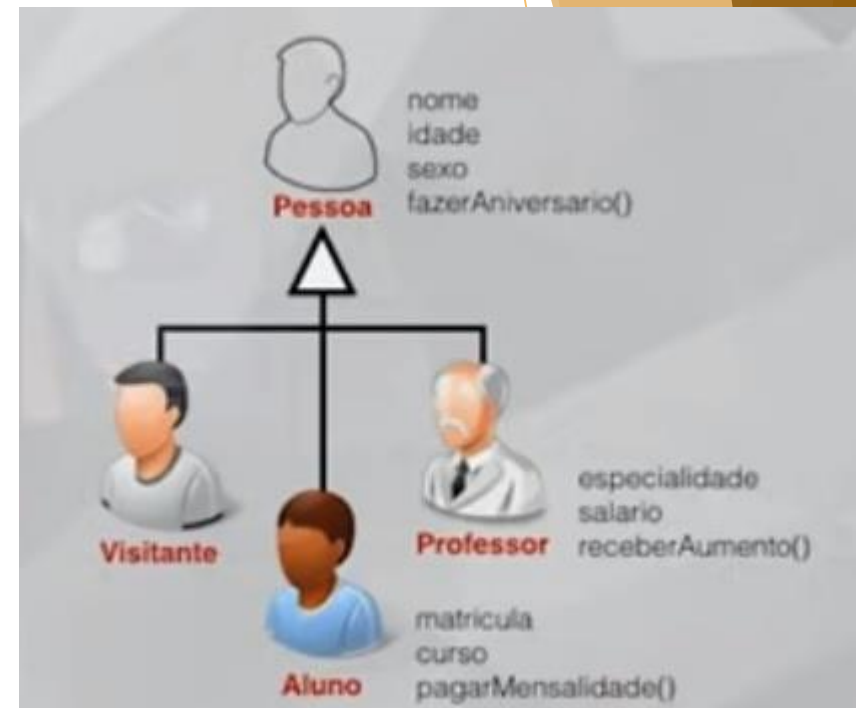
► Herança para Diferença.

É o tipo de herança mais completa que existe. Iremos abordar em vários momentos do curso.

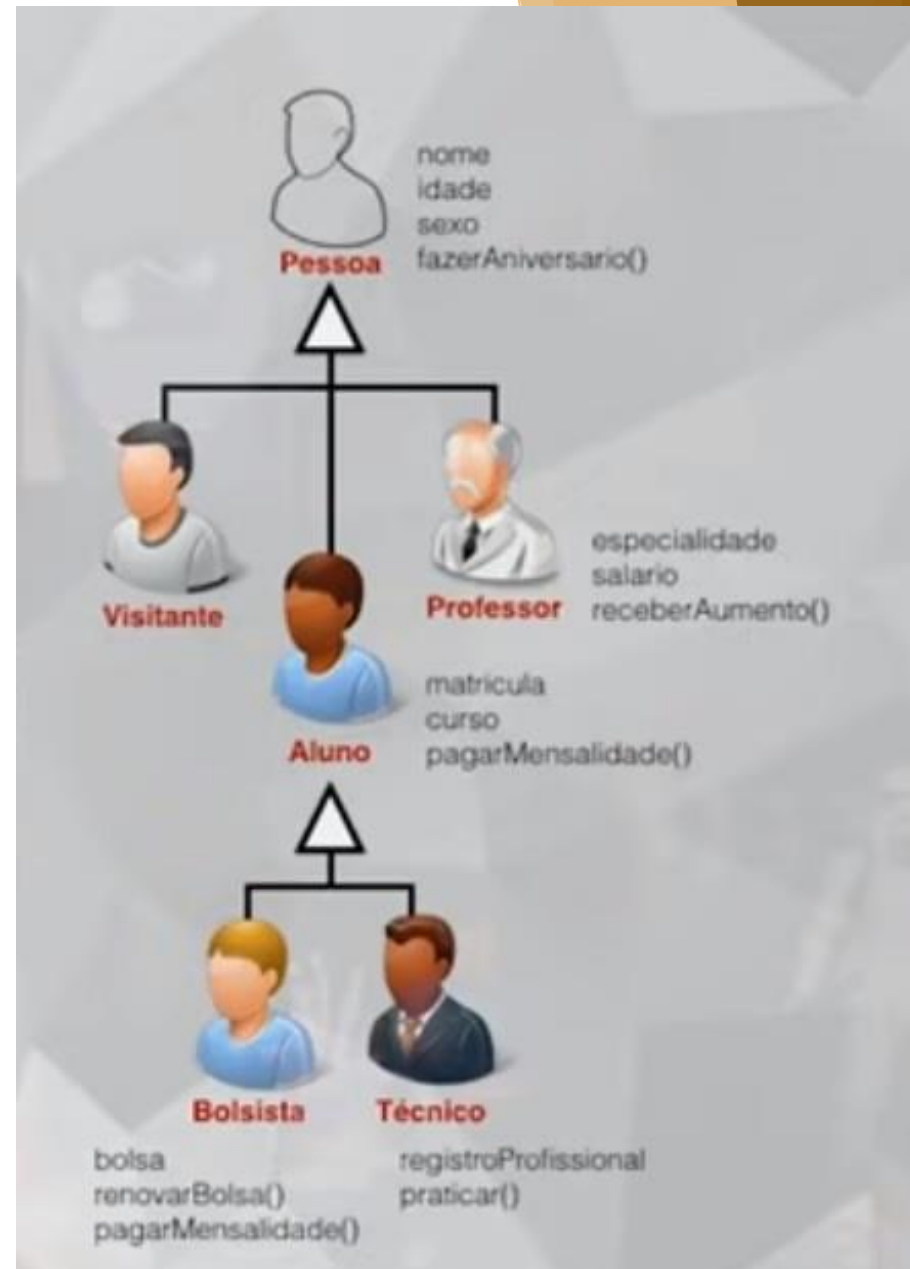
OBS: Neste curso definimos dois tipos de heranças. No entanto, isso não pode ser considerada uma solução definitiva. Você pode encontrar em outros cursos ou livros, uma divisão de tipos de heranças maior.

Vamos
analisar
um exemplo?

- ▶ A subClasse “Visitante”, “Professor” e “Aluno” herdam da superclasse “Pessoa” atributos e métodos.
- ▶ A subclasse “Visitante” é o exemplo de tipo de “herança de implementação”.
- ▶ Já as demais subclasses “Aluno” e “Professor” são denominadas “Herança para Diferença”



- ▶ Veja que agora, “Aluno” pode ser uma subclasse ou superClasse. Depende do ponto de vista.
- ▶ As classe “Bolsista” e “Tecnico” herdam do “Aluno” e “Pessoa” todos os atributos e métodos.
- ▶ No “Bolsista” é implementado novamente o método “pagarMensalidade” porque ele terá uma ação diferente. Portanto, será sobrescrito.



Pontos importantes

- ▶ **Classe Abstrata**

Não pode ser instanciada. Só pode servir como progenitora.

- ▶ **Método Abstrato**

Declarado, mas não implementado na progenitora.

- ▶ **Classe Final**

Não pode ser herdada por outra classe. Obrigatoriamente é uma folha.

- ▶ **Método Final**

Não pode ser sobrescrito pelas suas subclasses. Obrigatoriamente herdado.

Como
apresentar
o código final?

- ▶ Veja que pessoa é uma classe abstrata. É uma superClasse.
- ▶ As classes “Visitante”, “Aluno” estendem para a classe “Pessoa”.
- ▶ A classe “Bolsista” estende para classe “Aluno”.
- ▶ O método “pagarMensalidade” é sobreposto. @sobrepôr

```
classe abstrata Pessoa
    privado nome: Caractere
    privado idade: Inteiro
    privado sexo: Caractere
    publico metodo final fazerAniv()
        (...)
    fimMetodo
FimClasse

classe Visitante estende Pessoa
    (...)
FimClasse

classe Aluno estende Pessoa
    privado matricula: Inteiro
    privado curso: Caractere
    publico metodo PagarMensalidade()
        (...)
    fimMetodo
FimClasse

classe Bolsista estende Aluno
    privado bolsa: Inteiro
    publico metodo RenovarBolsa()
        (...)
    fimMetodo
    @Sobrepôr
    publico metodo PagarMensalidade()
        (...)
    fimMetodo
FimClasse
```

- ▶ A classe “Pessoa” é abstrata. Portanto, ela não pode instanciar um objeto.
- ▶ A primeira linha do programa principal estaria errado.
- ▶ @sobrepôr é um tipo de polimorfismo.

```
classe abstrata Pessoa
    privado nome: Caractere
    privado idade: Inteiro
    privado sexo: Caractere
    publico metodo final fazerAniv()
        (...)
    fimMetodo
FimClasse

classe Visitante estende Pessoa
    (...)
FimClasse

classe Aluno estende Pessoa
    privado matricula: Inteiro
    privado curso: Caractere
    publico metodo PagarMensalidade()
        (...)
    fimMetodo
FimClasse

classe Bolsista estende Aluno
    privado bolsa: Inteiro
    publico metodo RenovarBolsa()
        (...)
    fimMetodo
    @Sobrepôr
    publico metodo PagarMensalidade()
        (...)
    fimMetodo
FimClasse

// Programa Principal
p1 = novo Pessoa()
v1 = novo Visitante()
a1 = novo Aluno()
```


Atividade

- ▶ 1) Crie o código do exemplo apresentado na aula.
- ▶ 2) Faça uma classe “Animal” com um método “falar” e acrescente os atributos “Nome” e “Idade”. Faça as classes Homem, Cão e Gato, herdando de animal, redefinindo o método “falar” para retornar “Oi”, “Au au” e “Miau”, respectivamente. Instancie Homens, Cães e Gatos. Ao instanciar apresente o objeto e faça ele “Falar”.

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!!