



ACH2002

Herança em Programação OO

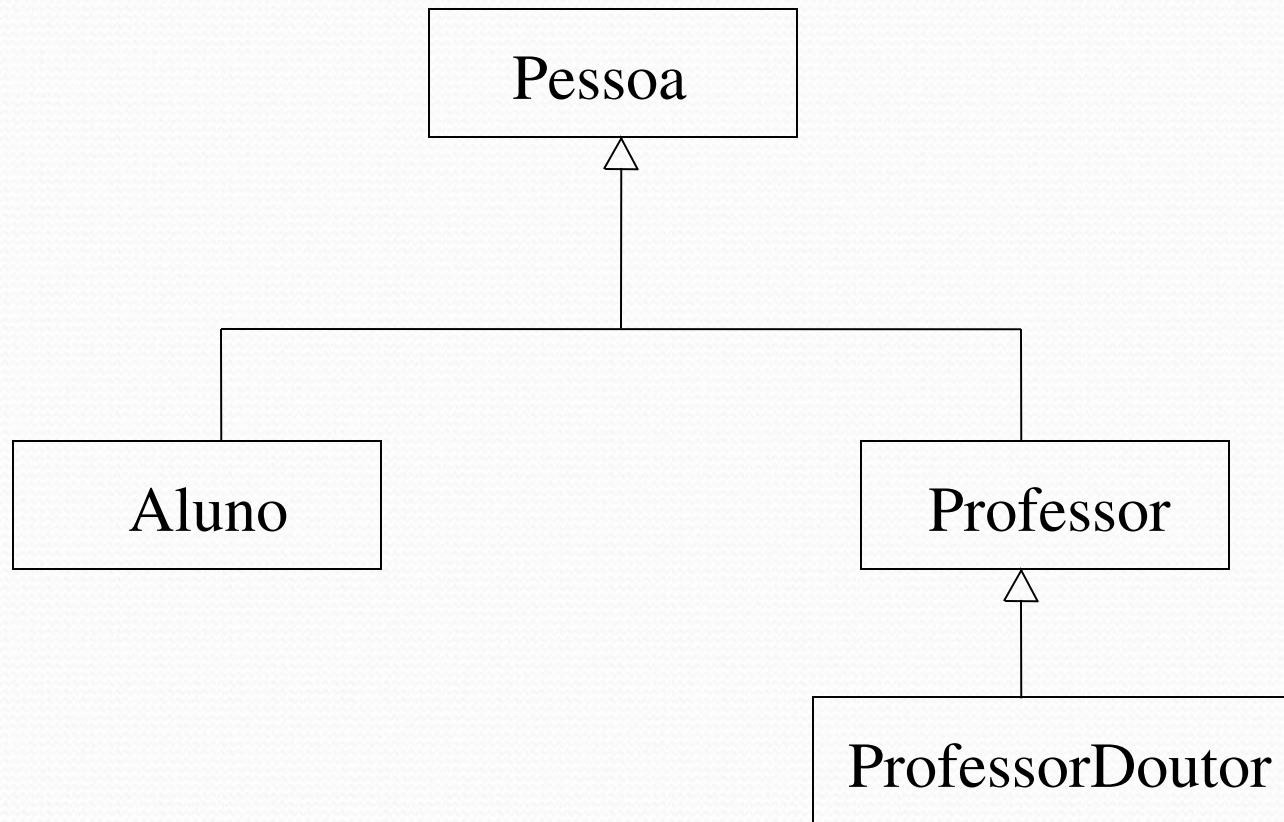
Delano Medeiros Beder

EACH – USP

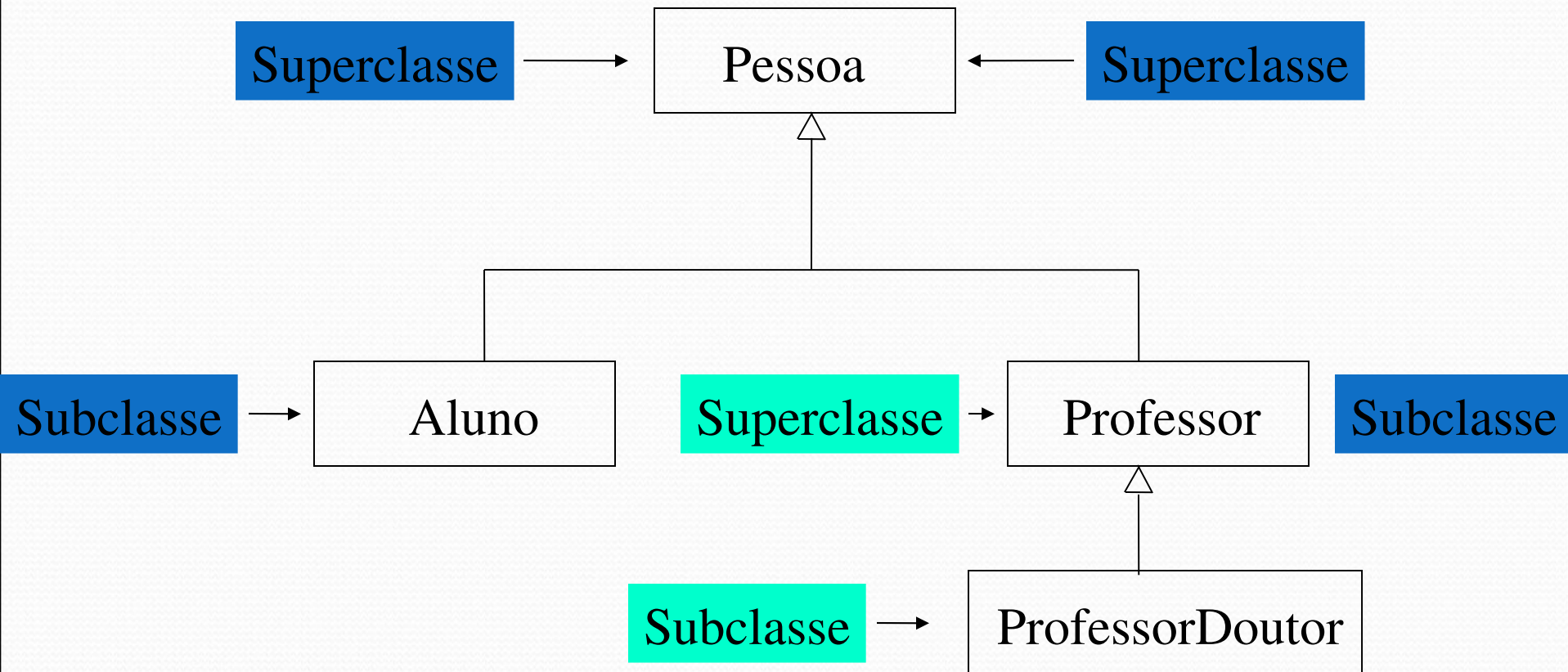
Herança

- Técnica que permite reutilizar código comum em várias classes.
- Objetivo:
 - Aumentar a produtividade, fazendo com que os programadores se concentrem em coisas realmente novas.
- Como reutilizar código?
 - Criando classes genéricas.

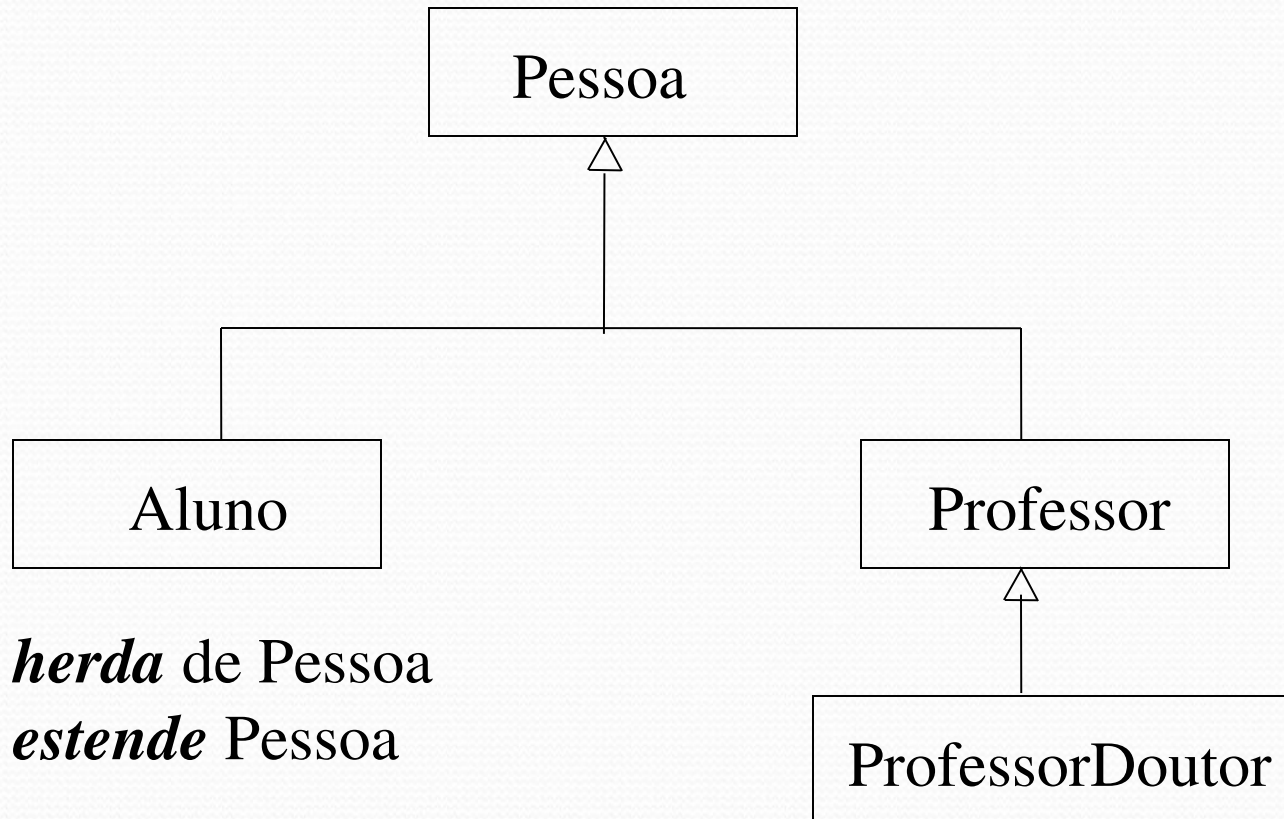
Herança



Herança



Herança

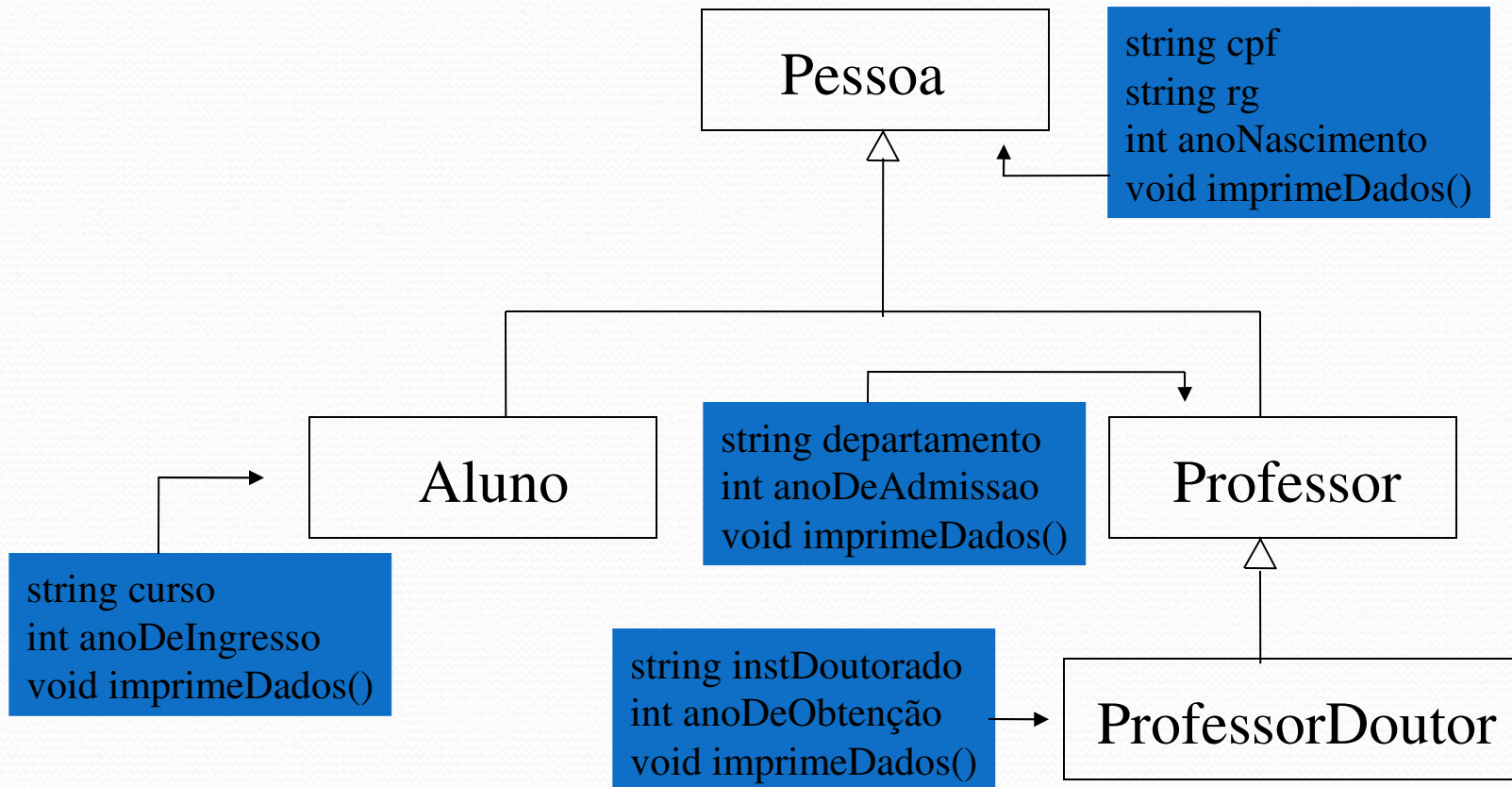


Aluno *herda* de Pessoa

Aluno *estende* Pessoa

ProfessorDoutor

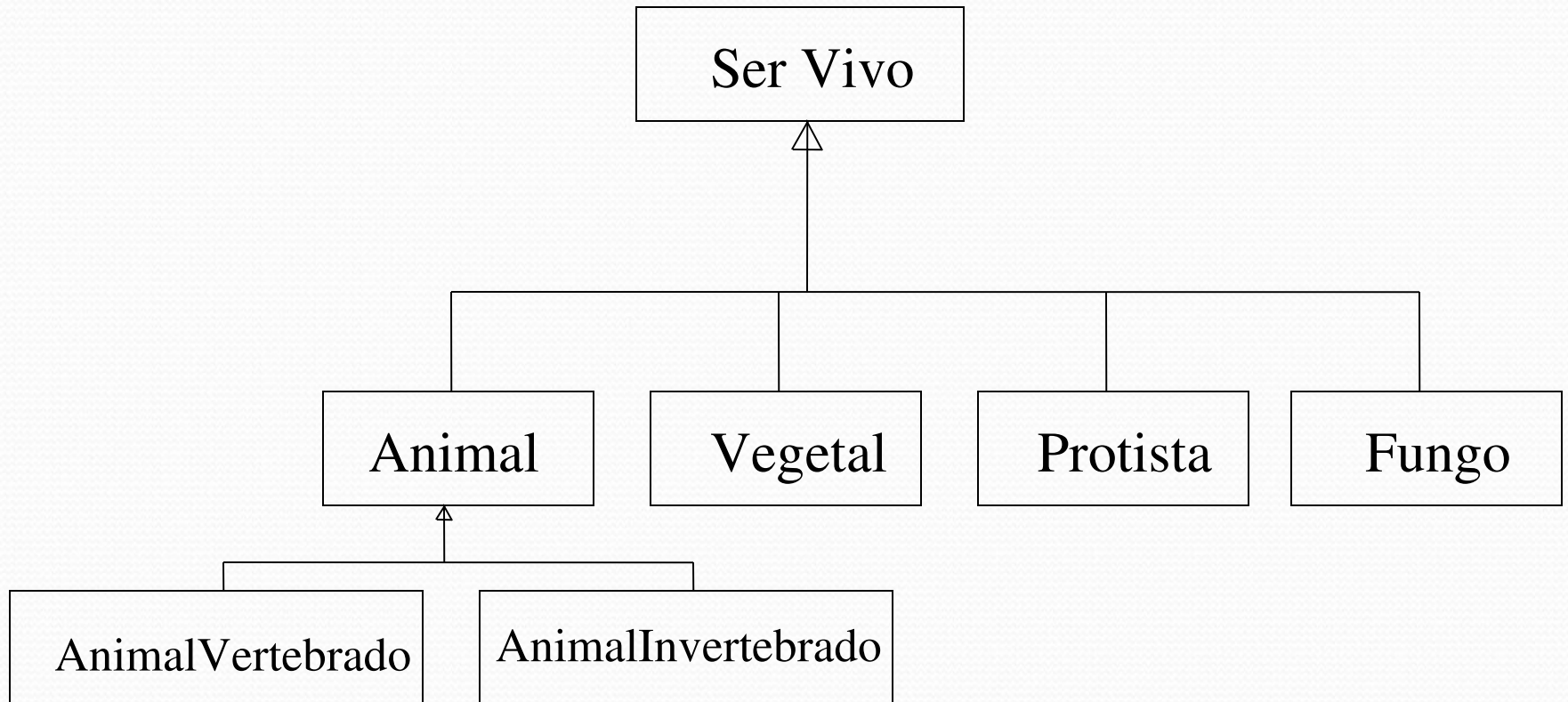
Herança



Herança -- propriedades

- A *subclasse* **herda** os atributos e métodos da *superclasse*.
- No entanto, pode **criar** atributos e métodos próprios.
- Pode ainda **especializar** os métodos herdados, criando versões próprias.
- Exemplo página 122 do livro.

Hierarquia de classes



Relacionamento “é um”

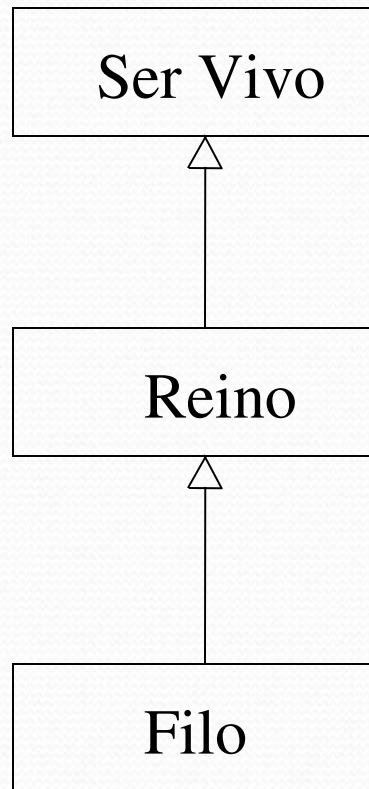
- A relação superclasse-subclasse é uma relação “é um”.
- Ou seja, se **B** é uma subclasse de **A** então todo **B** é um **A**.
- Exemplo:
 - Todo **Animal** é um **Ser Vivo**,
 - Mas nem todo **Ser Vivo** é um **Animal**.

Hierarquia de classes

Hierarquia
Errada

Reino não é
um Ser vivo.

Um Filo não é
um reino



Interfaces

- Uma interface é uma coleção de declarações de métodos sem dados e sem corpo
 - Os métodos de uma interface são sempre vazios (ou seja, são simples assinaturas de métodos)
- Quando uma classe implementa uma interface, ela deve programar todos os métodos declarados na mesma
 - A interface impõe que a classe que a implementa tenha métodos com assinaturas específicas

Classes Abstratas

- Uma classe abstrata é aquela que contém pelo menos uma declaração de método vazia (isto é, uma declaração de método sem implementação) e definições concretas de métodos e atributos
- Da mesma forma que uma interface, uma classe abstrata não pode ser instanciada, ou seja, nenhum objeto pode ser criado a partir de uma classe abstrata
- Uma subclasse de uma classe abstrata deve prover a implementação dos métodos abstratos de sua superclasse ou será também considerada abstrata
 - Uma classe abstrata pode ser subclasse de outra classe abstrata !!

Resumo

- Uma classe pode herdar de outra seus atributos e métodos.
- Uma subclasse pode estender a funcionalidade de sua superclasse acrescentando novos atributos e métodos.
- Métodos especializados nas subclasses prevalecem sobre os métodos da superclasse.
- O método da subclasse **sobrepõe** (*overrides*) o método da superclasse.

Bibliografia

- Camarão, C.; Figueiredo, L. “Programação de Computadores em Java”, Livros Técnicos e Científicos Editora, 2003.
- Goodrich, M. T., Tamassia, R. Estruturas de dados e algoritmos em Java, 2a. Edição, Bookman, 2001.
- Kon, F.; Goldman, A.; Silva, P.J.S. “Introdução à Ciência de Computação com Java e Orientado a Objetos”, IME - USP, 2004.