



POO

Herança

POO – Herança

- A herança é um princípio da POO que permite a criação de novas classes a partir de outras previamente criadas.
- Essas novas classes são chamadas de subclasses, ou classes derivadas; e as classes já existentes, que deram origem às subclasses, são chamadas de superclasses, ou classes base.
- Uma subclasse herda métodos e atributos de sua superclasse; apesar disso, pode escrevê-los novamente para uma forma mais específica de representar o comportamento do método herdado.

POO - Herança

C Aluno		
f	nome	String
f	endereco	String
f	telefone	String
f	cpf	String
f	cursos	String[]
f	notas	String[][]
m	calcularMedia()	double
m	verificarAprovado()	double

C Professor		
f	nome	String
f	endereco	String
f	telefone	String
f	cpf	String
f	departamento	String
f	nomeCurso	String
f	salario	double
m	calcularSalarioLiquido()	double

POO – Herança

- Analizando professores e alunos, vemos que todos podem (devem) ter nome, endereço, telefone e cpf; portanto, nada mais justo que criar subclasses de Pessoa para representá-los.

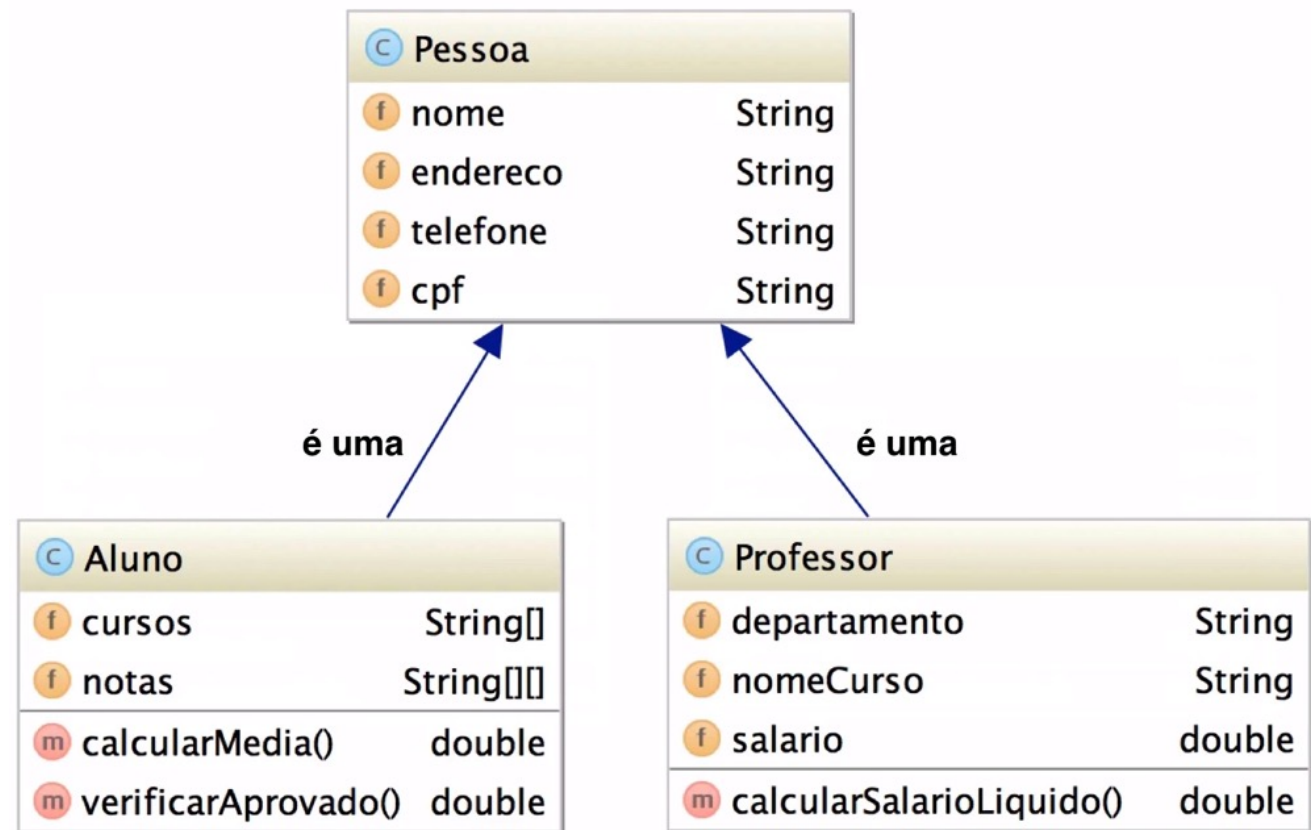
Aluno	
f nome	String
f endereco	String
f telefone	String
f cpf	String
f cursos	String[]
f notas	String[][]
m calcularMedia()	double
m verificarAprovado()	double

Professor	
f nome	String
f endereco	String
f telefone	String
f cpf	String
f departamento	String
f nomeCurso	String
f salario	double
m calcularSalarioLiquido()	double

POO – Herança

- Porém, existem algumas características que os alunos tem que um professor não tem: nota, por exemplo.
- Também existem características que os professores tem que os alunos não tem: departamento, salário, etc.
- Para resolver esse problema, e deixar a aplicação MUITO MAIS ORGANIZADA, vamos fazer duas classes: "Aluno" e "Professor".
- Cada uma dessas irá herdar a classe "Pessoa", pois também são um tipo de pessoa.
- Dizemos que "Pessoa" é a superclasse, "Aluno" e "Professor" são subclasses.

POO - Herança



POO – Herança

- Em Java, criamos classes derivadas utilizando a palavra ***extends***, seguida do nome da superclasse.
- Classes herdam características de outras classes
- Podemos adicionar mais atributos a classe
- No exemplo, professores e alunos são pessoas. Crie a classe “Pessoa”.
- O que essas pessoas tem em comum?
Nome, telefone, endereço, cpf
- Todos os objetos da classe “Pessoa” tem essas características.

POO – Herança

- Para saber quando usar, e detectar o uso de herança, use a relação 'é um'.
No exemplo anterior:
“Aluno” é uma “Pessoa”, e “Professor” é uma “Pessoa”.

POO – Comando *super*

- A palavra *super* representa uma chamada de método ou acesso a um atributo da superclasse, por isso tem esse nome.
- Devemos saber que todos os métodos e atributos públicos e protegidos da superclasse serão herdados, caso utilize-se o *extends* Classe na definição de uma nova classe derivada. Portanto, os métodos e atributos privados (*private*) não serão herdados, e não teremos acesso a eles nem com a utilização da palavra especial *super*.

POO – Comando *super*

- Apesar de a classe derivada herdar os comportamentos públicos da classe base, nada (ou quase nada) impede que implementemos de novo os métodos e atributos que quisermos na subclasse. Podemos sobrescrever métodos das superclasses, criando assim um novo comportamento para funções específicas.
- Se usarmos o comando ***final*** na declaração de métodos e atributos da super classe é proibido reescrevê-lo na subclasse.

POO - Herança - Exercício

Super Classe Pessoa - Atributos: Nome, endereço telefone, cpf

Métodos:

Construtor: carregar informações pessoais

tirarCopia: quando a pessoa por Aluno, custo de R\$ 0,10 por cópia; se for professor custo de R\$ 0,05 por cópia

Classe Aluno - Atributos: curso, nota1,nota2,nota3

Métodos:

Construtor: utilizar comando super para carregar dados pessoais do aluno

acessores e modificadores

calcularMedia: somas as 3 notas e dividir por 3

verificarAprovado: média ≥ 7.0 : aluno aprovado

Classe Professor – Atributos: curso, salario

Métodos:

Construtor: utilizar comando super para carregar dados pessoais do professor

acessores e modificadores

calcularSalarioLiquido: salario bruto - inss - IR

