blue

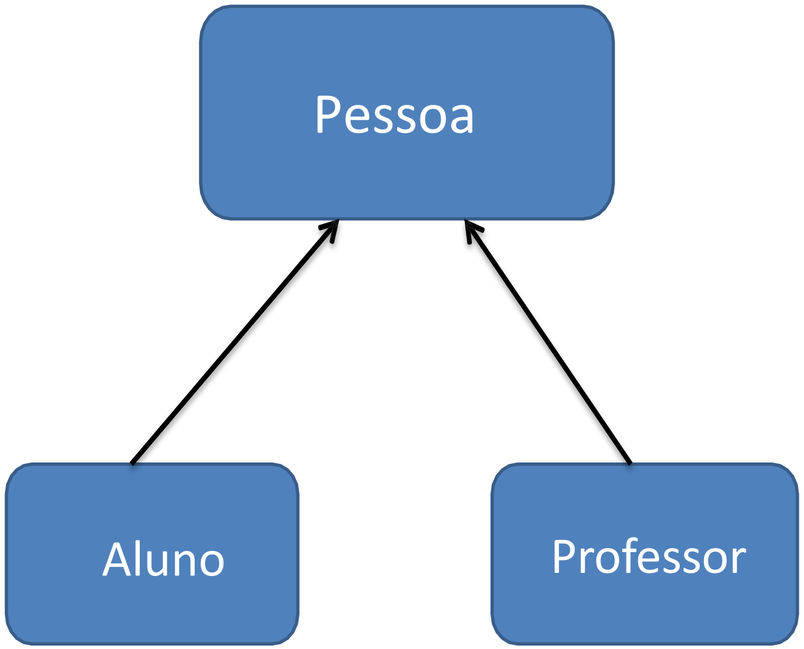
Orientação a Objetos (OO)

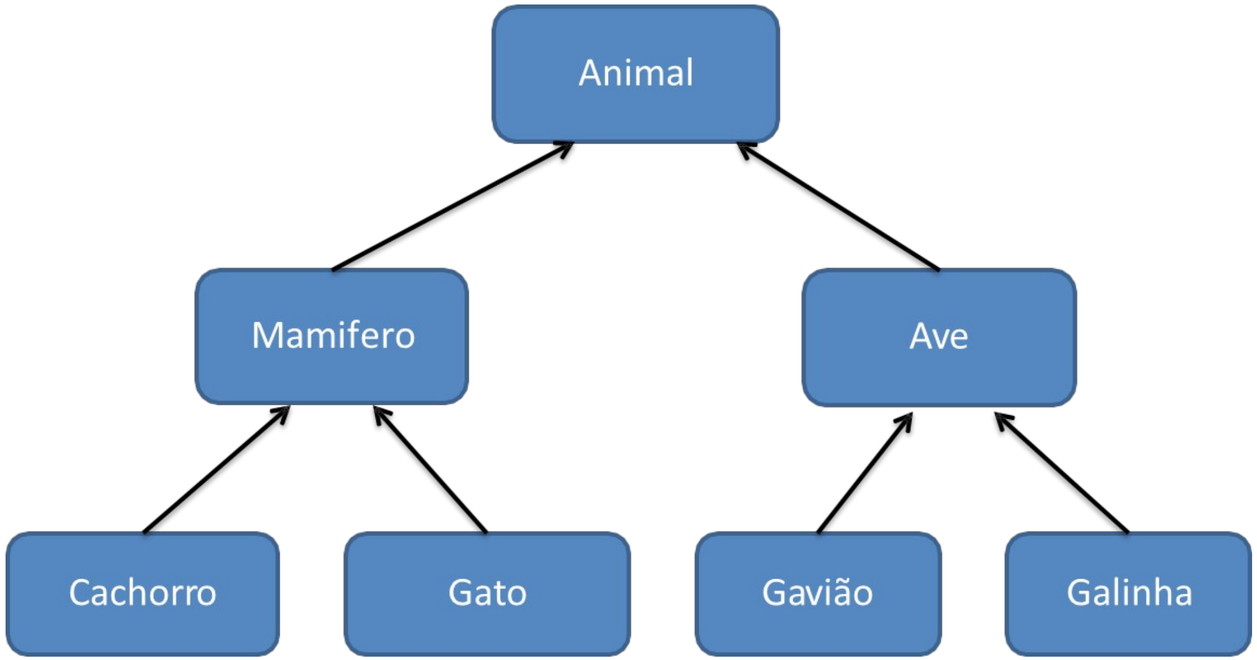
Herança / Polimorfismo

Prof. Gustavo Molina

< thefutureisblue.me />

Herança

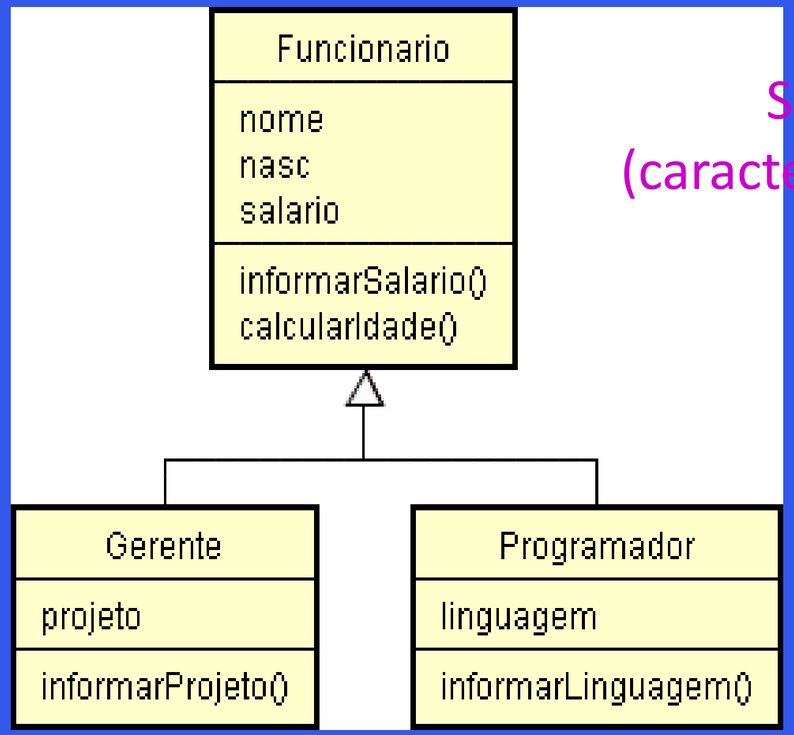

Pessoa
Aluno
Professor


Animal
Mamifero
Ave
Cachorro
Gato
Gavião
Galinha

Herança- Definições

- Uma classe pode ser **definida** a partir de outra já existente.
- Abstrai classes genéricas **(superclasse),** a partir de classes com propriedades (atributos e operações) **semelhantes.**
- As **subclasses herdam** todas as **propriedades** de sua **superclasse**, porém apresentam também suas próprias propriedades.

Herança



Superclasse
(características comuns)

Subclasses
(características específicas)

Exemplo – Prático Herança

Vamos criar um programa que criará 2 listas: uma lista de médicos e uma lista de advogados. Médicos e Advogados são subclasses de Pessoa, ou seja, herdam suas características.

Classe Pessoa: nome, idade, cpf, telefone.

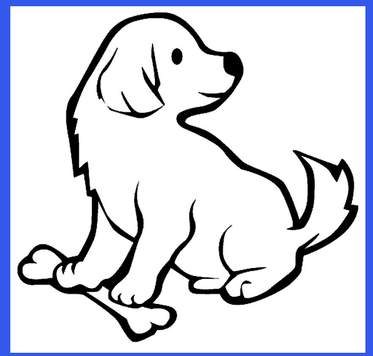
Classe Advogado: oab, nome, idade, cpf, telefone.

Classe Médico: crm, nome, idade, cpf, telefone.

Exemplo – Prático Polimorfismo

Criaremos as classes: Mamífero, Homem, Cachorro e Gato. A classe Mamífero é a superclasse e as demais são subclasses (herança). Para todas as classes criaremos o método som, que será personalizado (cada classe emitirá um som diferente).

Crie um objeto de cada classe e chame o método som. Observe o resultado.



Exercícios de Fixação

1. Crie uma classe chamada Ingresso, que possui um valor em reais e um método imprimirValor(). Crie uma classe IngressoVIP, que herda de Ingresso e possui um valor adicional. Crie um método que retorne o valor do ingresso VIP (com o adicional incluído). Crie um programa para criar as instâncias de Ingresso e IngressoVIP, mostrando a diferença de preços.

Exercícios de Fixação

2. Considere a classe ContaBancaria com os seguintes atributos: número, agência, saldo e código_tipo (código 1 = Conta Corrente, código 2 = Conta Poupança). Crie uma classe ContaImposto que herda de conta e possui um atributo percentualImposto. Esta classe também deve possuir um método calcularImposto() que subtrai do saldo, o valor do próprio saldo multiplicado pelo percentual do imposto. Crie um programa para criar as instâncias de ContaImposto e utilizar todos os métodos das 3 classes (ex.: sacar, depositar, calcularImposto).

Exercícios de Fixação

3. Crie uma classe Elevador para armazenar as informações de um elevador de prédio. A classe deve armazenar o andar atual (térreo = 0), total de andares no prédio (desconsiderando o térreo), capacidade do elevador e quantas pessoas estão presentes nele. A classe deve também disponibilizar os seguintes métodos:

Inicializar: que deve receber como parâmetros a capacidade do elevador e o total de andares no prédio (os elevadores sempre começam no térreo e vazio);

Entrar: para acrescentar uma pessoa no elevador (só deve acrescentar se ainda houver espaço);

Sair: para remover uma pessoa do elevador (só deve remover se houver alguém dentro dele);

Subir: para subir um andar (não deve subir se já estiver no último andar);

Descer: para descer um andar (não deve descer se já estiver no térreo);

Obs.: Encapsular todos os atributos da classe (criar os métodos set e get).

Exercícios de Fixação

4. Crie uma classe para representar um jogador de futebol, com os atributos nome, posição, data de nascimento, nacionalidade, altura e peso. Crie os métodos públicos necessários para sets e gets e também um método para imprimir todos os dados do jogador. Crie um método para calcular a idade do jogador e outro método para mostrar quanto tempo falta para o jogador se aposentar. Para isso, considere que os jogadores da posição de defesa se aposentam em média aos 40 anos, os jogadores de meio-campo aos 38 e os atacantes aos 35.

Por hoje é só!

Obrigado! =)

Prof. Gustavo Molina
gmolina@thefutureisblue.me

the future is blue

blue