



PARADIGMAS E LINGUAGENS DE PROGRAMAÇÃO

ENGENHARIA DA COMPUTAÇÃO – UFC/SOBRAL

Prof. Danilo Alves

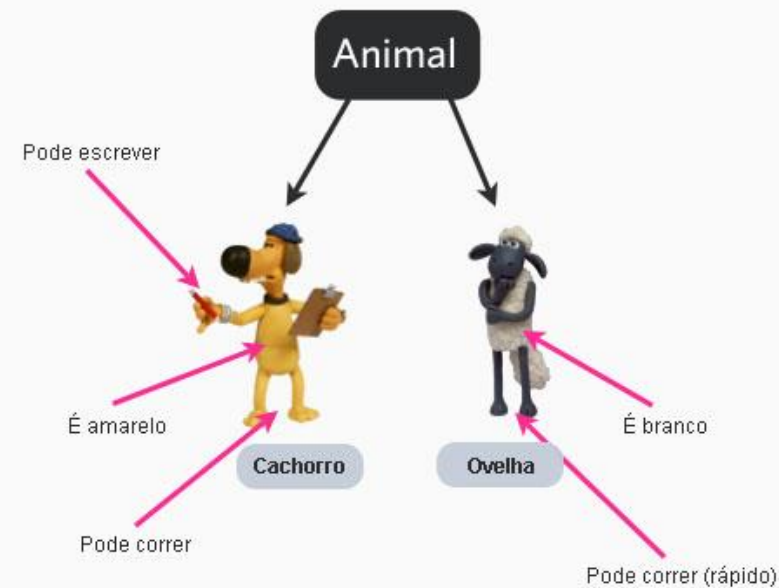
`danilo.alves@alu.ufc.br`

INTRODUÇÃO DE ORIENTAÇÃO A OBJETOS

- Colaboração de componentes individuais;
- Relacionamento do ser humano com o mundo;
- Ideia básica de aproximação do mundo real;
- Prestação de serviços;
- Ex: Biblioteca;

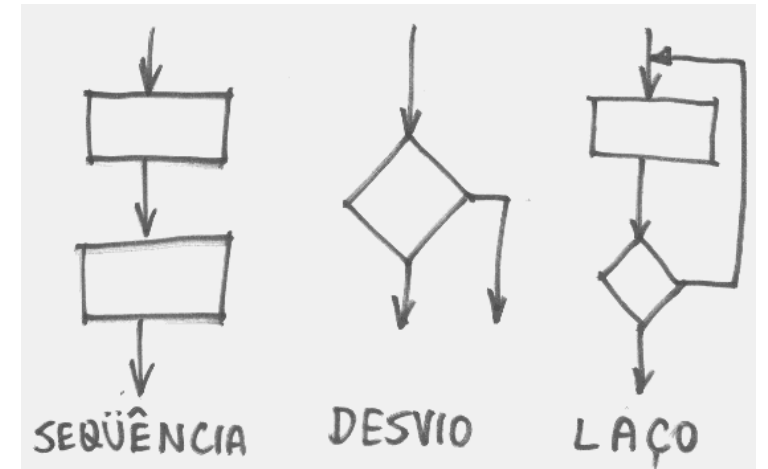
INTRODUÇÃO DE ORIENTAÇÃO A OBJETOS

- Objetos possuem responsabilidades pelos serviços;
- Analisar modelos;



HISTÓRICO

- Final da década de 60, com Simula67;
- Smaltalk, em 1970;
- C++, em 1980;
- Necessidade de modelar sistemas mais complexos;
- Anteriormente, somente com o paradigma estruturado;



IMPORTÂNCIA

- Imagine um sistema...
 - Dezenas de formulários;
 - CPFs são validados – função `validar(cpf)` deve ser chamada em cada formulário;
 - Todos os clientes são responsáveis!
 - Define-se que idade deve ser ≥ 18 . (Validação);

IMPORTÂNCIA

- Facilidade na escrita do programa;
- Facilidade de entendimento do código;
- Menor quantidade de erros (no caso geral);
- Alto índice de reuso de código;

OBJETOS

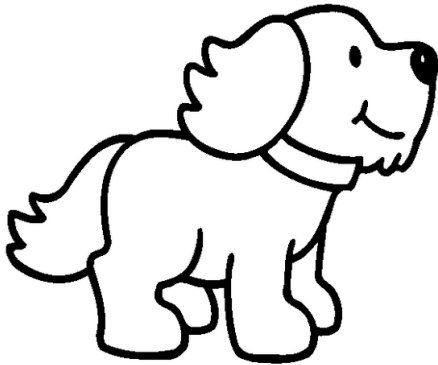
- Elementos abstratos responsáveis pelo seu estado e regras de seu estado;
- Olhando em nossa volta, encontramos vários exemplos de objetos;
- Os objetos reais possuem duas características:
 - Estado;
 - Comportamento;

Cachorros → **estado**: nome, cor, raça;
comportamento: latir, correr;

Bicicletas → **estado**: marcha atual, velocidade atual;
comportamento: trocar marcha, aplicar freios;

ATRIBUTOS E MÉTODOS

- Objetos armazenam seu estado em **atributos**;
- Objetos expõem seu comportamento através de **métodos**;



Atributos: Nome, Cor, Raça;
Métodos: Latir e Correr;

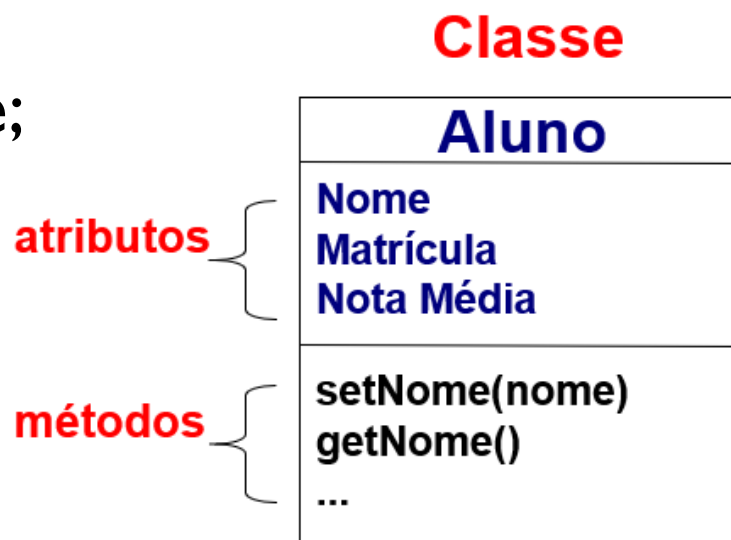
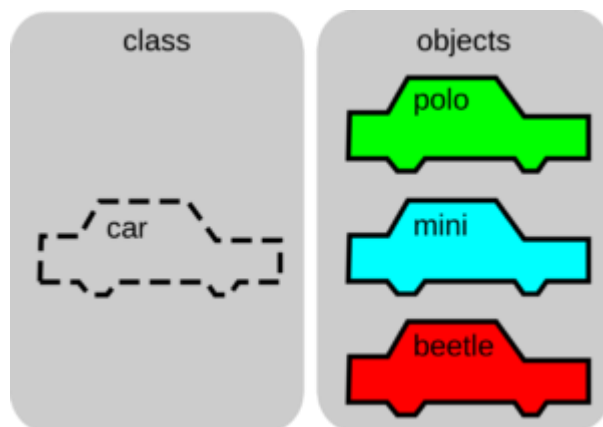
- Objetos com características semelhantes;

CLASSES

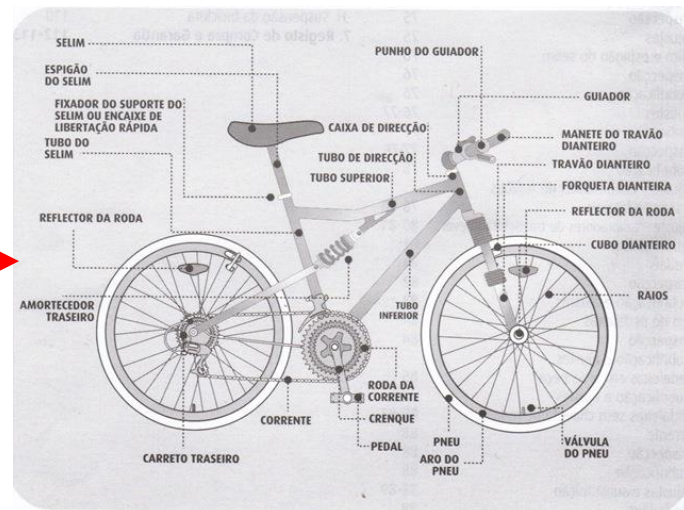


CLASSES

- Objetos são agrupados em classes;
- O projeto a partir do qual objetos individuais são criados;
- Define atributos e métodos;
- Objeto é a instância da classe;

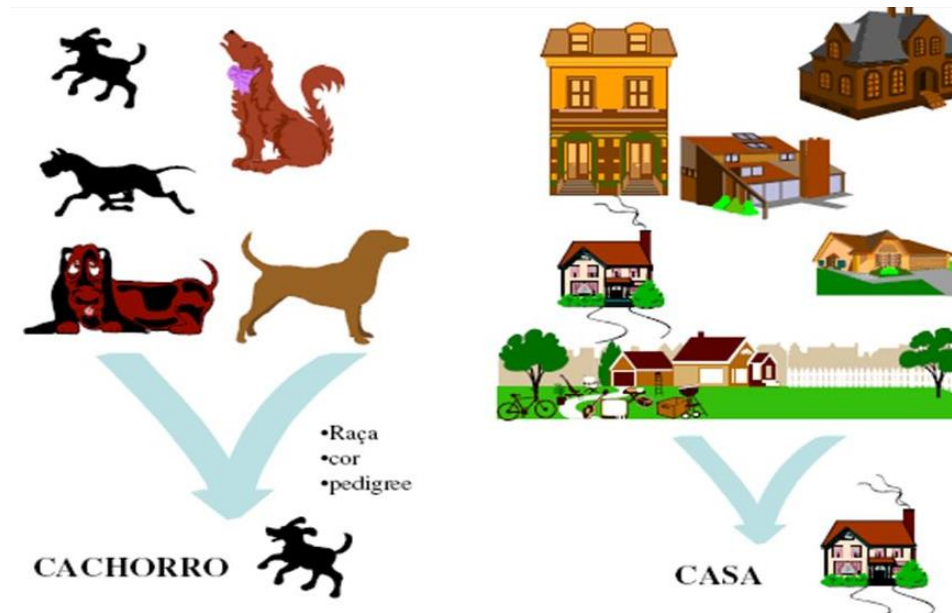


CLASSES

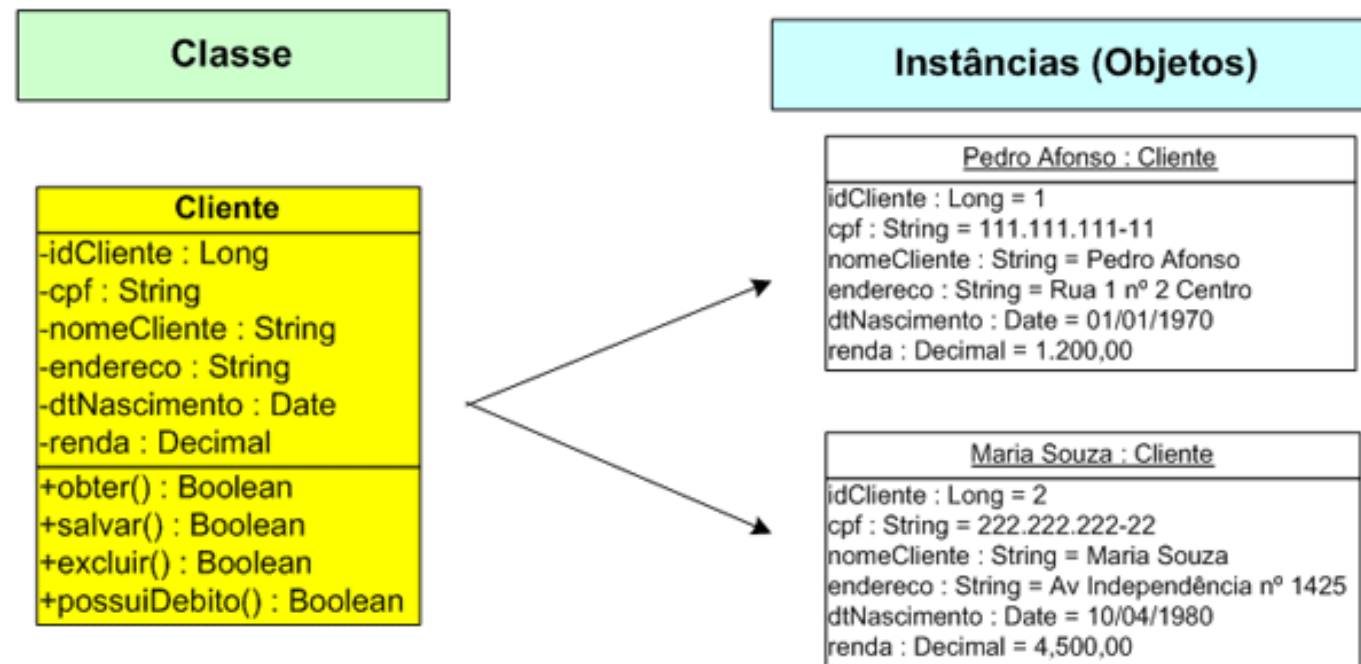


CLASSES

- Como abstrair uma classe?
- Analisando um conjunto de objetos;



CLASSES



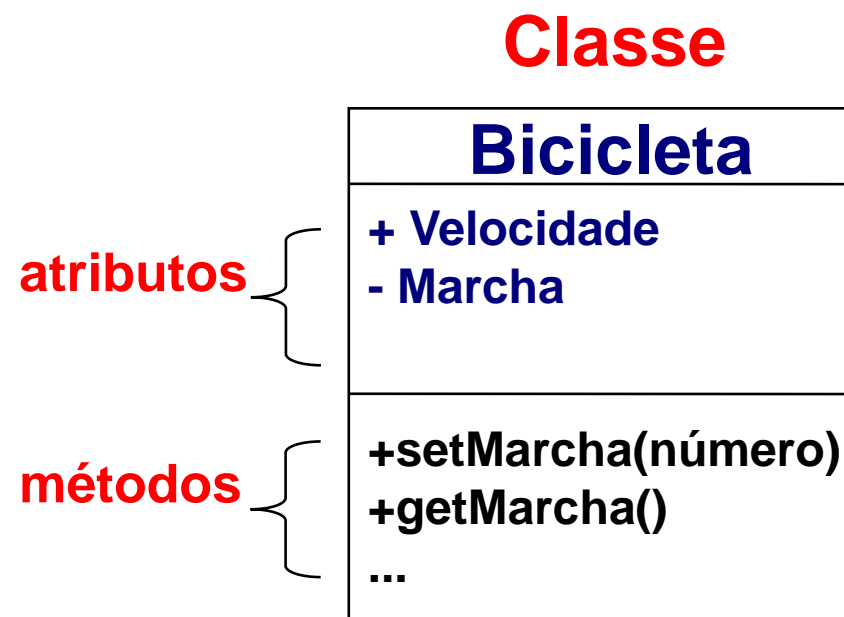
Objetos (Instâncias) da classe Cliente

ENCAPSULAMENTO

- Métodos e comunicação;
- Proteger o estado interno e modificar por métodos;
- Evitar alterações acidentais;
- Proteger informações;
- Objeto determina como o estado se modifica;
- Ex: Marcha da bicicleta;

ENCAPSULAMENTO

- Modificadores de acesso;
- Público, privado e protegido;
- Métodos *get* e *set*;



HERANÇA

- Diferentes tipos de objetos tem semelhanças entre si;

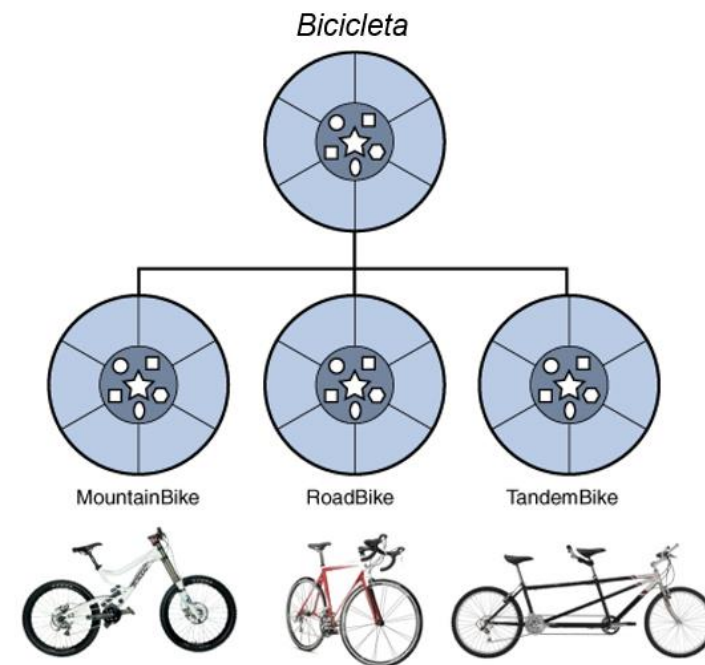
- Bicicletas *Tandem*;
- *Mountain bikes*;
- Bicicletas de corrida;

- Características comuns;
- Características diferentes;



HERANÇA

- Herdam características comuns de um modelo genérico;
- Capacidade de herdar os estados e comportamentos;
- Evitar repetir códigos;
- Implementa a diferença;
- Superclasse e subclasse;



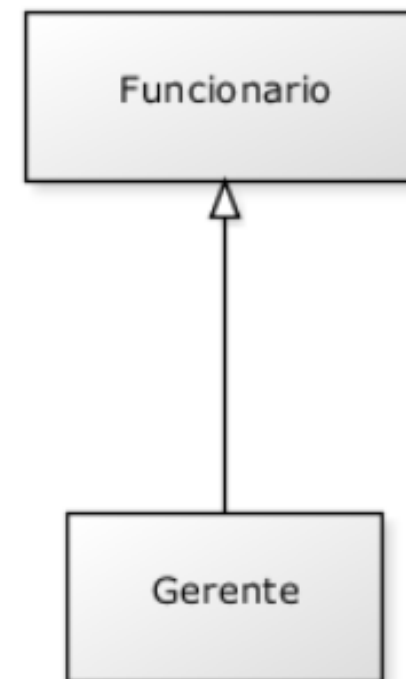
HERANÇA

- Abstraindo a relação de herança;



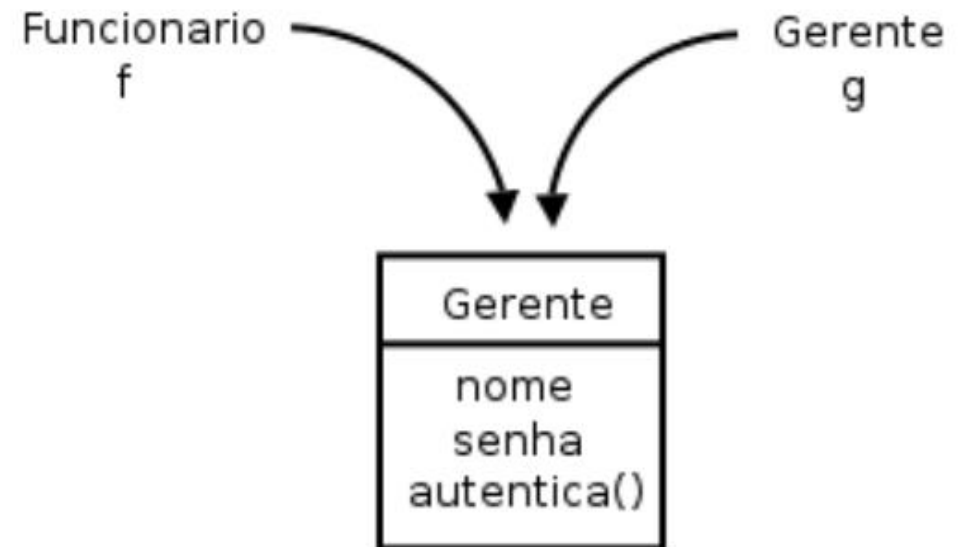
HERANÇA

- Generalização e especialização;
- Código não disponível na subclasse;
- Reescrita;
- Herança múltipla;



POLIMORFISMO

- Conceito biológico;
- Único objeto pode assumir tipos de referencia similares;
- Paramétrico (genérico);
- Sobrecarga;



RELEMBRANDO

