

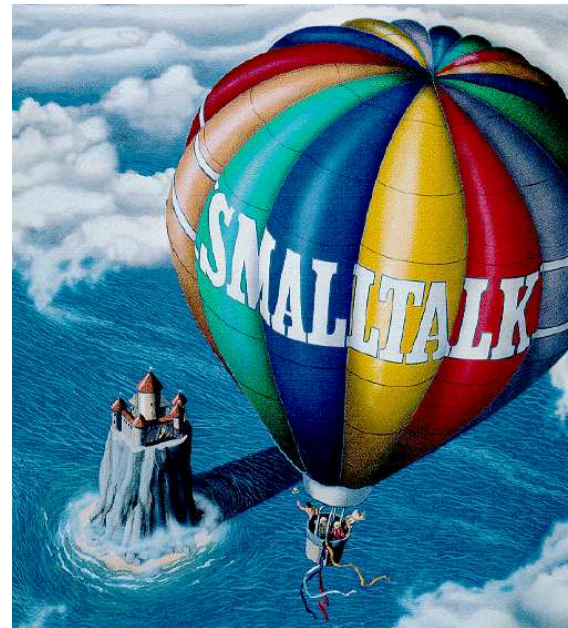
# Análise Orientada a Objetos

Conceitos de Orientação a Objeto

Prof.: MSc Eduardo de Souza Santos  
*eduardo@mestrado.ufu.br*

# Histórico

- Estudos sobre orientação a objetos iniciados na década de 70;
- Origem com a linguagem Simula-67 (alguns conceitos de O.O.);
- Smalltalk: incentivou e popularizou a O. O.



# Histórico

- Atualmente: C++, C#, Java, Python...
- Java é uma das principais linguagens, popularizando o paradigma da orientação a objetos.



# Conceitos básicos da O. O.

- Abstração;
- Classes e objetos;
- Encapsulamento;
- Herança;
- Polimorfismo.

# Abstração

→ Definição: construir um modelo para representar algum elemento do mundo real.

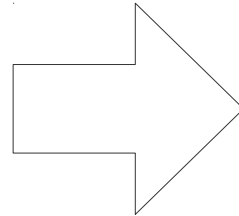
→ Uma **classe** é uma **abstração** de entidades existentes no domínio do sistema de software!

**\*\* Ressaltar aspectos essenciais do objeto;**

**\*\* Descartar aspectos desnecessários.**

# Abstração

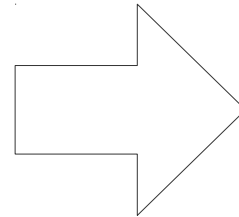
→ Exemplo: representar um carro para um sistema de estacionamento:



Carro
- cor - modelo - placa
+ chegou + saiu

# Abstração

→ Exemplo: representar um carro para um sistema de direção autônoma:



Carro
- nível do combustível - potência - temAbs
+ acelerar + frear + mudar de direção

# Objetos

→ são entidades em sistema de software que representam instâncias de entidades do mundo real;

## ***Características:***

- + Um objeto é uma abstração de uma coisa, com limites e significados bem definidos em relação ao problema considerado;
- + É geralmente identificado por um substantivo;
- + Possui estrutura e comportamento.



# Objetos

→ Exemplos:

Livro



Pessoa



Animal



# Objetos

→ Exemplos:

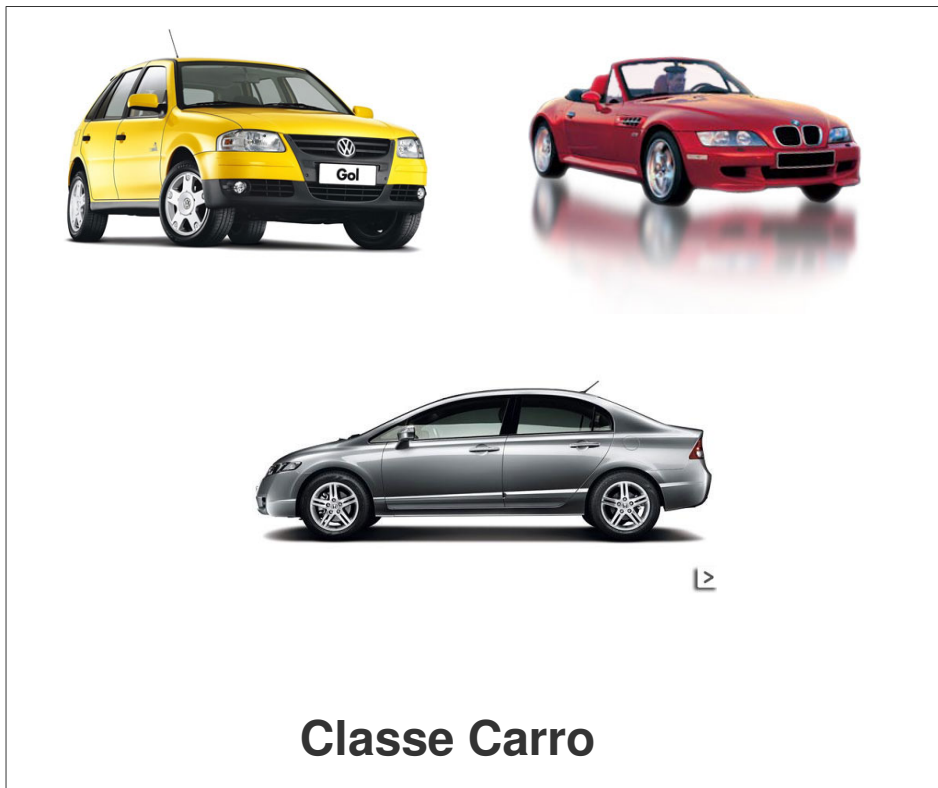
Objeto	Parte estrutural	Parte comportamental
Pessoa	Nome, idade, peso	Pode mudar de emprego, pode mudar de residência
Livro	Número de páginas, título	Pode ser lido, pode ser emprestado
Animal	Raça, cor	Pode andar, comer, brincar

# Classe

- Define os atributos e comportamentos que serão compartilhados por um tipo de objeto;
- Os objetos de certo tipo ou classificação compartilham os mesmos comportamentos e atributos;
- As classes são utilizadas para criar ou instanciar objetos;
- A definição de uma classe é arbitrária e depende da aplicação em desenvolvimento.

# Classe

→ Exemplos:



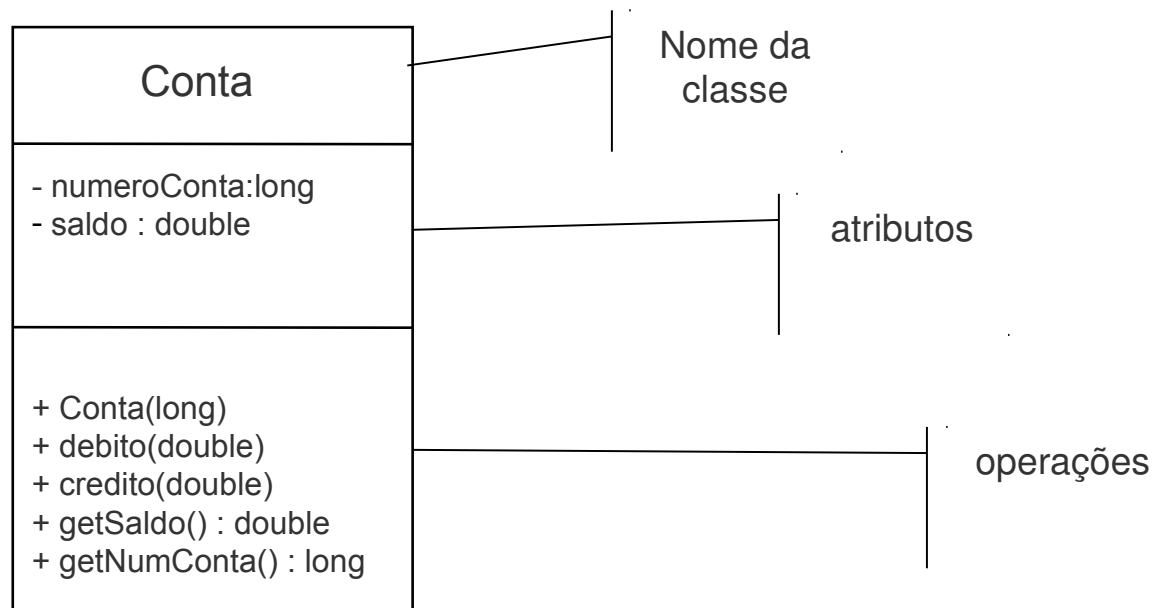
# Classe - outros conceitos

- Atributos: características da classe.
  - Ex.: cor, peso, altura
- Comportamento: diz respeito às ações que podem ser executadas por um objeto quando é passada alguma mensagem ou ocorre uma mudança de estado (o que o objeto faz)
  - Ex.: cair, voar, desligar, andar

# Classes em UML (*Unified Modeling Language*)

Uma classe é representada por um retângulo, subdividido em três áreas:

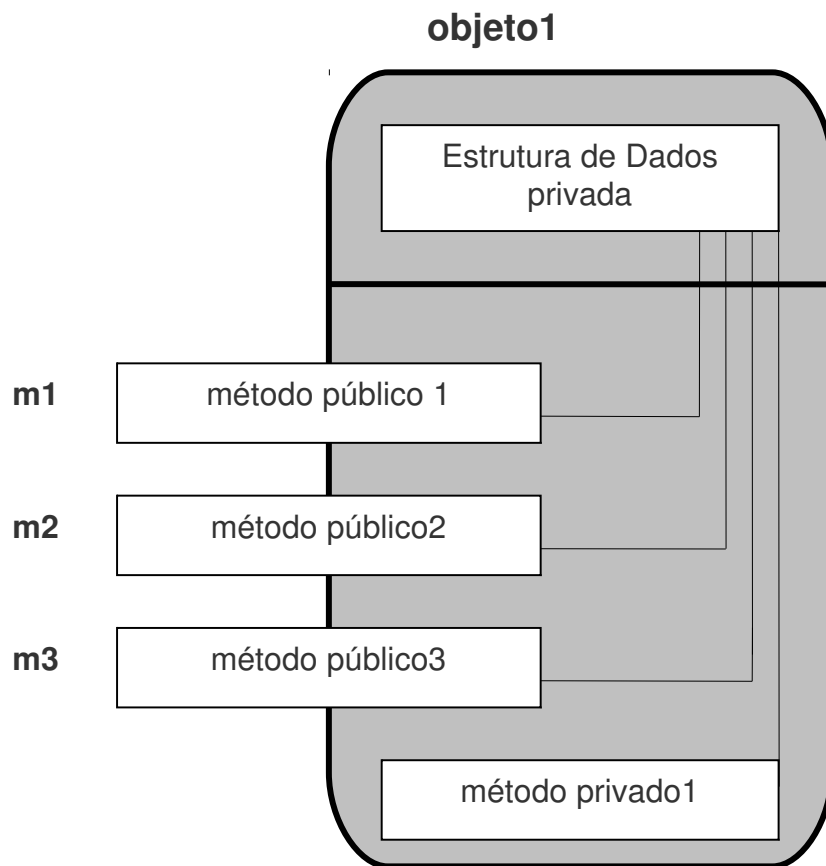
- + A primeira contém o nome da Classe;
- + A segunda contém seus atributos;
- + A terceira contém suas operações.



# Encapsulamento

- Esconde a implementação interna, permitindo o acesso através de uma especificação externa (API);
- Oculta detalhes de implementação de outras classes/objetos;
- Faz com que a utilização de uma determinada operação não dependa da implementação interna.

# Encapsulamento

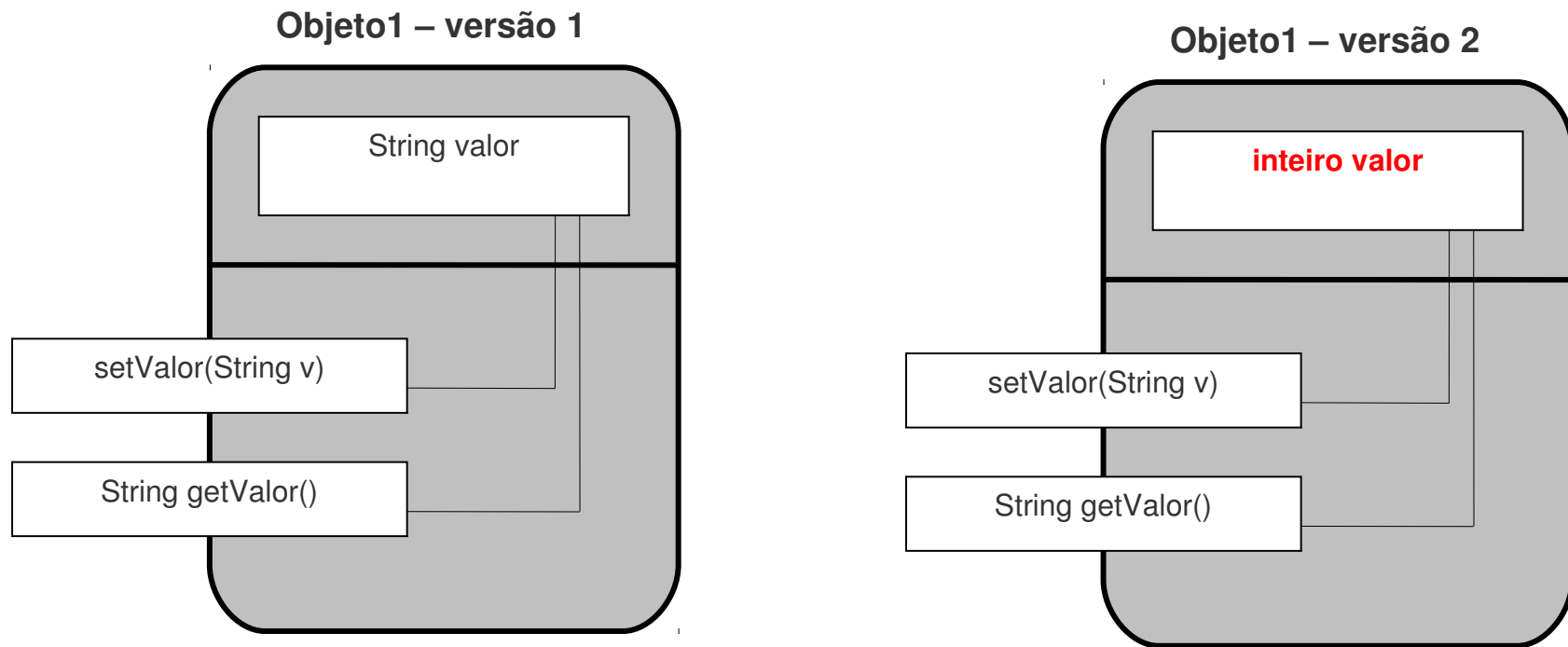


**Um objeto é uma capsula que possui:**

- Estrutura de dados privada;
- Uma API constituída por métodos públicos;
- Um conjunto de métodos privados.



# Encapsulamento - exemplo

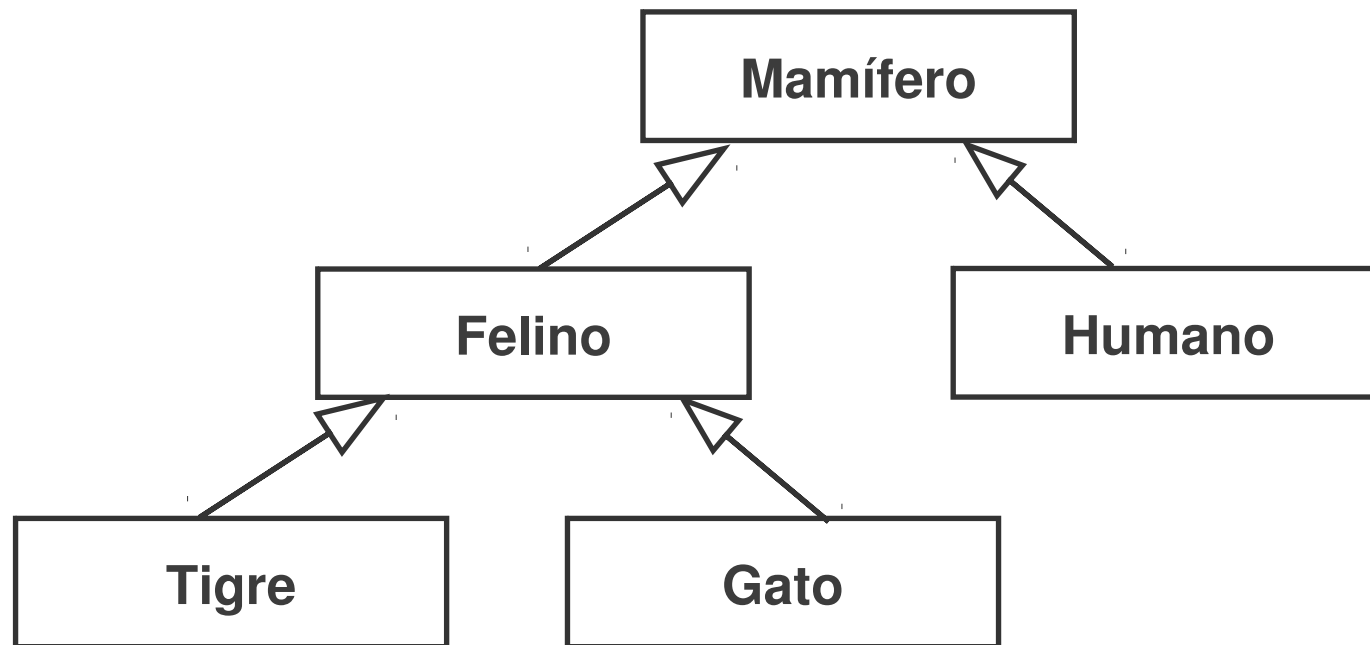


A modificação do tipo de dados privado não exige que todas as classes que se relacionavam com esse tipo de objeto sejam modificadas **desde que seja mantida a forma de acesso.**

# Herança

- Mecanismo que permite que uma classe (filha) seja baseada em uma outra classe (mãe);
- A classe FILHA herda **todos** os atributos e comportamentos da classe MÃE;
- Qual o benefício???

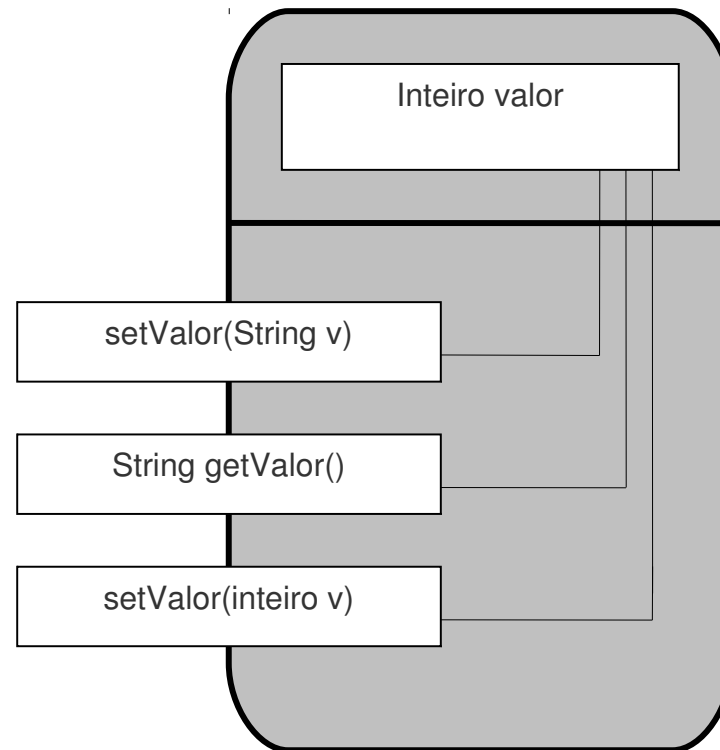
# Herança - exemplo



# Polimorfismo

- Significa ter várias formas;
- Em termos de programação, significa que o nome de um método pode representar comportamentos diferentes;
- O polimorfismo permite que um único nome expresse comportamentos diferentes.

# Polimorfismo - Exemplo



**Dois métodos “setValor” ??**