

Orientação a Objetos

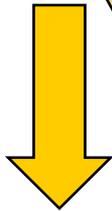
- Surgiu na tentativa de solucionar problemas existentes no desenvolvimento de Softwares **Complexos e Confiáveis** com baixo custo de desenvolvimento e manutenção
- Mundo Real é formado por objetos que se interagem
- Representar esses objetos em um software é mais natural e permanente do que representar a sua funcionalidade (decomposição funcional), pois essa é mutável

Mundo Computacional

Mundo Real



Gap Semântico



Aspectos mais importantes do mundo
real para fins de representação
no computador

DADOS

ESPAÇO DE PROBLEMAS

RESULTADOS

Objetos e Operações do mundo real

Algoritmo do mundo real

Objetos do mundo real

Mapeamento do domínio de soluções

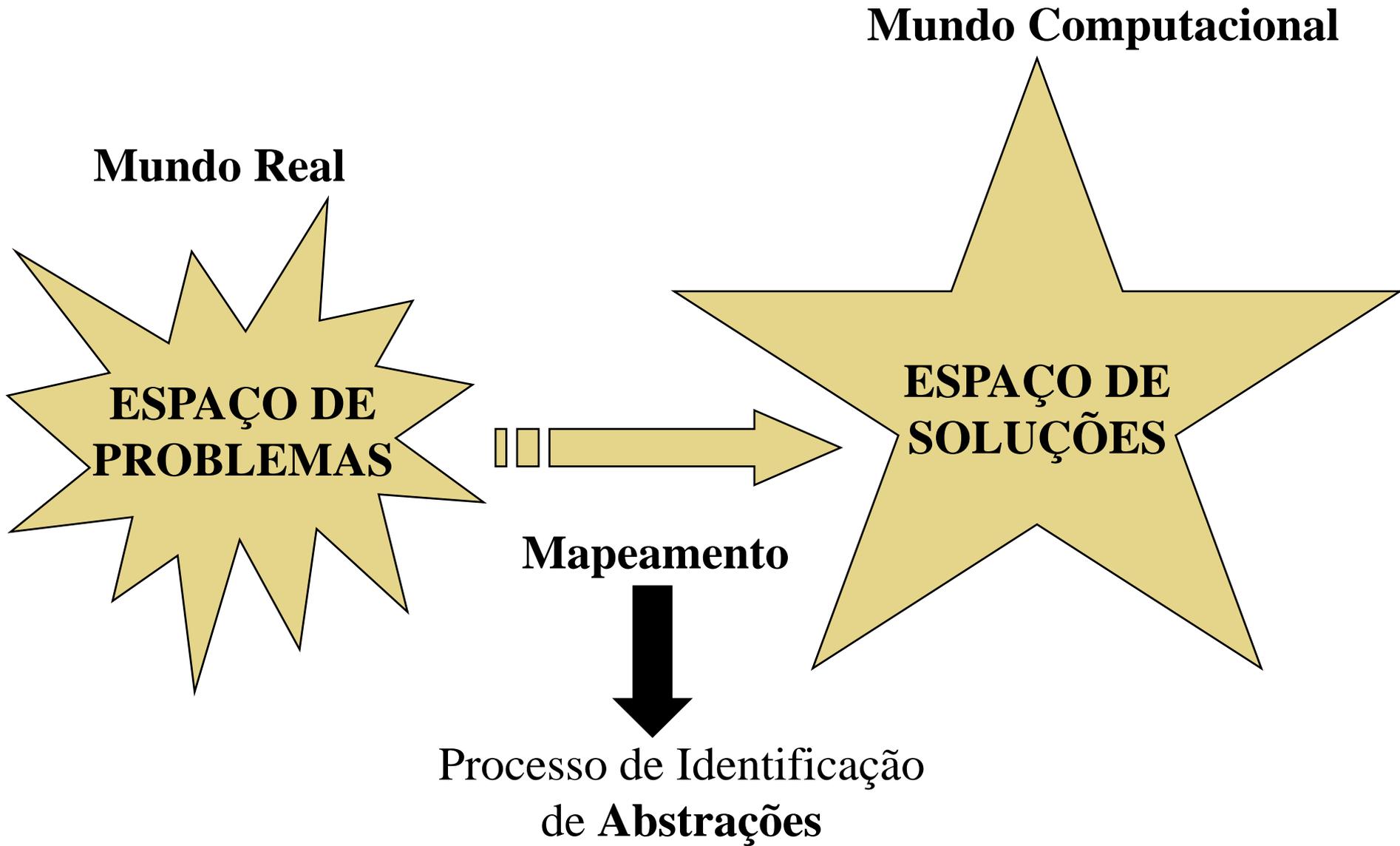
Interpretação Humana dos resultados

Objetos e Operações abstratas

ESPAÇO DE SOLUÇÕES

Dados de Saída

Algoritmo computacional

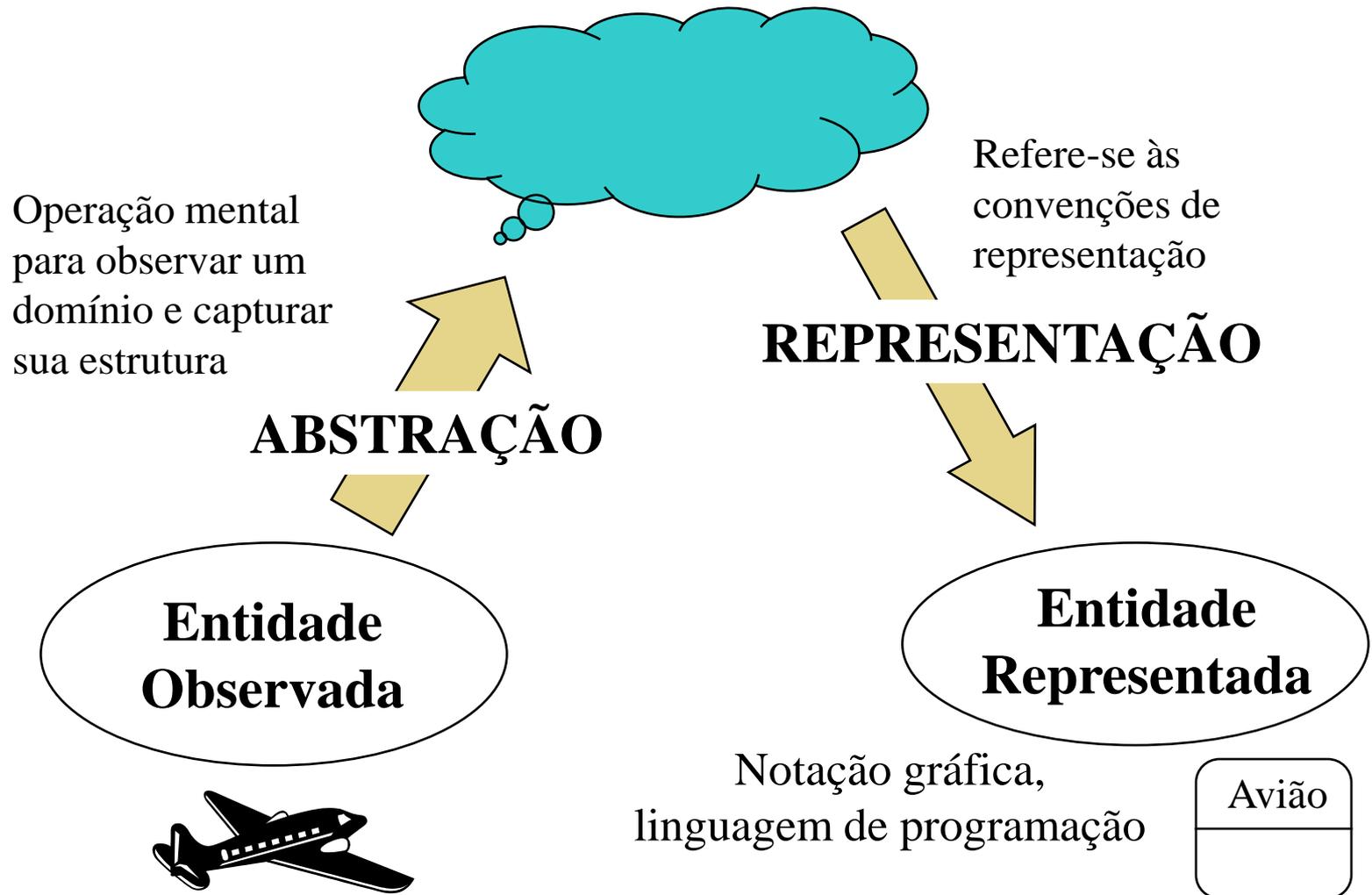


- Se essas abstrações não tiverem uma expressão direta (ou próxima) do mundo computacional, a complexidade da solução será aumentada

Modelagem Conceitual

- Tarefa mais importante de um processo de desenvolvimento de software
 - realiza-se a análise do domínio da aplicação e a modelagem das entidades e fenômenos desse domínio que o projetista considera importante, independentemente da implementação
- A tarefa de modelagem conceitual envolve dois mecanismos:
 - Abstração
 - Representação

Modelagem Conceitual



Abstração

- Mecanismo utilizado na análise de um domínio
- Através dela, o indivíduo observa a realidade e dela abstrai entidades, ações, etc. consideradas essenciais para uma aplicação, excluindo todos os aspectos julgados irrelevantes
- Exemplo:
 - **Fotografia por satélite** (imagem da realidade), despida de alguns aspectos (por exemplo, cor, movimento)
 - Da foto, pode-se abstrair um **mapa**, que elimina diversas propriedades da foto (detalhes particulares de um edifício ou praça)

Abstração

“a beleza está nos olhos de quem a vê”

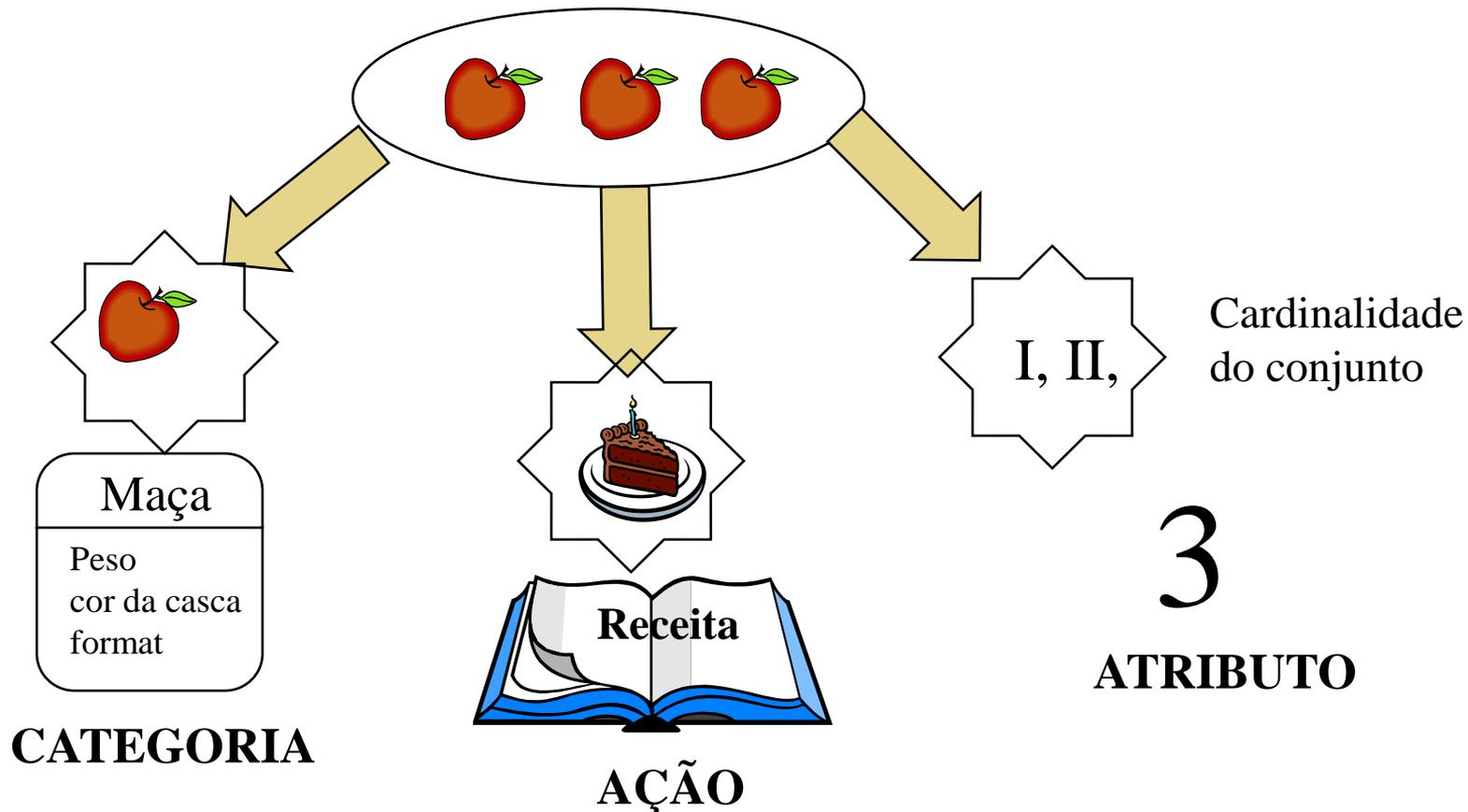
- O resultado de uma operação mental de abstração depende não tanto do fenômeno observado, mas do interesse do observador



Abstração

“a beleza está nos olhos de quem a vê”

- Diferentes abstrações a partir de um mesmo objeto do mundo real



Operações de Abstração relevantes para a modelagem conceitual

- **Classificação (Instanciação)**

- categorização dos objetos em **grupos** e/ou **classes (categoria)**, com base em algum conjunto de propriedades comuns

- **Generalização (Especialização)**

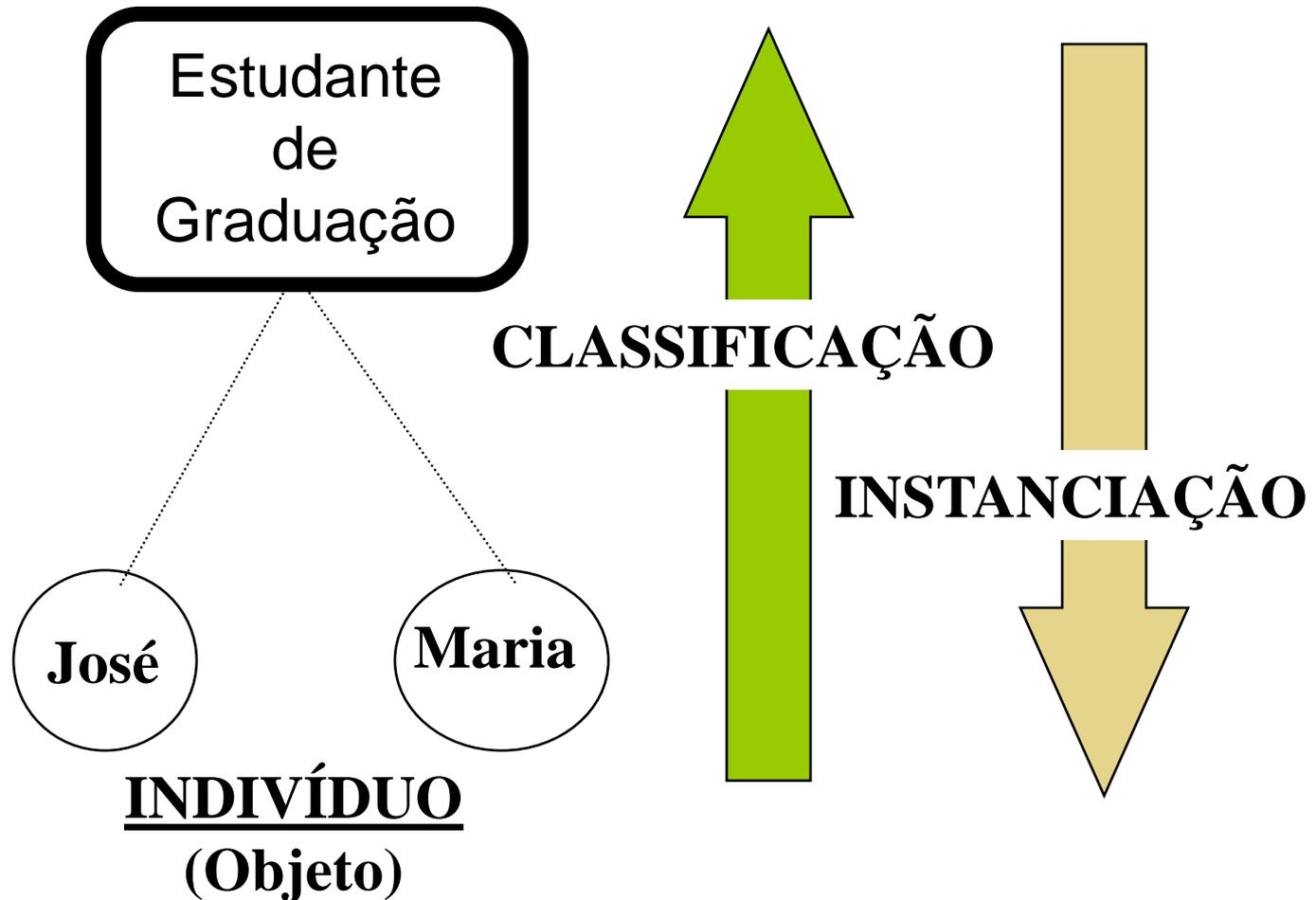
- a partir de duas categorias abstrai-se uma categoria mais genérica
- sub-categorias satisfazem todas as propriedades das categorias de que elas constituem especializações
- deve existir pelo menos uma propriedade que distingue duas categorias especializadas

Operações de Abstração relevantes para a modelagem conceitual

- **Agregação (Decomposição)**
 - composição de uma nova categoria como um agregado de categorias pré-existentes
 - instâncias de uma categoria (Estudante) são compostas por instâncias de outras categorias (Nome, Endereço)

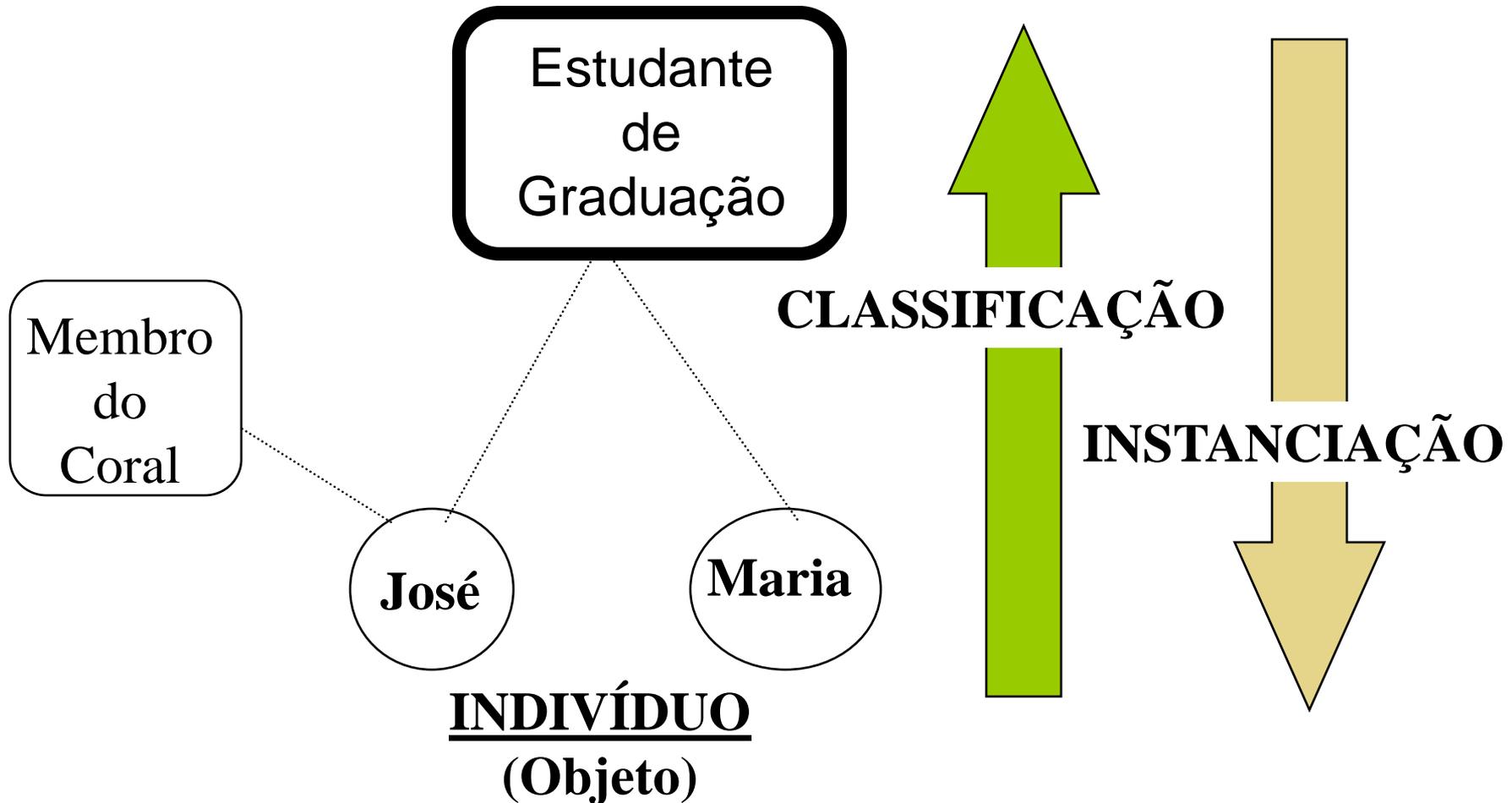
Classificação/Instanciação

CATEGORIA



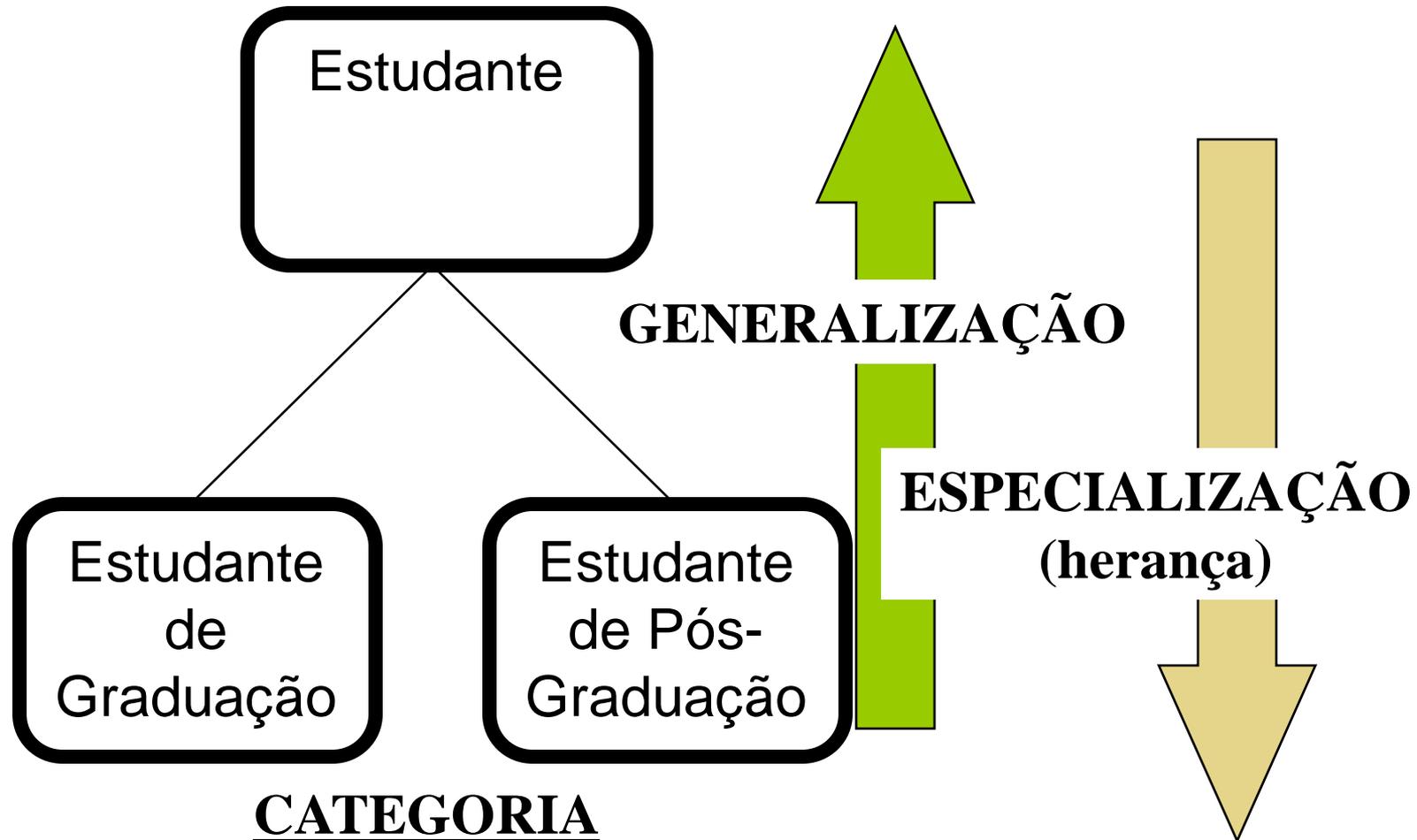
Classificação/Instanciação

CATEGORIA



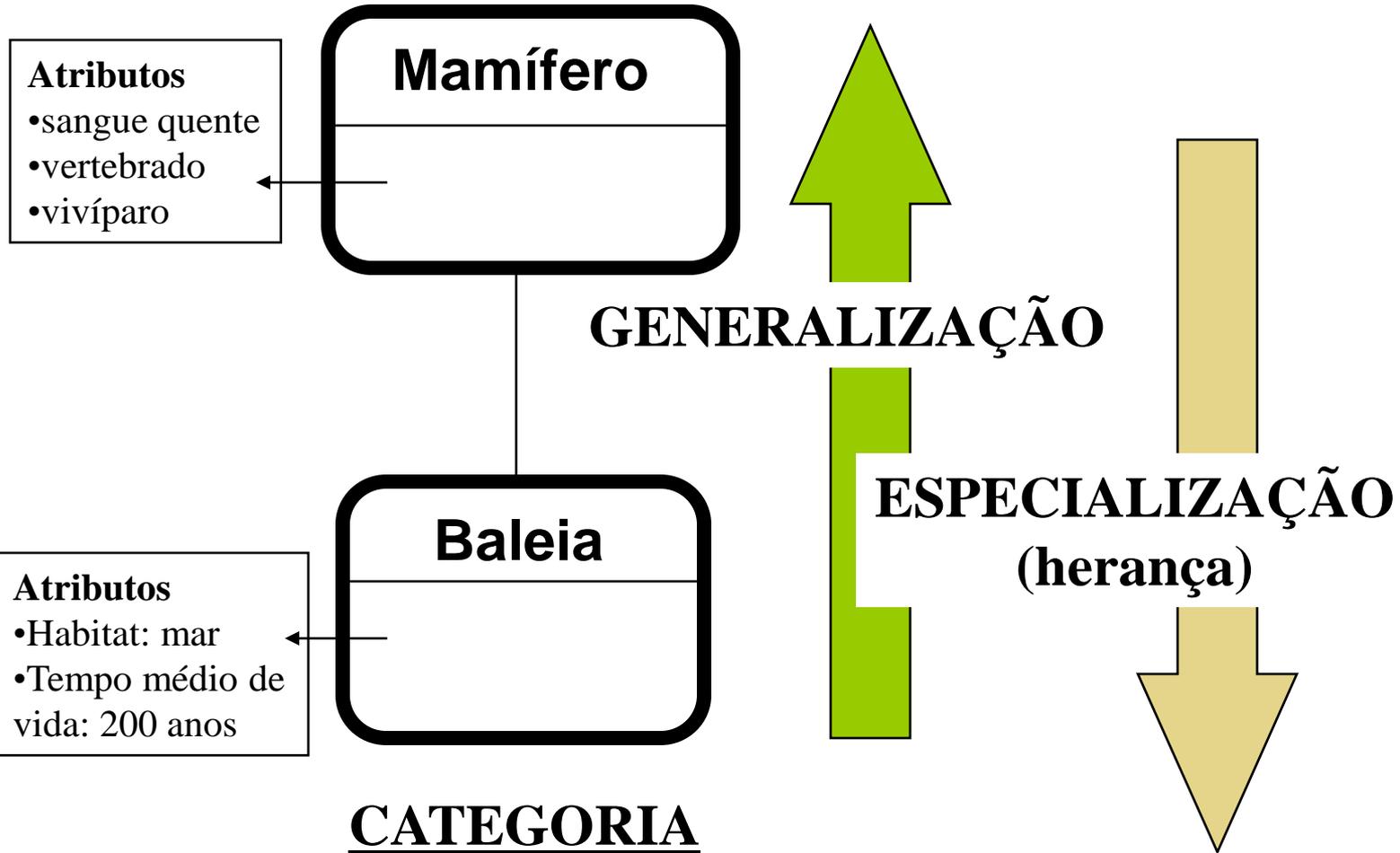
Generalização/Especialização

CATEGORIA



Generalização/Especialização

CATEGORIA



Agregação/Decomposição

CATEGORIA



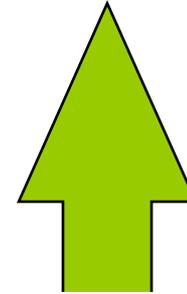
Nome

Endereço

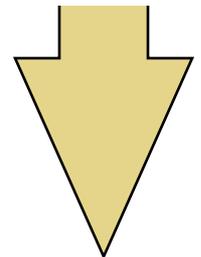
CATEGORIA

COMPONENTE

AGREGAÇÃO
(composição -
síntese)



DECOMPOSIÇÃO
(refinamento -
análise)



Paradigma de Orientação a Objetos (OO)

- Idéia básica

“Nós percebemos o mundo como uma coleção de objetos que interagem entre si”

“Maneira de ver as coisas, uma nova maneira de pensar”

- Permite que objetos do mundo real sejam mapeados em **Objetos** no computador

- Visão do DOMÍNIO da aplicação é composto por **OBJETOS** que se comunicam através de **MENSAGENS**

- *Paradigma de OO = objetos + classificação + herança + comunicação (Coad e Yourdon, 1990)*

- Modelo de Objetos

 - Objetos, Mensagens e Métodos

Paradigma de Orientação a Objetos (OO)

- Tradicionalmente
 - Softwares são formados por dados e procedimentos independentes
- Paradigma OO
 - Softwares são formados, principalmente, por:
 - **Objetos**
 - **parte interna** (define o comportamento do objeto, ou seja, especifica os métodos associados ao objeto)
 - **parte externa** (interface - maneira que o objeto se apresenta para outros objetos)
 - **Mensagens**
 - Diminuir o gap semântico

Paradigma de Orientação a Objetos

- O desenvolvimento orientado a objetos evoluiu no sentido

PROGRAMAÇÃO



ANÁLISE

- Conceitos desse paradigma foram introduzidos com o surgimento da linguagem Smalltalk
- Em seguida, surgiram métodos de projeto para dar suporte à implementação de sistemas orientados a objetos
- Recentemente, propostas de métodos para análise e projeto vêm emergindo

Histórico

- Métodos de Análise e Projeto OO
 - **CRC** (Class Responsibility Collaborator, Beecke e Cunningham, 1989)
 - **OOA** (Object Oriented Analysis, Coad e Yourdon, 1990)
 - **Booch** (1991)
 - **OMT** (Object Modeling Technique, Rumbaugh, 1991)
 - **Objectory** (Jacobson, 1992)
 - **Fusion** (1994)
 - **Unified Modeling Language** (UML, 1997)

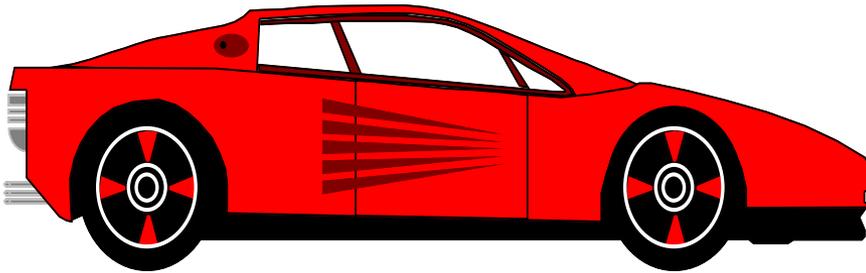
Conceitos de OO

- Modelo de Objetos
 - Objeto
 - Mensagem
 - Método
- Classe
- Atributo
- Protocolo
- Classificação
- Herança
- Encapsulamento/Ocultamento de Informação
- Polimorfismo

Objeto

- Tudo em Orientação a Objeto é **OBJETO**
- Objeto, no mundo físico, é tipicamente um produtor e consumidor de itens de informação
 - máquina, comandos, arquivos, pessoas, sinais
- Na modelagem de sistemas, Objeto é uma entidade que possui:
 - Estado
 - Comportamento
 - Identidade Única
- Definição
 - *“Qualquer coisa, real ou abstrata, a respeito da qual armazenamos dados e métodos que os manipulam”* Martin, Odell (1995)
 - **Abstração** de uma entidade do mundo real de modo que essa entidade possua várias características
 - Exemplo: objetos cachorro, casa, carro

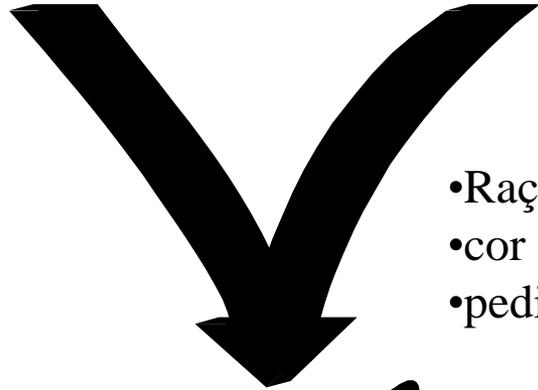
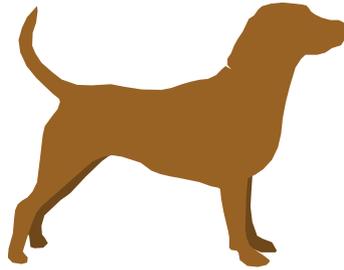
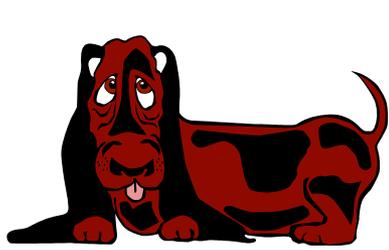
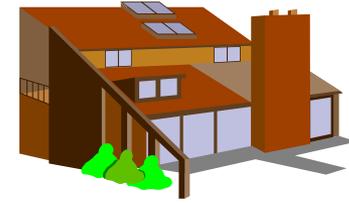
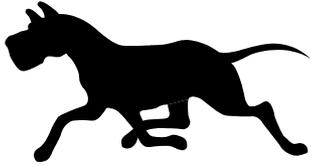
Estado do Objeto



Propriedades:

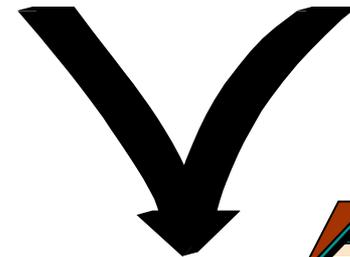
- Cor = Vermelha
- Ano = 2002
- Velocidade = 0 Km/h
- Combustível = Gasolina

- Conjunto de suas propriedades associadas a seus valores correntes
- Propriedades geralmente referenciadas como *Atributos*



- Raça
- cor
- pedigree

CACHORRO



CASA

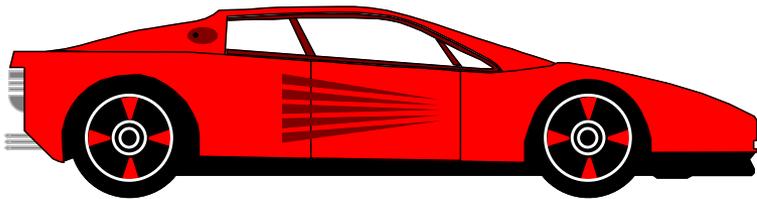


Comportamento do Objeto

- Conjunto de serviços ou operações que outros objetos, ditos clientes, podem requisitar
- Operações são usadas para alterar o estado do próprio objeto
- Operações geralmente referenciadas como *Métodos*

Comportamento do Objeto

- Comunicação entre objetos através de *Troca de Mensagens*
- Representa como o objeto reage às *Mensagens* a ele enviadas
- Ex



Métodos

- Acelerar ()
- Freiar ()
- Acender Faróis ()
- Virar a Direita ()

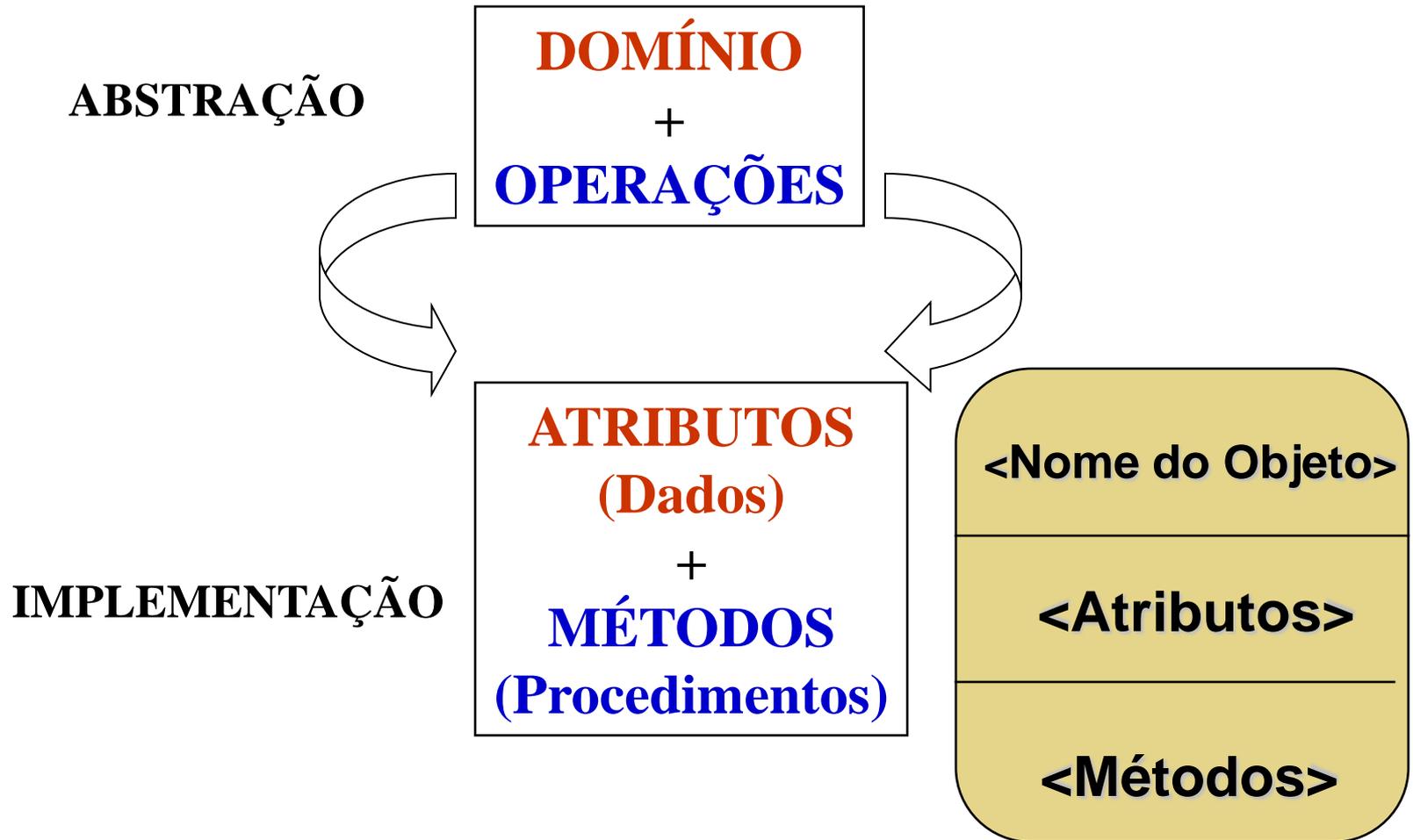
- **OBJETO**: Entidade independente, composta por um conjunto de elementos que a caracterizam (**domínio**) e as ações que agem sobre esse domínio (**operações**)

- **OBJETO** 



- **Visão implementacional** (Takahashi, 1990)
 - entidade composta por:
 - **ESTADO INTERNO**: é onde os valores podem ser armazenados e modificados ao longo da vida do objeto, caracterizando seu domínio
 - **COMPORTAMENTO**: é o conjunto de ações pré-definidas (**métodos**) através das quais o objeto responderá a demanda de processamento por parte de outros objetos

Abstração **Mapeamento** → Implementação



Objeto

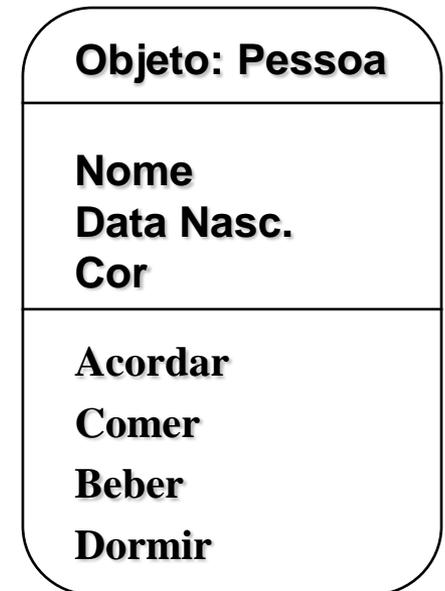
● Resumindo:

- Pacote de informações (atributos) e a descrição de suas operações (métodos), de modo que elas são intrínsecas ao seu domínio e este é formado pelos elementos que o caracterizam



Exemplo:

- **Objeto:** Pessoa
- **Atributos:** Nome, Data de Nascimento, Cor
- **Métodos:** Acordar, Comer, Beber, Dormir



Objeto

- Exemplo

- uma **Pessoa** e um **Carro** podem ser considerados como um objeto de dados no sentido de que qualquer um deles pode ser definido em termos de um conjunto de atributos e operações

- Objetos podem relacionar-se um com o outro

- Uma Pessoa pode possuir Carro, onde o relacionamento possuir define uma conexão específica entre Pessoa e Carro

Objeto

=

Atributos (Dados)

+

Métodos (Funcionalidade)

+

Encapsulamento

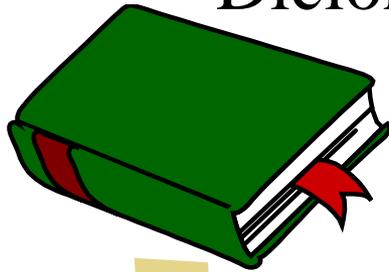
Exemplos de Objeto

- Uma fatura
- Uma organização
- Uma tela com a qual o usuário interage
- Um desenho de engenheiro
- Um avião
- Um vôo de avião
- Uma reserva num avião
- Um processo de atendimento de pedidos
- Um computador
- Um ser humano neurótico pode ser estereotipado como um objeto: possui memória em que valores podem ser armazenados e que condicionarão seu comportamento

Atributos

- Representam um conjunto de informações, ou seja, elementos de dados que caracterizam um objeto
- Descrevem as informações que ficam escondidas em um objeto para serem exclusivamente manipuladas pelas operações daquele objeto
- São variáveis que definem o estado de um objeto, ou seja, são entidades que caracterizam os objetos
- Cada objeto possui seu próprio conjunto de atributos
- Exemplos:
 - Objeto **Pessoa**
 - Atributos: Nome, Cor, Data de Nascimento
 - Objeto **Empregado**
 - Atributos: No do empregado, nome, nome do departamento, salário, endereço, dependentes

Dicionário



Objeto do mundo Real

Objeto: Dicionário

Estrutura de Dados

Adicionar Palavra

Eliminar Palavra

Encontrar Palavra

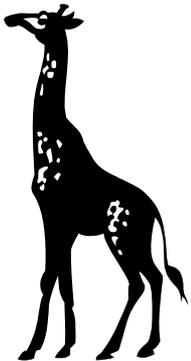
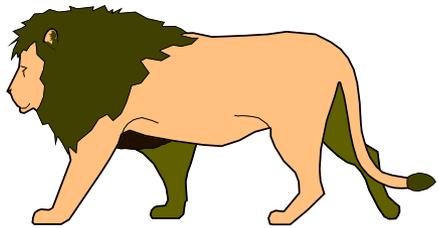
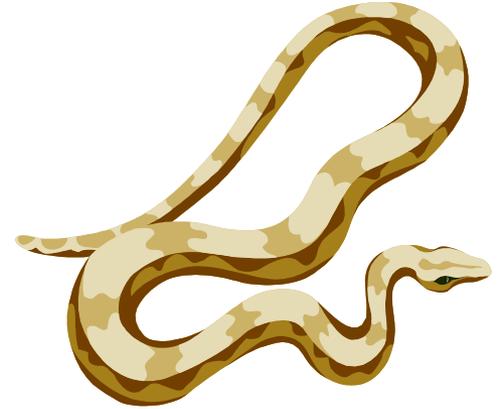
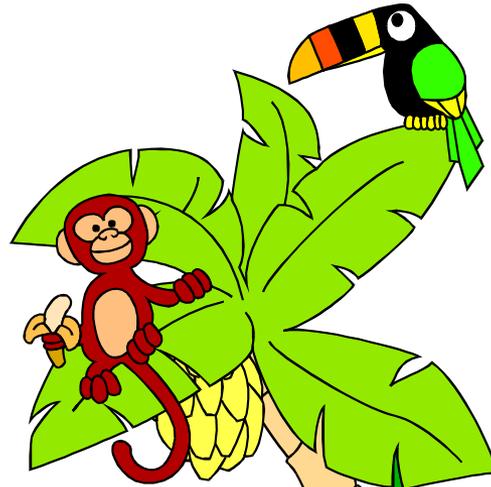
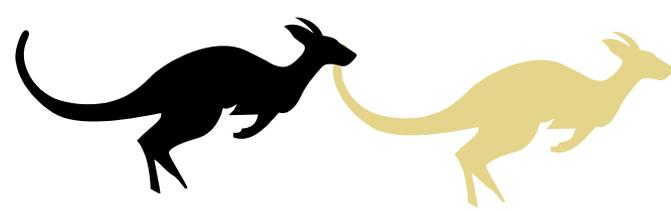
Imprimir definição

Atributo

- Um valor armazenado pelos objetos da classe
 - PESSOA: nome, idade, peso, cor
 - CARRO: cor, peso, ano, modelo
 - Cada atributo possui um valor para cada instância
 - João tem 24 anos e Maria tem 40
- Não se deve explicitar identificadores (características de programação - não têm significado intrínseco)
 - Não confunda identificadores internos com atributos do mundo real
 - O número do seguro social, número da chapa do carro, número do telefone, RG não são identificadores internos. ELES são atributos, pois têm significado no mundo real. **ATRIBUTOS LEGÍTIMOS**

Classificação

- Categorização em **Classes**
- Quando estamos modelando um domínio, observamos que há várias entidades similares que podem ser abstraídas em um conceito único que engloba esta similaridade
- Operação de abstração das **Similaridades**
- O POO incorporou esta operação através da abstração dos atributos e dos métodos que caracterizam objetos semelhantes, especificando-os na **CLASSE**
- **Exemplo:**
 - Classificação de espécies em Zoologia

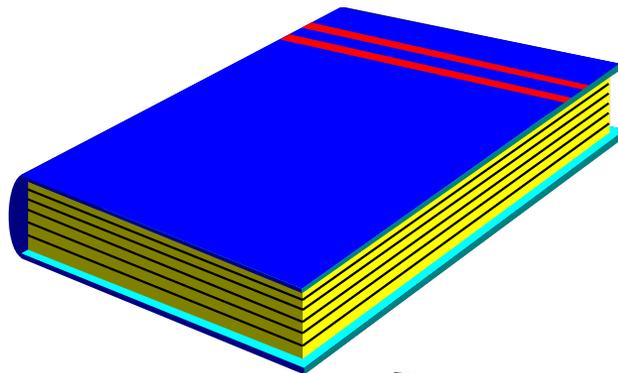


Classe

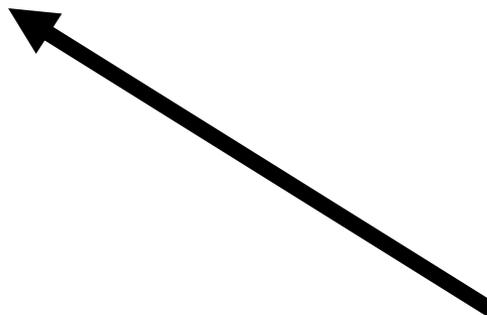
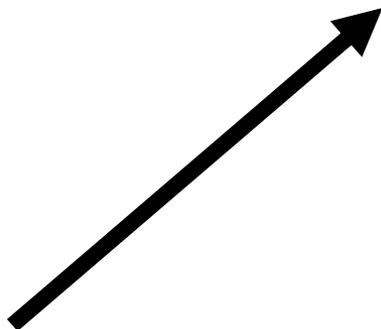
- **Objeto:** abstração de uma entidade do mundo real, através dos atributos e operações
- **Classe:** abstração de um conjunto de objetos similares do mundo real
- Agrupamento de objetos similares. Descreve a
 - estrutura de dados, e o
 - comportamento de objetos similares
- Todo objeto é uma instância de uma Classe
- Todas as instâncias de uma classe têm valores próprios para os atributos especificados na classe
- Os objetos representados por determinada classe diferenciam-se entre si pelos valores de seus atributos
- Sistema puramente OO, **Classe** é também um **Objeto**

Classe

- Conjunto de objetos que possuem propriedades semelhantes (ATRIBUTOS), o mesmo comportamento (MÉTODOS), os mesmos relacionamentos com outros objetos e a mesma semântica
 - PESSOA, CARRO, CASA, VEÍCULO
 - EMPRESA, PAÍS, SERES HUMANOS, HOMENS, MULHERES, MEIO DE TRANSPORTE



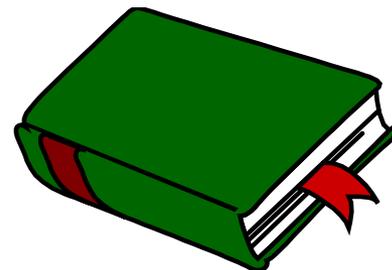
Livro



Bíblia



Relatório



Dicionário

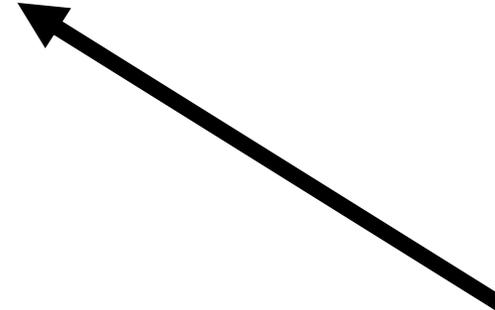
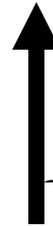
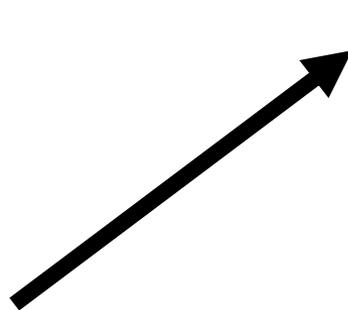
Reformar
Limpar
Pintar
Mobiliar



CASA

Portas
Salas
Cozinha

Quartos
Localização
Telhado



Generalização

Reformar
Limpar
Pintar
Mobiliar



Portas
Salas
Cozinha
Quartos
Localização
Telhado

(Superclasse)

CASA

PRAIA

FAVELA

MANSÃO

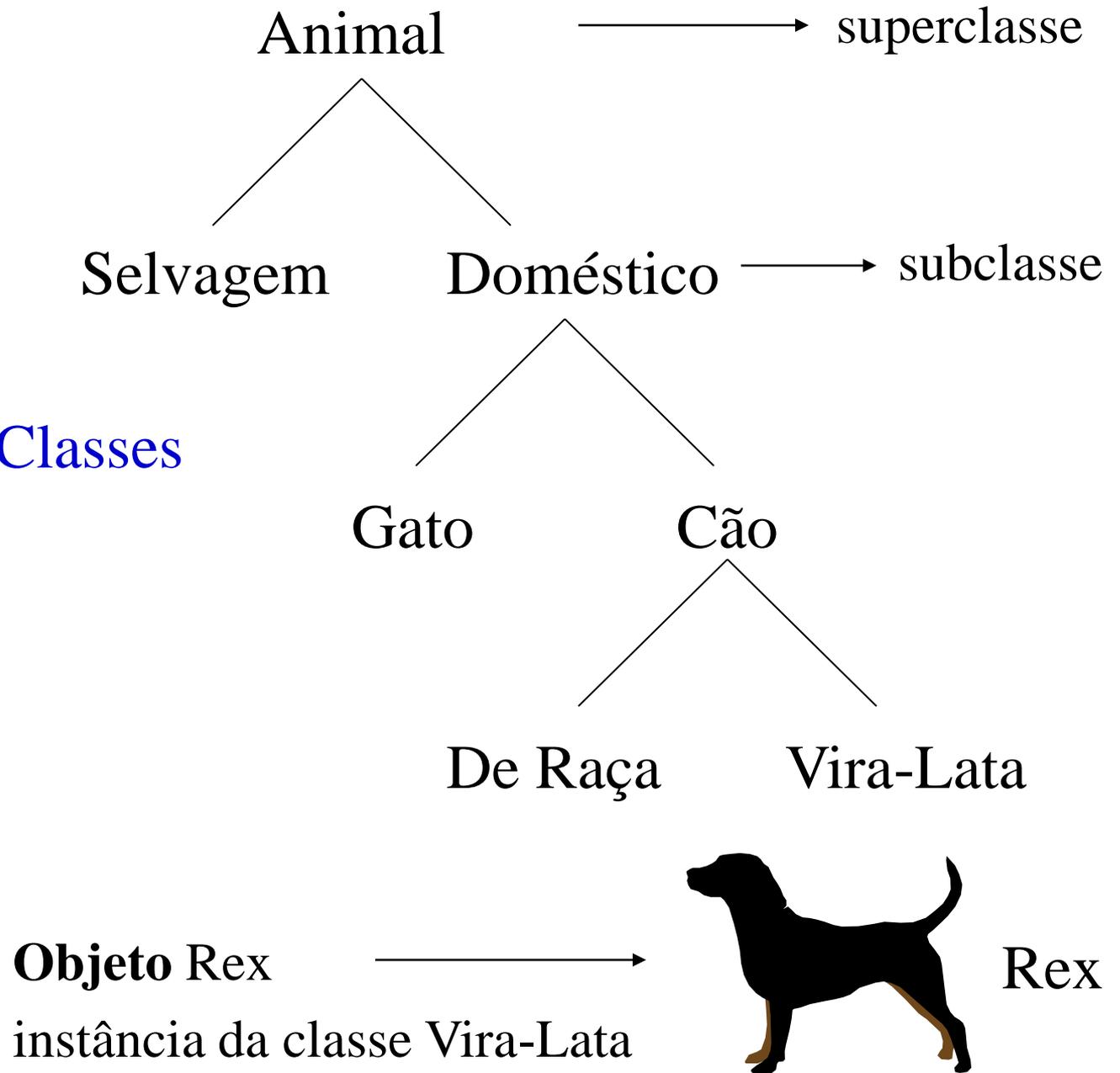
(Subclasse)

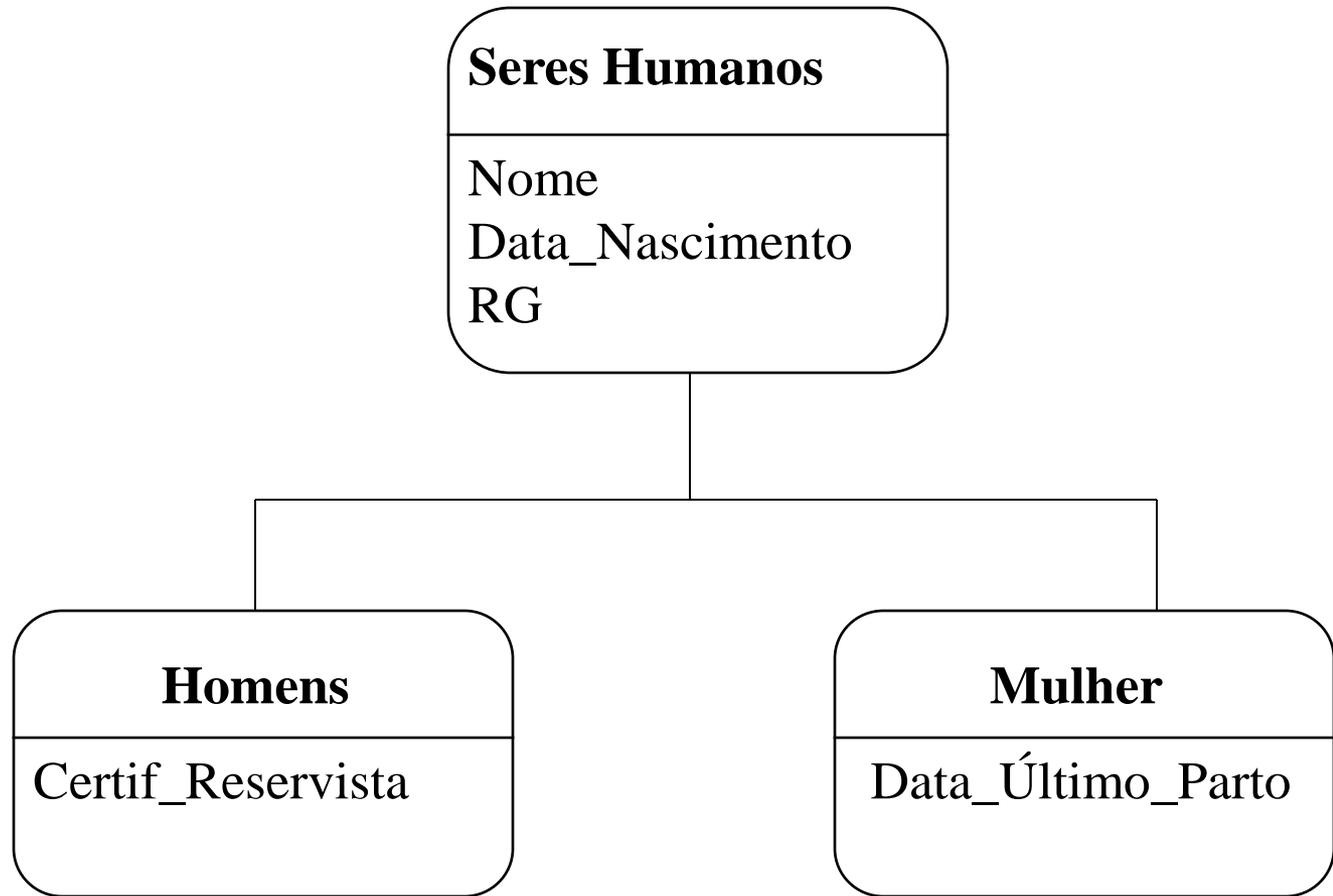
Limpar Piscina
Contratar Criadagem
Piscina
Quadras

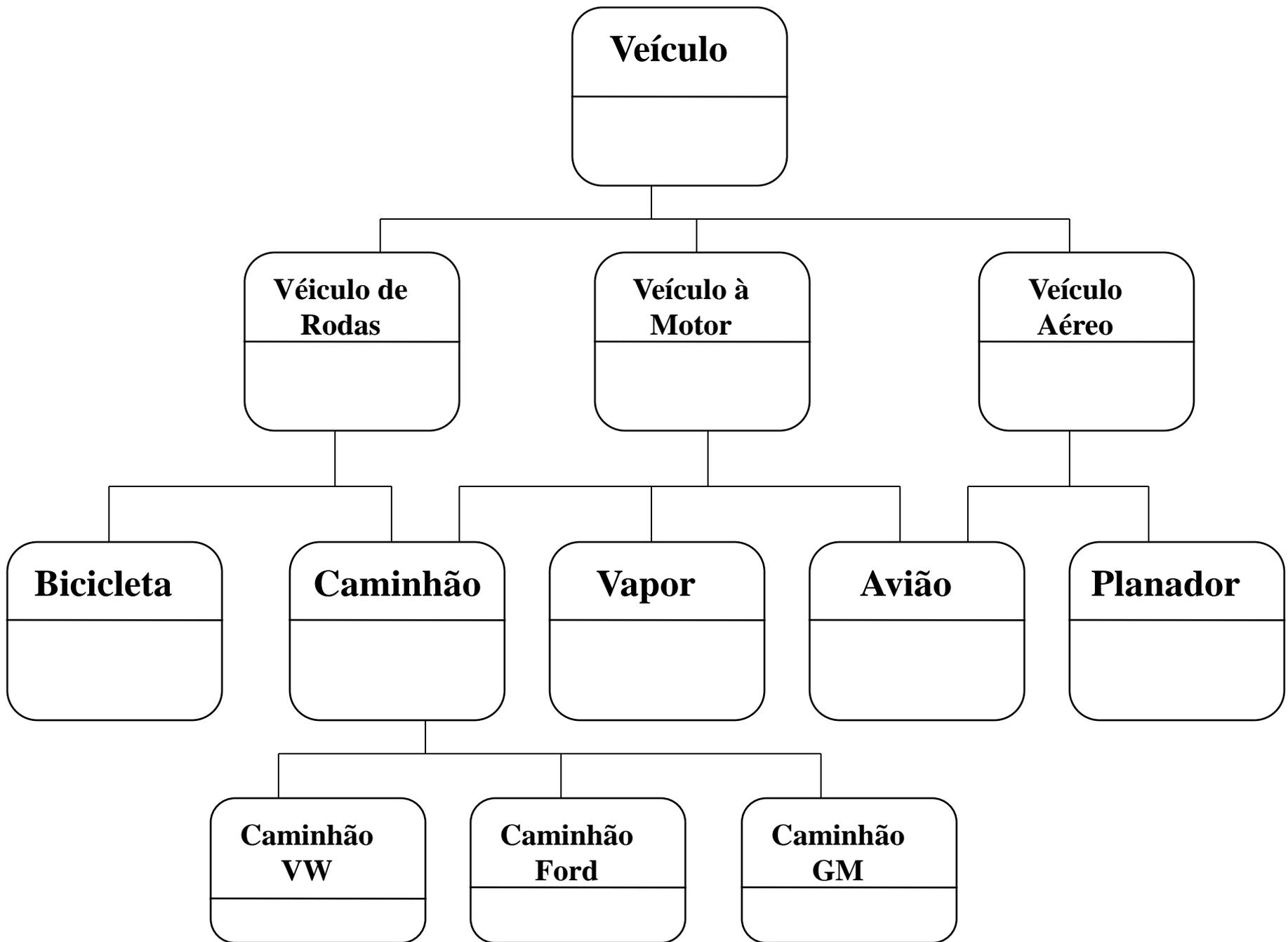
Especialização



Hierarquia de Classes







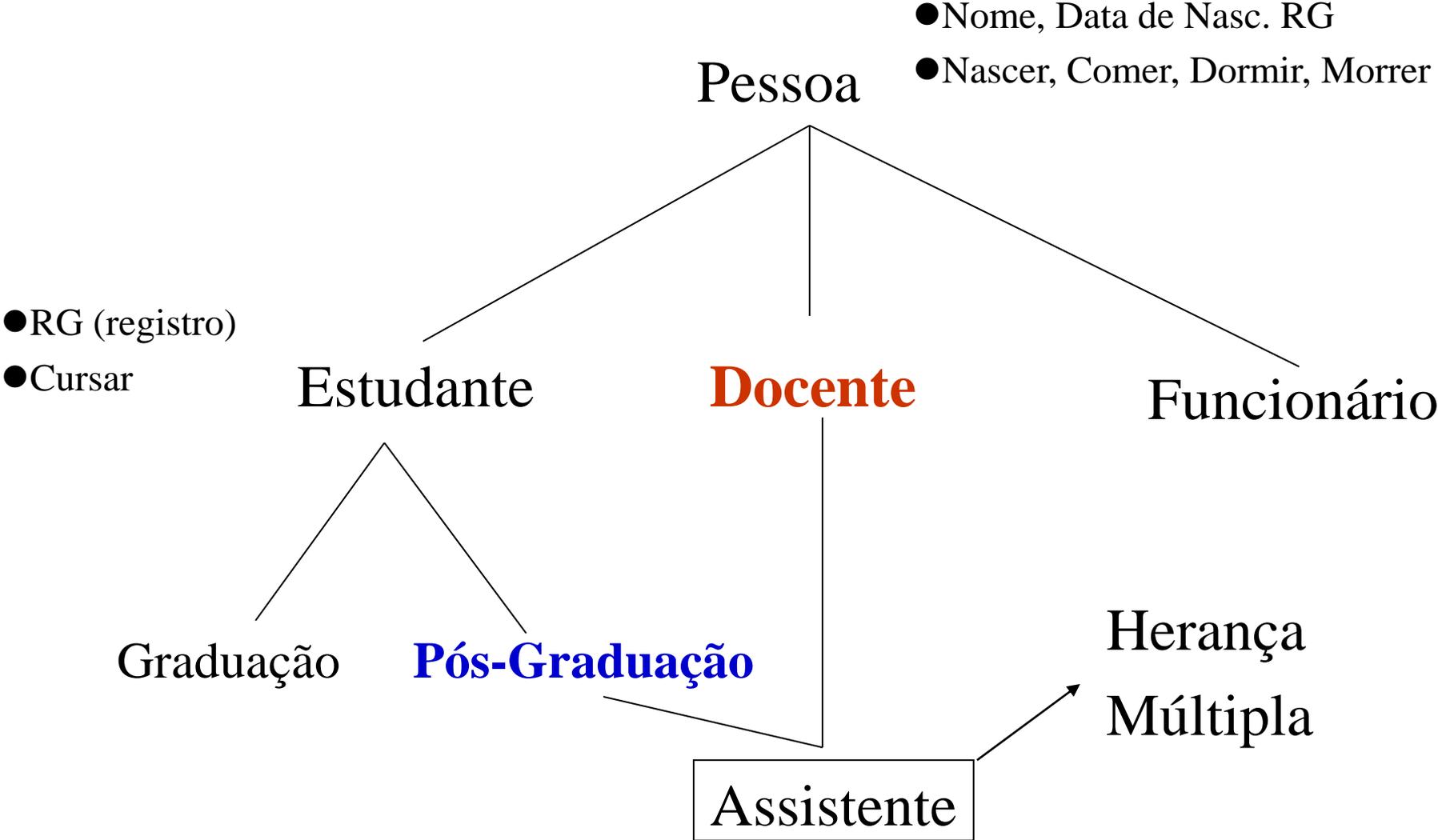
Herança

- Mecanismo que permite definir uma nova classe (subclasse) a partir de uma classe já existente (superclasse)
- Ao se estabelecer uma Especialização (subclasse) de uma classe, a subclasse herda as características comuns da superclasse, isto é, a especificação dos atributos e dos métodos da superclasse passam a fazer parte da especificação dos atributos e dos métodos da subclasse
 - A subclasse pode adicionar novos métodos, como também reescrever métodos herdados
- Habilidade de um objeto derivar seus atributos (dados) e métodos (funcionalidade) automaticamente de outro objeto

Herança

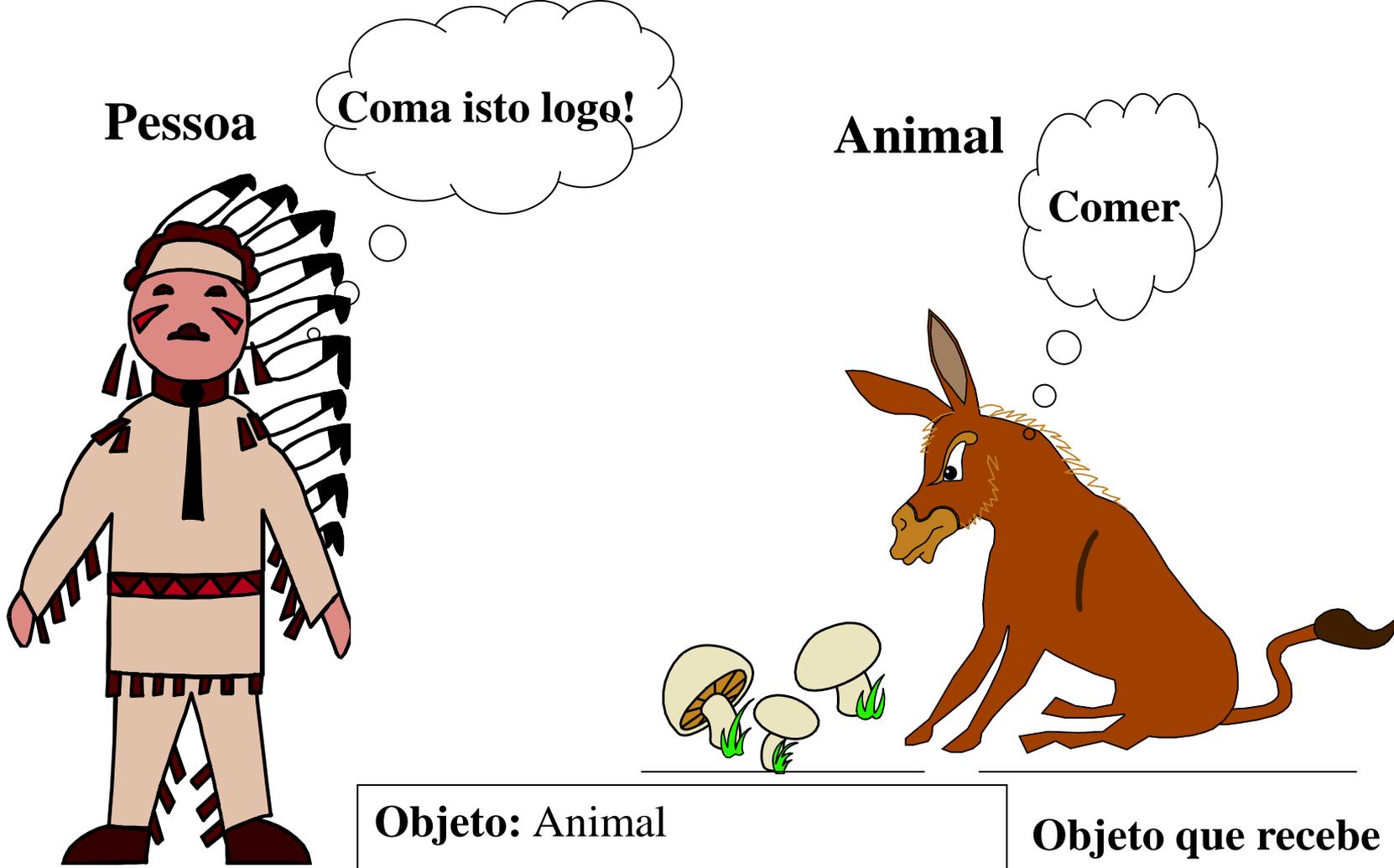
- Classes podem ser construídas a partir de outras já existentes; facilita extensibilidade
- Quando uma **mensagem** é enviada para um objeto
 - A procura do método correspondente começa pela classe do objeto
 - Se o método não for encontrado, a procura continua na superclasse
- A Herança pode ser de dois tipos:
 - **Herança Simples:** quando uma classe é subclasse de somente uma superclasse
 - **Herança Múltipla:** quando uma classe é subclasse de várias superclasses e conseqüentemente herda as características de cada uma delas

Exemplo: Sistema de Administração Universitária



Mensagem

- É o mecanismo através do qual os objetos se comunicam, invocando as operações desejadas
- Especificação de uma operação do objeto
- É composta por
 - **Seletor:**
 - nome simbólico que descreve o tipo da operação
 - descreve O QUE o objeto que envia quer que seja invocado, não como deveria ser invocado
 - o objeto receptor da mensagem contém a descrição de COMO a operação deveria ser executada
 - **Parâmetros:**
 - argumentos que uma mensagem pode conter que faz parte da operação e requer uma ordem única



Objeto que envia
EMISSOR



Determina o que deve ser manipulado e aguarda o retorno do objeto que foi ativado

Objeto: Animal
Mensagem: COMA ISTO LOGO
Seletor: COMA
Parâmetros: ISTO LOGO

Objeto que recebe
RECEPTOR



Determina como manipular

Mensagem

- Um objeto (**Emissor**) envia uma mensagem a outro (**Receptor**) que executará o serviço
- Métodos são invocados por Mensagens
- Exemplo
 - A chamada de um procedimento/função em LP é uma aproximação inicial de uma mensagem, como em $P(10,20)$, onde P é o **seletor** e os valores 10 e 20 são os **parâmetros**
 - Diferença:
 - a ação da mensagem a ser ativada depende essencialmente do objeto que receber a mensagem

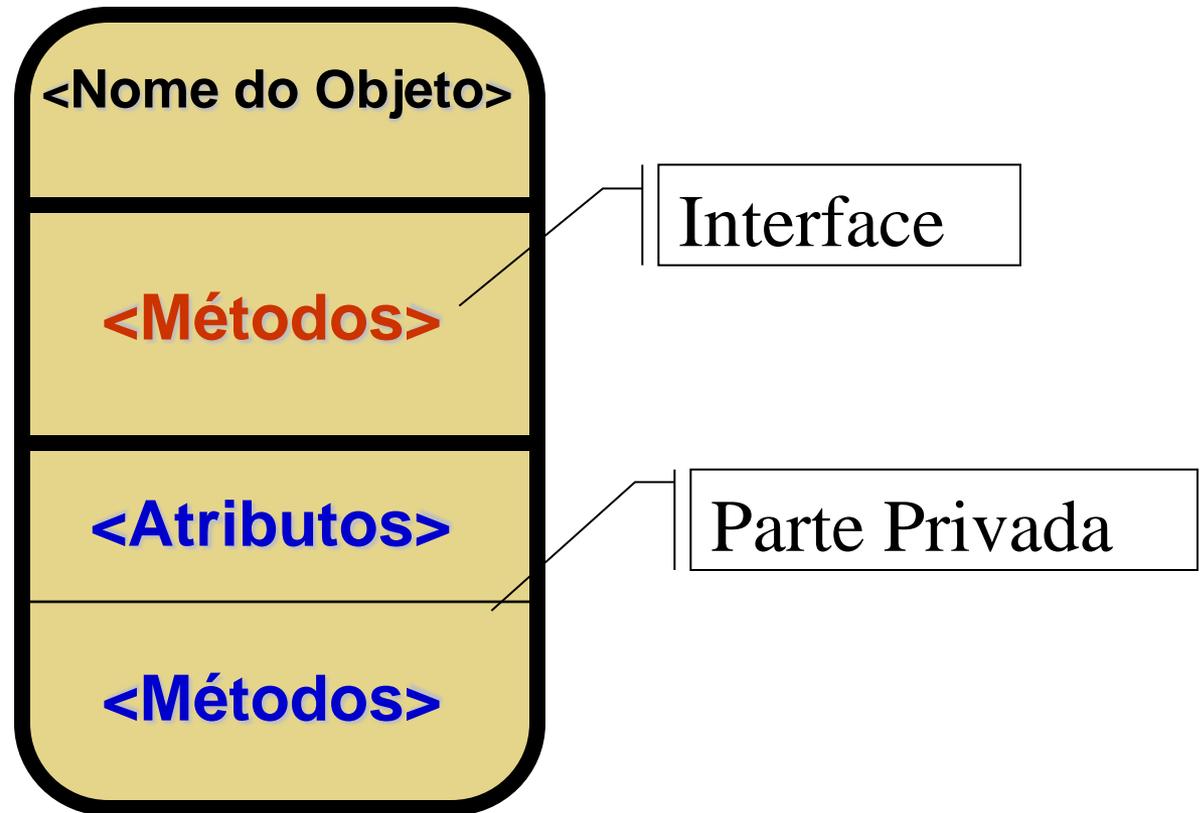
Método

- Origem: LP Smalltalk
- Quando um objeto é mapeado dentro do domínio do software, os processos que podem mudar a sua estrutura de dados são denominados **Operações** ou **Métodos**
- **Métodos** são invocados por **Mensagens**
- Cada objeto possui seu próprio conjunto de métodos
- Definições:
 - São procedimentos definidos e declarados que atuam sobre um objeto ou sobre uma classe de objetos
 - Descrição de uma sequência de ações a serem executadas por um objeto
 - Através dos métodos que especifica-se a um objeto **COMO FAZER** alguma coisa
 - São intrínsecos aos objetos e não podem ser separados

Interface (Protocolo)

- Objetos se comunicam através de **Mensagens**, que invocam os **Métodos** desejados
- **Parte Privada** do Objeto (Visão Interna)
 - Métodos
 - Atributos
- **Parte Compartilhada** do Objeto ou Interface
 - corresponde à parte externa do objeto, isto é, àquela que é vista por outros objetos com a finalidade de invocar os métodos que o objeto realiza
 - agrupa um conjunto de mensagens que o objeto pode responder
 - as mensagens especificam quais operações sobre o objeto podem ser realizadas, mas não como a operação será executada

Estrutura de um Objeto (classe)



Classes em C++

```
// classe chamada counter
class counter
{
    private:
        unsigned int value;
    public:
        counter( ) {value = 0}; // um construtor
        void increment ( )      {if (value < 65535) value++; };
        void decrement ( )     {if (value > 0) value--; };
        unsigned int access_value( ) {return value; }
};
```

Ocultamento de Informação

(Information Hiding)

- Característica que visa esconder detalhes de implementação
- É alcançado em OO, visto que o objeto, quando implementado, possui uma parte privada (atributos e métodos) e uma parte compartilhada (interface)
- Detalhes de implementação ficam escondidos na parte privada
- Programadores podem introduzir mudanças na implementação de um método sem afetar o comportamento externo desse método

Encapsulamento de Dados

- Objetos encapsulam suas estruturas de dados (atributos)
- Os atributos de um objeto só podem ser manipulados pelos próprios métodos do objeto
- Restringe a visibilidade do objeto mas facilita o reuso
- Os DADOS e os MÉTODOS são empacotados sob um nome e podem ser reusados como uma especificação ou componente de programa

Polimorfismo

- Ao receber uma mensagem para efetuar uma Operação, é o objeto quem determina como a operação deve ser efetuada, pois ele tem o comportamento próprio
- Como a responsabilidade é do **Receptor** e não do **Emissor**, pode acontecer que uma mesma mensagem ative métodos diferentes, dependendo da classe de objeto para onde é enviada a mensagem
- Permite a criação de várias classes com interfaces idênticas, porém objetos e implementações diferentes
- Capacidade de uma mensagem ser executada de acordo com as características do objeto que está recebendo o pedido de execução do serviço

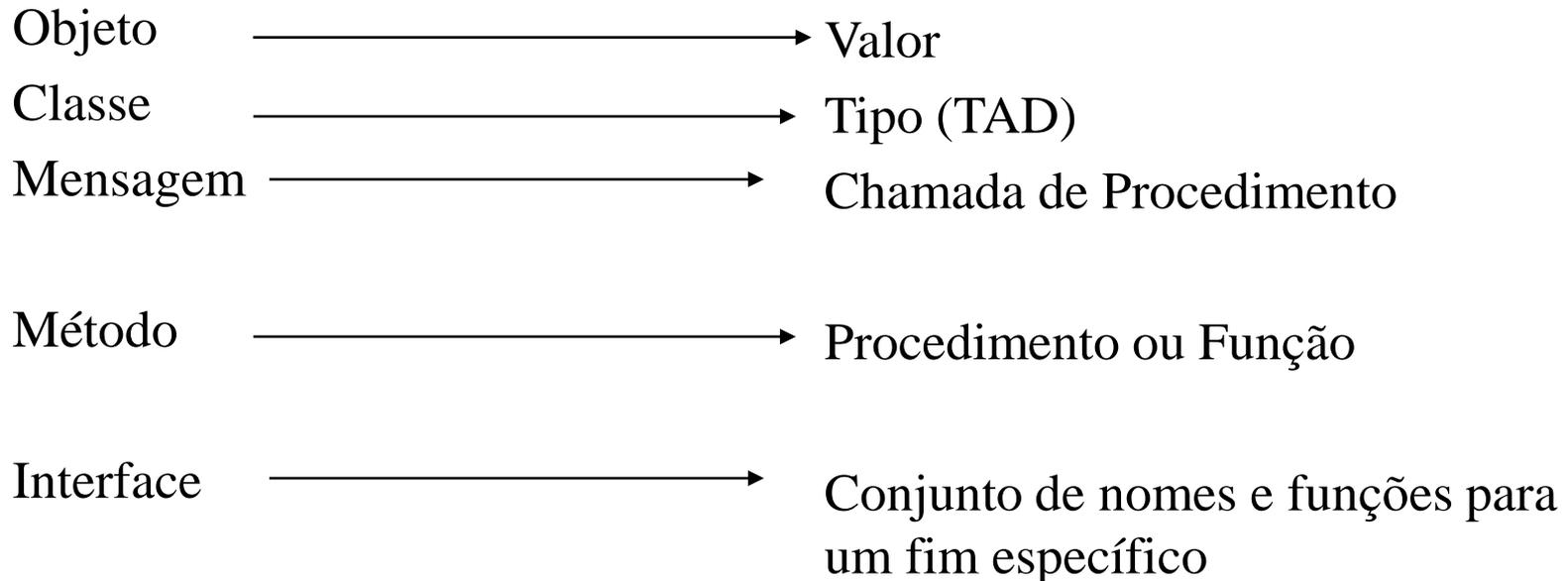
Polimorfismo

- Por exemplo, a operação “<” pode ser definida em várias classes de números: inteiro (método de comparação inteira), string .
- É comum em LPOOs

Analogia dos conceitos principais no POO e no paradigma tradicional de programação

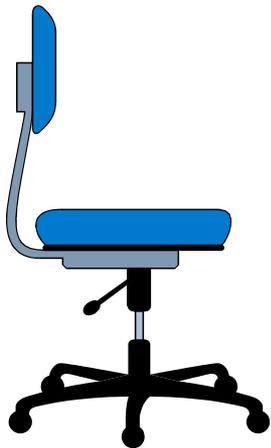
● Linguagens Orientadas a Objetos

● Linguagens Tradicionais



Características

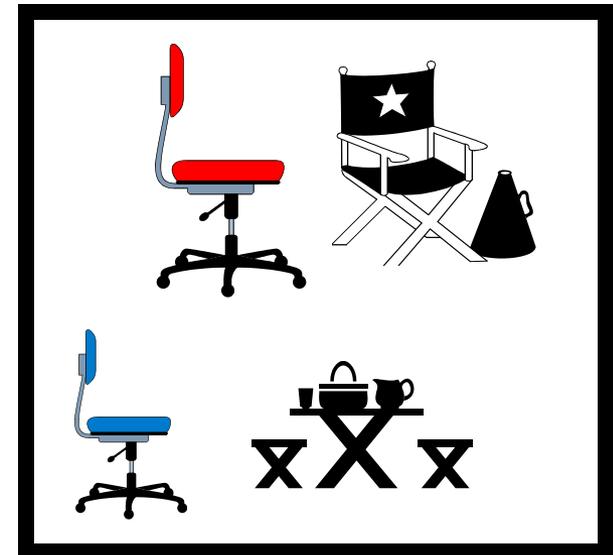
- Exemplo de objeto do mundo real
 - “a coisa em que você está sentado agora”



Cadeira

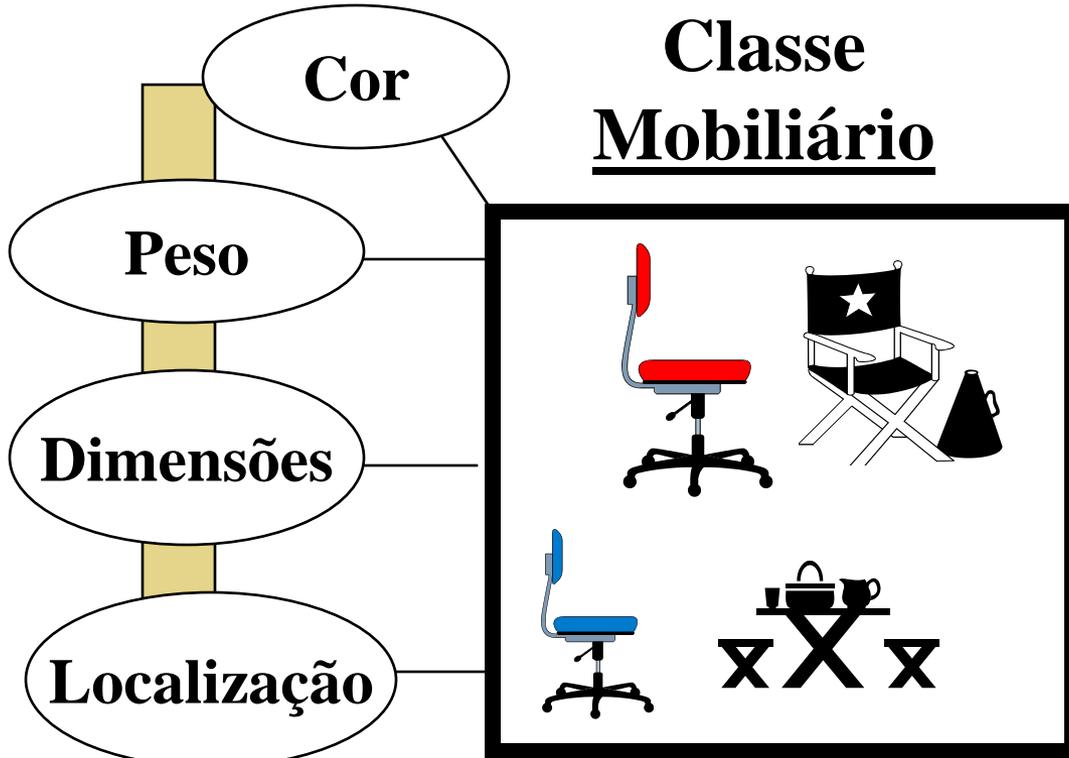


É membro - (instância)
de uma classe maior de
objetos



Mobiliário

Classe Mobiliário

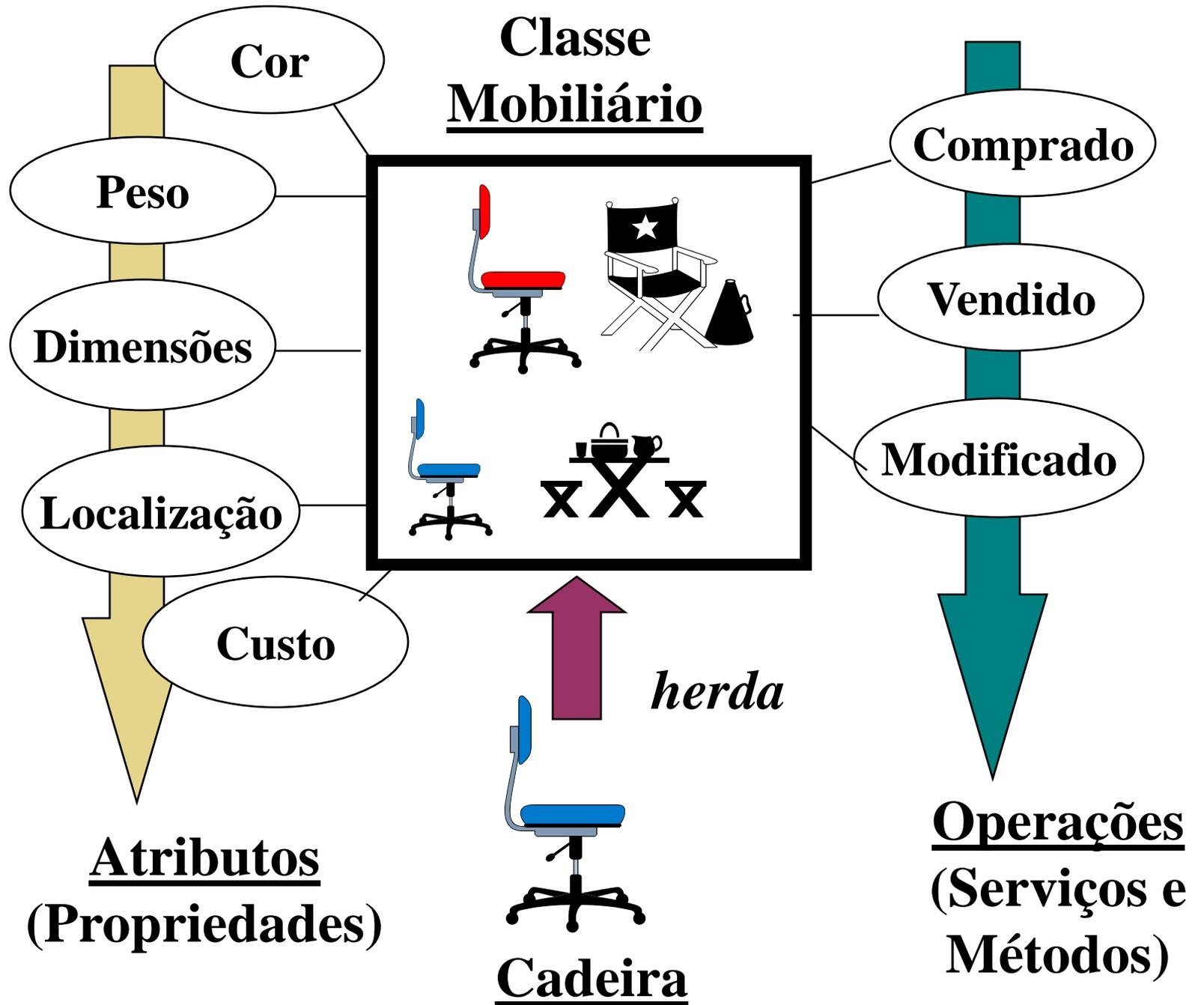


Atributos
(Propriedades)

herda

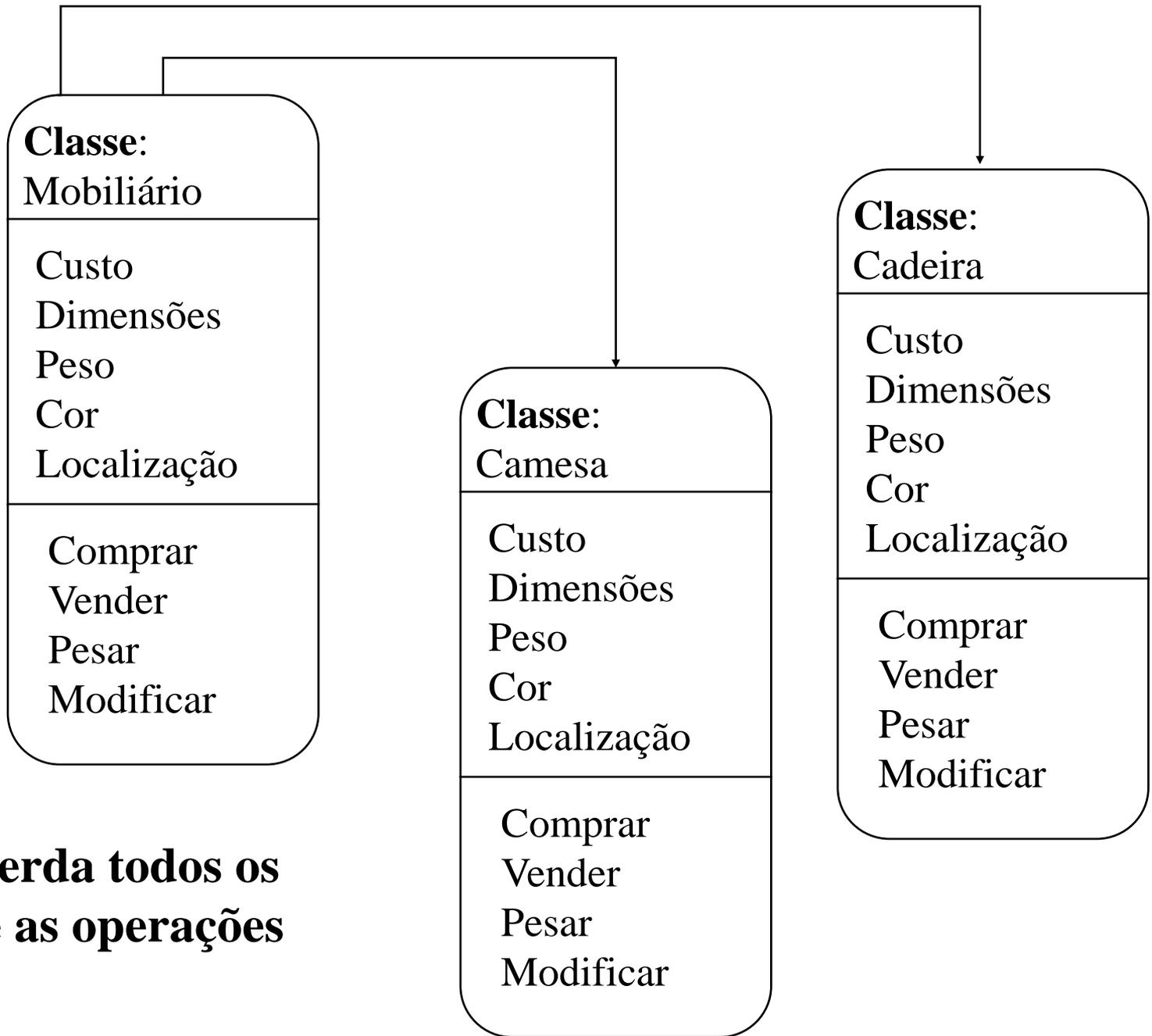


Cadeira



- O Objeto **Cadeira** é um membro (INSTÂNCIA) de uma CLASSE maior de objetos - MOBILIÁRIO
- CLASSE de objetos descreve um conjunto de objetos que possuem
 - propriedades semelhantes (ATRIBUTOS)
 - mesmo comportamento (MÉTODOS)
 - mesmos relacionamentos com outros objetos
 - mesma semântica
- Atributos GENÉRICOS da classe MOBILIÁRIO
 - custo, dimensões, peso, localização, cor
- Atributos se aplicam a todos os objetos da classe MOBILIÁRIO: mesa, cadeira, sofá, armário

- Uma vez que o objeto CADEIRA é um membro da classe MOBILIÁRIO, HERDA todos os atributos definidos para a classe
- Todo objeto da classe MOBILIÁRIO pode ser manipulado de diversas maneiras (OPERAÇÕES)
 - comprado
 - vendido, pesado
 - modificado (pintar de roxo, serrar uma perna)
- Cada uma dessas operações (SERVIÇOS ou MÉTODOS) modificará um ou mais atributos do objeto



O objeto herda todos os atributos e as operações da classe

<Nome da classe>

<Atributos>

<Operações>

- O objeto CADEIRA e outros da mesma classe
ENCAPSULA

- dados (os valores dos atributos que definem a cadeira)
- operações (as ações aplicadas para mudar os atributos da cadeira)

Aplicação Escritório Eletrônico

- Documento de Requisitos

“Pretende-se automatizar toda a entrada, processamento, saída e arquivamento de documentos em um escritório, onde um executivo e sua secretária desempenham os papéis típicos de suas funções. Por documento, entende-se qualquer tipo de informação que flui para/do escritório e entre o executivo e a secretária: cartas, anotações de telefonemas, documentos técnicos, etc.”

Identificação de Objetos

- O processo de análise para identificação dos objetos e classes \Rightarrow DIFÍCIL ☹
- Basicamente a mesma tarefa realizada na análise e no projeto de sistemas (Análise Estruturada)
- Os domínios de aplicação que naturalmente refletem entidades concretas do mundo real se beneficiam da abordagem OO

Identificação de Objetos

- Objetos podem ser:
 - Entidades Externas que produzem ou consomem informações (outros sistemas, pessoas)
 - Coisas que fazem parte do domínio (relatórios, displays, cartas)
 - Eventos que ocorrem no contexto do sistema (transferência de propriedade)
 - Papéis desempenhados por pessoas (gerente, engenheiro, vendedor)
 - Unidades organizacionais (grupo, equipe)
 - Lugares que estabelecem o contexto do problema (piso de fábrica, área de descarga)

Efetuar uma "análise gramatical" sobre a narrativa de processamento que descreva a bolha

- 👉 Isolar nomes (e locuções nominais), verbos (e locuções verbais)
- 👉 Os nomes e os verbos que são sinônimos ou que não tem nenhuma relação com o processo de modelagem são omitidos
- 👉 Todos os nomes são classes ou atributos (itens de dados)
- 👉 Todos os verbos são operações (métodos) - relacionamento entre classes