Universidade Federal de Mato Grosso do Sul

Sistemas de Informação - CPCX

Paradigma Orientado a Objetos

Professor Fernando Maia da Mota

Slides gentilmente cedidos por Profa. Dra.
Maria Istela Cagnin Machado UFMS/FACOM

Histórico de OO

- O termo OO surgiu no final da década de **60**, quando dois cientistas dinamarqueses criaram a linguagem Simula (*Simulation Language*)
  - **1967** - Linguagem de Programação Simula-67- *conceitos de classe e herança*
- O termo Programação Orientada a Objetos (POO) é introduzido com a linguagem Smalltalk (**1983**)
- FINS DOS ANOS **80** ⇨ Paradigma de Orientação a Objetos
  - abordagem poderosa e prática para o desenvolvimento de software
- **1983:** disponibilização da primeira versão do C++
- **1988:** lançamento da linguagem Eiffel (primeira linguagem OO "pura")
- **1995:** primeira versão da linguagem Java

Histórico de OO

- Na metade da década de 80, quando as linguagens orientadas a objetos começaram a fazer sucesso, surgiu a necessidade de processos para dar suporte ao desenvolvimento de software orientado a objetos.

Histórico de OO

- O surgimento da orientação a objetos exigiu a criação de processos que integrassem o processo de desenvolvimento e a linguagem de modelagem, por meio de técnicas e ferramentas adequadas

Histórico de OO

- Surgiram vários métodos de análise e projeto OO
  - CRC (*Class Responsability 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 (Coleman, **1994**)

Histórico de OO

- Depois de quase uma década do surgimento das linguagens OO, estabeleceu-se uma gama enorme de processos de desenvolvimento OO, o que passou a dificultar a comunicação entre analistas e projetistas de software

- A Linguagem de Modelagem Unificada (UML) – surgiu com o intuito de criar uma notação **completa e padronizada**, que todos pudessem usar para documentar o desenvolvimento de software OO

Histórico de OO

- No entanto, a UML não apresenta um processo, mas apenas a notação. Por isso, alguns anos depois de sua criação, passaram a surgir propostas de processos de desenvolvimento com base na UML

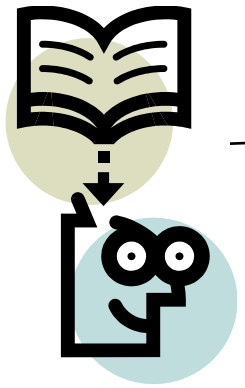
- **Exemplos:**
  - UP (Processo Unificado) e sua especialização pela Rational → o RUP (*Rational Unified Process*)

O que é desenvolvimento orientado a objetos ?

- Quatro grandes atividades:
  - Análise: investigação do problema
  - Projeto: solução lógica
  - Construção: código
  - Teste: verificação e validação do produto
- Ênfase na representação de objetos

Conceito do Domínio

(Biblioteca)

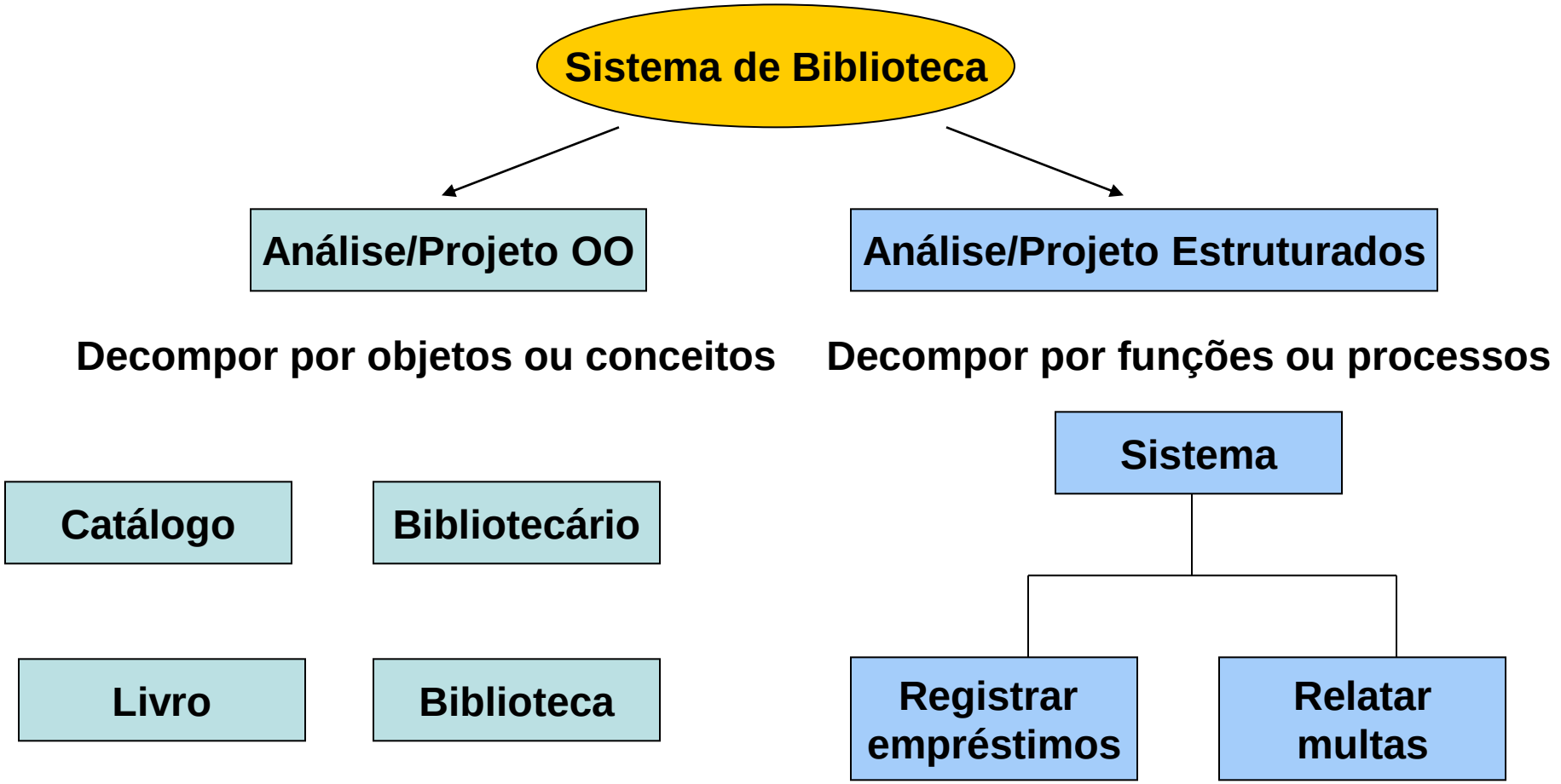


Representação do conceito na análise

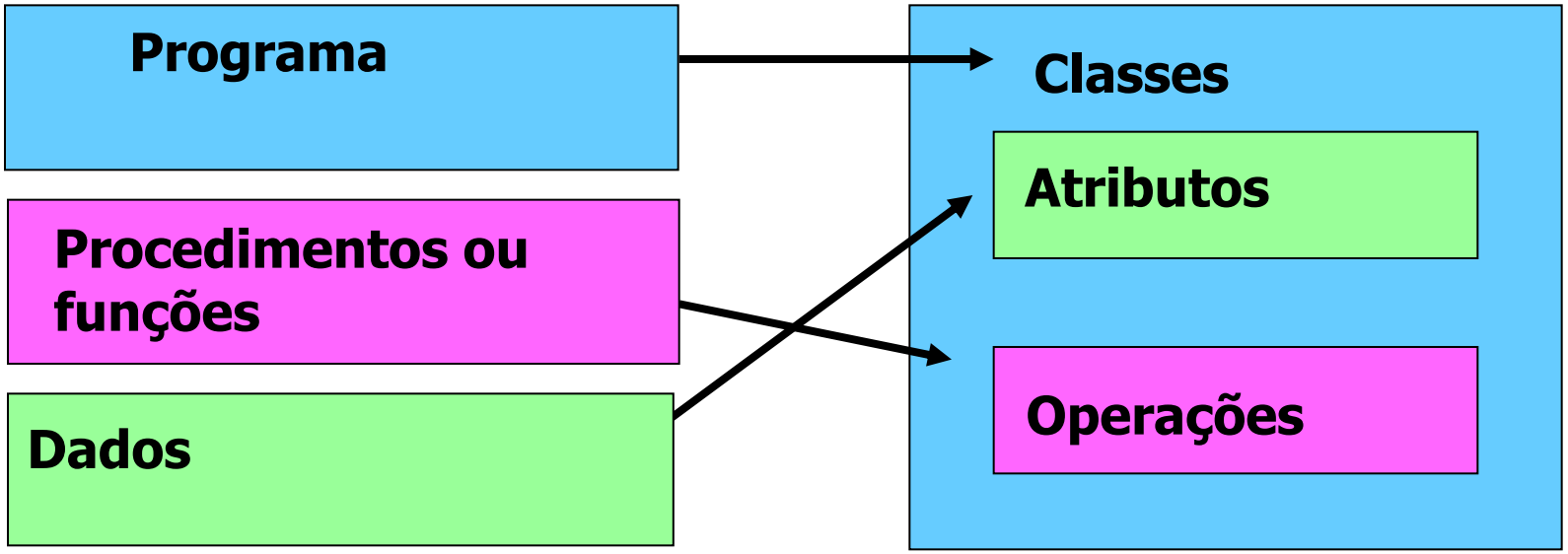
Livro
título
emprestar

```
public class Livro
{
  private String titulo;
  public void emprestar();
}
```

Desenvolvimento OO X Estruturado



Desenvolvimento OO X Estruturado



Foco em Sistema (operações ou funções)

Foco em Objeto (coisas)

Algumas Vantagens de OO

- reutilização: propiciada pelo encapsulamento dos métodos e dos dados nas classes
  - Aumenta a produtividade de sistemas
- extensibilidade: facilidade de estender o software devido a duas razões:
  - herança: novas classes são construídas a partir das que já existem
  - baixo acoplamento: as classes formam uma estrutura fracamente acoplada o que facilita alterações
- manutenibilidade: a modularização natural em classes facilita a realização de alterações no software

Algumas Vantagens de OO

- melhora de comunicação entre desenvolvedores e clientes
- maior dedicação à fase de análise, preocupando-se com a essência do sistema
  - Pode reduzir a quantidade de erros em fases posteriores
- mesma notação é utilizada desde a fase de análise até a implementação

Frente a essas vantagens, a tecnologia de OO tem provado ser "popular" e eficaz

Linguagens OO

- Existem diversas linguagens OO, tais como:
  - Smalltalk (1972)
  - Ada (1983)
  - Eiffel (~1985)
  - Object Pascal (1986)
  - Common Lisp (1986)
  - C++ (~1989)
  - Java

Linguagens orientadas a objetos

- "puras" – tudo nelas é tratado consistentemente como um objeto
  - Exemplos: Smalltalk, Eiffel, Ruby
- Projetadas para OO, mas com alguns elementos procedimentais
  - Exemplos: Java, Python
- Linguagens historicamente procedimentais, mas que foram estendidas com características OO
  - Exemplos: C++, Fortran 2003, Perl 5.

Conceitos Básicos OO

- **Orientação a Objetos** (OO): abordagem de desenvolvimento que procura explorar nosso lado intuitivo
  - Os objetos da computação são análogos aos objetos existentes no mundo real
    - Os objetos trocam mensagens entre si
    - Mensagens resultam na ativação de métodos, os quais realizam as ações necessárias
    - Os objetos que compartilham uma mesma interface, ou seja, respondem as mesmas mensagens, são agrupados em classes

Conceitos Básicos - Abstração

- **Abstração**
  - Mecanismo utilizado na análise de um domínio
  - O analista observa a realidade e dela abstrai entidades, ações, etc consideradas essenciais para uma aplicação, excluindo todos os aspectos julgados irrelevantes
  - O resultado de uma operação mental de abstração depende não tanto do fenômeno observado, mas do interesse do observador

Conceitos Básicos - Abstração

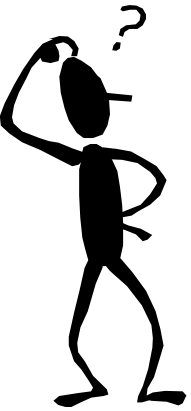
- **Abstração: Exemplo 1**

Propriedades:

Modelo,
Descrição,
Custo,
Preço Venda

Propriedades:

Tamanho, Número de Instruções, Velocidade



Serviços (Operações):

Comprar

Vender

Serviços (Operações):

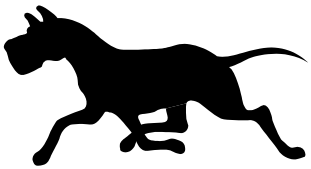
+ * / - =

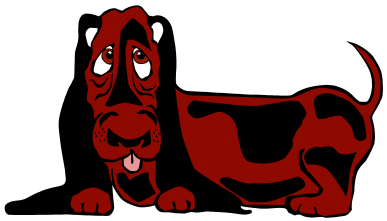
Conceitos Básicos - Abstração

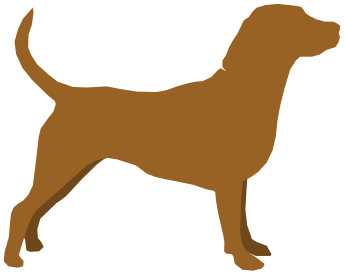
- **Abstração: Exemplo 2**

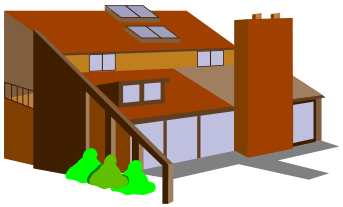
Conceitos Básicos - Objeto

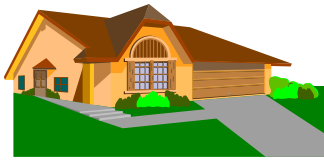
- **Objeto**
  - Tudo em OO é objeto
  - Definição (mundo do software)
    - “Qualquer coisa, real ou abstrata, a respeito da qual armazenamos dados e métodos que os manipulam” (Martin e Odell, 1995)
    - Abstração de uma entidade do mundo real de modo que essa entidade possua várias características e serviços
    - Exemplos: objeto casa, objeto cachorro











- Raça
- cor
- pedigree

CACHORRO

CASA

Conceitos Básicos - Objeto

- Objeto é algo dinâmico
- É criado por alguém, tem uma vida e morre ou é morto por alguém
- Durante a execução do sistema, os objetos podem:
  - ser construídos
  - executar ações
  - ser destruídos
  - tornar inacessíveis

Conceitos Básicos - Objeto

- **Resumindo:**
  - **Objeto:** 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

Conceitos Básicos - Métodos

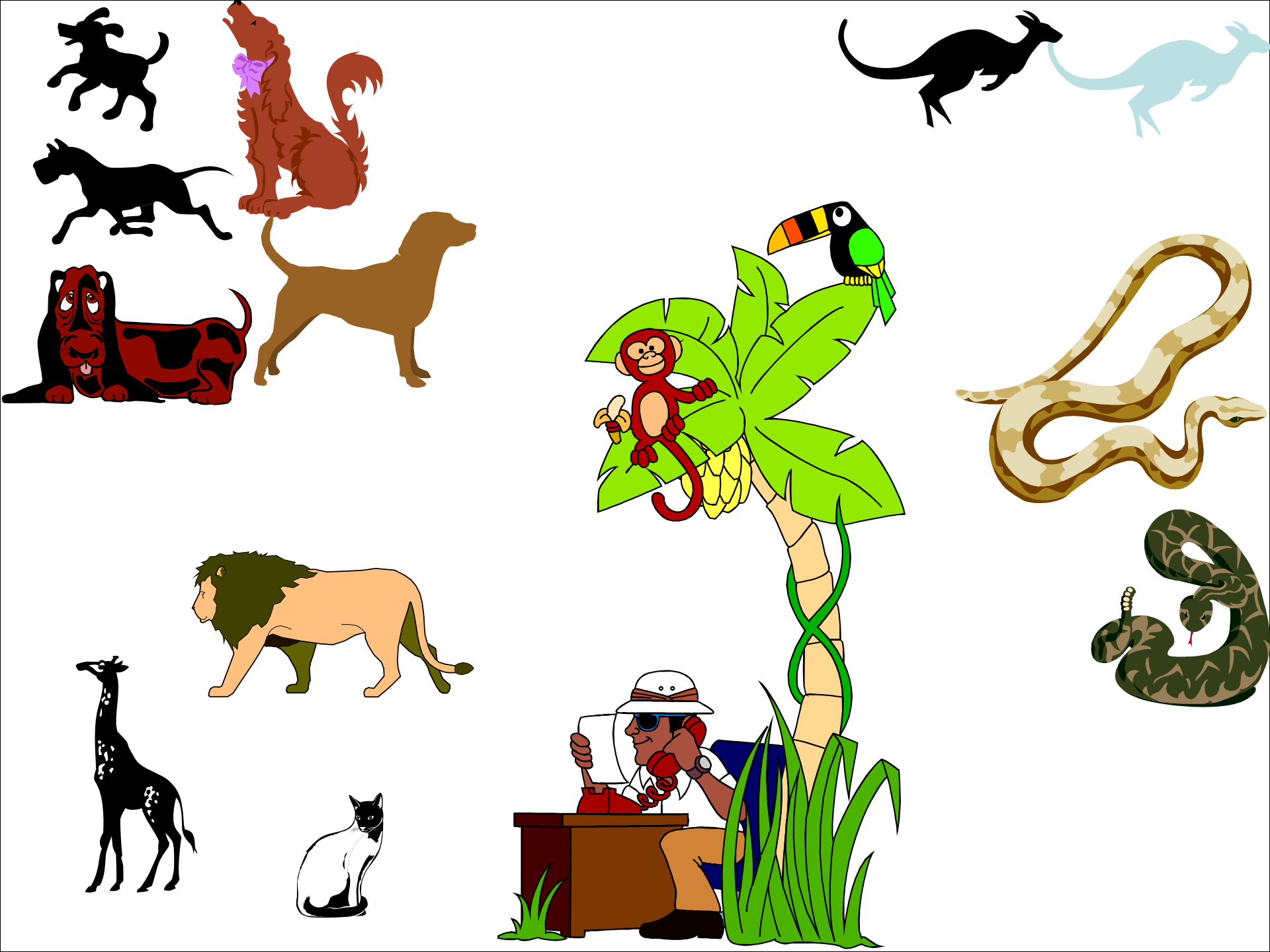
- **Métodos ou Operações:** podem mudar o estado dos objetos
  - Métodos são invocados por mensagens
  - Cada objeto possui seu próprio conjunto de métodos
- **Definições**:
  - São procedimentos definidos e declarados que atuam sobre um objeto
  - Descrição de uma sequência de ações a serem executadas por um objeto
  - Por meio dos métodos que especifica-se a um objeto COMO FAZER alguma coisa
  - São intrínsecos aos objetos e não podem ser separados

Conceitos Básicos - Classe

- Classe
  - abstração de um conjunto de objetos similares do mundo real
  - conjunto de objetos que possuem propriedades semelhantes (ATRIBUTOS), o mesmo comportamento (MÉTODOS), os mesmos relacionamentos com outros objetos e a mesma semântica

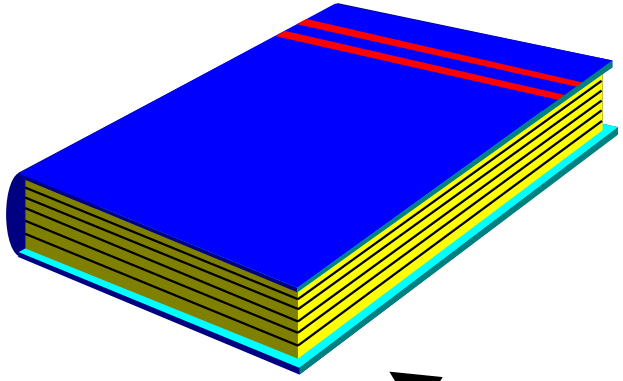
Conceitos Básicos - Classe

- 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
- Exemplo
  - Classe de espécies em Zoologia



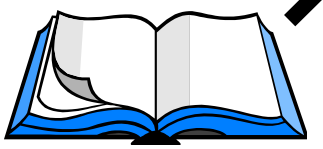
Conceitos Básicos - Classe

- Exemplo 1

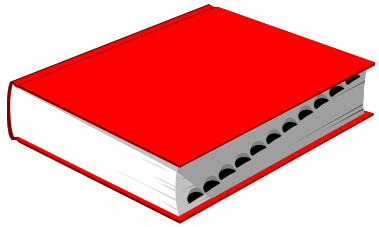


Livro

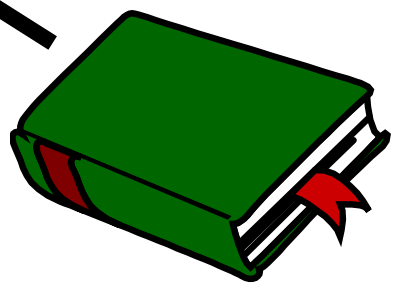
Objetos



Bíblia



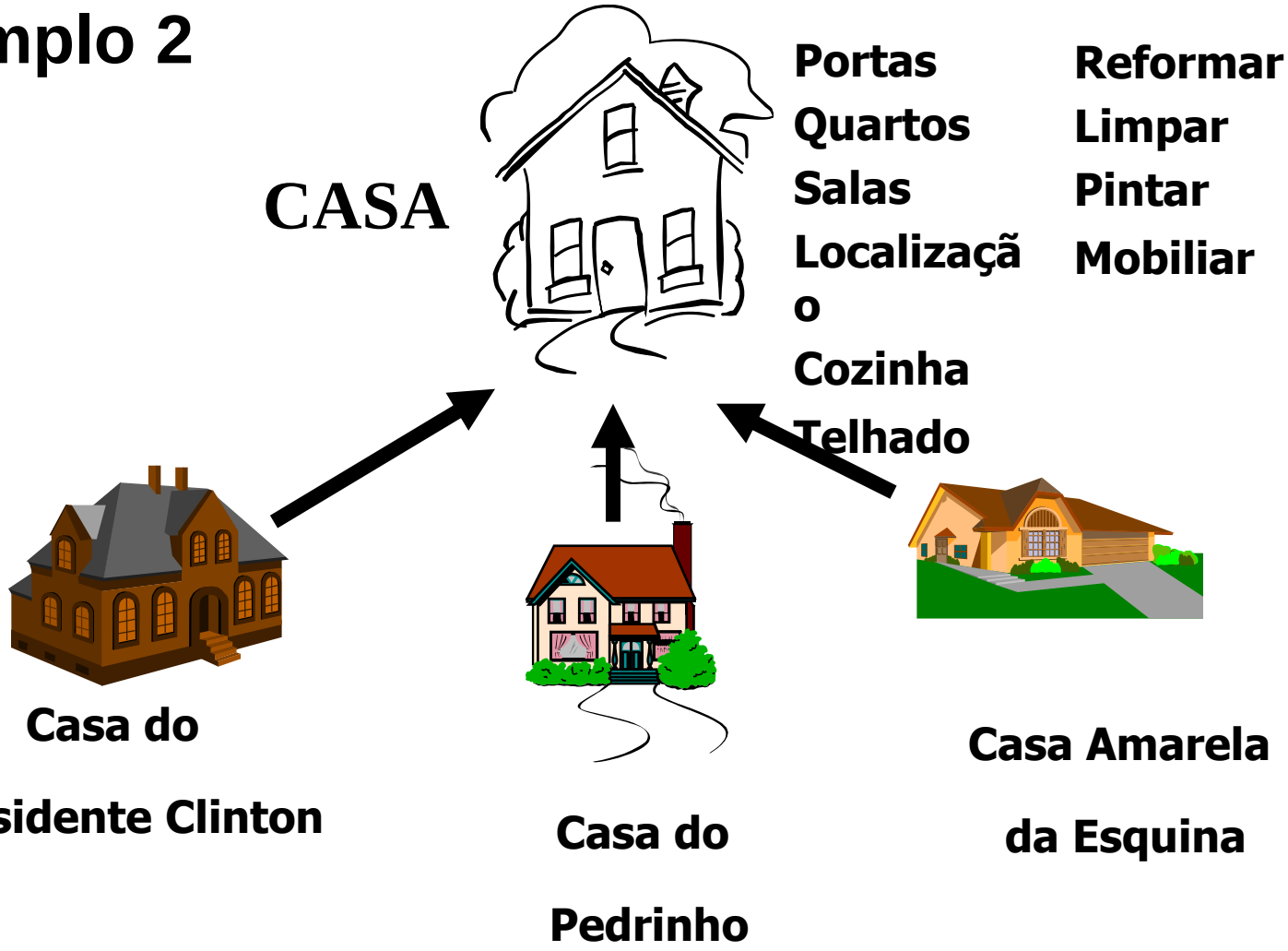
Relatório



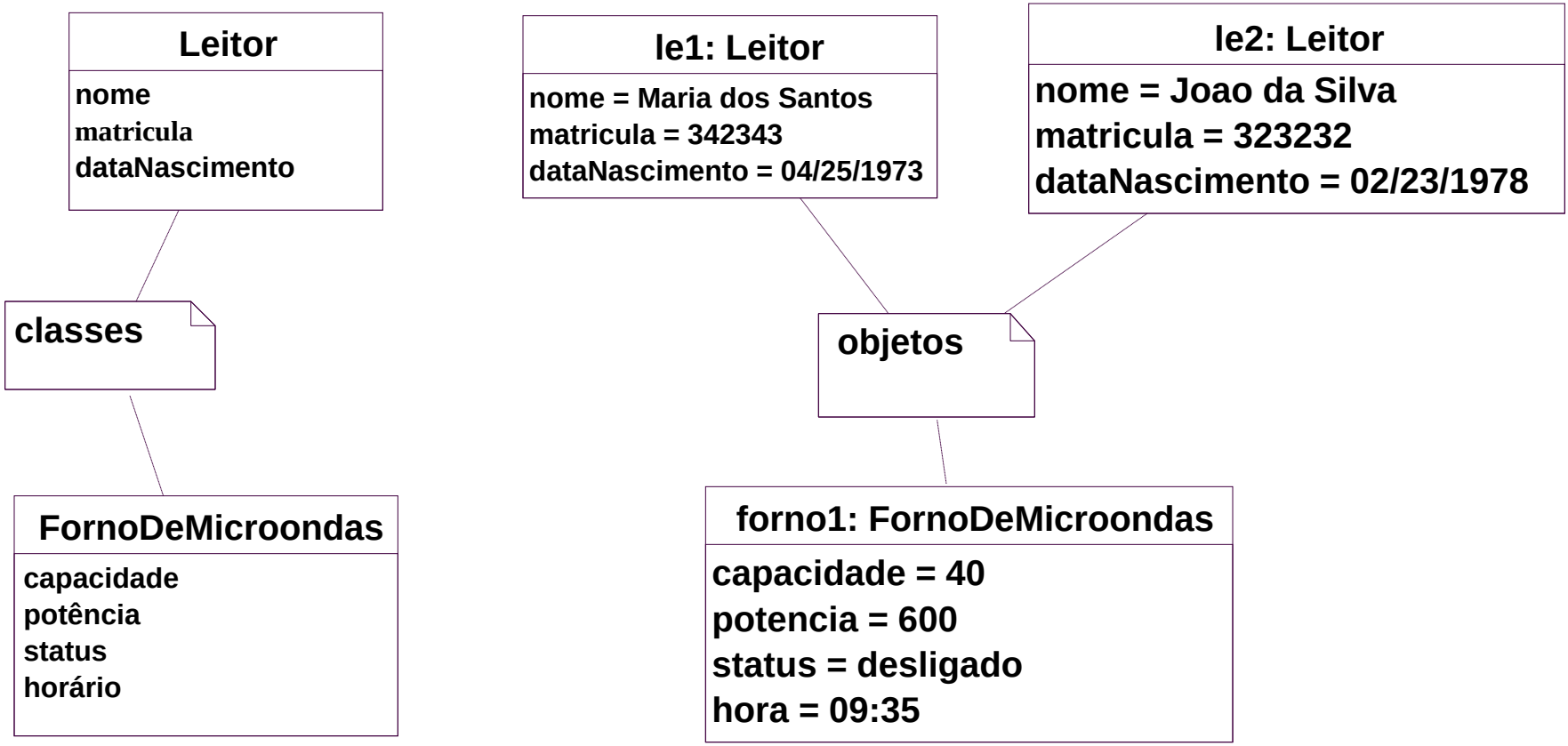
Dicionário

Conceitos Básicos - Classe

- **Exemplo 2**

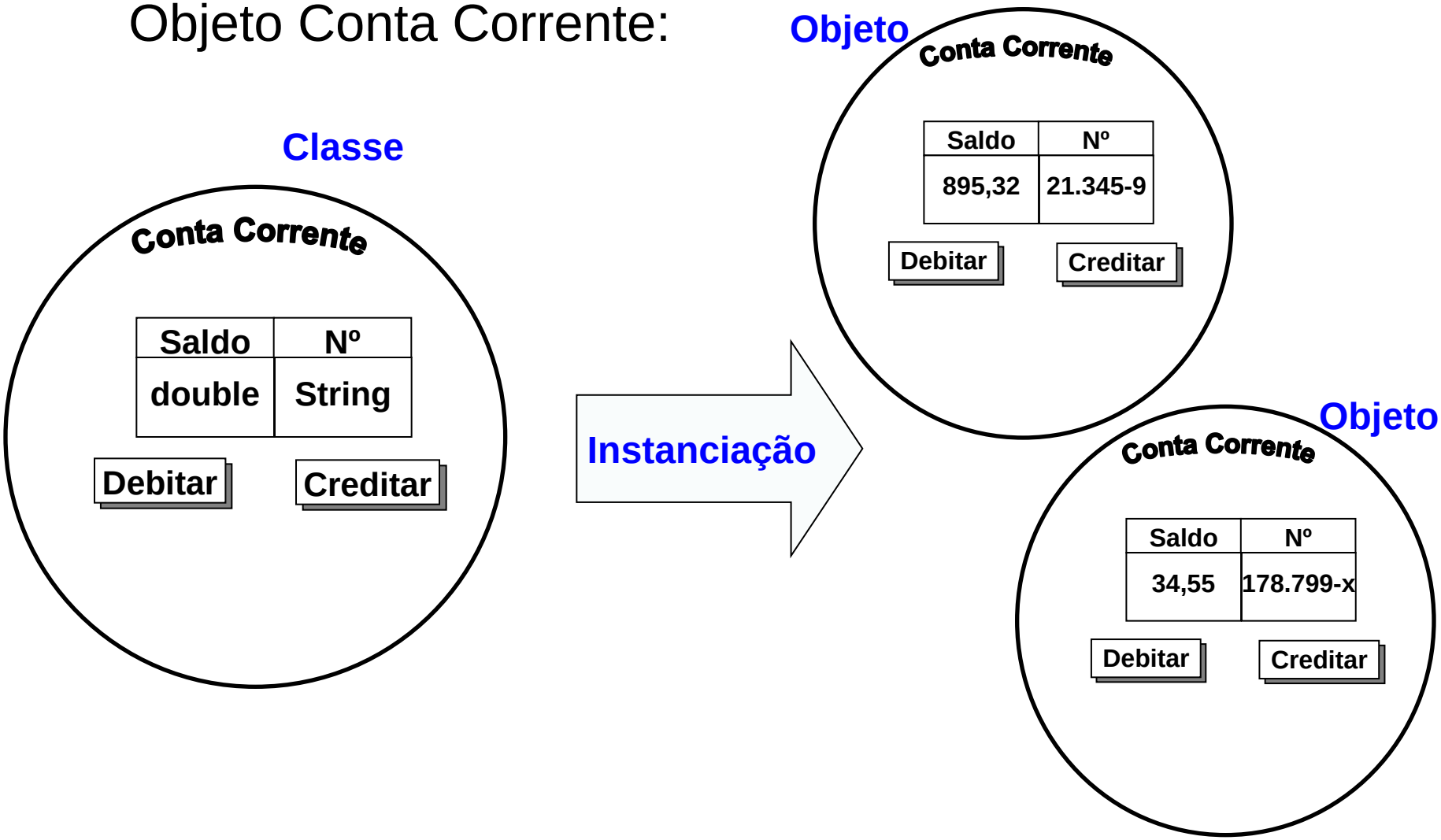


Objetos e Classes: Exemplos



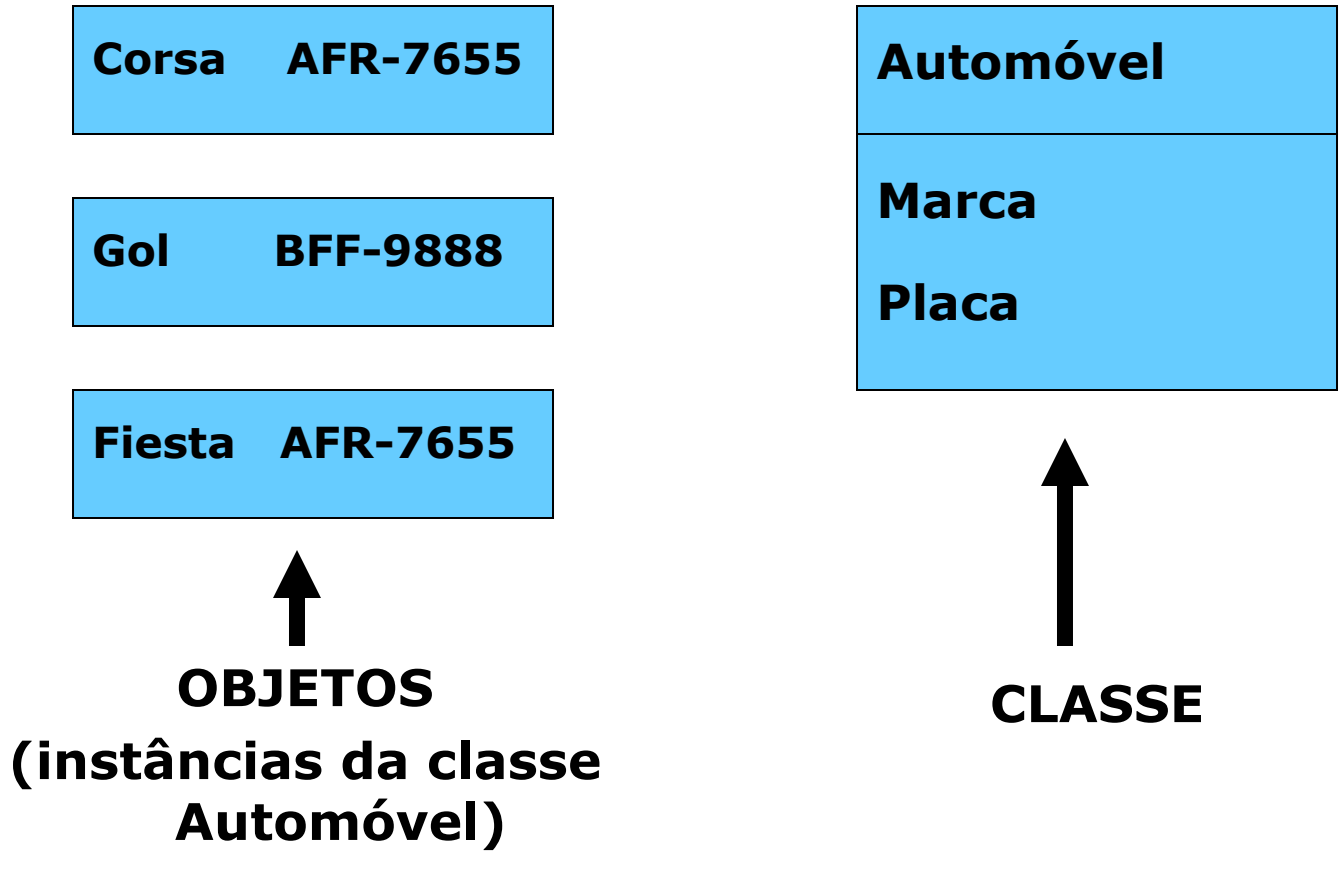
Classes e Objetos

Objeto Conta Corrente:



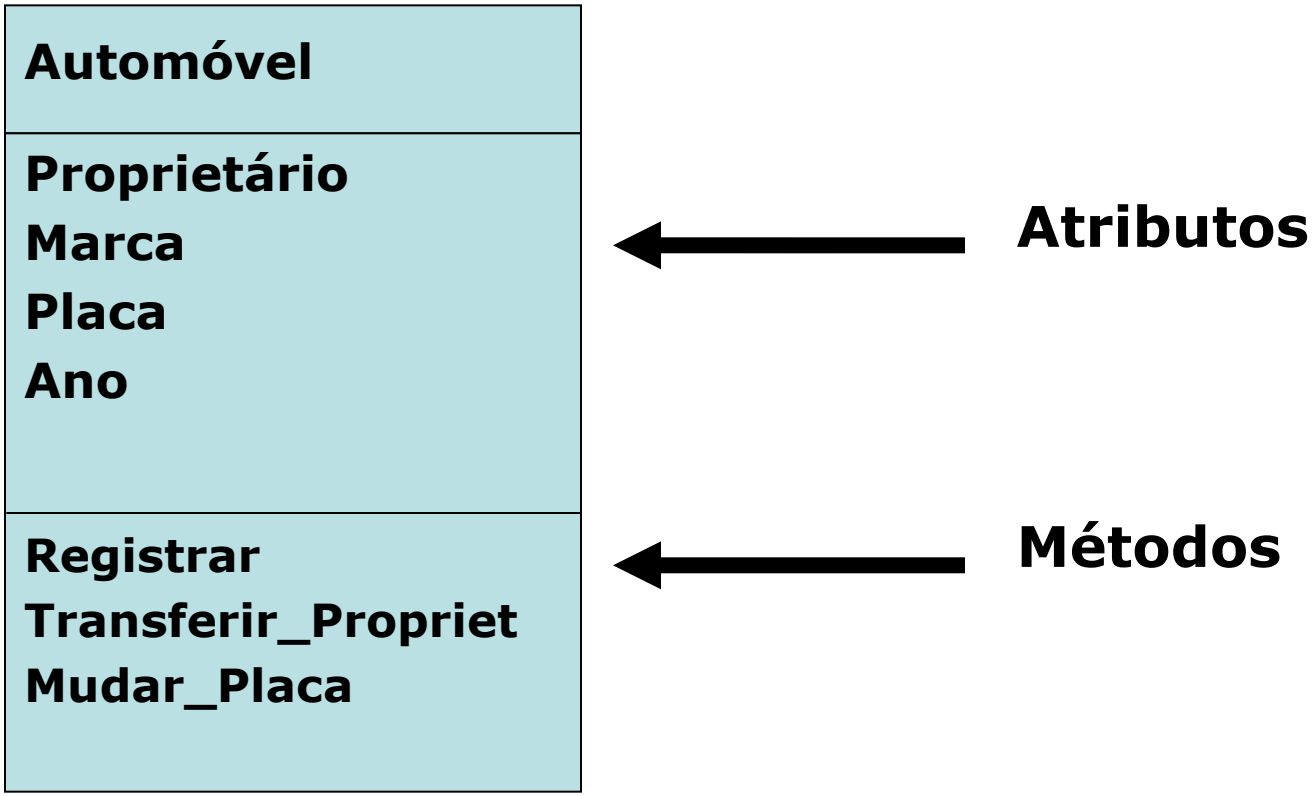
Conceitos Básicos – Exemplos

- **Exemplo 3**



Conceitos Básicos - Exemplos

- Atributos e Métodos



Conceitos Básicos – Exemplo

- Descrição da Classe Automóvel em Java

```
public class Automovel{
    private String proprietario;
    private String marca;
    private String placa;
    private int ano;

    public Automovel ();
    public boolean registrar ();
    public void transferir_proprietario(String
      novoProprietario);
    public void mudar_placa (String novaPlaca);
}
```

Conceitos Básicos - Encapsulamento

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

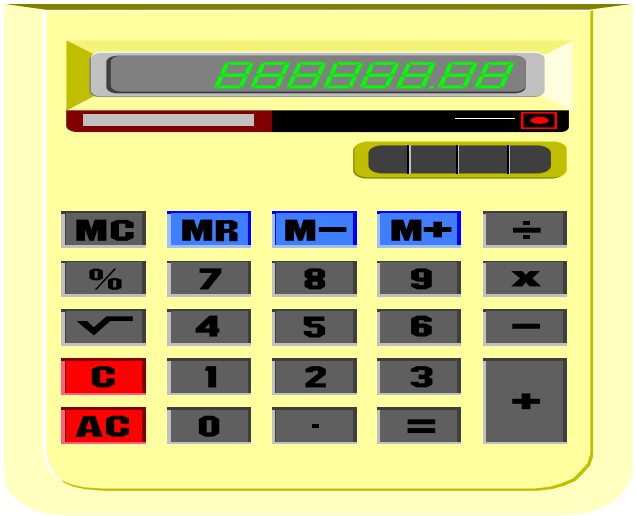
Conceitos Básicos - Encapsulamento

- Objetos encapsulam seus atributos
  - atributos de uma classe são acessíveis apenas pelos métodos da própria classe
  - outras classes só podem acessar os atributos de uma classe invocando os métodos públicos (métodos *getters* e *setters*)

- Restringe a visibilidade do objeto mas facilita o reúso, aumenta a legibilidade e manutenibilidade

Conceitos Básicos - Encapsulamento

- Exemplo



297 + 333 = 630

Conceitos Básicos - 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
    - 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

Conceitos Básicos - Mensagem

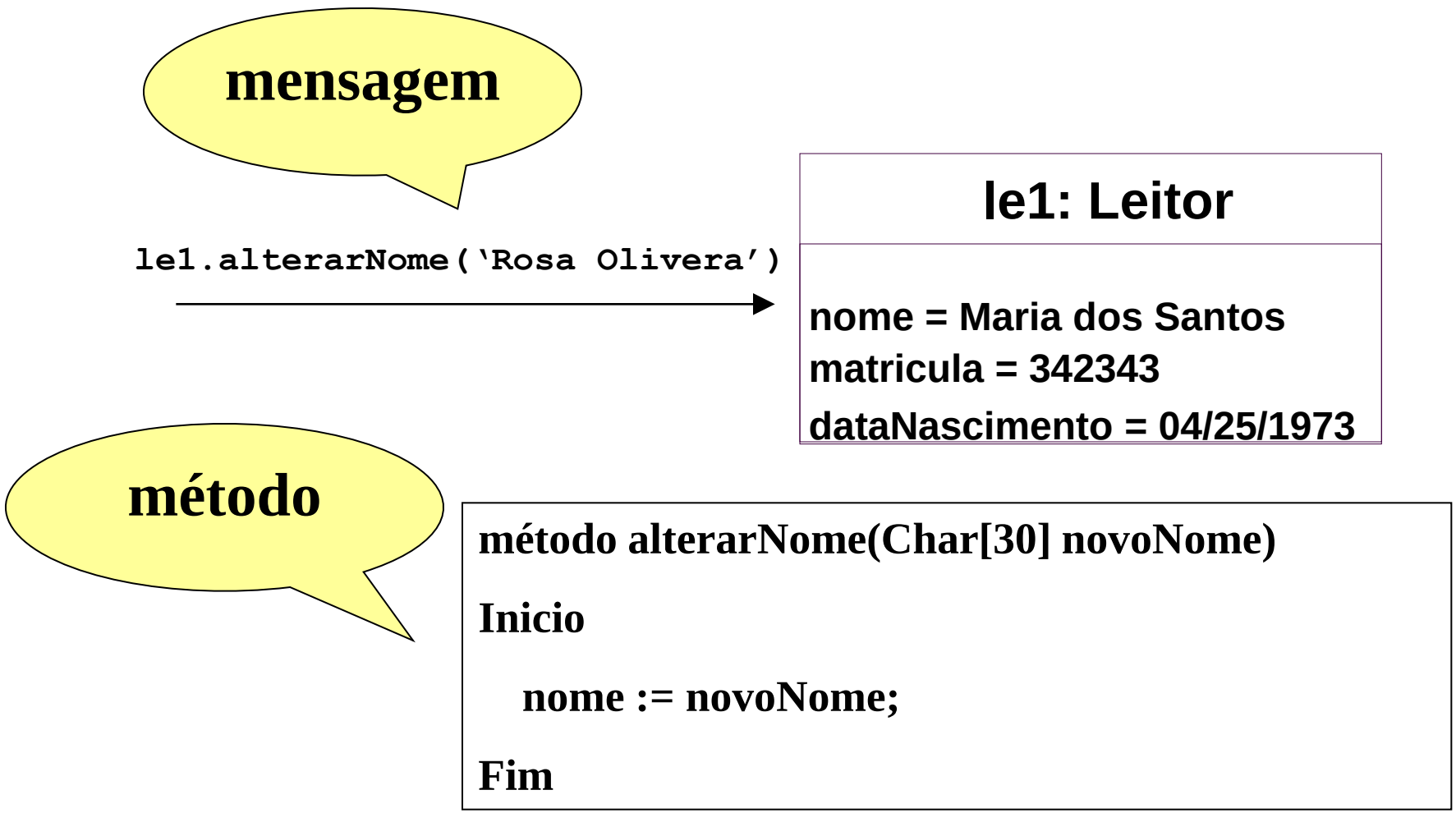
- Um objeto (Emissor) envia uma mensagem a outro (Receptor) que executará o serviço
- Métodos são invocados por Mensagens
- **Exemplo**
  - A chamada de um procedimento/função em LP é uma aproximação inicial de uma mensagem, como em:

    P(10,20), onde:

    P é o seletor e os valores 10 e 20 são os parâmetros

  - Diferença na OO:
    - a ação da mensagem a ser ativada depende essencialmente do objeto que receber a mensagem

Métodos X Mensagem

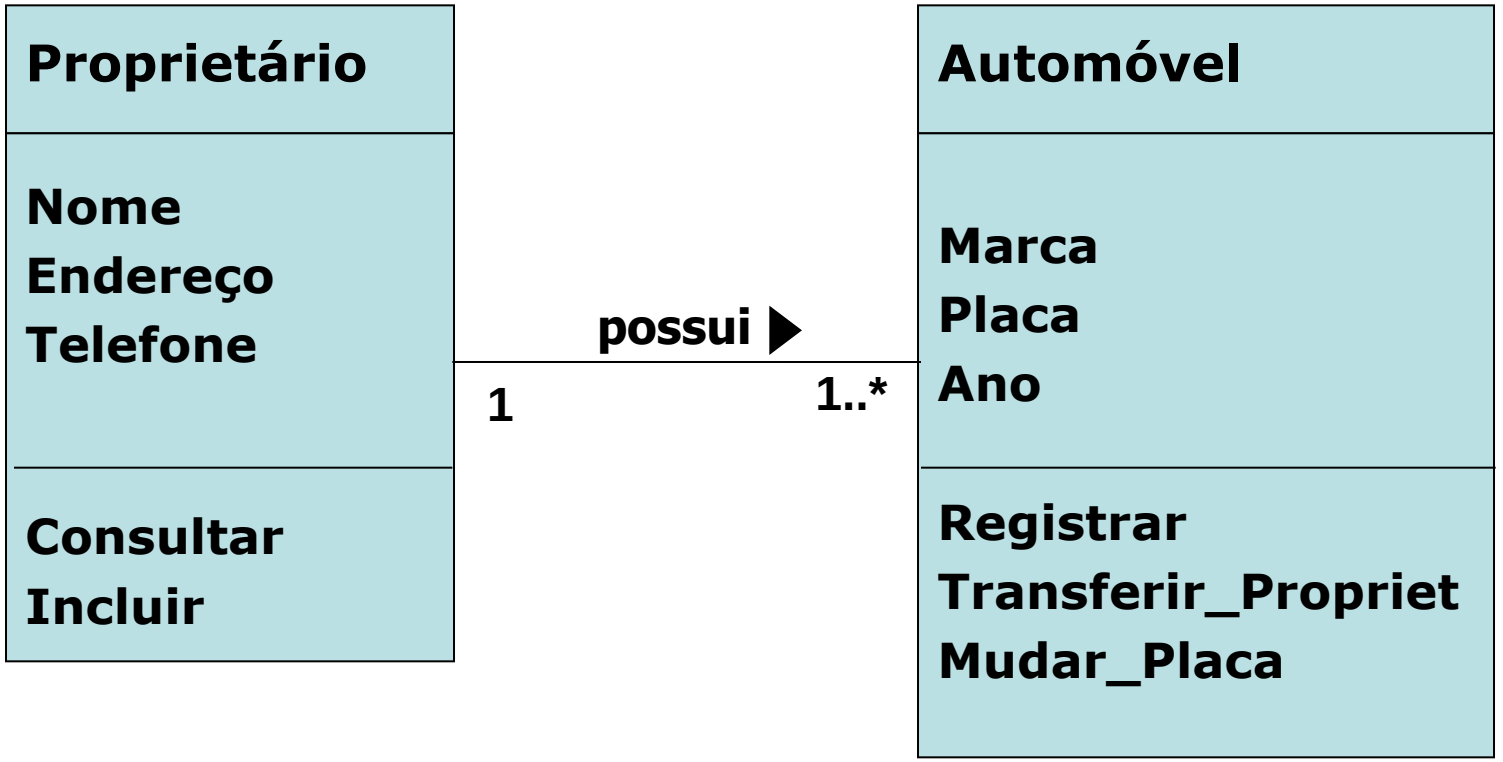


Conceitos Básicos - Relacionamento

- 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

Conceitos Básicos - Relacionamento

- Associação
  - um relacionamento que um **objeto** precisa ter com outro(s) **objeto(s)**, para cumprir suas responsabilidades



Conceitos Básicos - Herança

- Mecanismo que permite definir uma nova classe (subclasse) a partir de uma classe já existente (superclasse)

- A subclasse herda as características comuns da superclasse (atributos e métodos)
  - A subclasse pode adicionar novos atributos e métodos, como também reescrever métodos herdados

Conceitos Básicos - Herança

- 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

Conceitos Básicos - Herança

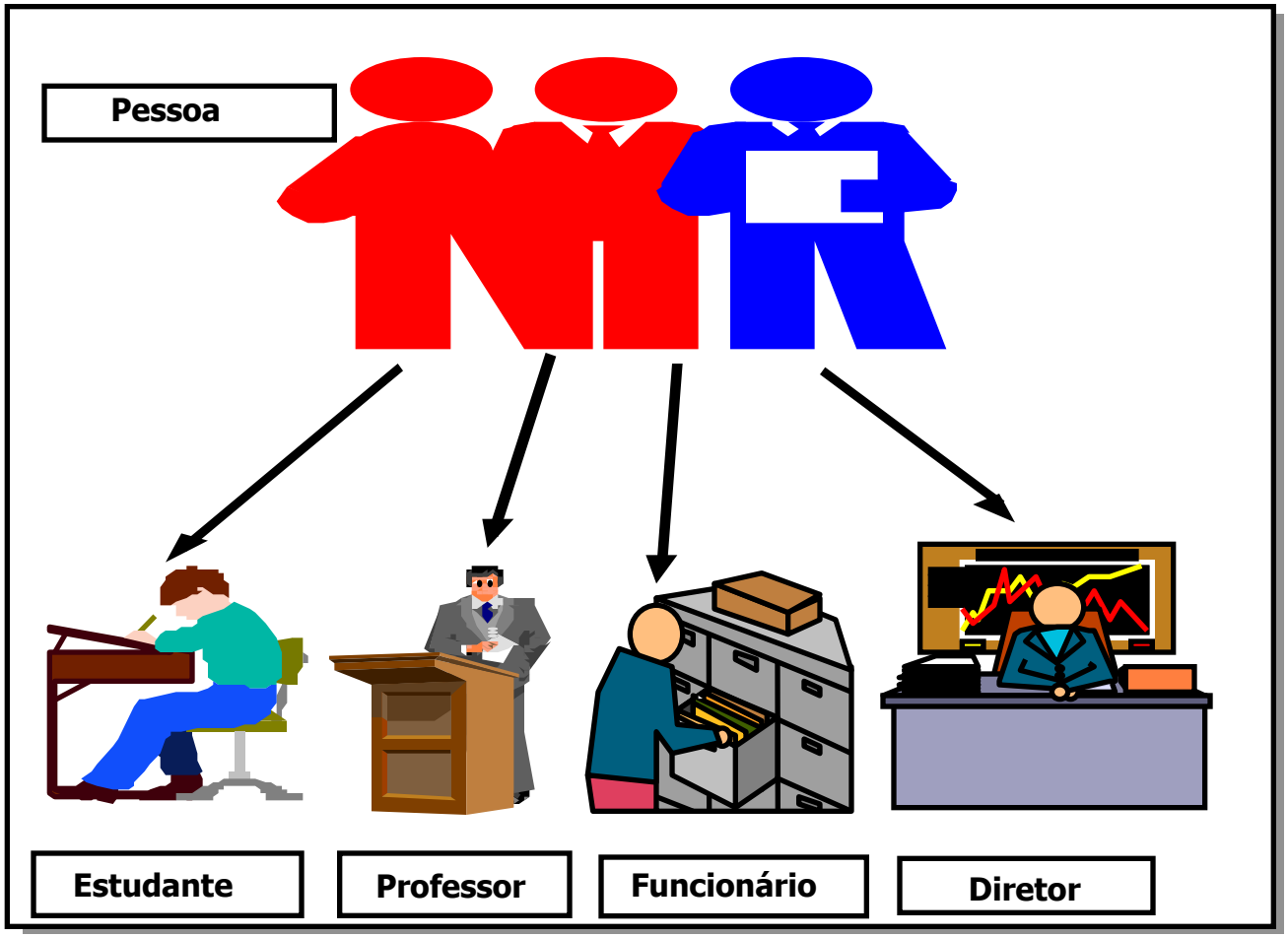
- A Herança pode ser de dois tipos:
  - Herança Simples: quando uma classe é subclasse de somente uma superclasse
  - Herança Múltipla: quando uma classe é subclasse de várias superclasses e, conseqüentemente, herda as características de cada uma delas

Conceitos Básicos - Herança

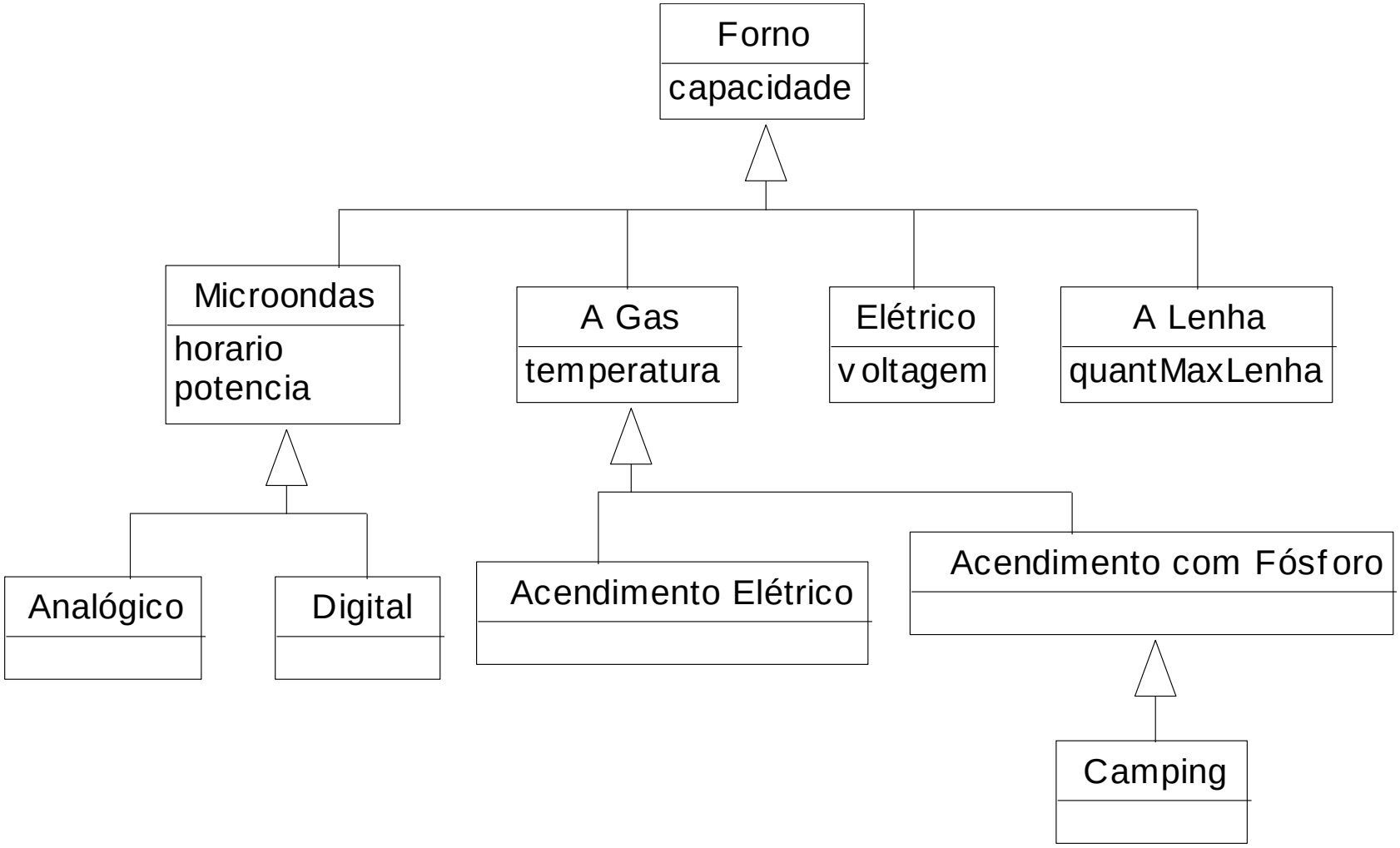
- Exemplo de Herança Simples
  - *Estudante* e *Professor* são subclasses de *Pessoa*
    - Herdam as propriedades de Pessoa.
    - Estudante possui características específicas: curso, ano letivo, boletim, etc.
    - Professor possui características específicas: titulação

Conceitos Básicos - Herança

- **Dica:**
  - **É um..**

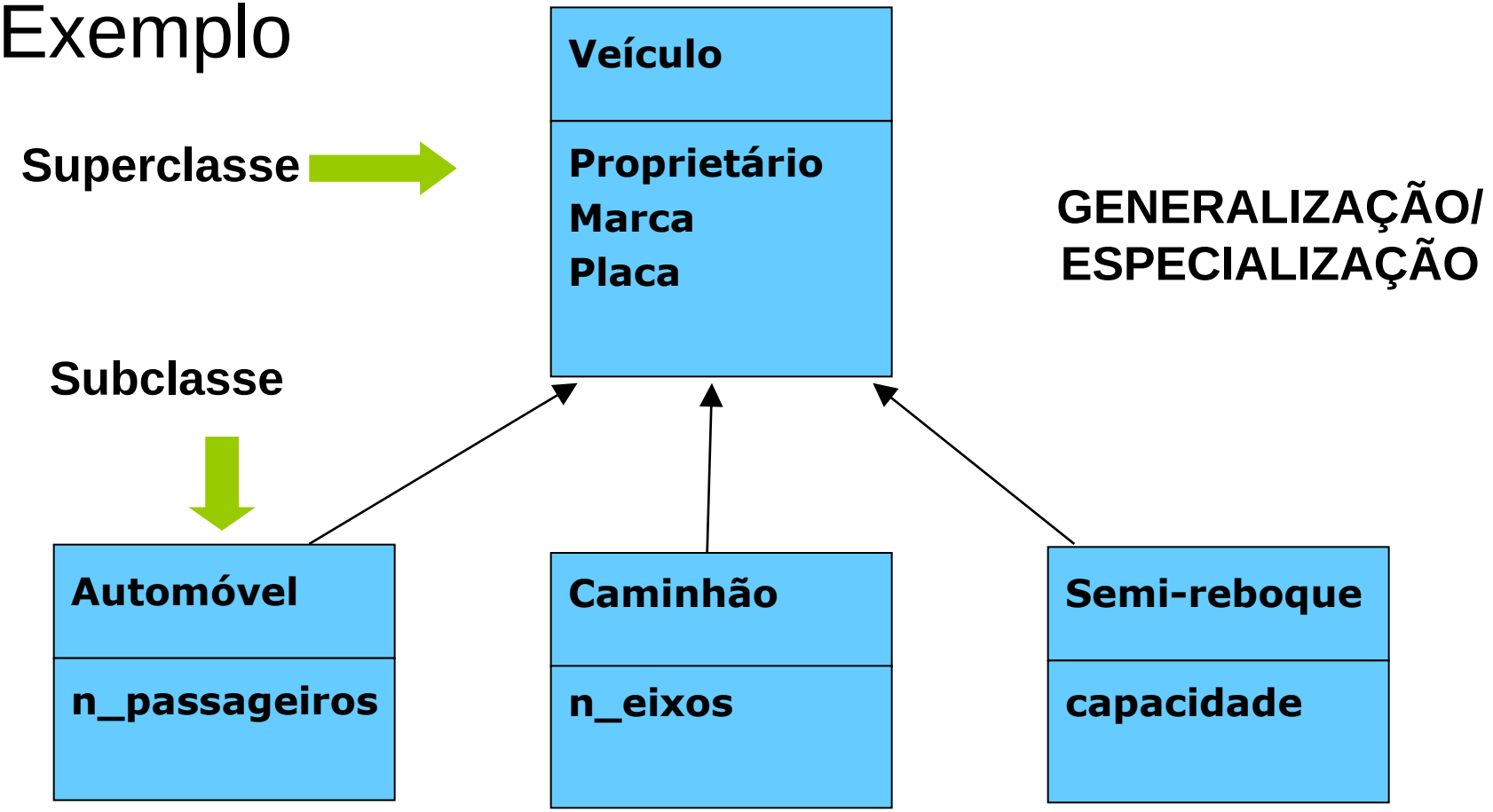


Conceitos Básicos - Herança



Conceitos Básicos - Herança

- Exemplo



Generalização

Reformar
Limpar
Pintar
Mobiliar

Portas
Salas
Cozinha

Quartos
Localização
Telhado

(Superclasse)

CASA

PRAIA

FAVELA

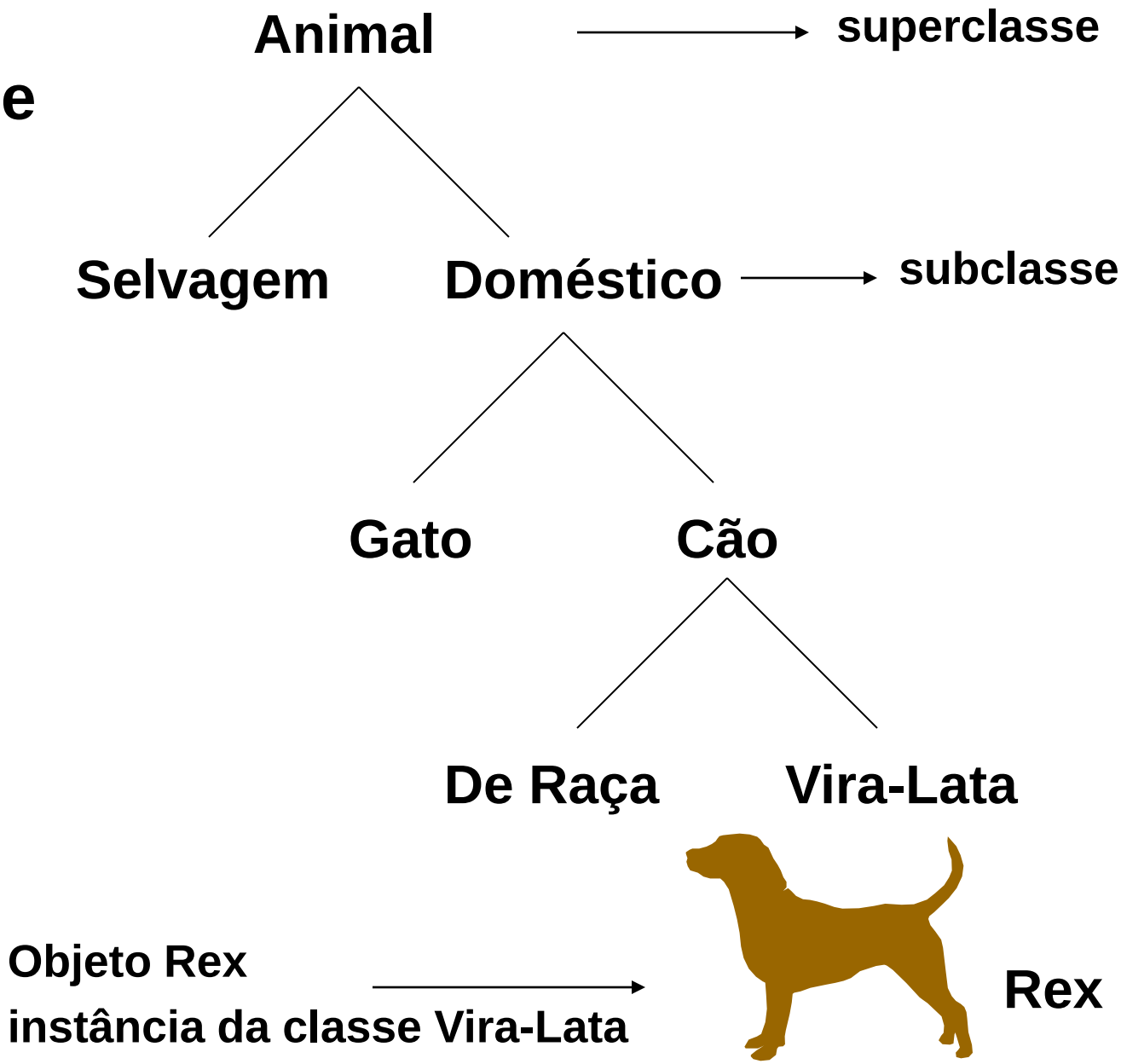
MANSÃO

(Subclasses)

Limpar Piscina
Contratar Criadagem
Piscina
Quadras

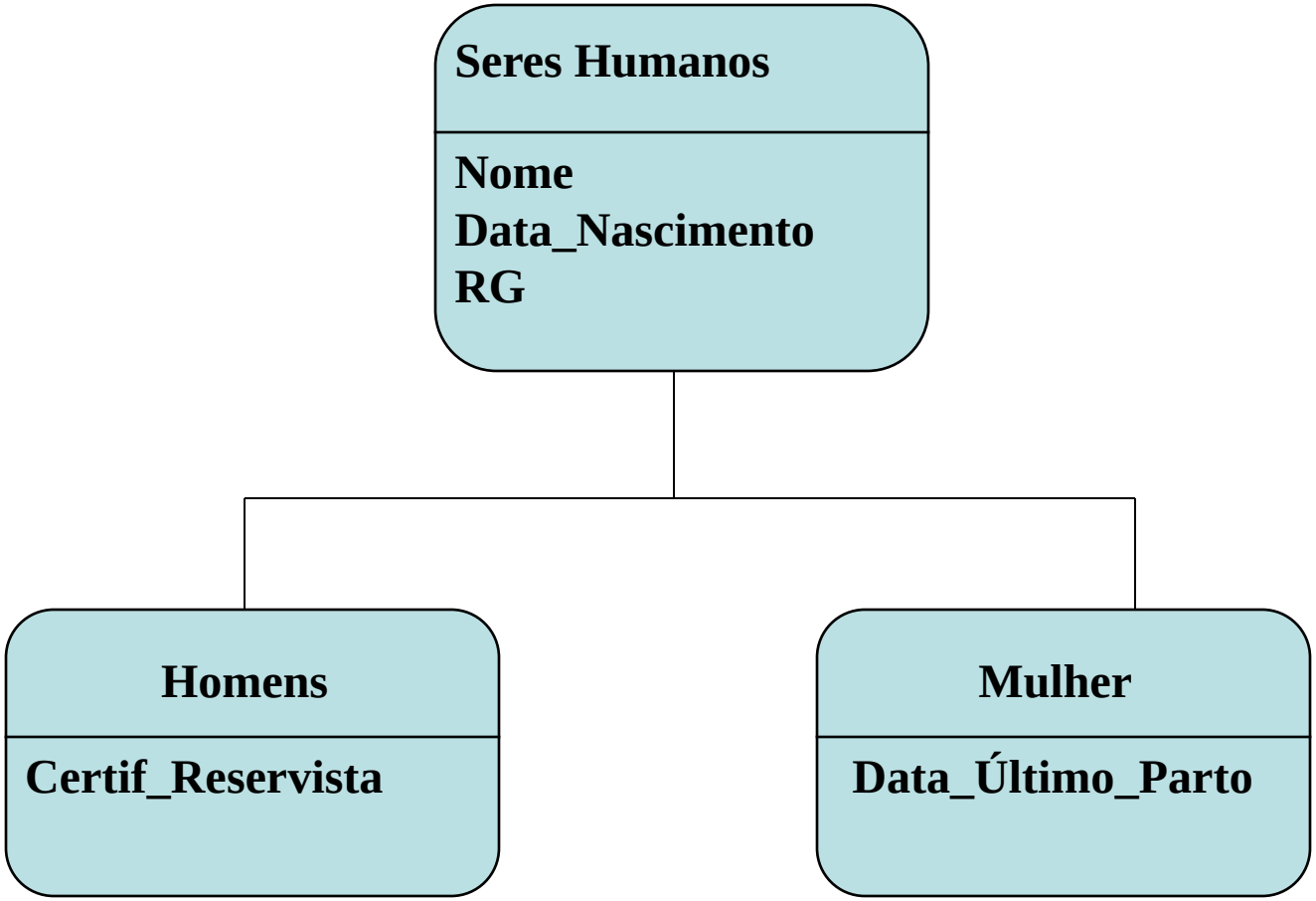
Especialização

Exemplo de Hierarquia de Classes



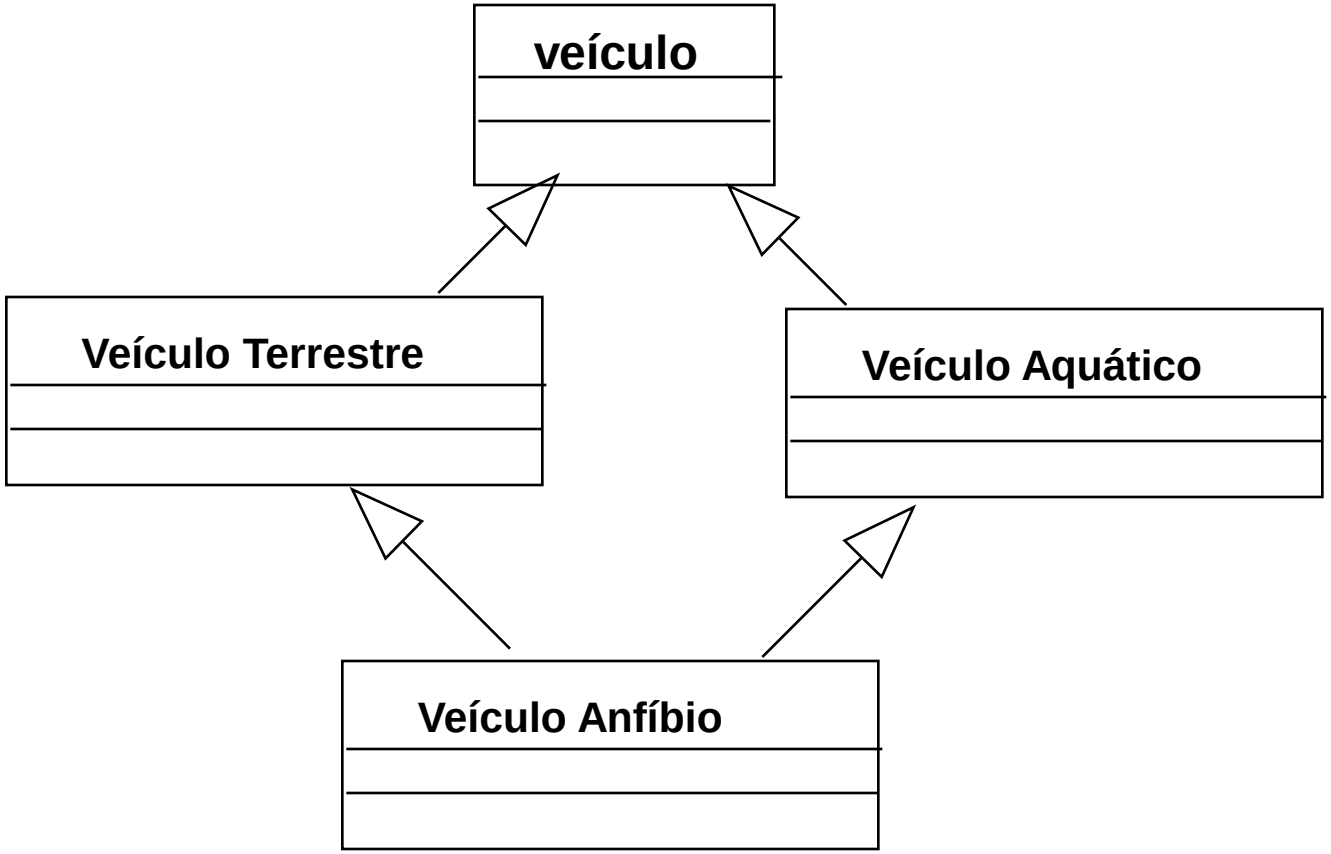
Conceitos Básicos - Herança

- **Exemplo**



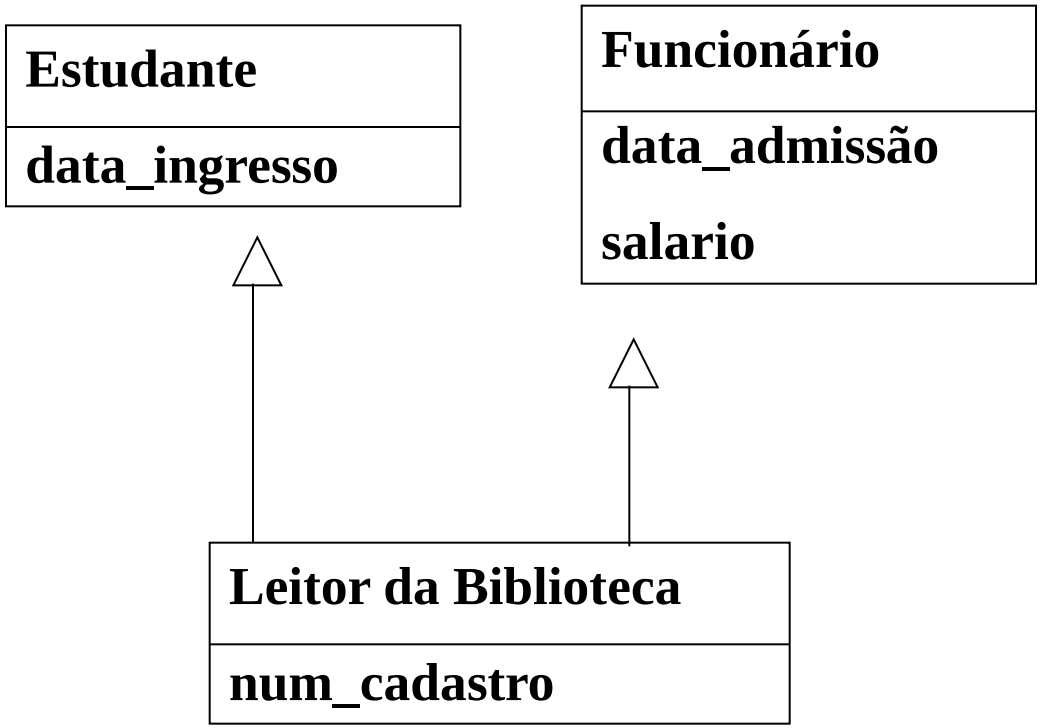
Conceitos Básicos - Herança Múltipla

- Exemplos de Herança Múltipla

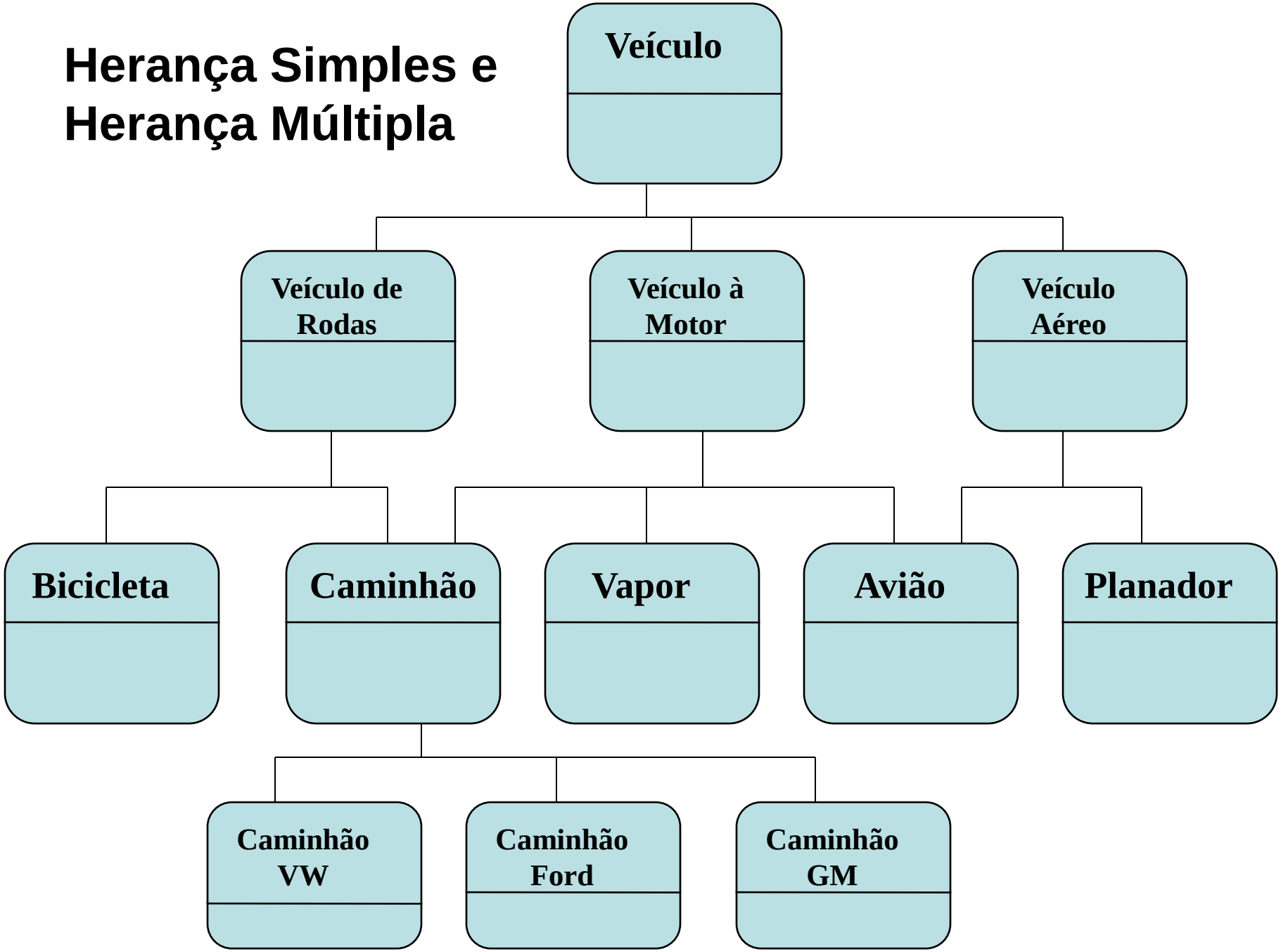


Herança Múltipla

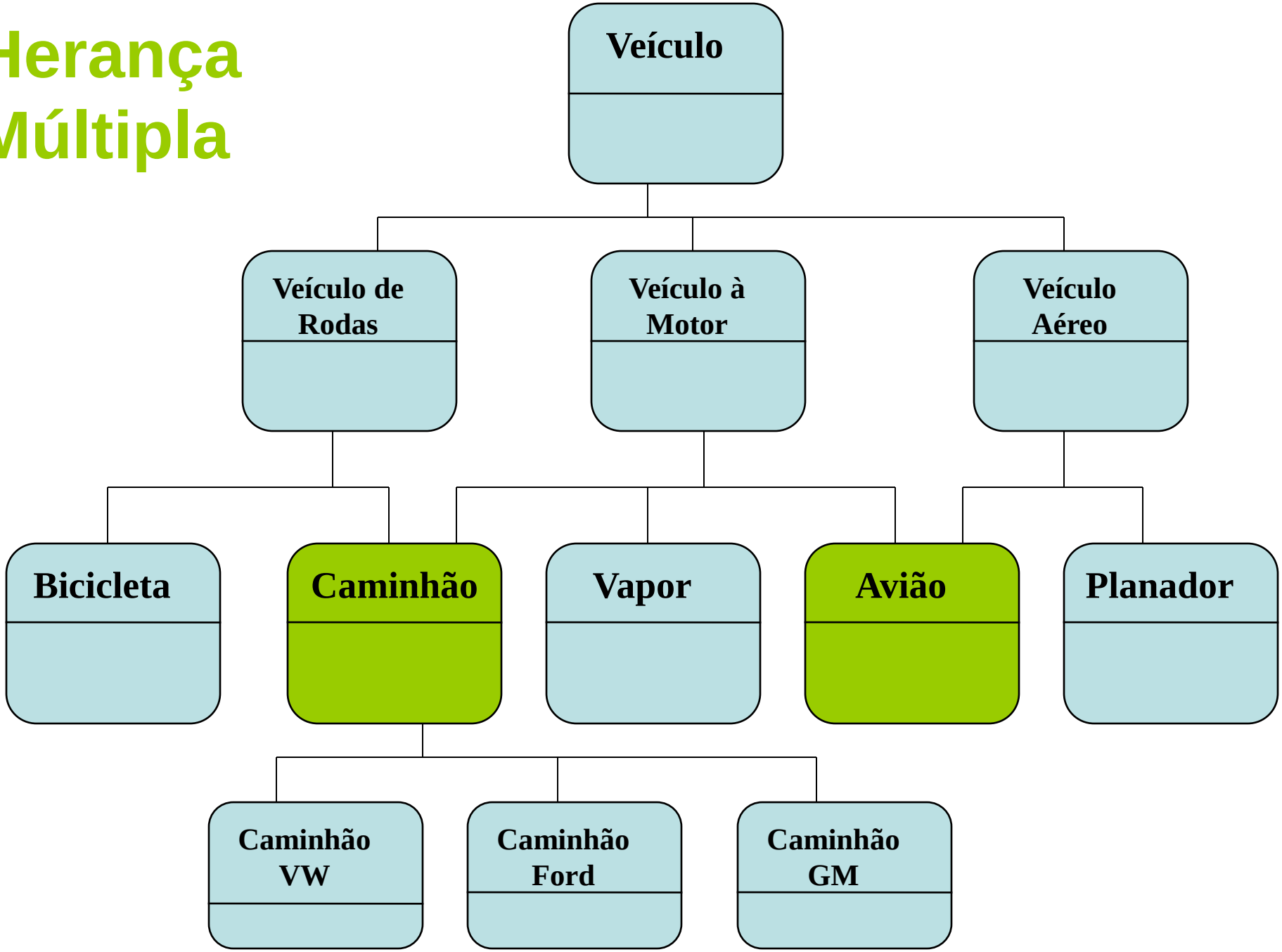
Existe mais de uma superclasse, ou seja, uma classe é declarada como uma subclasse de uma ou mais superclasses



Herança Simples e Herança Múltipla



Herança Múltipla

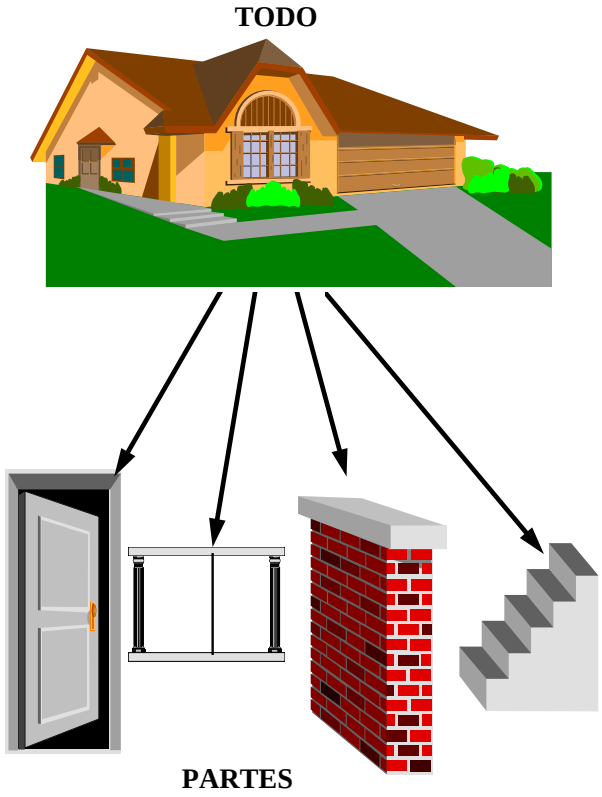


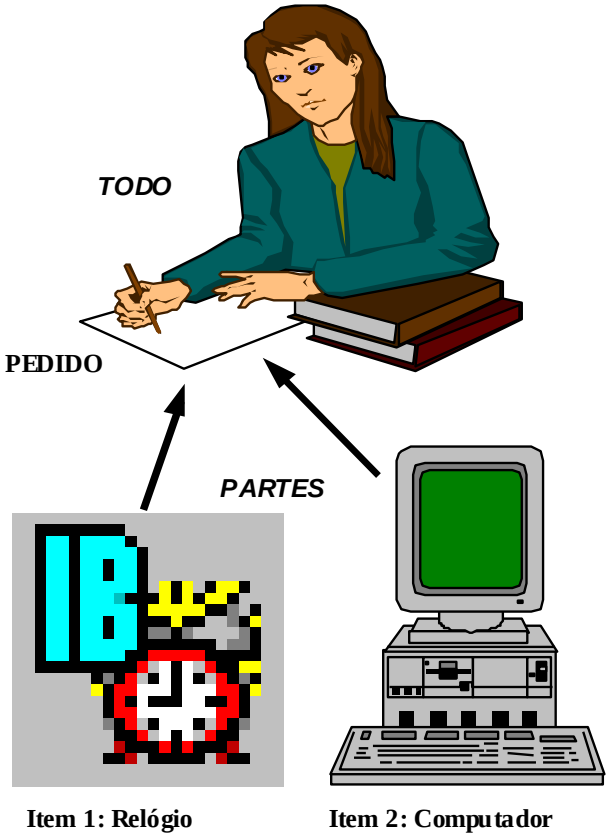
Conceitos Básicos – Todo-Parte

- Todo-Parte
  - permite a construção de uma classe agregada (todo) a partir de outras classes componentes (parte)
  - Dica: É parte de...

Conceitos Básicos – Todo-Parte

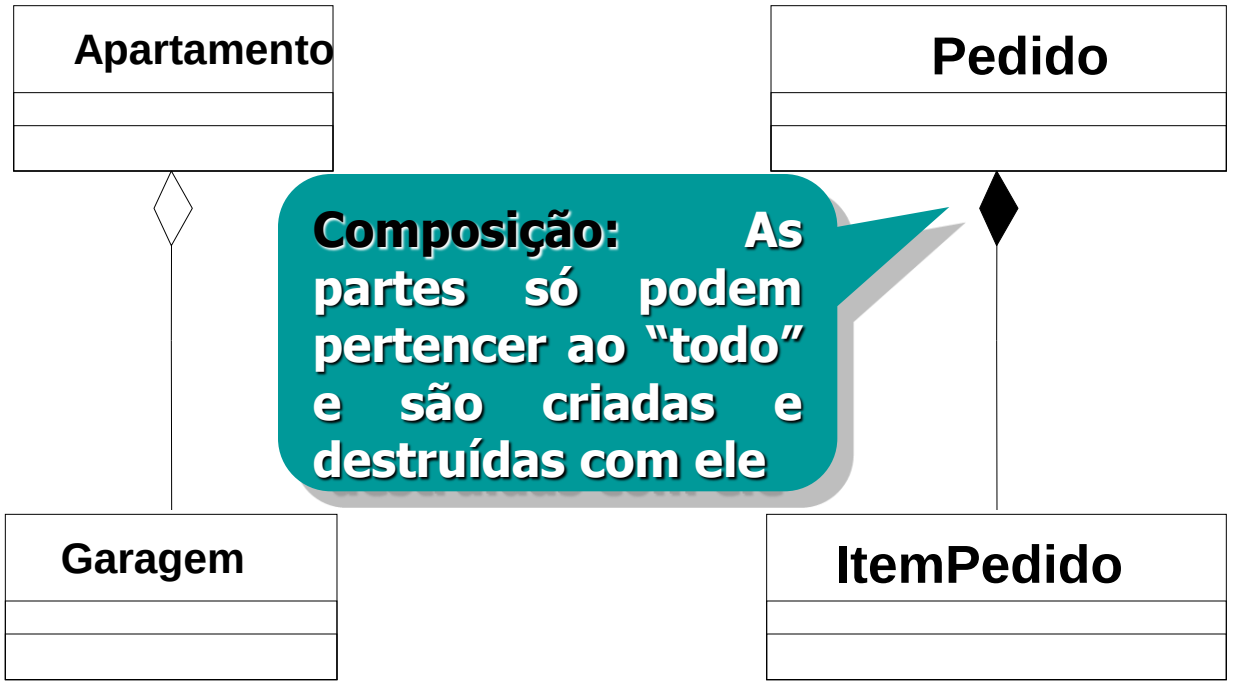
- Exemplo





Conceitos Básicos – Todo-Parte

- **Relacionamento de Agregação/Composição**
  - **Tipo especial de associação (é parte de)**
  - **Agregação (ou agregação simples)**
  - **Composição (ou agregação composta)**



Conceitos Básicos
Polimorfismo

- A palavra “polimorfismo” é derivada do grego e significa “muitas formas” ou “tendo muitas formas”

Conceitos Básicos
Polimorfismo de Inclusão (*Overriding*)

- Redefinição de um método em classes diferentes (dentro de uma hierarquia de herança) com a mesma assinatura
  - métodos possuem comportamento diferente (implementação diferente)
  - ao receber uma mensagem para efetuar uma operação, é o objeto quem determina como a operação deve ser efetuada

Conceitos Básicos
Polimorfismo de Sobrecarga (*Overloading*)

- Capacidade de fornecer o mesmo nome a mais de um método em uma mesma classe
  - possível desde que a assinatura seja diferente, podendo ocorrer entre métodos da mesma classe

- O método a ser executado é selecionado em tempo de execução
  - a assinatura do método chamado é observada e seleciona-se algum que satisfaça a assinatura

Conceitos Básicos
Polimorfismo de Sobrecarga (*Overloading*)

- **Exemplos**

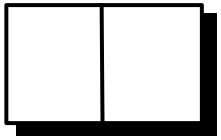
```
public int calcularQuadrado(int x)
{
      return x * x;
}

public double calcularQuadrado(double y)
{
      return y * y;
}
```

Conceitos Básicos
Polimorfismo de Sobrecarga (*Overloading*)

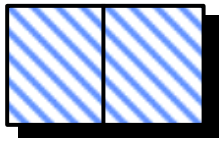
- **Exemplos**

Janela ( )

Janela (1 x 2 , 2)



Janela (1 x 2 , 2, Azul)



Conceitos Básicos – Analogia

- Analogia dos conceitos principais dos paradigmas: OO e tradicional