

# Paradigmas de Programação

---

## Programação Orientada a Objetos

Prof. Saulo Popov Zambiasi  
*saulopz@gmail.com*

# Histórico

---

- # Final da década de 60, quando dois cientistas dinamarqueses criaram a linguagem Simula.
- # 1967 - Linguagem de Programação Simula-67-  
conceitos de classe e herança
- # 1983 – Introdução do termo POO com a linguagem Smalltalk
- # FINS DOS ANOS 80: Paradigma de Orientação a Objetos
  - abordagem poderosa e prática para o desenvolvimento de software

# Histórico de OO

---

## # Surgiram linguagens híbridas:

- C++ (1986), Object-Pascal (1986)

## # Surgiram Métodos de Análise e Projeto OO

- 1989 **CRC** – Class Responsibility Collaborator, Beecke e Cunningham
- 1990 **OOA** – Object Oriented Analysis, Coad e Yourdon
- 1991 **Booch**
- 1991 **OMT** – Object Modeling Technique, Rumbaugh
- 1992 **Objectory** – Jacobson
- 1994 **Fusion** – Coleman
- 1997 **UML** – Unified Modeling Language

# Vantagens da OO

---

- # **Abstração de dados:** os detalhes referentes às representações das classes serão visíveis apenas a seus atributos;
- # **Compatibilidade:** as heurísticas para a construção das classes e suas interfaces levam a componentes de software que são fáceis de se combinar;
- # **Flexibilidade:** as classes delimitam-se em unidades naturais para a alocação de tarefas de desenvolvimento de software;

# Vantagens de OO

---

- # **Reutilização:** o encapsulamento dos métodos e representação dos dados para a construção de classes facilitam o desenvolvimento de software reutilizável, auxiliando na produtividade de sistemas;
- # **Extensibilidade:** facilidade de estender o software devido a duas razões:
  - **herança:** novas classes são construídas a partir das que já existem;
  - as classes formam uma estrutura fracamente acoplada o que facilita alterações;
- # **Manutenibilidade:** a modularização natural em classes facilita a realização de alterações no software.

# Vantagens de OO

---

- # Melhoria de comunicação entre desenvolvedores e clientes;
- # Redução da quantidade de erros no sistema, diminuindo o tempo nas etapas de codificação e teste;
- # Maior dedicação à fase de análise, preocupando-se com a essência do sistema;
- # Mesma notação é utilizada desde a fase de análise até a implementação.

# Conceitos Básicos

---

## # Orientação a Objetos

- procura explorar nosso lado intuitivo.
- objetos da computação => objetos no mundo real.

# Enfoque de OO => os objetos que trocam mensagens entre si.

# Mensagens ativam métodos, que realizam as ações.

# Objetos com a mesma interface são agrupados em classes.

# Interface: métodos.

# Conceitos Básicos

---

# Objeto é criado dinamicamente e tem um ciclo de vida:

- São construídos
- Executam ações
- São destruídos
- Também podem se tornar inacessíveis



# Conceitos Básicos

---

## # Objetos e Classes

**Gato Kuroneko**

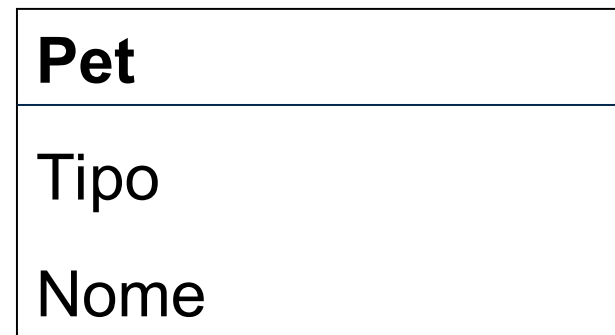
**Cão Bidu**

**Gecko Midori**



**OBJETOS**

(Instâncias da classe Pet)



**CLASSE**

# Conceitos Básicos

---

## # Objetos:

- Tudo em OO é OBJETO
- Objeto, no mundo físico, é tipicamente um produtor e consumidor de itens de informação
- Definição (mundo do software)
  - “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 possue várias características

# Conceitos Básicos

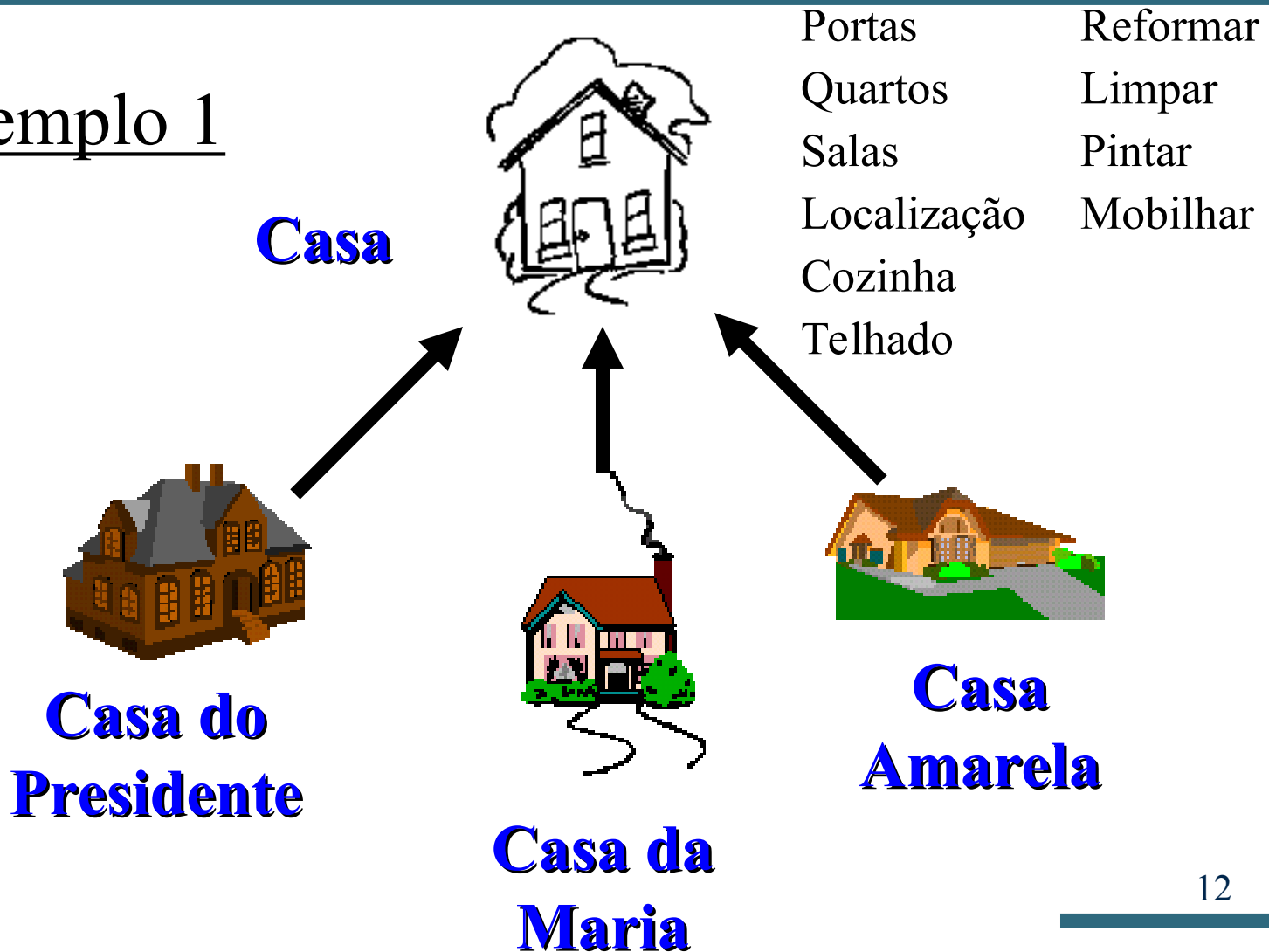
---

## # Classes:

- Agrupamento de objetos similares.
- Todo objeto é uma instância de uma Classe.
- Os objetos representados por determinada classe diferenciam-se entre si pelos valores de seus atributos.
- Conjunto de objetos que possuem propriedades semelhantes (ATRIBUTOS), o mesmo comportamento (MÉTODOS), os mesmos relacionamentos com outros objetos e a mesma semântica.

# Conceitos Básicos

## # Exemplo 1



# Conceitos Básicos

---

## # Atributos e Métodos: Exemplo 2

<b>Automóvel</b>
Proprietário Marca Placa Ano
Registrar Transferir_Proprietário Mudar_Placa



**ATRIBUTOS**



**MÉTODOS**

# Conceitos Básicos

---

## # Atributos e Métodos: Exemplo 3

<b>Figura</b>
Largura Altura Posicao_X Posicao_Y Cor_preenchimento
Mover Redimensionar



**ATRIBUTOS**



**MÉTODOS**

# Conceitos Básicos

---

## # 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

# Conceitos Básicos

---

## # Métodos:

- 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



# Conceitos Básicos

---

# Três elementos chaves de OO são:

- Encapsulamento
- Herança
- Polimorfismo

# Conceitos Básicos

---

## # Encapsulamento:

- Objetos encapsulam seus atributos;
- Propriedade segundo a qual os 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;
- 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.

# Conceitos Básicos

---

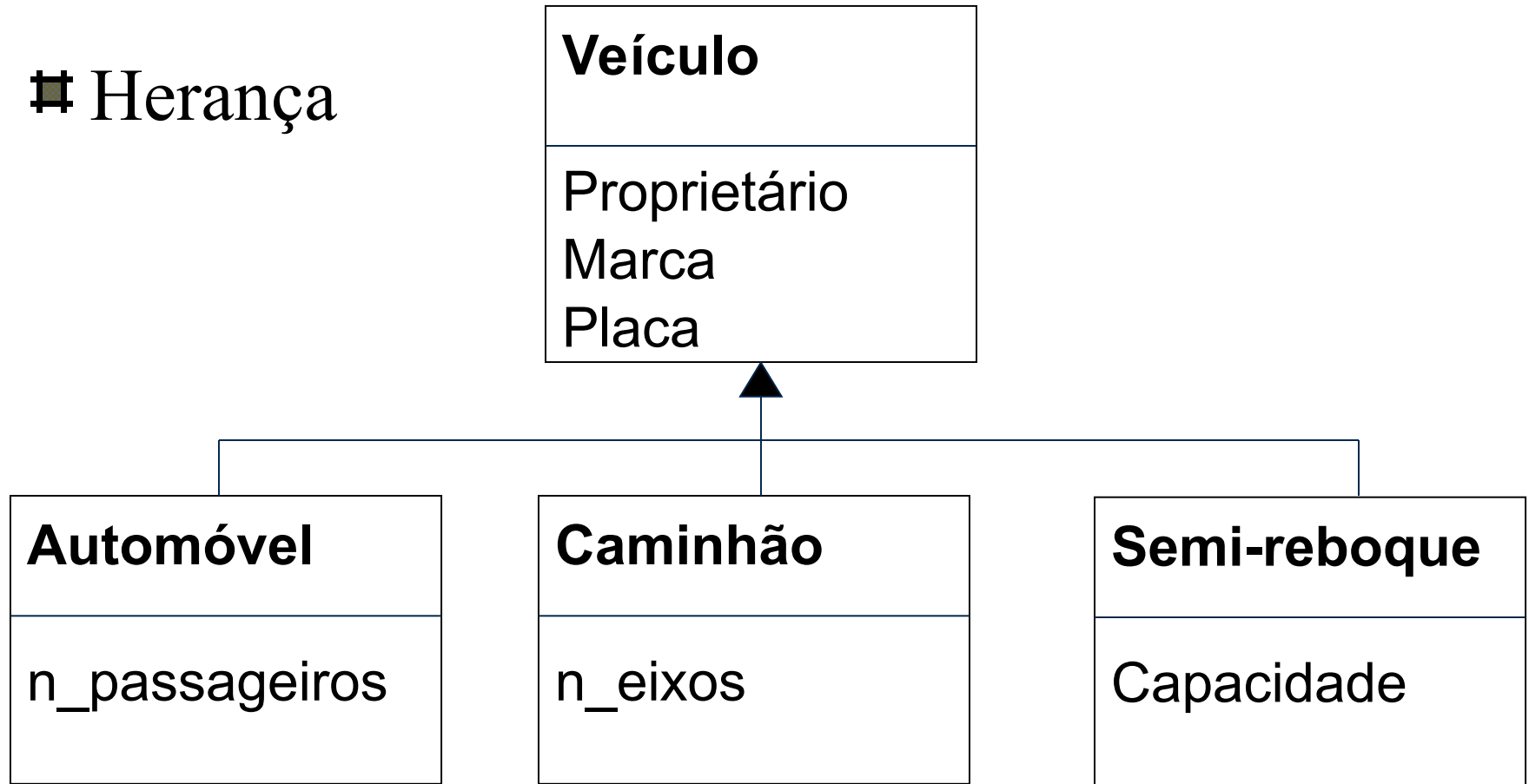
## # Herança:

- É o mecanismo pelo qual uma subclasse herda todas as propriedades da superclasse e acrescenta suas próprias e exclusivas características.
- As propriedades da superclasse não precisam ser repetidas em cada subclasse.
- Por exemplo, JanelaRolante e JanelaFixa são subclasses de Janela. Elas herdam as propriedades de Janela, como uma região visível na tela. JanelaRolante acrescenta uma barra de paginação e um afastamento.

# Conceitos Básicos

---

# Herança



# Conceitos Básicos

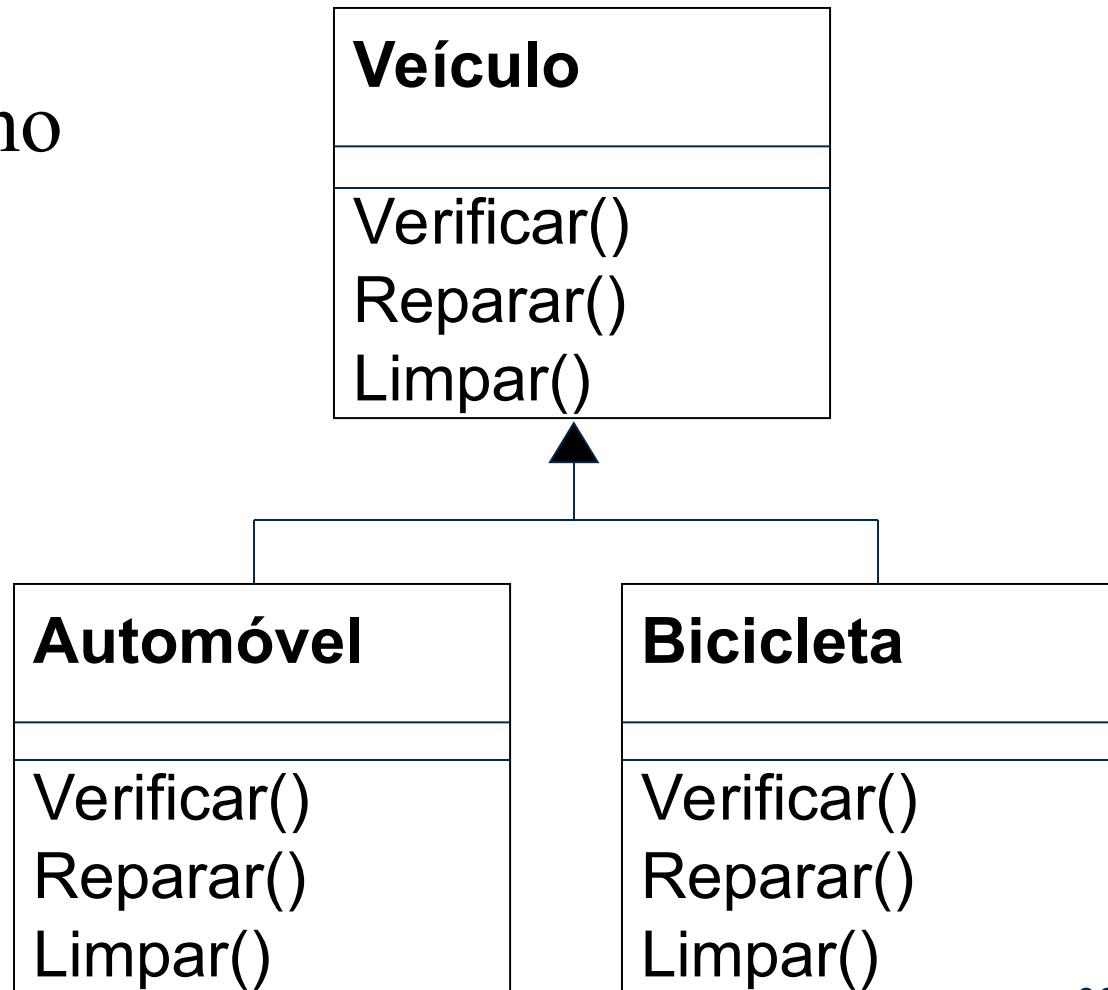
---

## # Polimorfismo:

- Geralmente representa a qualidade ou estado de um objeto ser capaz de assumir diferentes formas.
- Mais especificamente, propriedade segundo o qual vários métodos podem existir com o mesmo nome.
- Ao receber uma mensagem para efetuar uma Operação, é o objeto quem determina como a operação deve ser efetuada;
- Permite a criação de várias classes com interfaces idênticas, porém objetos e implementações diferentes.
- Exemplos:
  - O operador “+” pode ser usado com inteiros, pontos-flutuantes ou strings.
  - A operação mover pode atuar diferentemente nas classes Janela e Peça de Xadrez.

# Conceitos Básicos

## # Polimorfismo



# Conceitos Básicos

---

# Analogia dos conceitos principais no paradigma orientado a objeto e no paradigma tradicional de programação

## Linguagens Orientadas a Objetos

## Linguagens Tradicionais

Objeto	→	Valor
Classe	→	Tipo (TAD)
Mensagem	→	Chamada de Procedimento
Método	→	Procedimento ou Função
Interface	→	Conjunto de nomes e funções para um fim específico