



# **Programação II**

## **Aula 01 – Introdução a Orientação a Objetos**

### **Parte I**

Prof. Gustavo Callou

[gustavo.callou@ufrpe.br](mailto:gustavo.callou@ufrpe.br)

# Roteiro

- ▶ Conceitos de Orientação a Objetos
  - ▶ **Objeto**
  - ▶ **Abstração**
  - ▶ Classe
  - ▶ Herança
  - ▶ Modularização
  - ▶ Encapsulamento



# O paradigma da Orientação a Objetos (OO)

- ▶ *Um paradigma é uma forma de abordar um problema.*
- ▶ O paradigma OO surgiu no fim dos anos 60.
- ▶ Hoje em dia, praticamente suplantou o paradigma anterior, o *paradigma estruturado...*

# O paradigma da Orientação a Objetos

- ▶ Alan Kay, um dos pais do paradigma da orientação a objetos, formulou a chamada **analogia biológica**.
- ▶ “Como seria um sistema de software que funcionasse como um ser vivo?

# Analogia Biológica

- ▶ Cada “célula” interagiria com outras células através do envio de mensagens para realizar um objetivo comum.
- ▶ Adicionalmente, cada célula se comportaria como uma unidade autônoma.

# Analogia Biológica

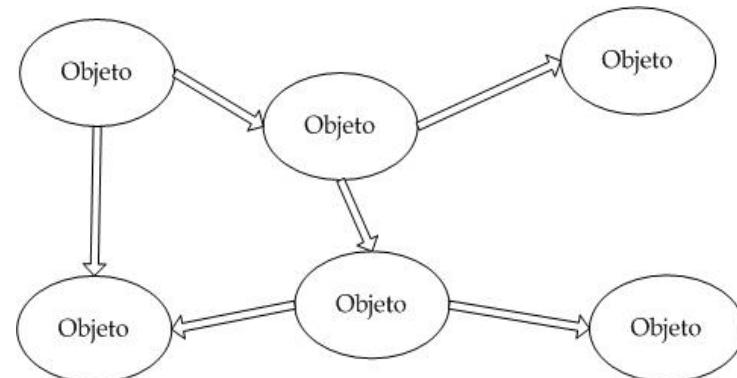
- ▶ De uma forma mais geral, **Kay pensou em** como construir um **sistema de software** a partir **de agentes autônomos que interagem entre si**.
  
- ▶ Com isso, ele estabeleceu os princípios da **orientação a objetos**.

# Orientação a Objetos - Princípios

- ▶ **Tudo é um objeto.**
- ▶ Pense em um **objeto** como uma **super variável**:
  - ▶ armazena dados,
  - ▶ requisições feitas a esse objeto são operações feitas sobre ele mesmo.
- ▶ Em teoria, você pode **representar** qualquer **elemento** conceitual no problema que você está tentando resolver (cachorros, livros, sócios, empréstimos, etc.) **como** um **objeto** no seu programa.

# Orientação a Objetos - Princípios

- ▶ **Um programa é uma coleção de objetos dizendo uns aos outros o que devem fazer.**
- ▶ Para fazer uma requisição a um objeto, pode-se “mandar uma mensagem” para este objeto.
- ▶ Uma **mensagem** pode ser pensada **como** sendo uma **chamada** a um procedimento ou função (**método**) pertencente ao objeto em particular.



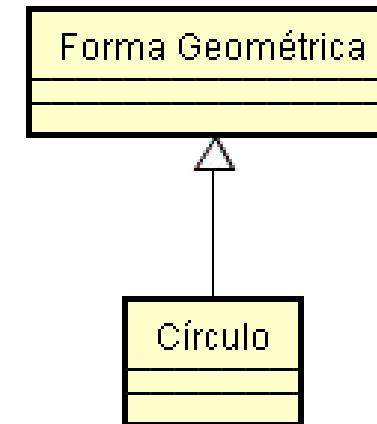
# Orientação a Objetos - Princípios

- ▶ **Um objeto pode ser composto por vários outros objetos**
  - ▶ Em outras palavras:
    - ▶ Pode-se criar um novo tipo de objeto empacotando objetos existentes.
    - ▶ Dessa forma, pode-se adicionar complexidade a um programa e escondê-la por trás da simplicidade do uso de objetos.

- ▶ **Todo objeto tem um tipo.**
- ▶ Usando as palavras certas, cada objeto é uma instância de uma classe, onde classe é um sinônimo de tipo.
- ▶ A questão mais importante relativa a uma classe é “que mensagens eu posso enviar para uma instância dessa classe?”

# Orientação a Objetos - Princípios

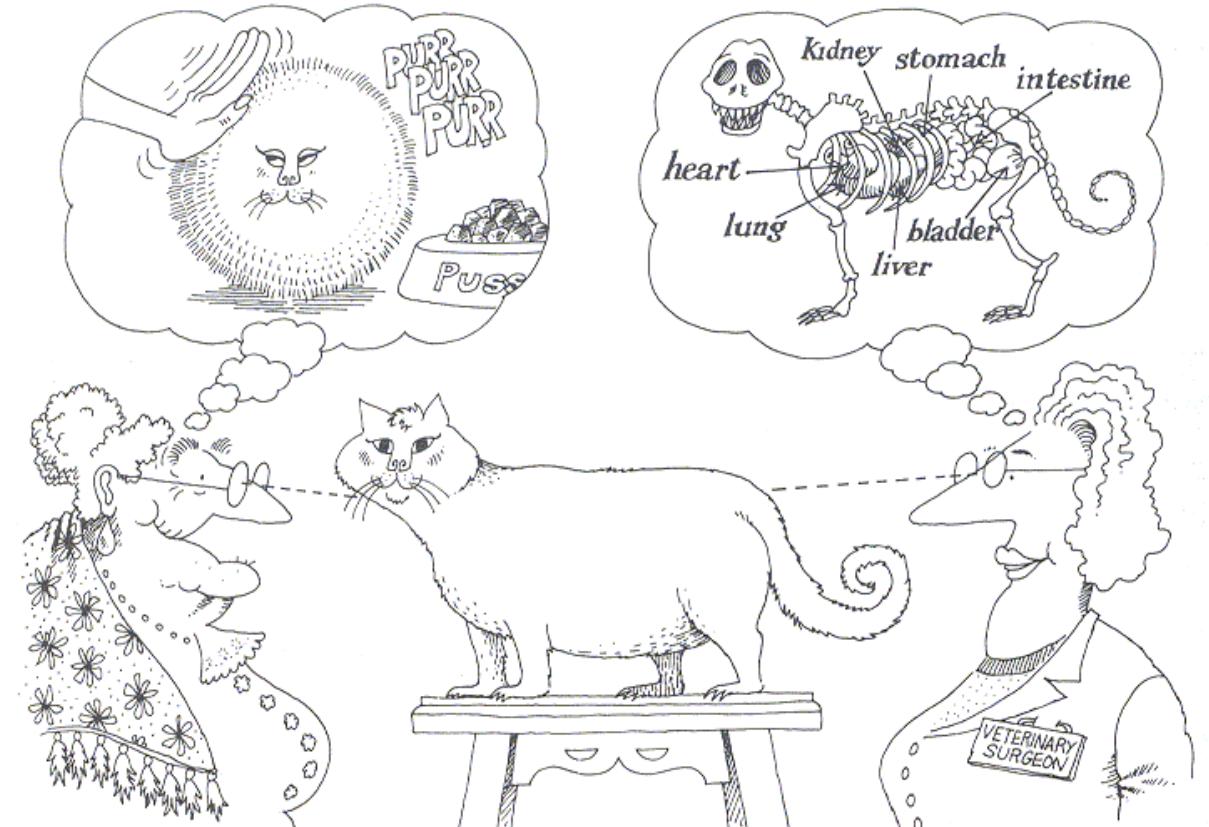
- ▶ **Todos os objetos de um dado tipo podem receber as mesmas mensagens.**
  - ▶ Por exemplo, um objeto do tipo “círculo” é também um objeto do tipo “forma geométrica”.
  - ▶ Então, o objeto “círculo” aceita qualquer mensagem endereçada a “forma geométrica”.
  - ▶ Essa capacidade de “substituição” de um objeto por outro é um dos mais poderosos conceitos em orientação a objetos.



# Objeto

- Definição:
  - Um **objeto** é qualquer **coisa**, real ou abstrata, **sobre a qual armazenamos dados e realizamos operações** que manipulam tais dados.
  - Um **objeto** é **composto** por:
    - **Atributos** → características ou propriedades que definem o objeto.
    - Comportamento → conjunto de ações pré-definidas (**métodos**).

# Abstração



Abstraction focuses upon the essential characteristics of some object, relative to the perspective of the viewer.

# Abstração

- ▶ Mecanismo → representa uma realidade complexa em termos de um modelo simplificado
  - ▶ detalhes irrelevantes possam ser suprimidos.
- ▶ Processo de **filtragem** de **detalhes sem importância** do objeto, para que apenas as características apropriadas que o descrevem permaneçam.

# Abstração

- ▶ Três abstrações de um carro



Officina

Consumidor

Detran

# Abstração

## ► Três abstrações de um carro



Officina

Placa, conserto,  
pagamento,  
etc

Consumidor

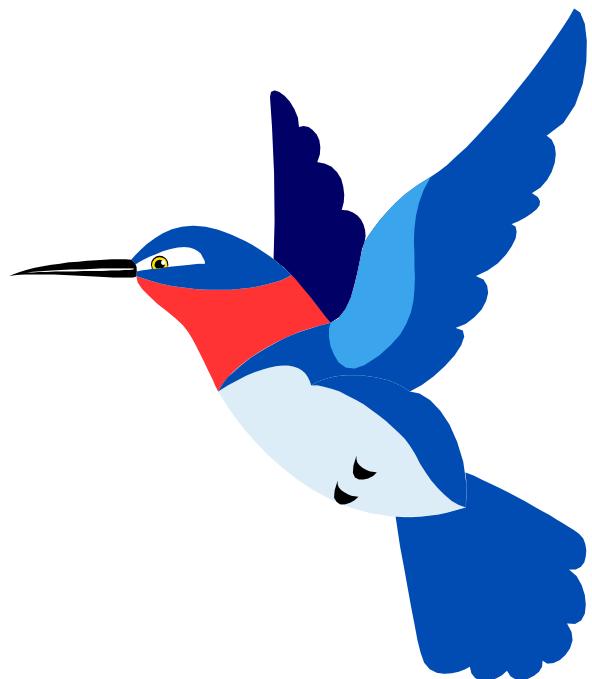
Consumo (Km/l),  
Manutenção,  
Conforto

Detran

Identificação,  
Impostos,  
Placa,

# Objetos - Exemplos

## ▶ Pássaro



*Identidade:* beija-flor

*Características:*

cores

forma do bico

tipo do vôo

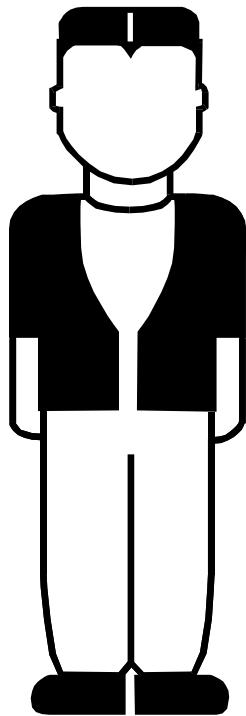
*Comportamento:*

voar

piar

# Exemplo

## ► Pessoa



*Identidade:* ‘Mário’

*Características:*

olhos pretos

nasceu em 16/02/70

pesa 70kg

mede 1,70m

*Comportamento:*

andar

falar

comer

rir

# Exemplo

## ► Ônibus



*Identidade:* placa LXY 7684

*Características:*

- cor amarela
- 30 assentos
- a diesel

*Comportamento:*

- frear
- andar
- correr
- buzinar
- acelerar

- ▶ O que significa dizer que um Sistema é orientado a objetos?

- ▶ O que significa dizer que um Sistema é orientado a objetos?
  - ▶ significa que o aplicativo é **organizado como uma coleção de objetos** que incorporam tanto a estrutura como o comportamento dos dados.

# Sistema de Controle de Pizzarias

- ▶ Sistema que informatiza os pedidos de pizza em um restaurante.
- ▶ Objetos:



- ▶ Sistema que informatiza os pedidos de pizza em um restaurante.

- ▶ Objetos:

- ▶ Pedido,
- ▶ Cardápio,
- ▶ Pizza,
- ▶ Caixa,
- ▶ Cliente,
- ▶ Garçom,
- ▶ Cozinheiro,
- ▶ etc



- ▶ Cardápio → armazenar os preços e mantê-los atualizados.
- ▶ Pedido → processamento dos pedidos feitos pelos clientes.
- ▶ Caixa → computa a conta a ser paga pelo Cliente.

- ▶ Utilidade de OO:
- ▶ Exemplo 1:
  - ▶ Caso houvesse alteração no sistema para atender a necessidade de atualização de preços →
  - ▶ Assim, os demais objetos não sofreriam alteração
- ▶ Exemplo 2:
  - ▶ Caso a forma de calcular a conta fosse modificada (exemplo: gorjeta) →
- ▶ Obs: Cada objeto tem a sua respectiva função

- ▶ Utilidade de OO:
- ▶ Exemplo I:
  - ▶ Caso houvesse alteração no sistema para atender a necessidade de atualização de preços → seria a responsabilidade do cardápio.
  - ▶ Assim, os demais objetos não sofreriam alteração
- ▶ Exemplo 2:
  - ▶ Caso a forma de calcular a conta fosse modificada (exemplo: gorjeta) →
- ▶ Obs: Cada objeto tem a sua respectiva função

- ▶ Utilidade de OO:
- ▶ Exemplo 1:
  - ▶ Caso houvesse alteração no sistema para atender a necessidade de atualização de preços → seria a responsabilidade do cardápio.
  - ▶ Assim, os demais objetos não sofreriam alteração
- ▶ Exemplo 2:
  - ▶ Caso a forma de calcular a conta fosse modificada (exemplo: gorjeta) → o caixa seria refeito.
- ▶ Obs: Cada objeto tem a sua respectiva função