



# Conceitos Básicos

---

Prof. MSc. Tiago Araújo

[tiagodavi70@gmail.com](mailto:tiagodavi70@gmail.com)

# Material

---

SCC0504 - Programação Orientada a Objetos (2018)

Luiz Eduardo Virgilio da Silva

# Sumário

---

Paradigmas de Programação

Os pilares da POO

Classes e Objetos

Tipos de acessos

# Paradigmas de programação

---

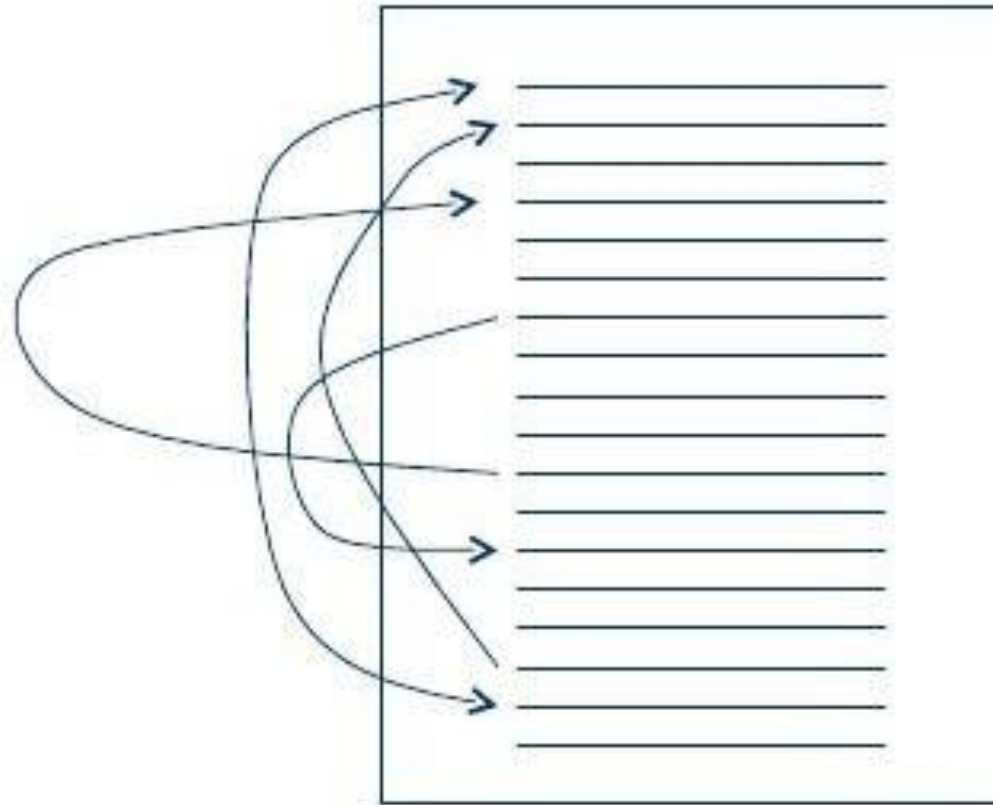
Programação Não Estruturada

Programação Estruturada

Programação Orientada a Objetos

# Paradigmas de programação

## Programação Não Estruturada



# Paradigmas de programação

---

O uso indiscriminado de transferência de controle era considerado a raiz de muitos problemas

- Legibilidade
- Podemos usar o comando GOTO (com cuidado)

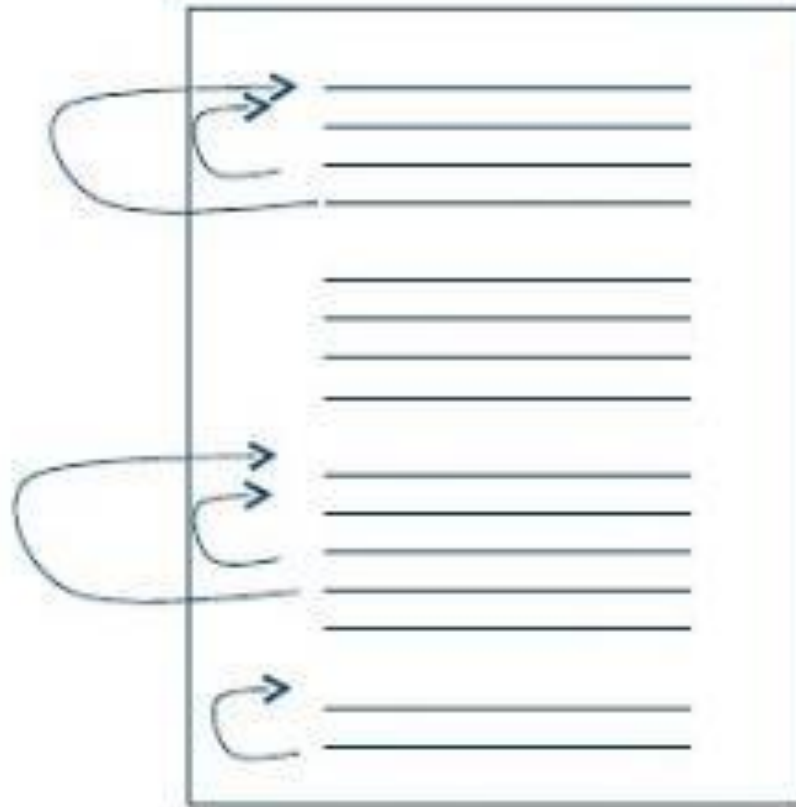
Na programação estruturada os comandos de um programa são executadas sequencialmente

Vários comandos permitem que essa sequência seja quebrada, causando o que é chamado de transferência de controle

- Porém, de forma organizada

# Paradigmas de programação

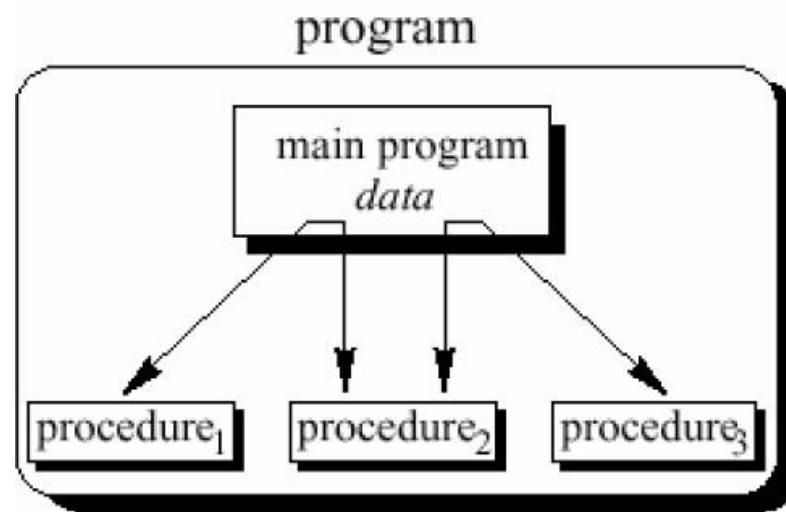
## Programação Estruturada



# Paradigmas de programação

Os procedimentos são combinados para prover a funcionalidade desejada

O programa pode ser visto como uma sequência de chamadas de procedimento. Programa principal é responsável por passar os dados para as chamadas individuais para serem processados



**Ex.: C, Pascal, Fortran**



# Programação Orientada a Objetos

---

Devido aos requisitos atuais, os softwares têm se tornado cada vez mais complexos e maiores

Isso tem levado a busca de meios para tornar a tarefa de programação mais produtiva

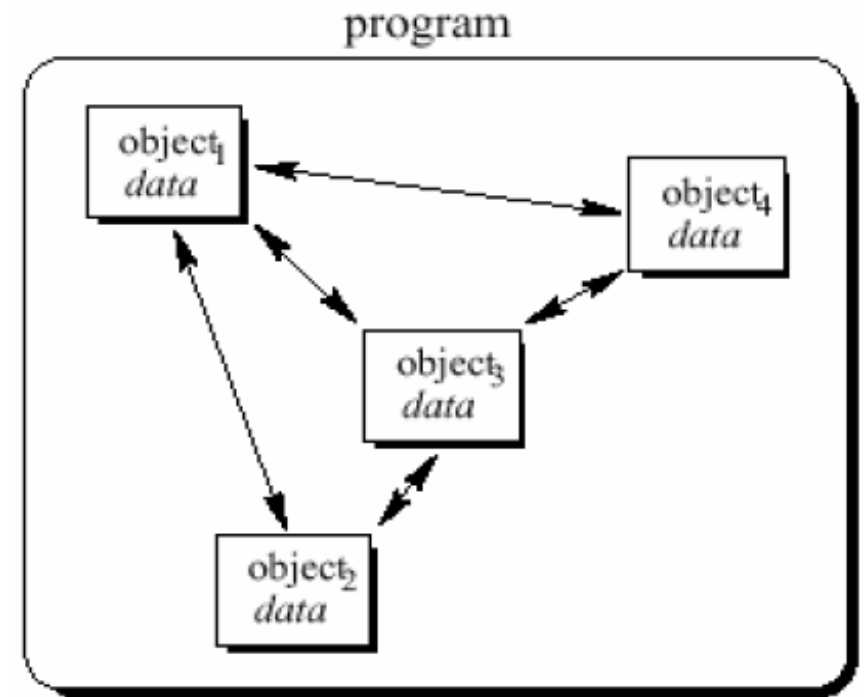
Ainda não existe uma resposta definitiva a essa busca, mas há um consenso de que a Programação Orientada a Objetos (POO) consegue produzir resultados mais competitivos do que as outras abordagens

Os programas OO comumente são mais fáceis de entender, corrigir e modificar

# Programação Orientada a Objetos

POO é caracterizado pelo uso de um conjunto de objetos interagentes, cada qual responsável pelo gerenciamento de seu estado interno

- Os objetos interagem uns com os outros através da troca de mensagens
- Cada objeto é responsável pela inicialização e destruição de seus dados internos



**Ex.: Eiffel, SmallTalk, C++, Java, C#**

# Programação Orientada a Objetos

---

## Na programação estruturada

- Procedimentos são implementados em blocos e a comunicação entre eles se dá pela passagem de dados
- Um programa estruturado, quando em execução, é caracterizado pelo acionamento de procedimentos cuja tarefa é a manipulação de dados

## Na programação orientada a objetos

- Dados e procedimentos são encapsulados em um só elemento denominado objeto
- O estabelecimento de comunicação entre objetos (envio e recebimento de mensagens) caracteriza a execução do programa

# Programação Orientada a Objetos

---

Vantagens da POO em relação à programação estruturada

- Maior índice de reaproveitamento de código
- Maior facilidade de manutenção
- Menos código gerado
- Maior confiabilidade no código
- Maior facilidade de gerenciamento do código (reduz grandes problemas para problemas menores)

# Classes e Objetos

---

Um objeto é uma entidade que formaliza o modo pelo qual compreendemos algo no domínio do problema

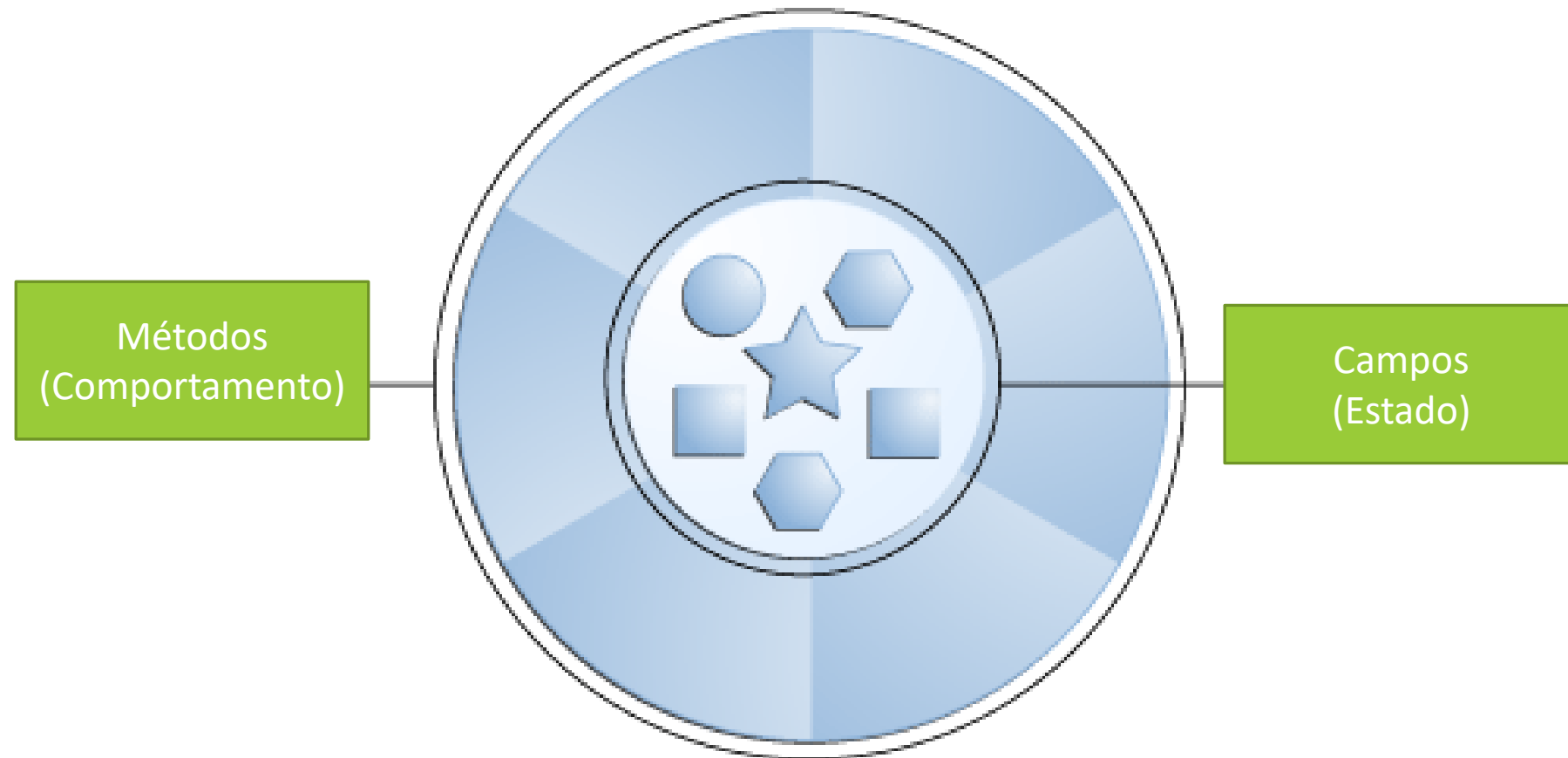
- Reflete a capacidade do sistema de guardar informações sobre o elemento abstraído e interagir com ele
- Entidade o mais próximo possível das entidades do mundo real - aquilo que é tangível ou visível
- Dessa forma, os objetos são uma forma de diminuir o gap semântico
- Diferença entre o domínio de problemas e soluções

# Classes e Objetos

---

A um objeto sempre estarão associados

- Estado
  - Definido pelas propriedades (atributos) que ele possui e pelos valores que elas estão assumindo
- Comportamento
  - Definido pela forma como ele age e reage, em termos de mudança de seu estado e o relacionamento com os demais objetos do sistema (métodos)
- Identidade
  - Cada objeto é único



# Classes e Objetos

---

Exemplos: Cachorro, mesa, televisão, bicicleta, lâmpada, ...

## Lâmpada

- Atributos: ligada, desligada
- Métodos: ligar, desligar

## Rádio

- Atributos: ligado, desligado, volume, estação, ...
- Métodos: ligar, desligar, aumentar/abaixar volume, sintonizar,...

## Cachorro

- Atributos: nome, cor, raça, peso, ...
- Métodos: latir, morder, correr, dormir, ...

## Bicicleta - ??

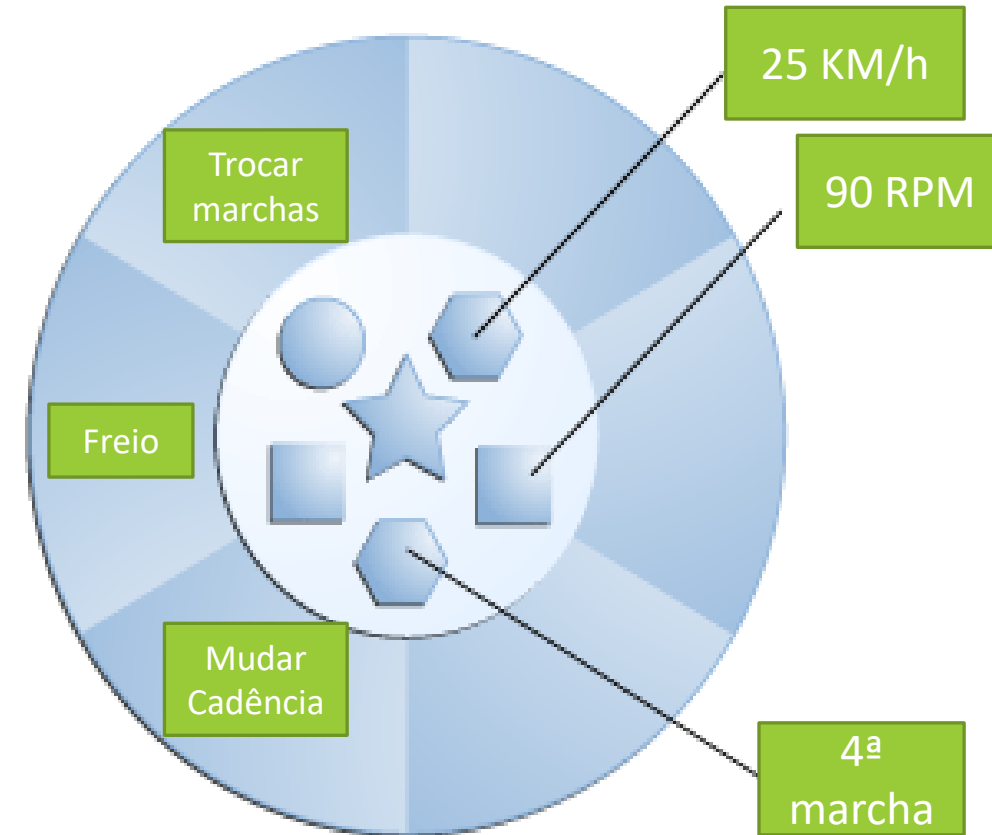


# Classes e Objetos

## Bicicleta

- Atributos: marcha, cadência do pedal, velocidade, ...
- Métodos: mudar marcha, mudar cadência do pedal, frear, ...

Objetos podem conter objetos como atributos



# Classes e Objetos

---

Uma classe descreve um conjunto de objetos semelhantes

- Atributos e métodos que resumem as características comuns de vários objetos

Diferença entre classe e objeto

- Objeto constitui uma entidade concreta com tempo e espaço de existência
- Classe é tão-somente uma abstração

Em termos de programação, definir uma classe significa formalizar um tipo de dado (TAD) e todas as operações associadas a esse tipo, enquanto declarar objetos significa criar variáveis do tipo definido

# Classes e Objetos

---

Classe é um template (“forma”) para a criação de objetos

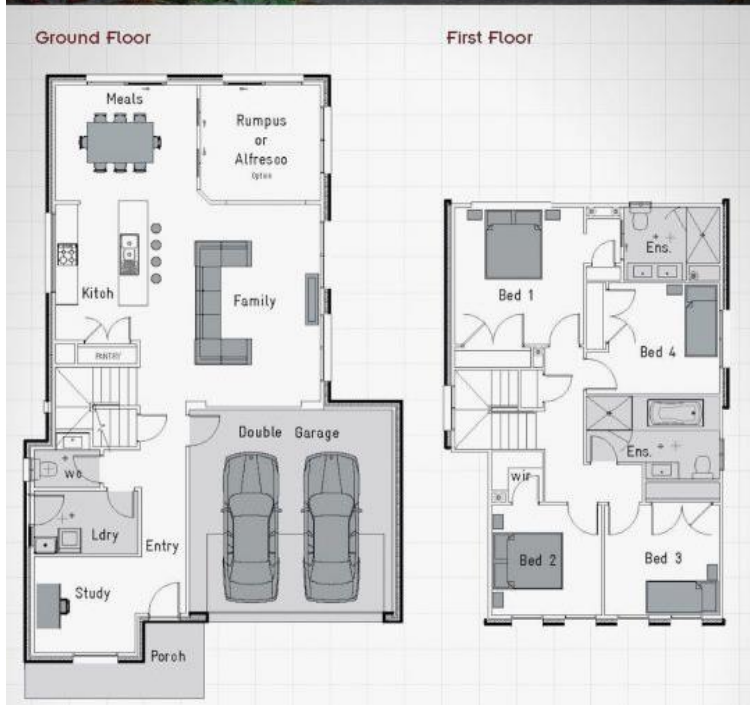
- Uma classe especifica os tipos de dados (atributos) e operações (métodos) suportadas por um conjunto de objetos

Um objeto é uma instância de uma classe

- Criação de um objeto a partir de uma classe é chamada de instanciação
- É muito comum que em um programa existam várias instâncias de uma mesma classe
  - O que diferencia cada uma?



= Objeto



= Classe

# Classes e Objetos

---

Cada instância é formada por valores de atributos únicos e um comportamento comum definido pela classe

Inúmeras instâncias podem ser criadas a partir de uma classe

O estado de cada instância é representado pelos valores de seus atributos, que podem ser diferentes

Diferentes objetos de uma mesma classe possuem suas próprias cópias de cada atributo

- A menos que isso seja desejado e explicitamente declarado
- Neste caso, um único atributo pode ser compartilhado para todas as instâncias

# Classes e Objetos

---

Os métodos são operações que podem ser executadas pelos objetos

- Valores dos atributos são (normalmente) acessados através dos métodos definidos pela classe
  - *Information-hiding*
- O serviço oferecido pelos método é um comportamento específico, residente no objeto, que define como ele deve agir quando exigido

# Tipos de Acesso

---

Uma classe pode definir o tipo de acesso à seus membros (atributos e métodos)

## Público

- Atributo ou método da classe pode ser acessado por todas as demais entidades do sistema

## Protegido

- Atributo ou método da classe pode ser acessado somente por classes da mesma hierarquia e mesmo pacote

## Privado

- Atributo ou método da classe pode ser acessado somente por métodos da própria classe

# Tipos de Acesso

---

A escolha dos tipos de acesso é muito importante na POO

- Define o escopo dos atributos e métodos

Em geral, atributos são declarados privados

- Métodos da própria classe são responsáveis por modificar e recuperar o estado dos atributos
  - Tais métodos são públicos
  - Setters e getters
  - ◗ Garantem a estabilidade e segurança
  - Information-hiding

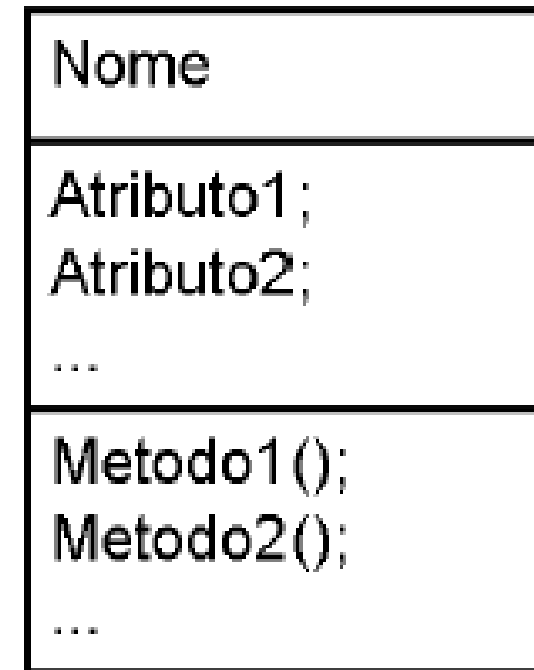


# Representação Gráfica

---

A notação gráfica de uma classe permite visualizar uma abstração independente de qualquer linguagem de implementação específica, dando ênfase às partes mais importantes: seu nome, atributos e métodos (operações)

- Também é possível representar tipos de acesso



# Pilares da POO

---

O paradigma orientado a objetos define alguns princípios básicos que devem ser seguidos

- Abstração
- Encapsulamento
- Herança
- Polimorfismo
- Mensagens

# Abstração

---

Consiste em identificar os requisitos de software e modelá-los em classes

- Ignorar aspectos não-relevantes, concentrando-se apenas nos aspectos principais do problema

Classes são abstrações de conceitos

Consiste basicamente no processo de retirar do domínio do problema os detalhes relevantes e representá-los na linguagem de solução (ex.: Java)

Classes (objetos) podem ser qualquer entidade reconhecida como um elemento da solução

- Objeto real ou não

# Encapsulamento

---

A propriedade de implementar dados e procedimentos correlacionados em uma mesma entidade recebe o nome de encapsulamento

A ideia por trás do encapsulamento é a de que um sistema orientado a objetos não deve depender de sua implementação interna, e sim de sua interface

- *Information-hiding*

## Exemplo: objeto Bicicleta

- Atributos (estados) não são alterados diretamente pelas outras entidades
- Métodos da própria classe são definidos para fazê-lo
  - Permite controle total de como os atributos variam
  - Ex: limite para número de marchas

# Encapsulamento

---

## Transparência

Não importa como os métodos são implementados

Para as outras entidades, o importante é saber como se comunicar com o objeto

- Quais métodos estão disponíveis
- Assinatura dos métodos
  - Interface

Isso permite que a implementação de um método seja facilmente reescrita, sem prejuízo para as outras entidades

- Ex: carros

# Herança

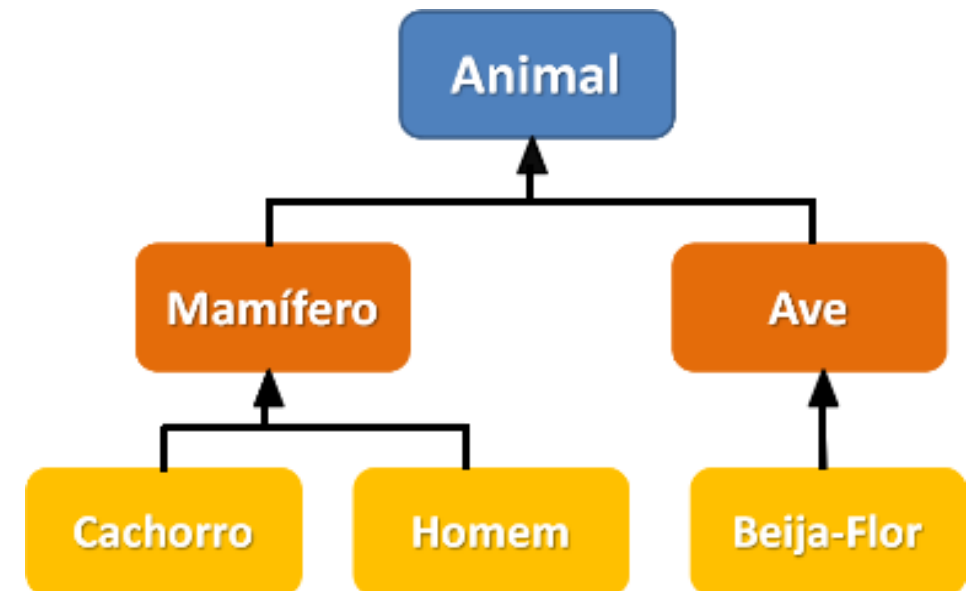
Permite a hierarquização das classes em um sistema

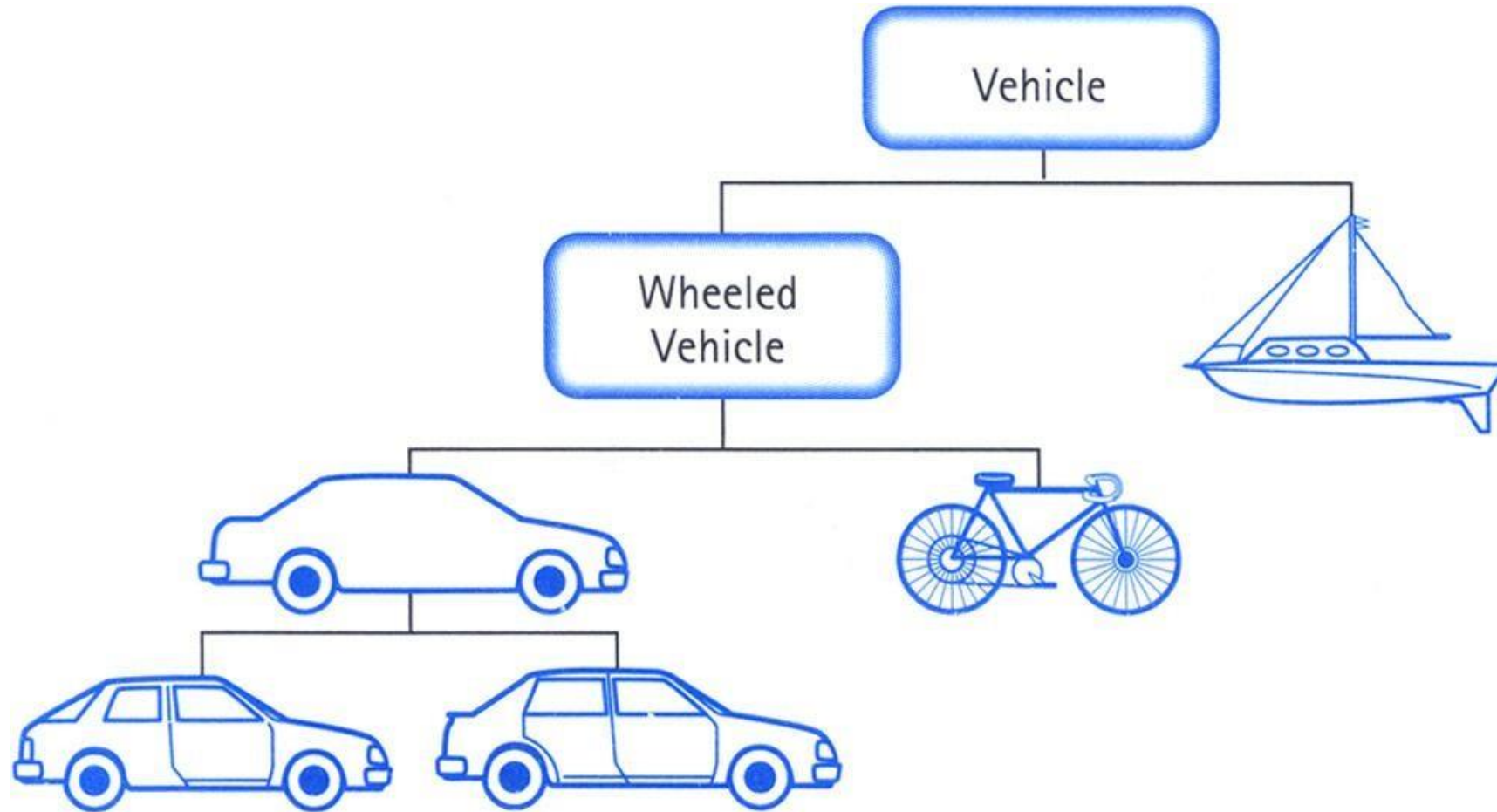
Uma classe mais especializada (sub-classe ou classe-derivada) herda as propriedades (métodos e atributos) de uma classe mais geral (super-classe ou classe-base)

Uma sub-classe pode sobrescrever o comportamento de uma super-classe (polimorfismo)

Promove reuso

Novos atributos e métodos podem ser definidos nas sub-classes, além dos herdados





# Polimorfismo

---

Polimorfismo, em biologia, é um princípio no qual um organismo pode surgir de formas diferentes.

- Indivíduos de uma mesma espécie possuem muitas características similares.
- Contudo, algumas características são peculiares.





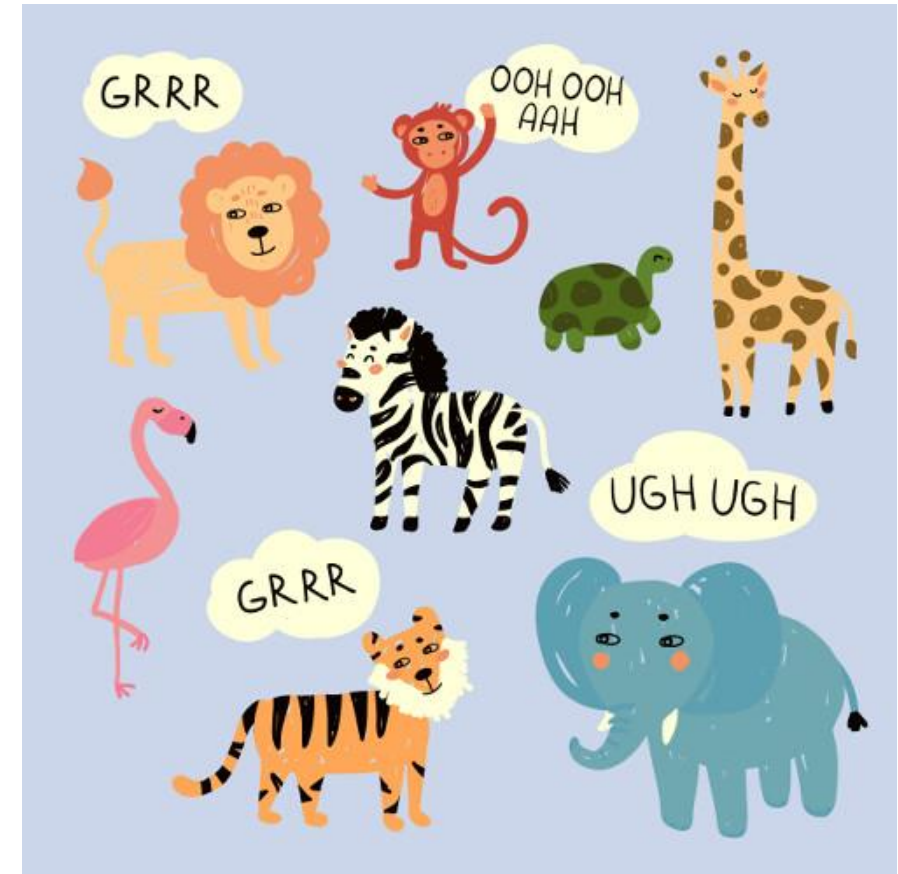
# Polimorfismo

O mesmo princípio se aplica no paradigma de POO

- Herança permite que subclasses herdem as características de sua classe pai (mãe).
- Contudo, propriedades particulares da subclasse podem ser redefinidas.

Em POO, polimorfismo é a capacidade de uma mesma operação (método) comportar-se de maneira diferente nas diferentes classes de uma hierarquia

- Método com a mesma assinatura, porém com serviços diferentes



# Interface

---

Pontos de interação entre dois meios, objetos, etc.

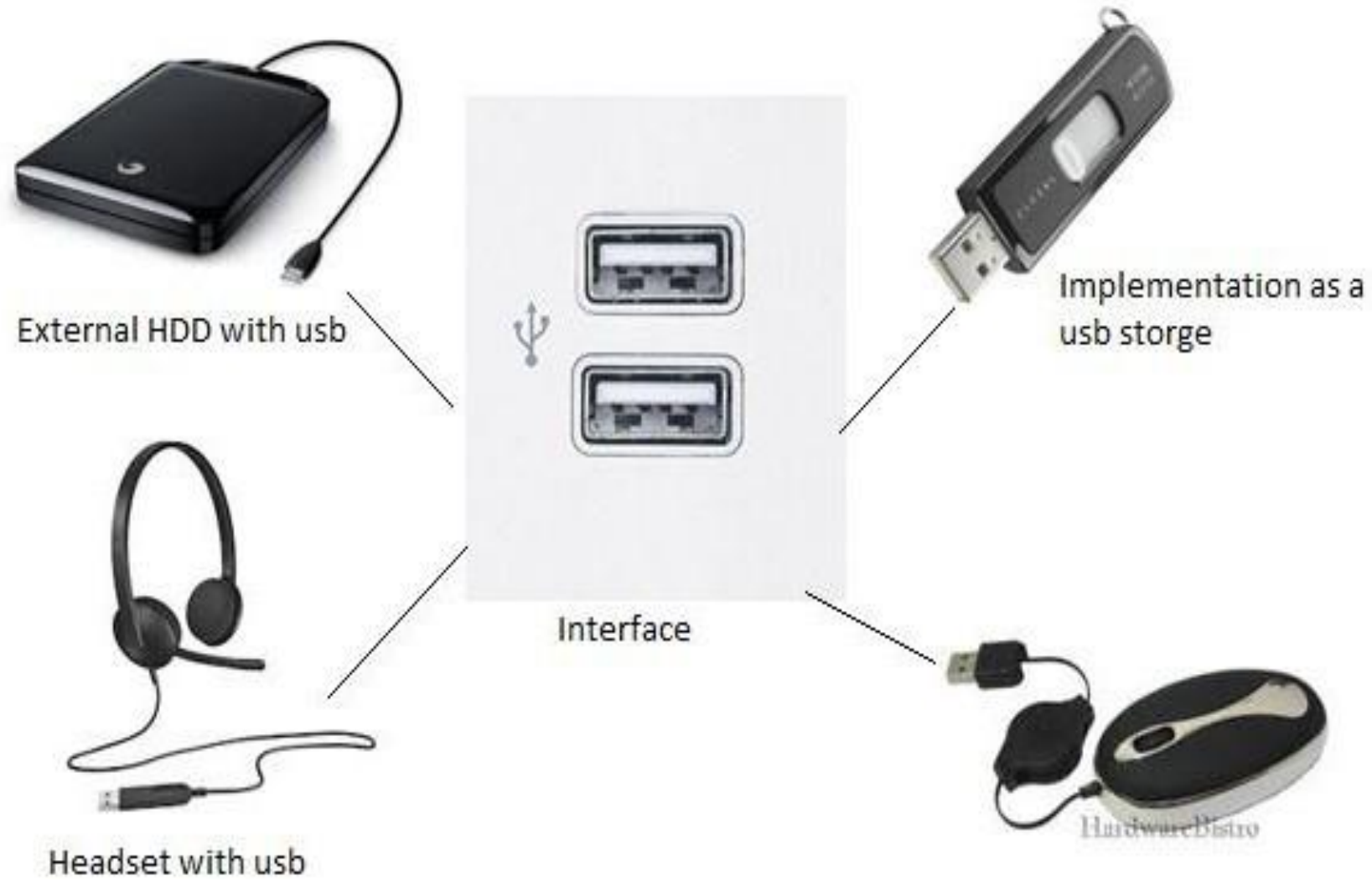
Ex: Televisão

- As interfaces da TV são os botões de ligar/desligar, mudar canais, ...

Pontos de interação da entidade com o mundo externo

Uma interface em Java representa uma declaração de um conjunto público de operações

- Define um contrato
- Não pode ser instanciada
- Garante que os objetos terão esta forma de comunicação



# Pacotes

---

Pacote é um conjunto de classes e interfaces relacionadas de alguma forma

Biblioteca é o conjunto de pacotes

API (Application Programmin Interface)

- Muito útil: reuso de código
- API Java

Antes de definir uma nova classe, verifique se já não existe uma solução

- A validação do código é demorada

# Resumo

---

Paradigmas de Programação

Os pilares da POO

Classes e Objetos

Tipos de acessos

Interfaces

# Exercício

---

Definir as classes abaixo utilizando diagramas

- Animal, Homem, Cachorro, Carro, Casa

Defina atributos e métodos destas classes, pensando na interação entre objetos desses tipos

- Lembre que atributos podem ser objetos

Onde pode haver herança?

Simule troca de mensagens entre os objetos