



"Antes de imprimir pense em sua responsabilidade e compromisso com o **MEIO AMBIENTE**."

*Engenharia de Software*

# *Análise Orientada a Objetos (AOO)*



**Givanaldo Rocha de Souza**

[givanaldo.rocha@ifrn.edu.br](mailto:givanaldo.rocha@ifrn.edu.br)

<http://docente.ifrn.edu.br/givanaldorochoa>

Material original gentilmente cedido pelo professor Fábio Procópio



## Introdução

- ❑ Para Rumbaugh, Orientação a Objetos é *“uma nova maneira de pensar os problemas utilizando modelos organizados a partir de conceitos do mundo real (...)”*;



- ❑ *“Orientação a Objetos é um paradigma de análise, projeto e programação de sistemas de software baseado na composição e interação entre diversas unidades de software chamadas de objetos.”*  
WIKIPÉDIA;

### AOO

- ➔ Introdução
- Classificação
- Abstração
- Objeto
- Classes
- Atributos
- Métodos
- Visibilidade
- Encapsulamento
- Herança
- Polimorfismo
- UML
- Referências



## Classificação

- ❑ Quando éramos crianças, os adultos nos ensinaram a pensar de forma orientada a objetos;
- ❑ Por exemplo, pensávamos em conceitos simples como pessoa, carro, mala e coelho;
- ❑ Quando as pessoas pensam assim (sejam elas crianças ou não), são definidas classes, ou seja, um conjunto de objetos;
- ❑ O nosso aprendizado é obtido por meio da classificação, isto é, formar grupos de objetos com características e comportamentos semelhantes.



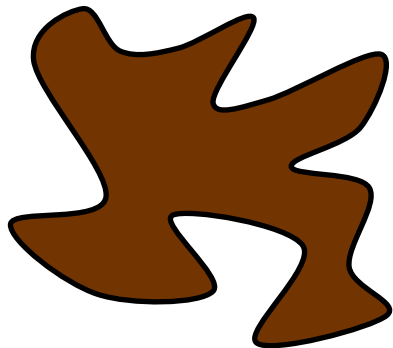
### AOO

- Introdução
- ➔ Classificação
- Abstração
- Objeto
- Classes
- Atributos
- Métodos
- Visibilidade
- Encapsulamento
- Herança
- Polimorfismo
- UML
- Referências

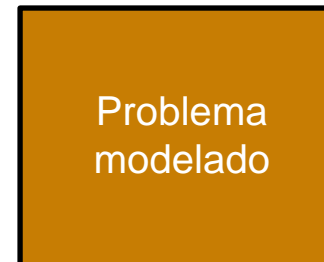
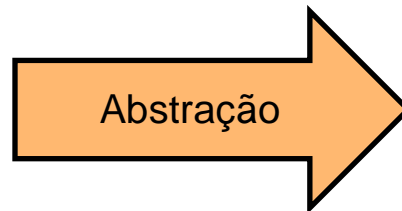


## Abstração

- ❑ A abstração é essencial para identificarmos classes;
- ❑ Consiste na seleção de alguns aspectos de domínio do problema a modelar, desconsiderando os irrelevantes para o nível de abstração em questão;
- ❑ Indispensável na modelagem de objetos reais porque, no mundo real, quase tudo é complexo:



Problema do mundo real



### AOO

- Introdução
- Classificação
- ➔ Abstração
- Objeto
- Classes
- Atributos
- Métodos
- Visibilidade
- Encapsulamento
- Herança
- Polimorfismo
- UML
- Referências



# Abstração

## Mundo REAL



CARRO DE SOM

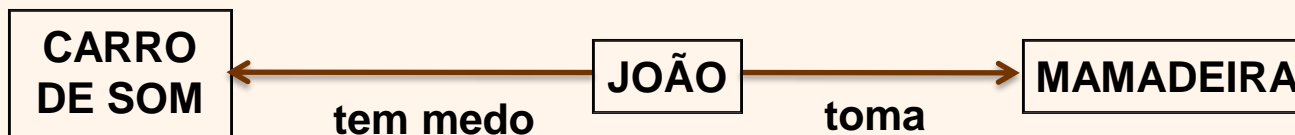


JOÃO



MAMADEIRA

## Problema MODELADO



### AOO

- Introdução
- Classificação
- Abstração
- Objeto
- Classes
- Atributos
- Métodos
- Visibilidade
- Encapsulamento
- Herança
- Polimorfismo
- UML
- Referências



## Objeto

- ❑ É uma entidade real ou abstrata, que modela um conceito presente na realidade humana, ocupando espaço físico ou lógico;
- ❑ Um objeto é uma pessoa, um lugar, é a base para todos os outros conceitos da orientação a objetos;
- ❑ Facilita a compreensão do mundo real e oferece uma base real para implementação em computador;
- ❑ Um objeto denota uma entidade de natureza física, conceitual ou de software:
  - ❑ Entidades físicas: um carro, uma pessoa, um livro;
  - ❑ Entidade conceitual: um DER de uma sistema;
  - ❑ Entidade de software: um *radiobutton* em uma página web.



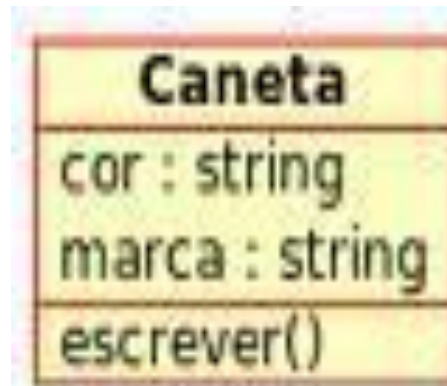
### AOO

Introdução  
Classificação  
Abstração  
Objeto  
Classes  
Atributos  
Métodos  
Visibilidade  
Encapsulamento  
Herança  
Polimorfismo  
UML  
Referências



# Classes

- ❑ Uma classe é o projeto de um objeto;
- ❑ Uma classe representa uma categoria e os objetos são membros dessa categoria;
- ❑ Classe é a representação de um conjunto de coisas reais ou abstratas que são reconhecidas como sendo do mesmo tipo;
- ❑ Uma classe é considerada uma fábrica de instâncias que inclui atributos e operações dessas instâncias;
- ❑ É importante evitar a criação de classes que tentem abranger tudo (ou várias coisas).



## AOO

- Introdução
- Classificação
- Abstração
- Objeto
- Classes
- Atributos
- Métodos
- Visibilidade
- Encapsulamento
- Herança
- Polimorfismo
- UML
- Referências



# Classes

## Classe Pessoa (grupo de objetos similares que compartilham atributos e comportamentos)



**Objetos:** ocorrências de uma classe

### AOO

- Introdução
- Classificação
- Abstração
- Objeto
- Classes
- Atributos
- Métodos
- Visibilidade
- Encapsulamento
- Herança
- Polimorfismo
- UML
- Referências



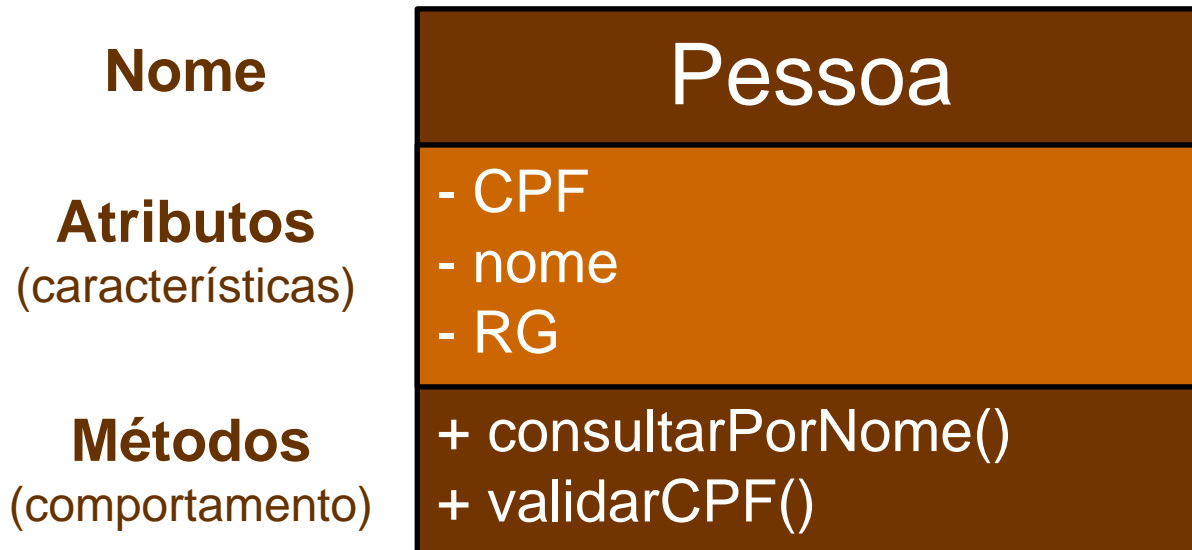


# Classes: representação

- ❑ Representada por um retângulo que pode possuir até três divisões:
  - ❑ Nome da classe
  - ❑ Atributos pertencentes à classe
  - ❑ Possíveis métodos da classe

## AOO

Introdução  
 Classificação  
 Abstração  
 Objeto  
 → Classes  
 Atributos  
 Métodos  
 Visibilidade  
 Encapsulamento  
 Herança  
 Polimorfismo  
 UML  
 Referências





# Atributos

- ❑ Também são conhecidos como propriedades;
- ❑ São as características de uma classe, ou seja, as peculiaridades que costumam variar de objeto para objeto;
- ❑ Alguns exemplos de atributos em uma classe chamada Pessoa seriam: altura, sexo, cor, idade;
- ❑ Alguns atributos permitem diferenciar um objeto do outro dentro de uma mesma classe.

## AOO

Introdução  
Classificação  
Abstração  
Objeto  
Classes  
Atributos  
Métodos  
Visibilidade  
Encapsulamento  
Herança  
Polimorfismo  
UML  
Referências



# Métodos

- ❑ Também são chamados de comportamentos;
- ❑ Representam as atividades que uma classe pode executar;
- ❑ Podemos comparar um método a uma função desenvolvida em uma linguagem de programação como, por exemplo, C#;
- ❑ Os métodos podem (ou não) receber parâmetros;
- ❑ Um método retornar (ou não) valores;
- ❑ Representam um conjunto de instruções que são executadas quando eles são chamados.

## AOO

- Introdução
- Classificação
- Abstração
- Objeto
- Classes
- Atributos
- ➔ Métodos
- Visibilidade
- Encapsulamento
- Herança
- Polimorfismo
- UML
- Referências



# Visibilidade

- ❑ Indica o nível de acessibilidade de um atributo ou método;
- ❑ Basicamente, há três modos de visibilidade:
  - ❑ Pública (+)  
Objetos de quaisquer classes podem acessar o atributo ou método
  - ❑ Privada (-)  
Apenas a classe possuidora do método ou do atributo pode ter acesso
  - ❑ Protegida (#)  
Apenas a classe e as subclasses possuidoras do método ou do atributo podem ter acesso

## AOO

- Introdução
- Classificação
- Abstração
- Objeto
- Classes
- Atributos
- Métodos
- ➔ Visibilidade
- Encapsulamento
- Herança
- Polimorfismo
- UML
- Referências



# Encapsulamento

- ❑ Em Programação Orientada a Objetos, significa separar o programa em partes, o mais isoladas possível;
- ❑ O encapsulamento almeja tornar o software mais flexível, fácil de alterar e de criar novas implementações;
- ❑ Quando houver código duplicado é recomendado procurar um lugar para encapsulá-lo.



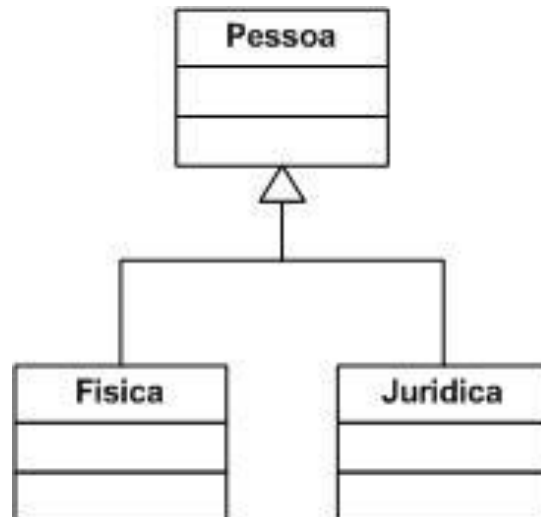
### AOO

Introdução  
Classificação  
Abstração  
Objeto  
Classes  
Atributos  
Métodos  
Visibilidade  
Encapsulamento  
Herança  
Polimorfismo  
UML  
Referências



## Herança

- ❑ Uma das características mais poderosas e importantes da Orientação a Objetos;
- ❑ Permite o reaproveitamento de atributos e de métodos otimizando, assim, o tempo de construção do código;
- ❑ Trabalha com os conceitos de superclasse e subclasse:
  - ❑ Superclasse – também chamada de “classe mãe”, possui classes derivadas dela que são chamadas de subclasses;
  - ❑ Subclasse – também chamada de “classe filha”, herda os métodos e os atributos da sua “classe mãe”.



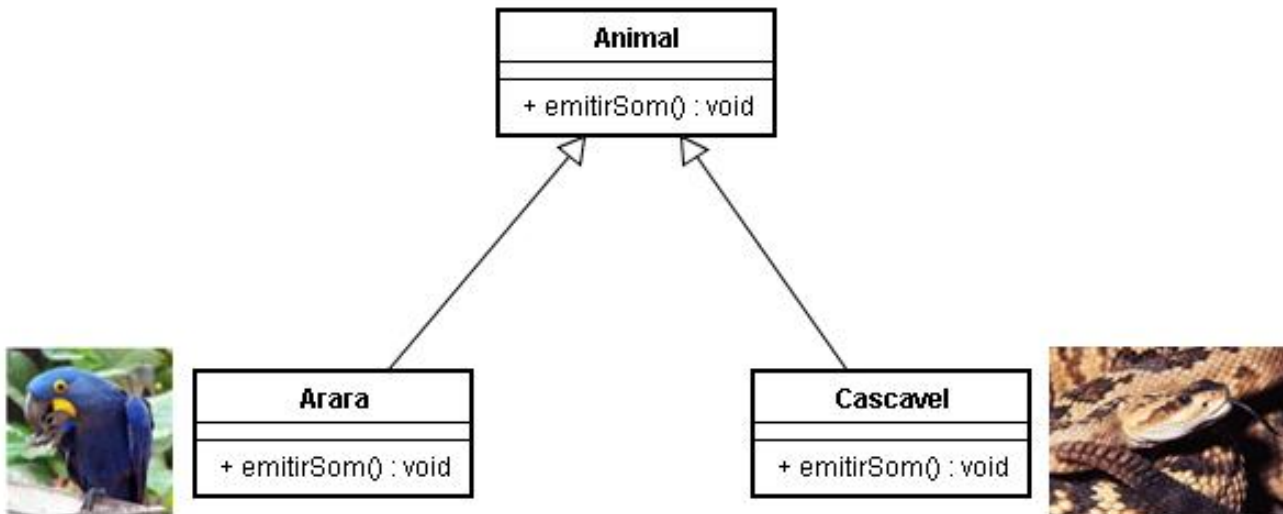
### AOO

- Introdução
- Classificação
- Abstração
- Objeto
- Classes
- Atributos
- Métodos
- Visibilidade
- Encapsulamento
- Herança
- Polimorfismo
- UML
- Referências



# Polimorfismo

- ❑ Seu conceito está associado ao de Herança;
- ❑ Trabalha com a redeclaração de métodos previamente herdados por uma classe;
- ❑ Os métodos, apesar de semelhantes, diferem de alguma forma da implementação utilizada na superclasse. Assim, é necessário a implementação na subclasse:



## AOO

Introdução  
 Classificação  
 Abstração  
 Objeto  
 Classes  
 Atributos  
 Métodos  
 Visibilidade  
 Encapsulamento  
 Herança  
 ➔ Polimorfismo  
 UML  
 Referências



## UML – *Unified Modeling Language*

- ❑ Uma linguagem visual utilizada para modelar sistemas de informação baseado no paradigma Orientação a Objetos;
- ❑ Oferece visualização, especificação, construção e documentação de artefatos de sistema;
- ❑ Tem como objetivo auxiliar:
  - ❑ Requisitos
  - ❑ Comportamento
  - ❑ Estrutura lógica
  - ❑ Dinâmica dos processos
  - ❑ Necessidades físicas da implantação
- ❑ A UML tornou-se uma norma industrial para o desenvolvimento de softwares OO;
- ❑ **Observação:** A UML não é uma linguagem de programação e sim uma linguagem de modelagem.



### AOO

- Introdução
- Classificação
- Abstração
- Objeto
- Classes
- Atributos
- Métodos
- Visibilidade
- Encapsulamento
- Herança
- Polimorfismo
- UML
- Referências





## Referências

- [1]. SIERRA, Katy; BATES, Bert. **Use a cabeça JAVA**. Ed 2, Editora Altabooks.
- [2]. GUEDES, Gilleanes. **UML Uma Abordagem Prática**. Editora Novatec.
- [3]. FURLAN, José. **Modelagem de Objetos através da UML**. Editora Makron Books.
- [4]. CASTRO, Maurício. **Orientação a Objetos**. Solis/Univates (internet).
- [5]. BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML Guia do Usuário**. Editora Campus.

### AOO

- Introdução
- Classificação
- Abstração
- Objeto
- Classes
- Atributos
- Métodos
- Visibilidade
- Encapsulamento
- Herança
- Polimorfismo
- UML
- ➔ Referências