

# Programação Orientada a Objetos

*Modelagem Orientada a Objetos*

Rone Ilídio  
Thiago Oliveira



# Classes

- **As classes formam o núcleo de um programa OO.**
- **Classes suportam os conceitos de:**
  - Abstração;
  - Encapsulação;
  - Proteção de dados;
  - Hierarquia;
  - Polimorfismo.

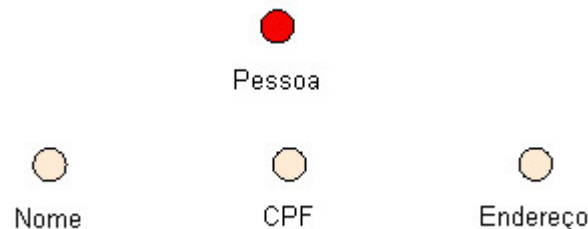
# Tipos Abstratos de Dados

- Abstração é correlacionada aos objetivos de quem se abstrai.
- A abstração do TAD Pessoa depende de seu uso no sistema.

□ Pessoa sob a perspectiva de um médico:



□ Pessoa sob a perspectiva de cadastro de dados:





## Classes - Atributos

- Um atributo é uma propriedade de uma classe que descreve um conjunto de valores que as instâncias da classe, objetos, podem atribuir.
  - Uma classe pode ter zero ou mais atributos.
  - Propriedade persistida x atributo derivado.
- Atributo estático é um atributo cujo valor é compartilhado por todas as instâncias, objetos;
- Atributo não estático possui um valor único para cada objeto, instância da classe.



## Classes - Operações

- Operação é um serviço que pode ser requisitado por qualquer objeto da classe para obter um comportamento.
  - Uma classe pode ter zero ou mais operações;
- Operação abstrata é aquela que não possui um método que a implemente na classe;
  - Uma classe que possui uma ou mais operações abstratas é dita classe abstrata.



## Classe abstrata

- É uma classe que não possui instâncias diretas, objetos, apenas suas classes descendentes.
- São úteis para definir uma estrutura comum a várias classes.
- Facilitam a reutilização de código.
- Uma operação abstrata numa classe define apenas a sua forma, não a sua implementação.

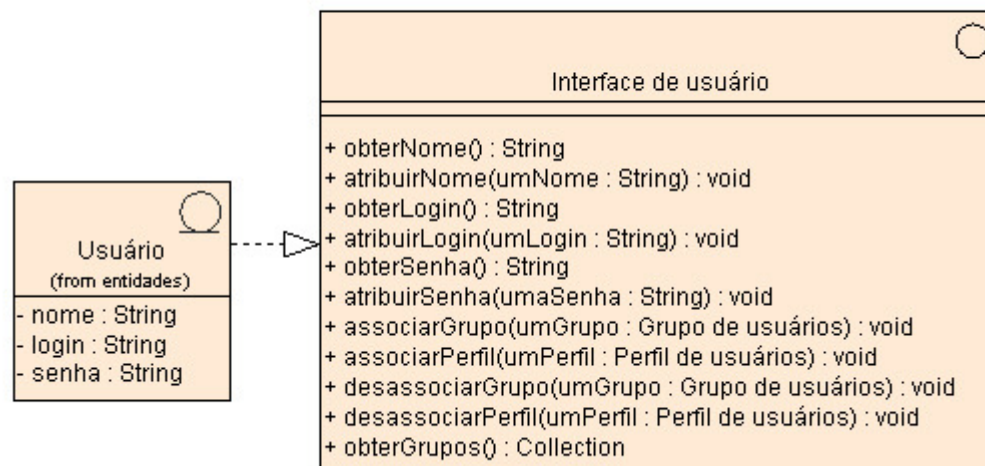


# Interface

- O propósito de uma interface é encapsular um conjunto de operações oferecidas pela classe.
- É comum apresentarmos na interface apenas parte das operações.
- A interface especifica a assinatura das operações.

# Encapsulação

- Encapsulação consiste em ocultar o funcionamento interno de uma classe.
- O acesso aos serviços oferecidos é feito via interface, independentemente do funcionamento interno de uma classe.







# Relacionamentos

- Classes isoladas não compõem um Sistema OO.
- A interação entre os objetos define o comportamento de um Sistema OO.
- As classes colaboram umas com as outras através de relacionamentos:
  - Associações;
  - Agregações e Composições;
  - Herança;
  - Dependência.

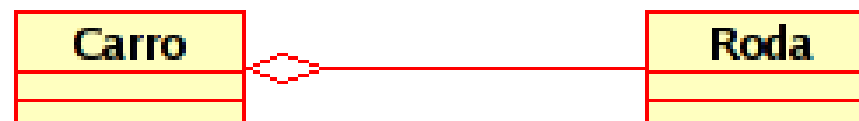
# Associações

- Associações são relacionamentos persistentes entre classes, objetos.
  - Representam relações conceituais entre classes.
- Associações reflexivas são associações de uma classe com ela própria.
  - Não é uma associação do objeto com ele mesmo.



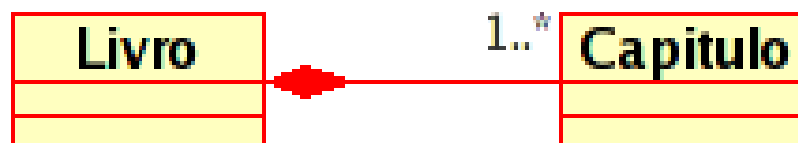
# Agregações

- Agregação é uma forma especial de associação onde o todo está relacionado às suas partes.
  - Representa uma propriedade fraca.
- A frase “parte de” é utilizada para descrever:
  - Grupo de usuários x Usuário.
- O todo é chamado de classe forte e a parte de classe fraca e os ciclos de vida de objetos todo e parte são independentes.



# Composições

- Composição é uma forma especial de agregação onde o todo está relacionado às suas partes.
  - O todo é chamado de classe forte e a parte de classe fraca.
- A frase “composto por” é utilizada para descrever:
  - Acervo x Livros.
- A parte depende do todo para existir:
  - Criar/excluir um objeto todo e então criar/excluir um objeto relacionado parte.





# Herança

- Relacionamento entre classes onde uma classe compartilha a estrutura e o comportamento de uma ou mais classes.
  - É um relacionamento descrito “é um tipo de”.
- Define uma hierarquia de abstrações na qual a subclasse herda de uma ou mais superclasses.
- A subclasse herda os atributos, operações e relacionamentos da superclasse.



# Dependência

- Indica que a mudança em uma classe pode causar mudanças na outra.
  
- Fatores que levam à dependência entre classes:
  - Troca de mensagens entre os objetos das classes;
  - Uma classe tem como atributo outra classe;
  - Uma classe aparece como parâmetro na assinatura da operação de outra classe.