



Programação Orientada a Objetos

Aula 06 – Introdução e conceitos básicos

Paulo Vinicius Vieira

paulo.vieira@faculdadeimpacta.com.br

Objetivos

- Compreender o que é Programação Orientada a Objetos
- Compreender os conceitos de abstração, classe e objeto
- Compreender métodos e atributos
- Aplicar os conceitos na prática

Paradigmas de Programação

- Programação Estruturada
- Programação Funcional
- Programação Orientada a Objetos

Paradigmas de Programação

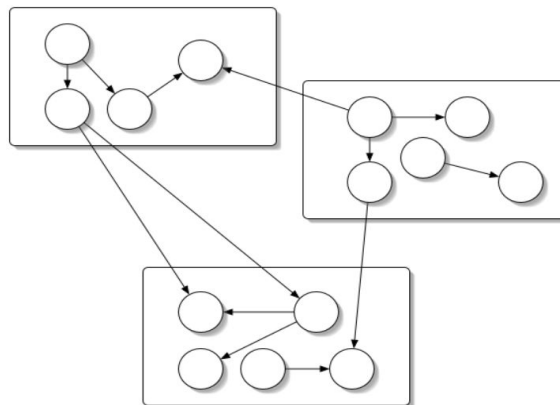
- Programação Estruturada
 - Segue uma sequência de instruções
 - Utiliza variáveis e estruturas de dados para armazenar informações
 - Os programas geralmente são organizados em funções
 - A comunicação entre as funções ocorre pela passagem de dados (parâmetros)

Paradigmas de Programação

- Programação Funcional
 - Baseada em expressões matemáticas
 - Um programa é composto de múltiplas funções
 - Utiliza a composição de funções
 - Uma nova função é construída a partir da composição de outras funções

Paradigmas de Programação

- Programação Orientada a Objetos (POO)
 - É um paradigma onde um programa é estruturado em componentes individuais chamados de objetos
 - Os objetos interagem entre si por meio da troca de mensagens.
 - O principal objetivo da POO é aproximar o mundo digital com o mundo real.



Programação Orientada a Objetos

- Os primeiros conceitos de Orientação a Objetos começaram a ser desenvolvidos na década de 60
- A primeira linguagem de programação Orientada a Objetos foi a linguagem Smalltalk, desenvolvida durante a década de 70.
- Se popularizou com o desenvolvimento das linguagens C++ e Java
- Atualmente, a maioria das linguagens de programação de uso comercial possibilita a Programação Orientada a Objetos

Programação Orientada a Objetos

- Vantagens:

- Reaproveitamento de Código
 - podemos usar o objeto de um sistema que criamos em outro sistema
- Facilidade de Manutenção
 - uma pequena modificação beneficia todas as partes do sistema que usarem o objeto
- Maior confiabilidade no código
 - ao alterar um objeto, nenhum outro objeto é afetado
- Maior robustez
 - O implementação de um sistema se assemelha com o mundo real

Programação Orientada a Objetos

- O que é um **objeto**?

“É a representação de uma coisa do mundo real.”
(BARNES, 2009)
- **Objetos** representam entidades do mundo real
 - Física ou
 - Conceitual

Princípios básicos da POO

A Programação Orientada a Objetos se baseia em 4 pilares:

- Abstração
 - Processo de representar entidades do mundo real
- Encapsulamento
 - Adiciona segurança à aplicação
- Herança
 - Facilita o reuso do código
- Polimorfismo
 - Adiciona a possibilidade de alteração no funcionamento interno dos objetos

Abstração

- A abstração permite modelar determinado objeto do mundo real na forma computacional
- Mapeamos apenas as características relevantes e úteis para a solução de determinado problema
 - Detalhes irrelevantes podem ser omitidos
- Exemplo:
 - A altura de uma pessoa pode não ser útil em um sistema de folha de pagamento

Atributos e Métodos

- Um objeto é constituído por **atributos** e **métodos**
 - **Atributos** representam as características que descrevem um objeto
 - Exemplo: atributos de um objeto Cachorro:
 - nome, idade, cor dos pêlos, peso.

(Um cachorro chamado Rex, de três anos, pêlo de cor marrom, com 8 kg”)
 - **Métodos** são as ações e comportamentos do objeto
 - Exemplo: comportamentos de um objeto Cachorro:
 - andar, latir, comer, dormir, morder, etc.

Atributos

- Os atributos são “Variáveis” ou “campos” que armazenam os valores que descrevem as características dos objetos
- Diferentes cachorros podem ter o mesmo conjunto de atributos, porém com valores distintos.



Cachorro	
Nome:	Rex
Idade:	3 anos
Comprimento o pelos:	Curtos
Cor pelos:	Marrom
Peso:	8 kg



Cachorro	
Nome:	Snoopy
Idade:	2 anos
Comprimento o pelos:	Curtos
Cor pelos:	Branca
Peso:	8 kg

Métodos

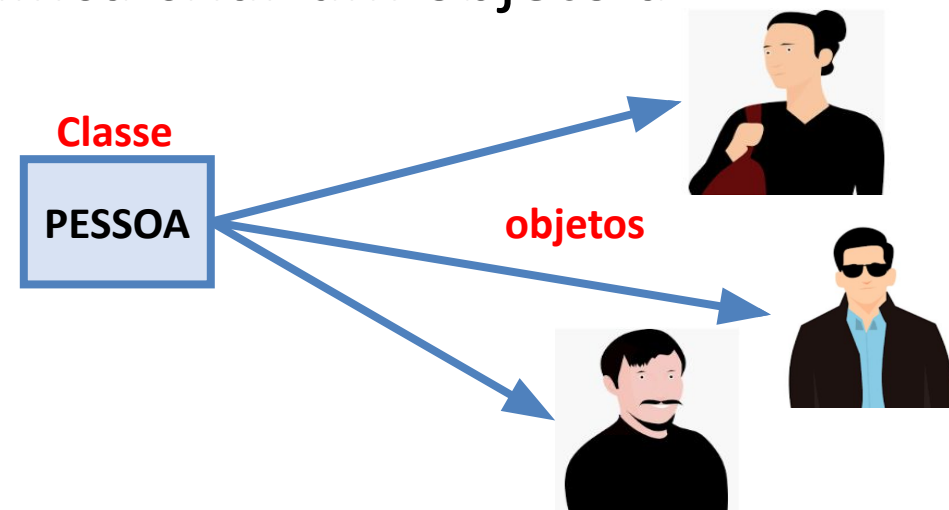
- **Métodos** são os comportamentos (ações) que um objeto pode executar.
- A única maneira de interagir com os objetos é através dos métodos que ele disponibiliza.
- Exemplo: comportamentos de um objeto Cachorro:
 - andar, latir, comer, dormir, morder, etc.
 - Diferentes cachorros podem ter o mesmo conjunto de métodos, porém executados de forma diferente.

Classes

- Todo objeto possui uma **Classe**
 - Classe define o modelo de um objeto.
 - Classe representa a estrutura de um objeto
- A classe define propriedades e comportamentos comuns de um grupo de objetos
 - Uma classe pode agrupar objetos similares, que compartilham os mesmos atributos e métodos

Classes Vs. Objeto

- Os objetos são criados a partir das classes
 - A classe serve de modelo para a criação de novos objetos.
 - João é uma instância da classe Pessoa
 - Maria é outra instância da classe Pessoa
- Um objeto é uma **instância de uma classe**
 - Instanciar um objeto significa criar um objeto a partir de uma classe



Classes

- Uma classe é composta por:
 - **Atributos** – variáveis que caracterizam o estado do objeto
 - **Métodos** – ações que podem ser realizada pelo objeto

Classe Cachorro

Atributos:

- Nome
- Peso
- Cor

Métodos:

- Latir
- Andar
- Dormir

Classe Pessoa

Atributos:

- Nome
- Idade
- Altura

Métodos:

- Andar
- Falar
- Comer

Classes

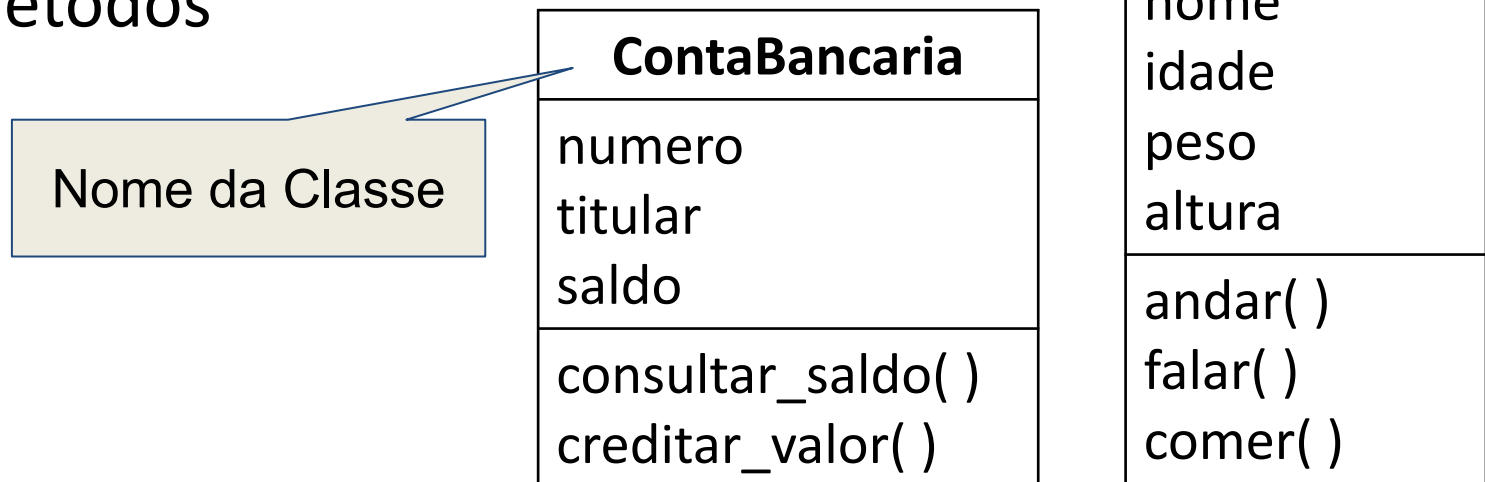
- Representação Gráfica (UML):
- Uma classe pode ser representada graficamente por um retângulo dividido em três seções:
 - Nome da classe
 - Atributos
 - Métodos

ContaBancaria
numero titular saldo
consultar_saldo() creditar_valor()

Pessoa
nome idade peso altura
andar() falar() comer()

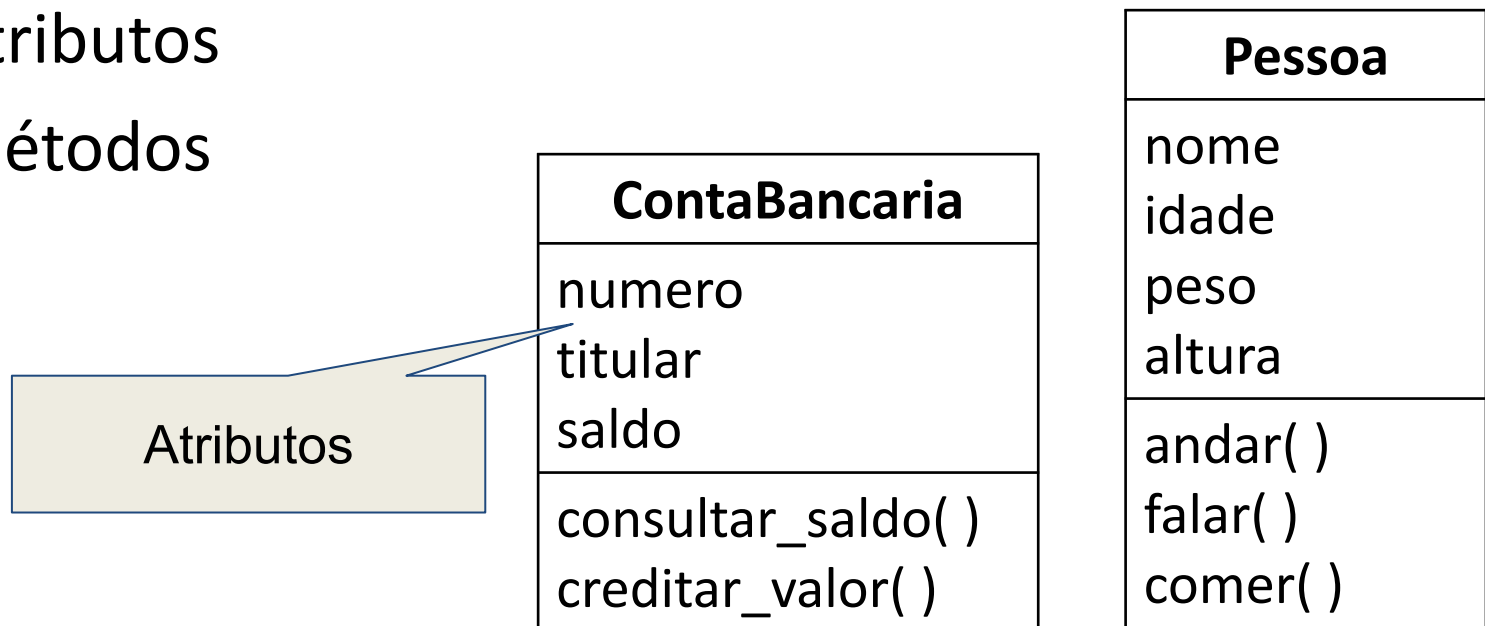
Classes

- Representação Gráfica (UML):
- Uma classe pode ser representada graficamente por um retângulo dividido em três seções:
 - Nome da classe
 - Atributos
 - Métodos



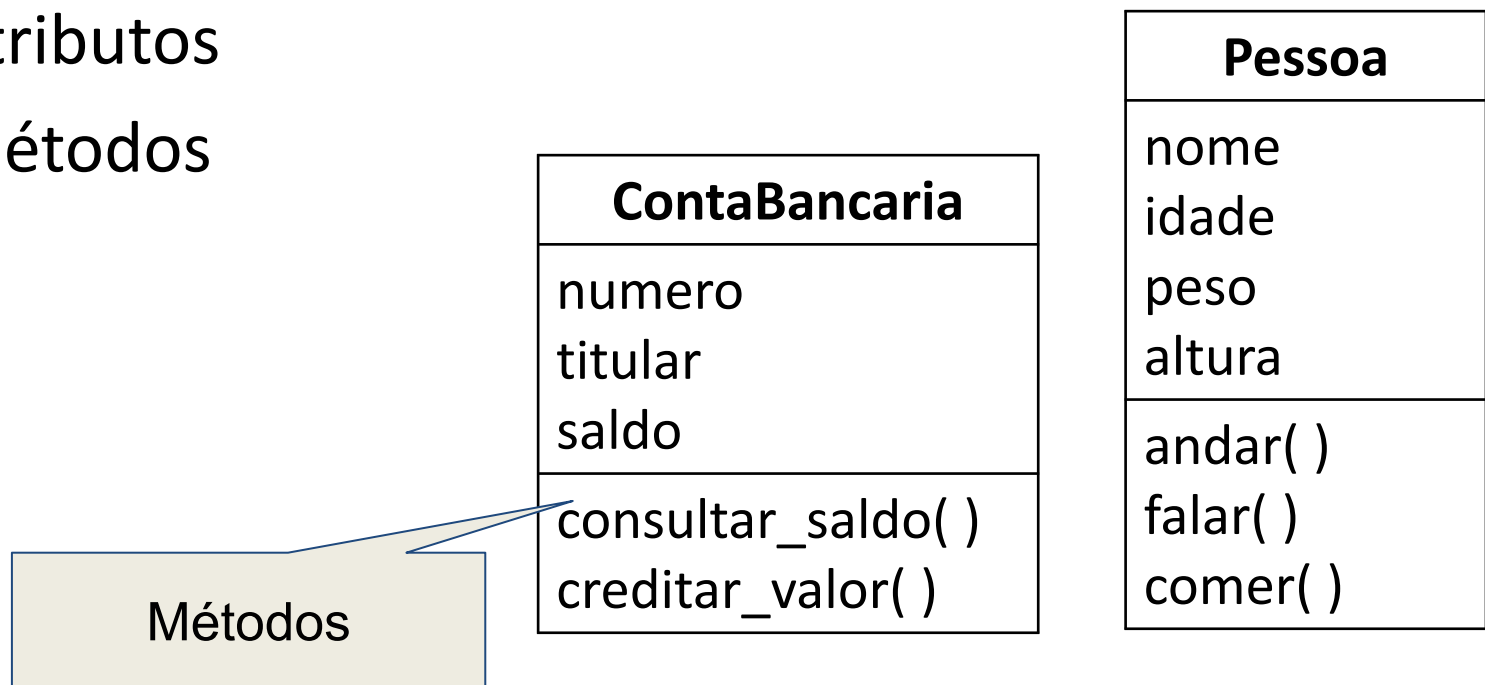
Classes

- Representação Gráfica (UML):
- Uma classe pode ser representada graficamente por um retângulo dividido em três seções:
 - Nome da classe
 - Atributos
 - Métodos



Classes

- Representação Gráfica (UML):
- Uma classe pode ser representada graficamente por um retângulo dividido em três seções:
 - Nome da classe
 - Atributos
 - Métodos



Classes

- Representação Gráfica (UML):
- Uma classe pode ser representada graficamente por um retângulo dividido em três seções:
 - Nome da classe
 - Atributos
 - Métodos

ContaBancaria
numero: int titular: str saldo: float
consultar_saldo() creditar_valor()

Pessoa
nome: str idade: int peso: float altura: float
andar() falar() comer()

Classes

- Representação Gráfica (UML):
- Uma classe pode ser representada graficamente por um retângulo dividido em três seções:
 - Nome da classe
 - Atributos
 - Métodos

ContaBancaria
numero: int titular: str saldo: float
consultar_saldo(): float creditar_valor(valor: float): None

Pessoa
nome: str idade: int peso: float altura: float
andar() falar() comer()