



Computação Orientada a Objetos II

Prof. Dr. Rodrigo Duarte Seabra

Universidade Federal de Itajubá
Bacharelado em Ciência da Computação/Sistemas de Informação

Orientação a Objetos

Aula 02

Prof. Dr. Rodrigo Duarte Seabra

Universidade Federal de Itajubá
Bacharelado em Ciência da Computação/Sistemas de Informação

- ✓ UML está totalmente inserida no **paradigma da orientação a objetos**
- ✓ Compreender o **conceito** de orientação a objetos

✓ **Classes:** grupos de objetos

✓ Cada objeto possui as mesmas características e comportamentos de qualquer objeto do grupo em questão

- pessoa, casa, carro etc...

✓ **Instanciação:** criar um exemplo de um tipo, um grupo, uma classe

- mesmas características e comportamentos de todos os outros objetos já instanciados

✓ **Apesar de possuírem os mesmos atributos, os objetos de uma classe não são exatamente iguais**

- cada objeto pode armazenar **valores diferentes** em seus atributos

- ex: carro azul, verde, preto...

✓ **Classe:** representa uma **categoria**, e os objetos são os membros ou exemplos dessa categoria

✓ **Na UML:**

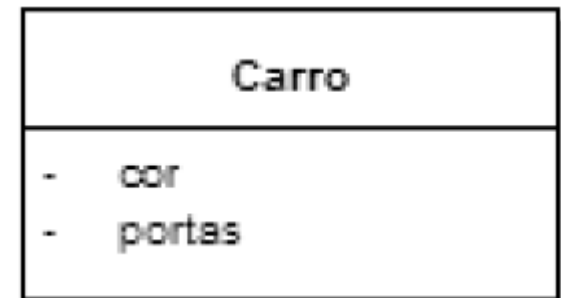
- classe é representada por um retângulo
- pode ter até três divisões



Exemplo de uma classe

Atributos ou Propriedades

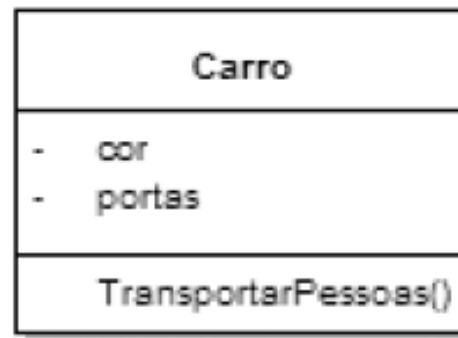
- ✓ **Classes** costumam definir **atributos** (propriedades)
- ✓ **Atributos** representam as **características de uma classe**, ou seja, as particularidades que costumam variar de um objeto para outro
 - ex: altura em um objeto da classe **Pessoa**
- ✓ **Atributo** possui um nome e o tipo de dado que o atributo armazena (inteiro, real, string etc.)
- ✓ Objetos têm os atributos relativos à classe à qual pertencem
- ✓ Todos as **instâncias de uma mesma classe** têm exatamente os **mesmos atributos** que, por sua vez, podem assumir valores diversos



Exemplo de classe com atributos

Métodos, Operações ou Comportamentos

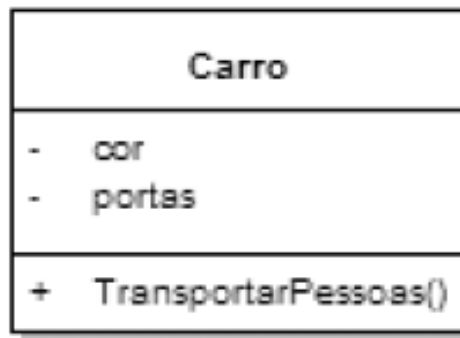
- ✓ Classes costumam ter **métodos**, também conhecidos como operações ou **comportamentos**
- ✓ Um método representa uma **atividade** que um objeto de uma classe pode executar
 - ex: um objeto da classe **Carro** pode executar a atividade de **transportar** **pessoas**
- ✓ Grande parte da codificação propriamente dita dos sistemas orientados a objetos está contida nos métodos definidos em suas classes



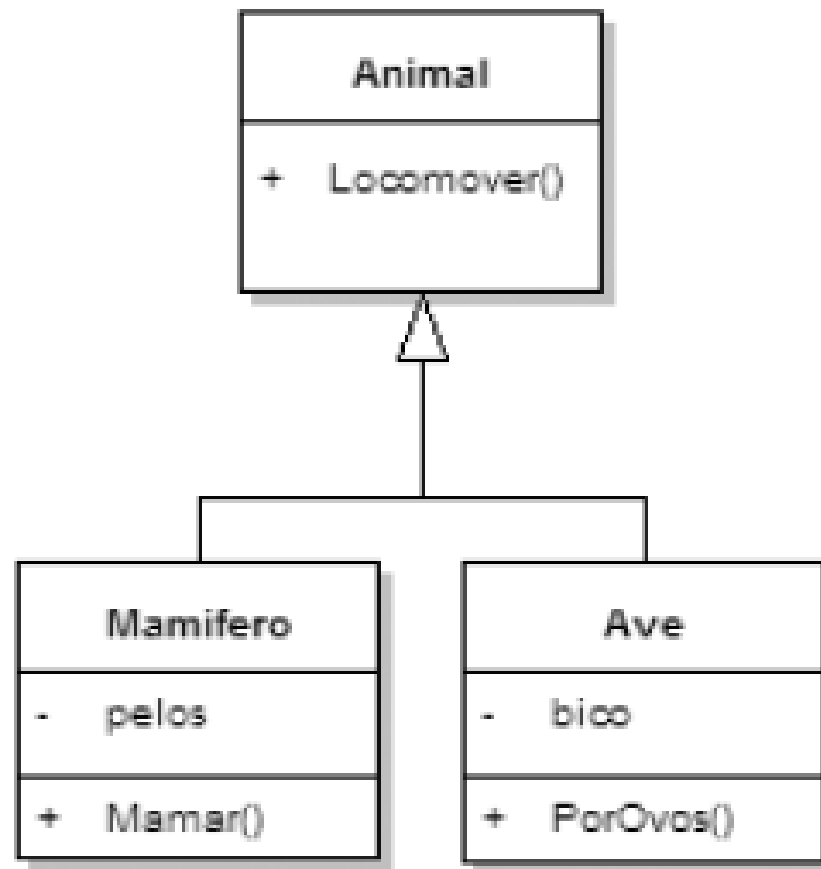
Exemplo de classe com métodos

✓ Indica o **nível de acessibilidade** de determinado atributo ou método, sendo representada à esquerda destes:

- **privada**: (-) somente os objetos da classe detentora do atributo ou método poderão enxergá-lo ou utilizá-lo
- **protegida**: (#) além dos objetos da classe detentora do atributo ou método também os objetos de suas subclasses poderão ter acesso ao mesmo
- **pública**: (+) o atributo ou método pode ser utilizado por qualquer objeto
- **pacote**: (~) o atributo ou método é visível por qualquer objeto dentro do pacote. Somente elementos que fazem parte de um pacote podem ter essa visibilidade



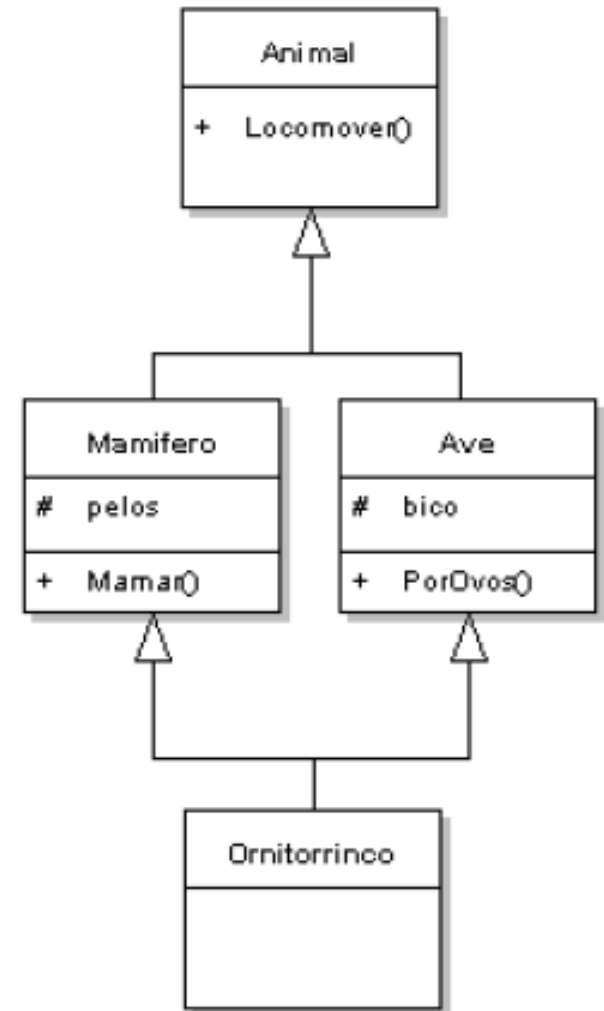
- ✓ Permite o **reaproveitamento de atributos e de métodos**, otimizando o tempo de desenvolvimento, além de permitir a diminuição de linhas de código
- ✓ **Conceito de superclasse (classe-mãe)**: contém classes derivadas a partir dela, chamadas **subclasses** (classes-filha)
- ✓ Subclasses ao serem derivadas a partir de uma superclasse **herdam** suas características, ou seja, seus atributos e métodos
- ✓ A alteração dos métodos da superclasse atualiza imediatamente as subclasses
- ✓ **Conceito de especializações**: classes com características compartilhadas por muitas classes, mas que tenham **pequenas diferenças** entre si



Exemplo de Herança

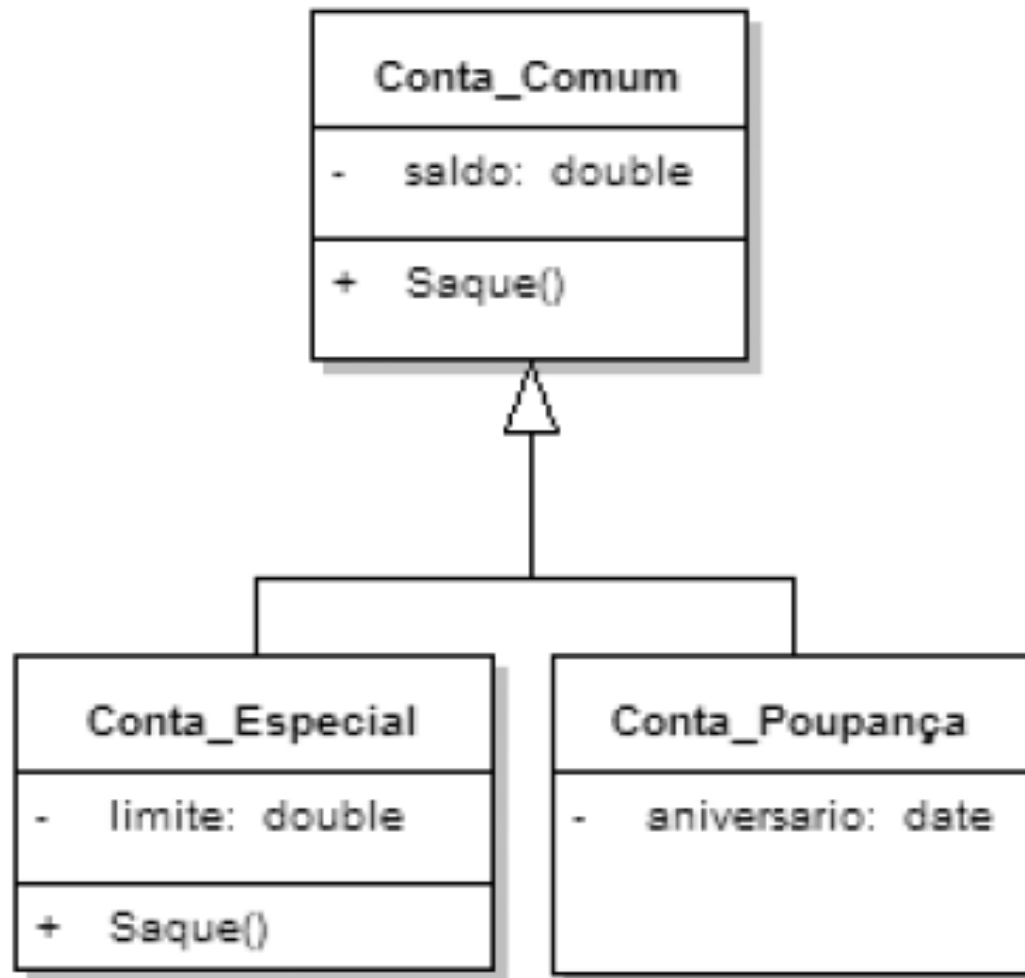
Herança Múltipla

- ✓ Ocorre quando uma **subclasse herda características de duas ou mais superclasses**
- ✓ Uma subclasse pode herdar atributos e métodos de diversas superclasses
- ✓ Nem todas as linguagens de programação oferecem esse tipo de recurso



Exemplo de Herança Múltipla

- ✓ **Conceito** associado à herança
- ✓ Trabalha com a **redeclaração de métodos** previamente herdados por uma classe
- ✓ Embora semelhantes, **diferem de alguma forma** da implementação utilizada na superclasse, sendo necessário, portanto, reimplementá-los na subclasse
- ✓ Podem existir dois ou mais métodos com a mesma nomenclatura, diferenciando-se na maneira como foram implementados
- ✓ O sistema verifica se a classe da instância em questão contém o método declarado nela própria ou se o herda de uma superclasse



Exemplo de Polimorfismo

- ✓ 1 – Defina 2 classes diferentes (com atributos e métodos) e instancie 3 objetos para cada uma
- ✓ 2 – Com base no exercício anterior, defina, para cada classe, 2 ou mais subclasses dando exemplos de instanciação de objetos
- ✓ 3 – Que superclasse poderíamos criar, num relacionamento de herança, para as classes Triângulo, Retângulo e Hexágono? Exemplifique uma operação polimórfica nessa relação.
- ✓ 4 – Defina uma superclasse que sirva para todas as subclasses de cada item a seguir:
 - a) Linha, Ponto, Polígono, Círculo
 - b) Creche, Escola de Nível Médio, Universidade, Curso de Idiomas
 - c) Aéreo, Fluvial, Terrestre
 - d) Prefeito, Governador, Presidente