

Programação Orientada a Objetos

Departamento de Computação
Universidade Federal de Sergipe

Orientação a Objetos

Prof. Kalil Araujo Bispo
kalil@dcomp.ufs.br



Introdução

- Já conhecemos o conceito de classe
- Já conhecemos o conceito de objetos
- O que vamos ver hoje
 - Classificação, Abstração e Instanciação
 - Classes de Objetos
 - Atributos
 - Métodos
 - Visibilidade
 - Herança
 - Polimorfismo

Classificação, Abstração e Instanciação

- O ser humano aprende a pensar de forma orientada a objetos
 - Abstrações
 - Classificações
 - Características
 - Comportamentos
- Ex: Casa, carro, pessoa...

Classificação, Abstração e Instanciação

- **Esforço de abstração**
 - Carro
 - Cores diferentes
 - Formatos diferentes
 - Estilos diferentes
 - Mas todos tem a mesma classificação: carro
 - Carro é um conceito geral que se refere a muitos objetos
 - Características e propriedades em comum

Classificação, Abstração e Instanciação

- **Carro é uma denominação de grupo**
 - Abstração de uma Classe
- **Sempre que for observado um objeto com as características determinadas**
 - O objeto é um exemplo do grupo carro
 - Ou seja, é uma Instância da classe Carro

Classificação, Abstração e Instanciação

- **Nosso modo de aprender OO**
 - Aprendemos por meio de classificações
 - Grupos de objetos com características iguais: Classes
 - Um novo conceito → Uma nova classe
 - Todo o objeto com características dessa classe é um exemplo: Instância da classe
 - Instância: criar um exemplo de um tipo, um grupo, uma classe.

Classificação, Abstração e Instanciação

- Depois de abstrair um conceito inicial, costumamos criar subdivisões dentro das classes
 - Carro: marca, modelo
 - Pessoa: profissão, parentesco
- Apesar de terem os mesmos atributos, objetos de uma classe não são iguais

Classificação, Abstração e Instanciação

- **Abstração**

- Habilidade de concentrar nos aspectos essenciais de um contexto qualquer, ignorando características menos importantes ou acidentais

- **Encapsulamento**

- Consiste na separação de aspectos internos e externos de um objeto

Classes de Objetos

- Uma classe representa uma categoria
- Os objetos são os membros ou exemplos dessa categoria



Atributos ou Propriedades

- Classes costumam definir atributos, também chamados de propriedades
- Representam a característica de uma classe
 - Peculiaridades que variam de um objeto para outro
- Não é a classe que contém os atributos, mas as instâncias
 - Não é possível trabalhar com as classes, mas com as instâncias

Atributos ou Propriedades

- **A classe carro**
 - Não existe realmente
 - É uma abstração
 - Uma forma de classificar e identificar um grupo de objetos semelhantes
- **Outro exemplo: Planta de uma casa**
 - Exprime o conceito de classe e a partir dela constroem-se casas
 - Apenas nas casas reais é onde se pode morar

Atributos ou Propriedades

- Os objetos têm os atributos relativos à classe à qual pertence
- São características do objeto
- Todas as instâncias de uma classe tem os mesmos atributos, mas valores diferentes

cor	Carro
portas	

Métodos, Operações ou Comportamentos

- **Classes costumam ter métodos, também conhecidos como operações ou comportamento**
 - Representa uma atividade que um objeto possa executar
 - Equivalente a uma função
 - Pode receber parâmetros de entrada
 - Pode retornar valores

Métodos, Operações ou Comportamentos

- Conjunto de instruções que são executados
- Grande parte da codificação encontra-se nos métodos

Carro
cor portas
transportar()

Visibilidade

- Indica o nível de acessibilidade de um atributo ou método
- Existem basicamente 4 tipos de visibilidade
 - Privada → Apenas objetos da classe detentora
 - Publica → Qualquer objeto
 - Protegida → Apenas objetos da classe detentora ou subclasses
 - Pacote → Qualquer objeto dentro do pacote

Herança

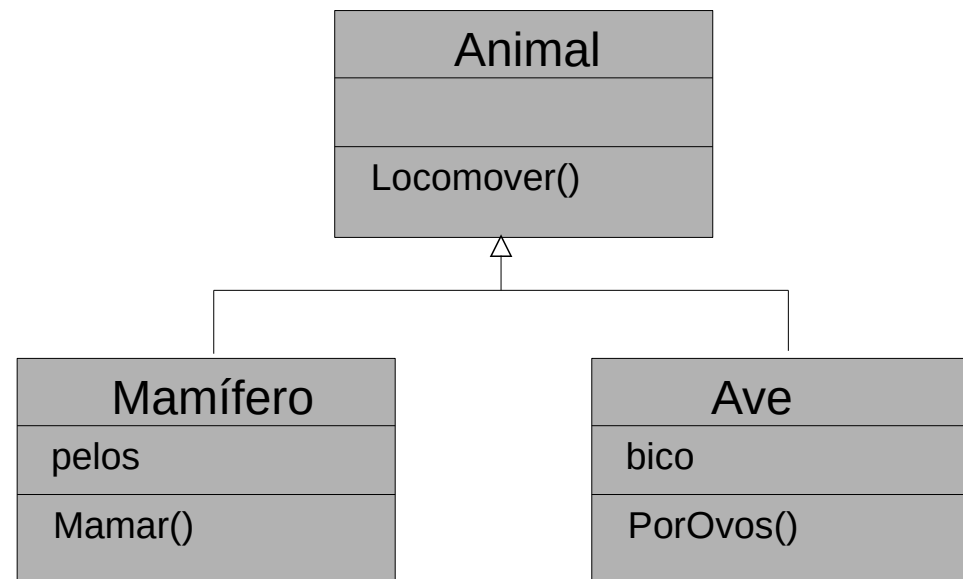
- Característica importante da orientação a objetos
- Permite o reaproveitamento de atributos ou métodos
- Permite a diminuição de linhas de código
- Facilita manutenções futuras

Herança

- Conceitos de superclasses e subclasses
- As subclasses, ao serem derivadas a partir de uma superclasse, herdam suas características
 - Permite reutilização de código pronto
 - Declaração de métodos e atributos exclusivos das subclasses
- Torna mais ágil o processo de desenvolvimento

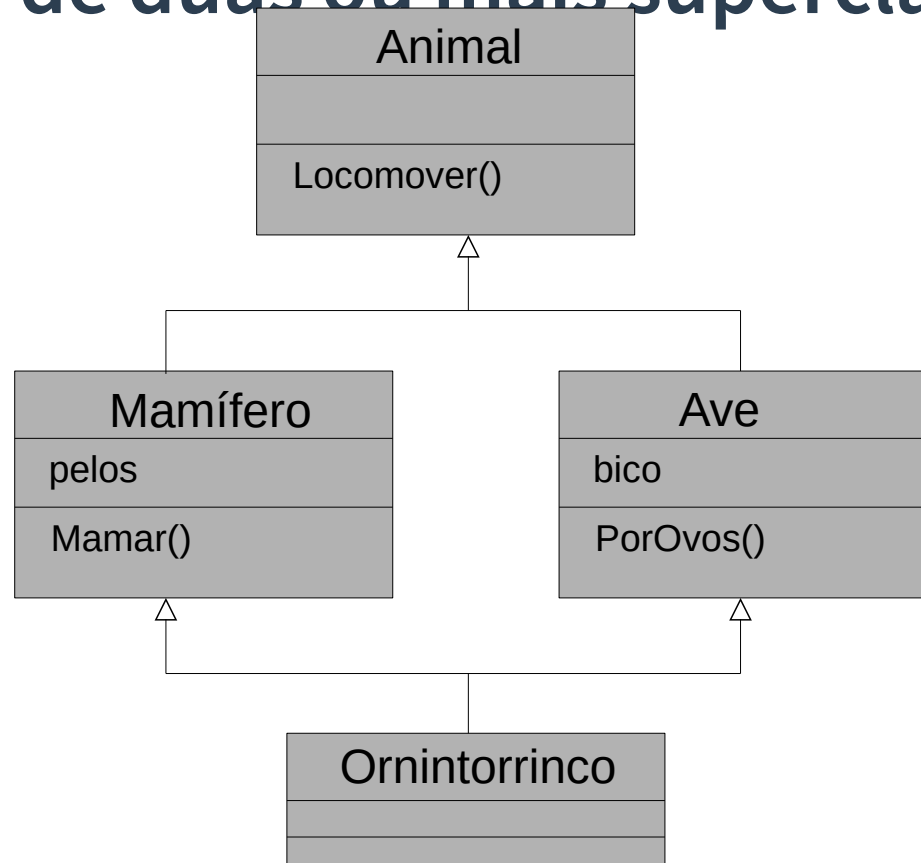
Herança

- Permite trabalhar com especializações
 - Classes gerais, com características comuns a muitas classes
 - Subclasses, detalhando características exclusivas das mesmas



Herança Múltipla

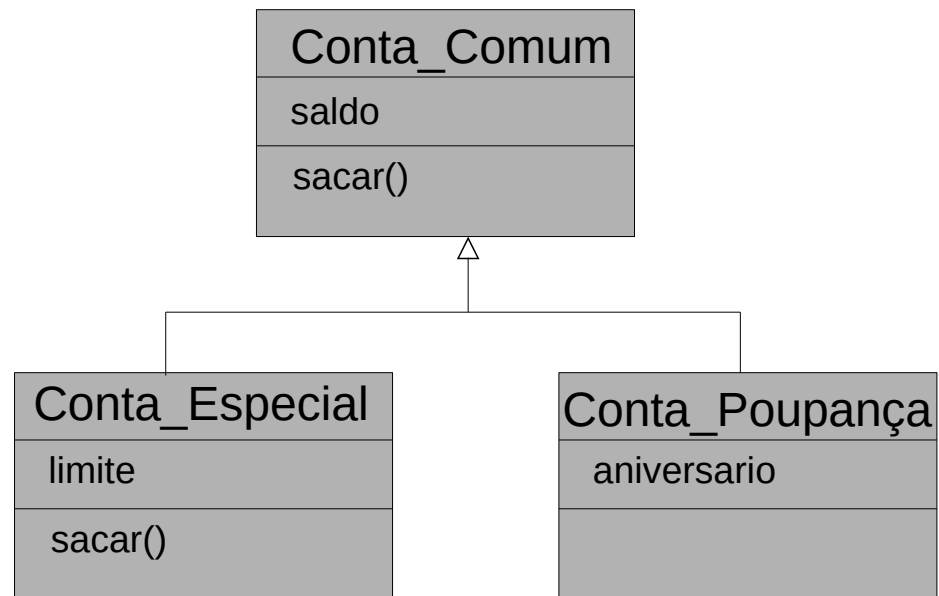
- Ocorre quando uma subclasse herda características de duas ou mais superclasses



Polimorfismo

- Conceito relacionado à herança
- Redecaração de métodos previamente herdados por uma classe
 - Diferem de alguma forma da implementação utilizada na superclasse
 - Evita-se modificar o código da superclasse
 - Pode então existir métodos com a mesma nomenclatura, mas com implementações diferentes

Polimorfismo



Polimorfismo

- Inclusão - um ponteiro para classe mãe pode apontar para uma instância de uma classe filha
- Paramétrico - se restringe ao uso de templates (C++, por exemplo) e generics (C#/Java)
- Sobrecarga - duas funções/métodos com o mesmo nome mas assinaturas diferentes
- Coerção - conversão implícita de tipos sobre os parâmetros de uma função

Referências

- UML 2: Uma Abordagem Prática. Gilleanes T. A. Guedes. Novatec Editora, 2009.