

# Aula 03

Site: MoodleWIFI  
Curso: Analise de sistemas  
Livro: Aula 03  
Impresso por: RIANE RUBIO  
Data: Friday, 12 Apr 2019, 19:54

# Sumário

1. Definição
2. Exemplos de classe
3. Classes e visões
4. Classificação e instanciação
5. Generalização / Especialização
6. Considerações
7. Herança
8. Composição
9. Classes abstratas
10. Relacionamentos

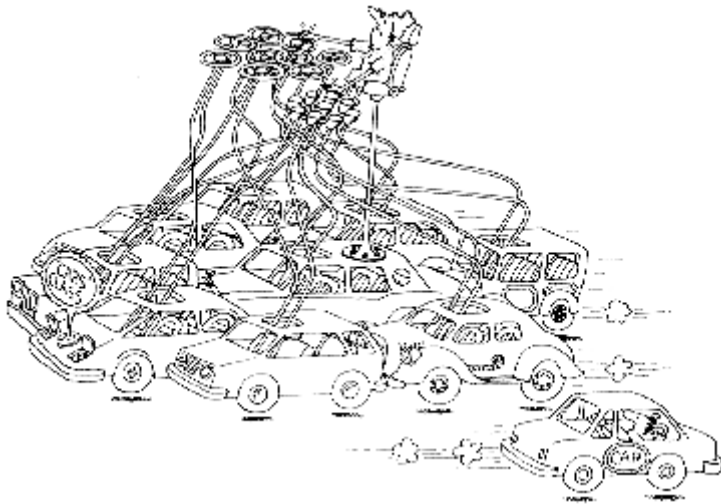
# 1. Definição

Todos os **objetos** estão armazenados em estruturas que identificam os seus **atributos** e os seus **métodos**.

Analogamente às tabelas, dos **bancos de dados relacionais**, onde os comandos **DDL** (*Em SQL*) permitem a criação de sua estrutura, as classes, em orientação a objetos, propiciam a montagem da estrutura dos objetos que a comporão.

As **classes** são estruturas **genéricas** de onde os objetos podem derivar.

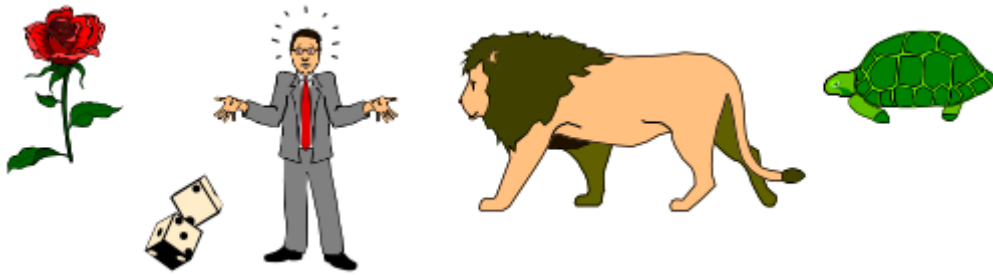
Cada **objeto** é uma linha de vida à parte e cada um possui um estado específico, sendo que cada objeto é diferente do outro (*Em seu conteúdo e estado*) mesmo que a sua interface seja a mesma.



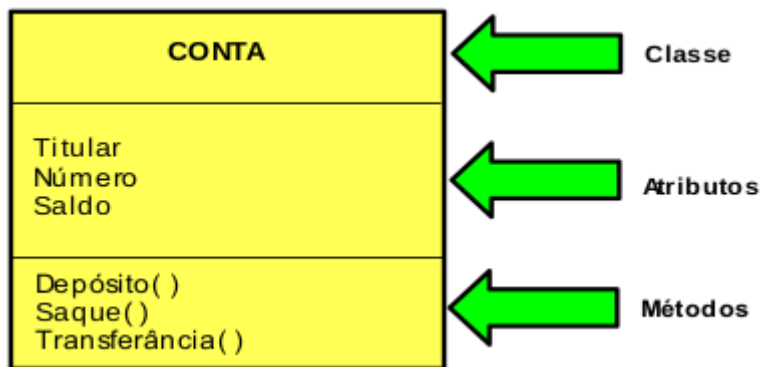
"Uma classe é um grupo de objetos que compartilham uma estrutura comum e um comportamento comum"

## 2. Exemplos de classe

Existem vários **tipos de objetos**, portanto existem vários **tipos de classes**:



Graficamente, podemos representar uma classe da seguinte forma:



- Todos os objetos são instâncias de uma classe.
- Classe é o que se projeta e programa
- Objeto é o que se cria durante o processamento

### 3. Classes e visões

A interface de uma classe provê a **visão externa** e, portanto, enfatiza a **abstração** enquanto esconde sua **estrutura** e segredos de seu **comportamento**.

A **implementação** de uma classe é sua **visão interna**, que engloba os segredos de seu **comportamento**.

A interface de uma classe divide-se em:

- **Pública** (Representada pelo símbolo “+”) → Uma declaração que forma parte da interface de uma classe e é visível para todos os clientes que são habilitados para isto;
- **Protegida** (Representada pelo símbolo “#”) → Uma declaração que forma parte da interface de uma classe, mas não é visível para quaisquer outras classes, exceto suas subclasses;
- **Privada** (Representada pelo símbolo “-”) → Uma declaração que forma parte da interface de uma classe, mas não é visível para quaisquer outras classes.

## 4. Classificação e instanciação

### Classificação

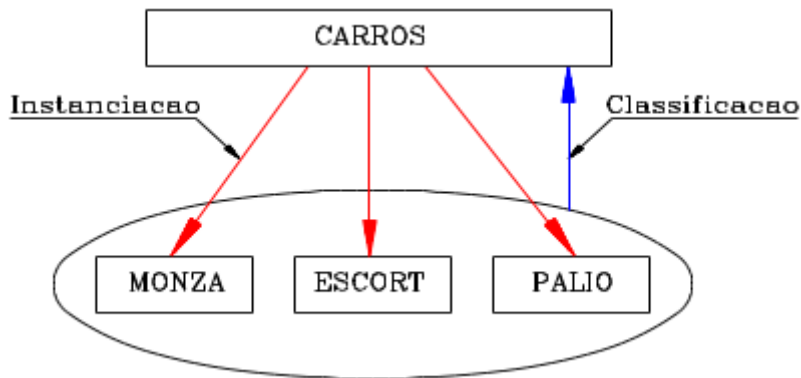
É a operação que divide em uma ou mais categorias (*classes*) indivíduos dentro de um domínio;

### Instanciação

É a operação que evidencia um indivíduo (*Objeto*) que pertence a uma categoria.

São operações opostas.

Quando se **une** vários indivíduos em uma categoria, é uma **classificação**, mas quando se **pega** um único indivíduo da categoria, é uma **instanciação**.



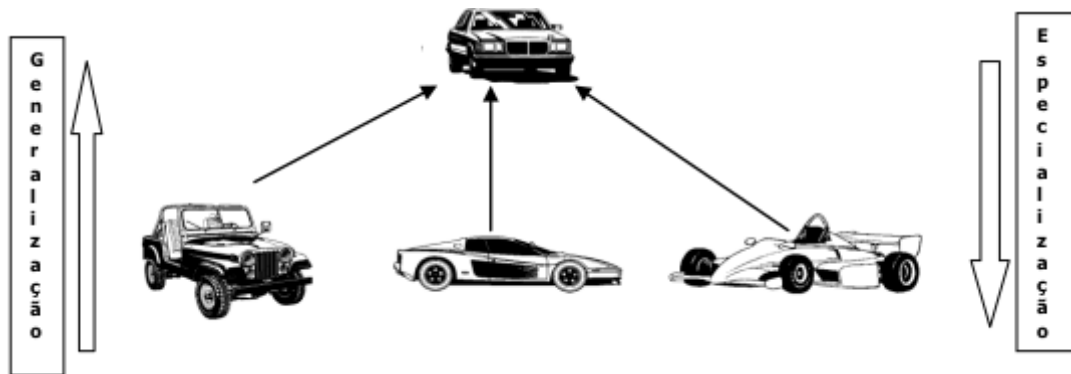
# 5. Generalização / Especialização

## Generalização

Ato de definir-se uma Classe (*supertipo*) que irá conter as características e o comportamento que outras Classes possuem em comum;

## Especialização

Ato de definir-se uma Classe que além de herdar as características e o comportamento do Supertipo ainda possui outras características que são sua especialidade (*somente ela tem*) e, portanto, os demais subtipos não possuem



## 6. Considerações

Uma Classe derivada herda a estrutura de dados e métodos de sua classe “*base*”, mas pode seletivamente:

- adicionar novos métodos;
- estender a estrutura de dados;
- redefinir a implementação de métodos já existentes.

Uma Classe “*base*” propicia a funcionalidade que é comum a todas as suas classes derivadas, enquanto que uma classe derivada proporciona a funcionalidade adicional que especializa seu comportamento.

A **Generalização** é uma noção **conceitual**.

**Herança** é a **implementação** de uma generalização.



## 7. Herança

**Herança** é o compartilhamento de **atributos** e **operações** entre classes com base em um relacionamento **hierárquico**.

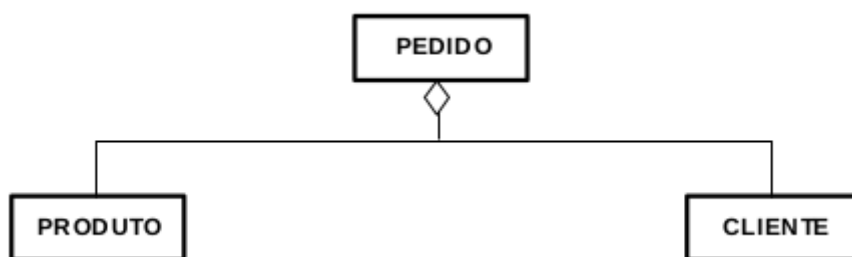
Cada **subclasse** herda todas as propriedades da **superclasse** e acrescenta suas próprias e exclusivas

características, permitindo assim, a reutilização de especificações comuns.

### Composição por agregação

Propriedade de uma Classe que forma suas instâncias a partir da reunião das características e do comportamento de outras instâncias da mesma Classe ou de Classes diferentes, sendo que os componentes têm **existência independente** em relação ao composto, ou seja, a eliminação do composto não afeta a existência dos elementos que o criaram.

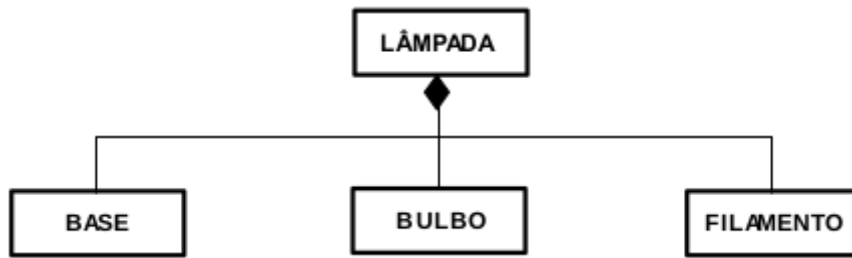
É uma associação entre um objeto e outros objetos que fizeram parte dele.



Se **A** agrega **B** então **B** é parte de **A**; mas seus tempos de vida são independentes.

## 8. Composição

Propriedade de uma Classe formar suas instâncias a partir da reunião das características e do comportamento de outras instâncias da mesma Classe ou de Classes diferentes, sendo que os componentes **não têm existência independente** em relação ao composto, ou seja, a eliminação do composto afeta a existência dos elementos que o criaram.



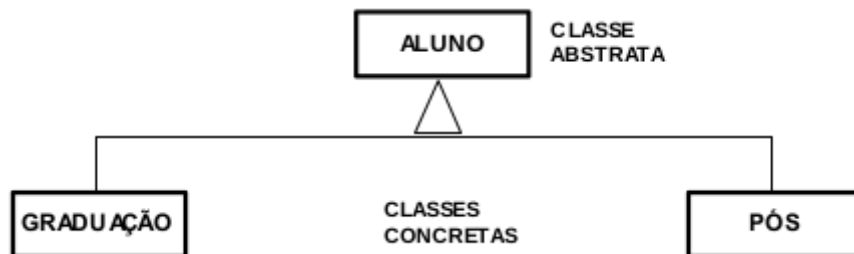
Se **A** é composto de **B** então **A** controla o tempo de vida de **B**.

## 9. Classes abstratas

São Classes utilizadas apenas para organizar a estrutura, já que ela não será instanciada.

Ela define um conjunto mínimo de atributos e métodos virtuais, que deverão ser implementados pelas **SubClasses**.

Uma Classe abstrata sugere uma técnica de “*design*” e muitas vezes aumenta a clareza da modelagem.



Uma Classe Abstrata é uma classe que:

- Provê organização;
- Não possui instâncias;
- Possui uma ou mais operações abstratas;

Possui subclasses que implementam estas operações.

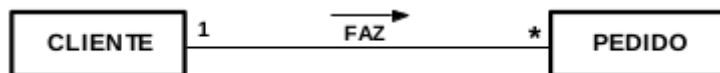
Uma operação abstrata só determina a existência de um comportamento não definindo uma implementação.

# 10. Relacionamentos

Um relacionamento modela uma conexão física ou conceitual entre objetos.

Os relacionamentos são sempre **bidirecionais** e permitem uma visualização do número de objetos que podem ser associados.

Todo relacionamento deve possuir um verbo que o identifica.



Relacionamentos possíveis:

