



UNIFACS
LAUREATE INTERNATIONAL UNIVERSITIES

Programação Orientada a Objetos

Aulas nº X: Relacionamento entre Classes

Prof. Luis Gustavo Araujo
2018

Objetivo

Compreender os conceitos e aplicações para os diversos tipos de relacionamento entre classes (Associação, Agregação e Composição).

Classes

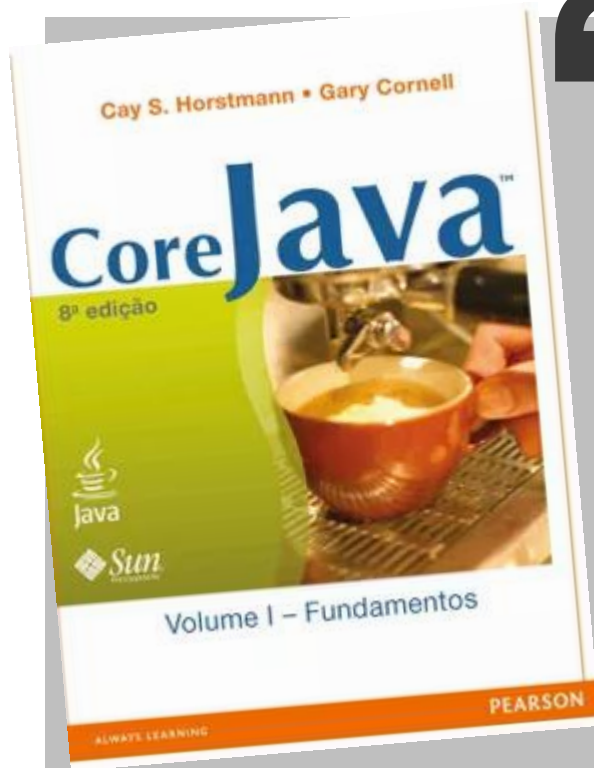
Associação

POO

Relacionamento entre Classes

Há diversos autores que falam sobre o tema, alguns tratam alguns conceitos como similares, outros os diferenciam.
Vejamos:

Relacionamento entre Classes



“

Os relacionamentos mais comuns entre classes são:

- *dependência (“usa um”)*
- *agregação (“tem um”)*
- *herança (“é um”).*

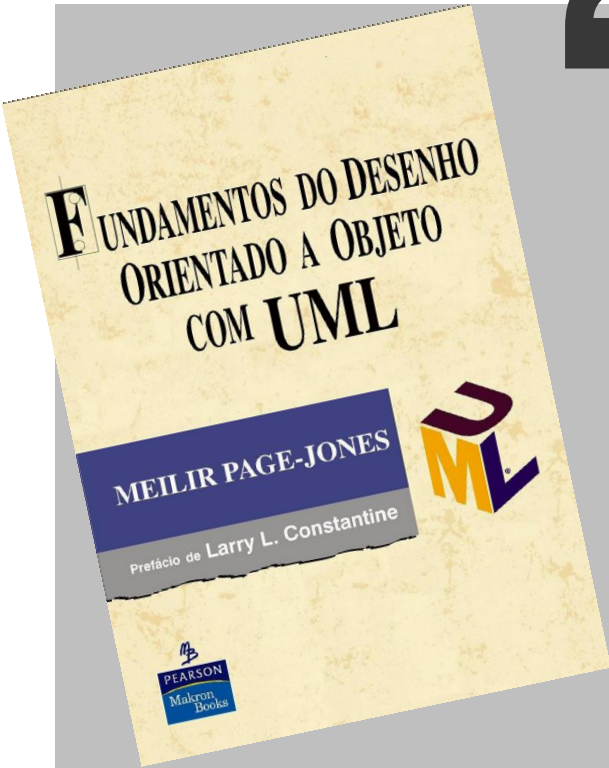
***Hortmann
& Cornell***

Core Java

Relacionamento entre Classes

“

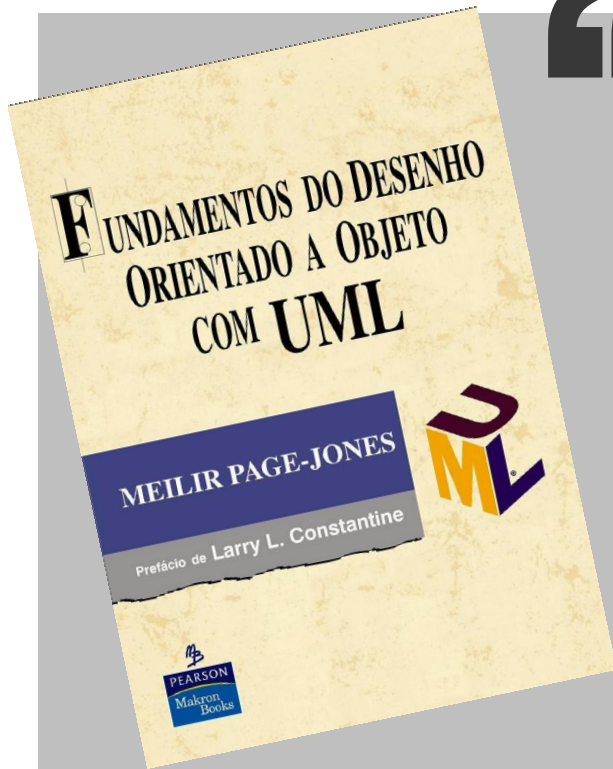
Uma associação representa uma população variada de vínculos (links) de relacionamento entre instâncias de classe.



**Meilir
Page-Jones**

Fundamentos do Desenho Orientada a Objeto com UML

Relacionamento entre Classes



“

Existem duas formas de associação todo/parte:

- *Composição;*
- *Agregação.*

***Meilir
Page-Jones***

Fundamentos do Desenho Orientada a Objeto com UML

Dependência (usa um)

É quando uma Classe precisa de outra classe para executar alguma tarefa. Ou seja, os elementos de uma Classe podem - esporadicamente - usar elementos de outras classes.

Dependência

```
1. public class Calc {  
2.  
3.     static double exp(int n1, int n2){  
4.         return Math.pow(n1, n2);  
5.     }  
6. }
```


Associação (*tem um*)

Agregação ocorre quando uma Classe contém atributos de outra Classe (*tem um*), mas não há relação todo/parte.

Ex.: Pessoa tem cachorro, mas cachorro não faz parte de pessoa.

Associação (*tem um*)

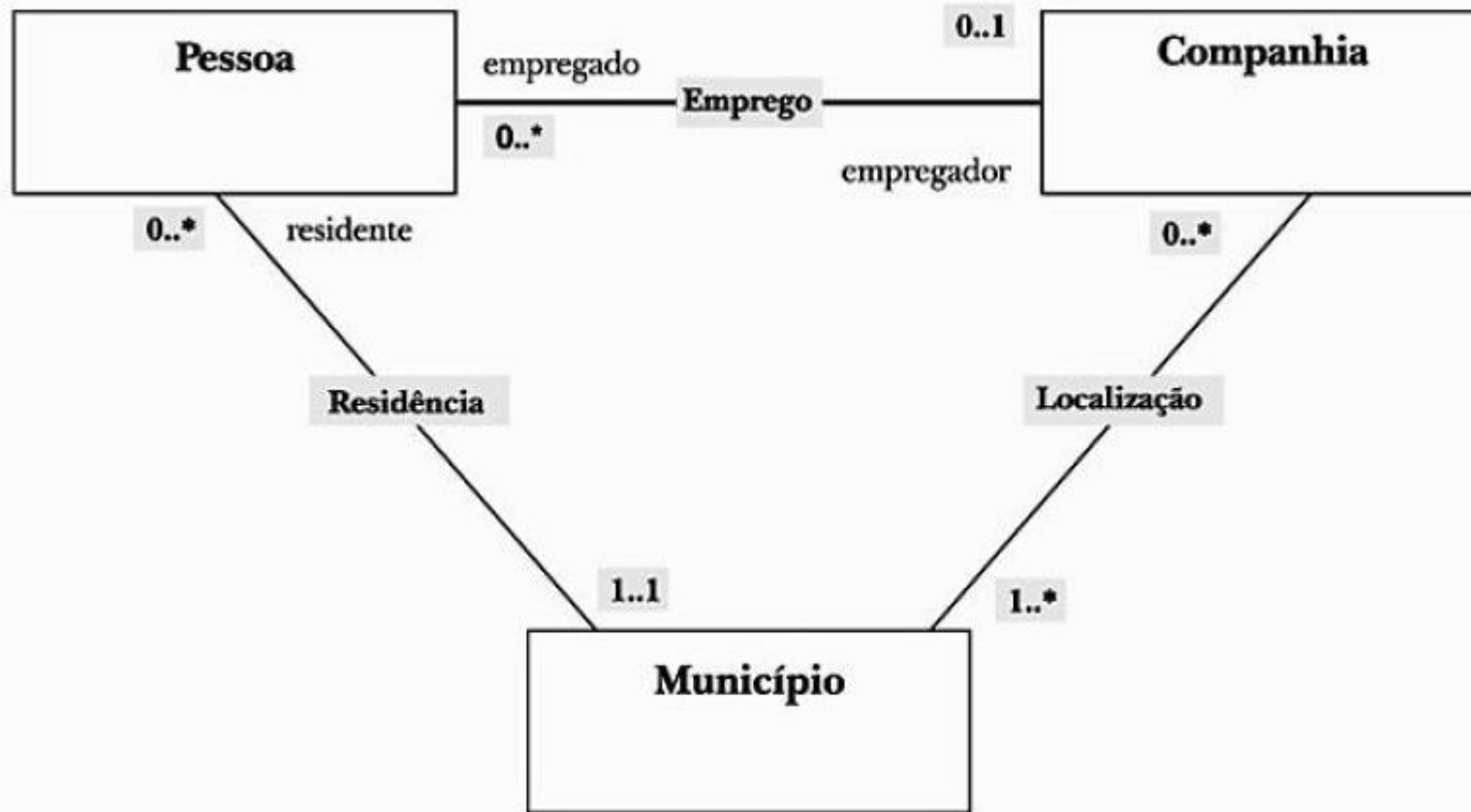


Figura. 4.11 Três associações entre pares de classes.

Associação (*tem um*)

```
1. public class Pessoa {  
2.  
3.     private Municipio residencia;  
4.     private Campanhia emprego;  
5.  
6. }
```

Agregação

Muito comum em sistemas, pois muitos elementos do dia-a-dia são compostos. *Ex: Uma cidade é um agregado de casas, Uma Florestas um agregado de árvores ...*

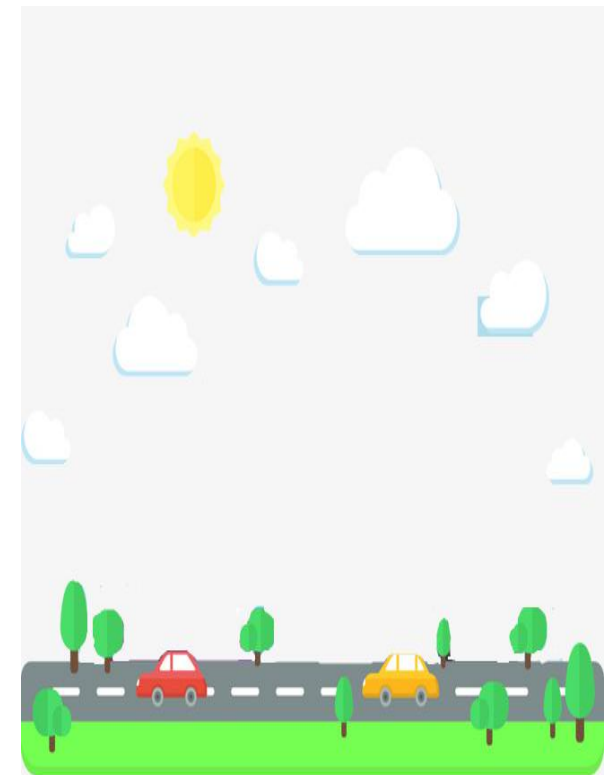
Na relação de Composição, o todo é denominado de *[objeto] Agregado* e as partes de *[objetos] Constituinte*.

Agregação (Características)

Existe três características principais, na agregação, a primeira é:

a) O objeto agregado pode potencialmente existir sem o seu objeto constituinte. *Mas nem sempre isso faz sentido no mundo real.*

Uma cidade devastada, sem casas, ainda é uma cidade!



Agregação (Características)

A segunda é:

b) Um objeto pode ser constituinte de mais de um agregado.

Uma casa pode estar em duas cidades?



Agregação (Características)

A segunda é:

b) Um objeto pode ser constituinte de mais de um agregado. *No mundo real, o objeto pode ou não fazer uso dessa propriedade.*

Uma casa pode estar em duas cidades?



Moradores do loteamento Marisol recebem boletos do IPTU de Salvador e Lauro de Freitas

MAIS INFORMAÇÕES |



0

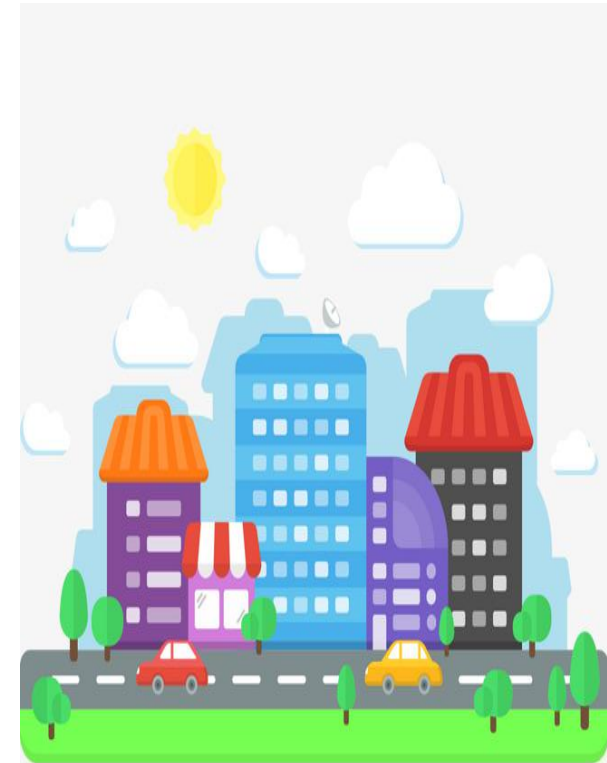


Agregação (Características)

A terceira é:

b) A agregação tende a ser homeômera (*Cuja as partes são semelhantes*).

Uma cidade é composta de várias casas (todas pertencem à mesma Classe).



Agregação (*tem um - fraco*)

Agregação ocorre quando uma Classe contém atributos de outra Classe (tem um), mas a sua relação é fraca.

*Cada classe funciona independentemente e tem o seu próprio tempo de vida.

Agregação (*todo/parte*)

Alguns autores tratam de Agregação como uma relação *todo/parte*, ou seja o atributo do tipo da Classe B, faz parte da Classe A.

Ex: Escola (todo), Professores (parte)

Agregação (*todo/parte*)

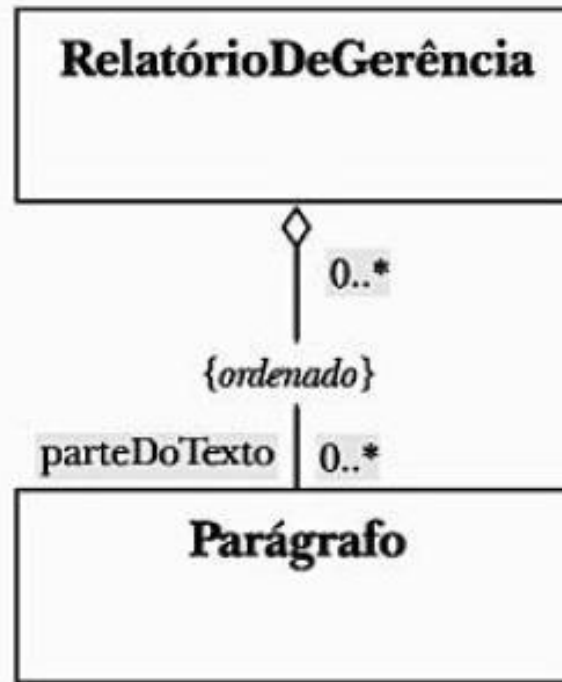


Figura 4.17 Objeto agregado e seus constituintes.

Agregação (*todo/parte*)

```
1. public class Relatorio{  
2.  
3.     private Paragrafo paragrafo;  
4.  
5. }
```

Composição

Muito comum em sistemas, pois muitos elementos do dia-a-dia são compostos. *Ex: Pessoa é composta por Perna, Cabeça, Braço...*

Na relação de Composição, o todo é denominado de *[objeto] Composto* e as partes de *[objetos] Componentes*.

Composição (Características)

Existe três características principais, na composição, a primeira é:

a) O objeto composto não existe sem os seus componentes.

Pessoa, não existe sem braços, pernas e etc (isso é questionável, professor!)

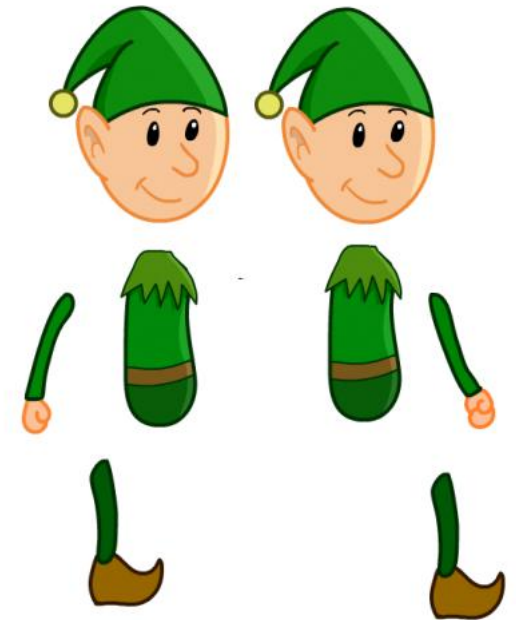


Composição (Características)

A segunda é:

b) Um objeto componente só pode ser parte de um objeto composto (todo).

Pessoas não compartilham braços, pernas e etc (isso é questionável, professor!)



Composição (Características)

A terceira e última é:

b) A Composição é normalmente heterômera (as partes não são semelhantes).

Pessoas não são formadas apenas de Braços (Não, isso não é questionável!)



Composição (*tem um- forte*)

Composição ocorre quando uma Classe contém atributos de outra Classe (tem um), mas a sua relação é forte.

*O tempo de vida da classe parte é limitado pelo tempo de vida da classe todo.

Composição (*todo/parte*)

Alguns autores tratam de Composição como uma relação *todo/parte*, ou seja o atributo do tipo da Classe B, faz parte da Classe A.

Ex: Escola (todo), Cursos (parte)

Composição (*todo/parte*)

126 FUNDAMENTOS DO DESENHO ORIENTADO A OBJETO COM UML

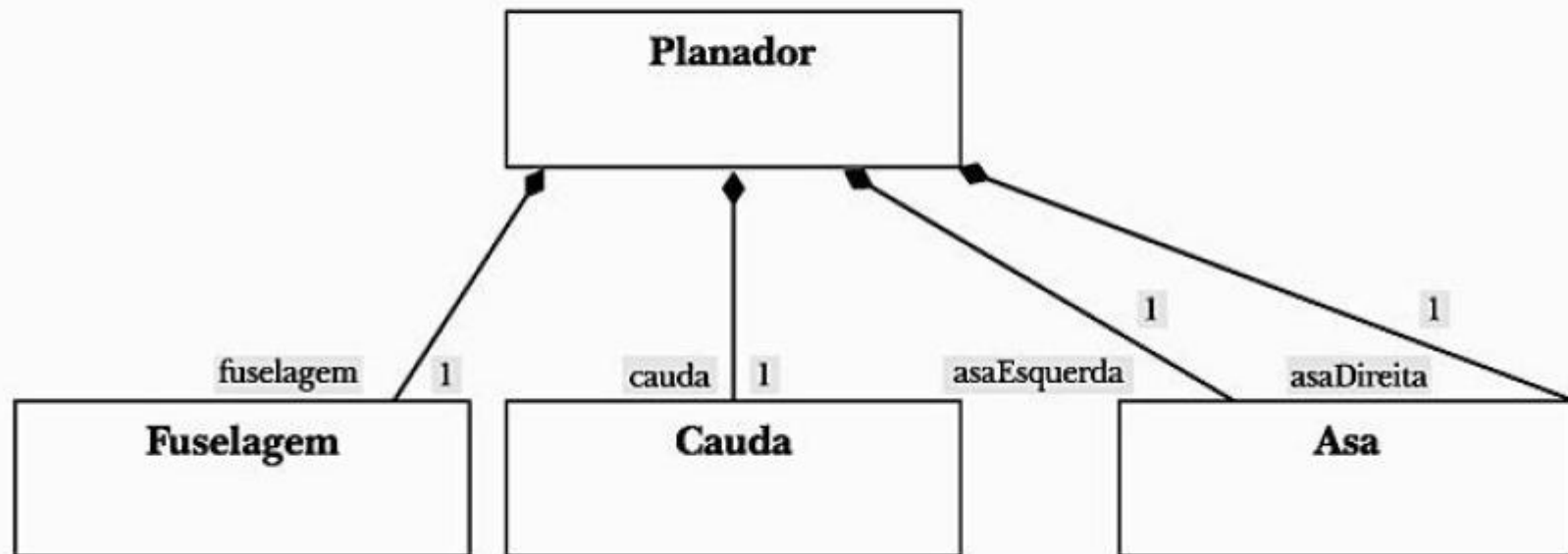


Figura 4.16 Objeto composto e seus componentes.

Composição (*todo/parte*)

```
1.  public class Planador {  
2.  
3.    private Fuselagem fuselagem;  
4.    private Cauda cauda;  
5.    private Asa asa;  
6.  
7.  }
```

Revisando...

Vamos pensar um
Pouco....



Por que usar POO?

1 Dependência (Parâmetro ou Retorno).

2 Associação (Tem Uma).

3 Agregação (Parte/Todo) com vida própria.

4 Composição (Parte/Todo) vida limitada.

Atividade

1 - Construa modelos que usem as relações vistas em sala, suas classes, atributos e métodos* (* caso necessário):

- Dependência;
- Associação;
- Agregação;
- Composição.

2 - Identifique no Projeto de Gestão Escolar, a presença dessas relações.

Referências Técnicas

ECKEL, B. **Pensando em Java** (tradução de Thinking in Java. 3. ed. Prentice-Hall, Dezembro 2002.) - Online.

HORSTMANN, S.; CORNELL, G.; **Core JAVA**. 1. vol. São Paulo: Pearson Education, 2010.

TURNI, R. **Desbravando Java e Orientação a Objetos**: um guia para o iniciante da linguagem. São Paulo: Casa do Código, 2016.

Livro: MEILIR, P. **Fundamentos do Desenho Orientado a Objeto com UML**. São Paulo: Maykon Books, 2001. 462p.



UNIFACS
LAUREATE INTERNATIONAL UNIVERSITIES

Programação Orientada a Objetos

Aulas nº X: Relacionamento entre Classes

Prof. Luis Gustavo Araujo
2018