

# Relacionamento entre Classes: Agregação e Composição



Prof. Jeferson Souza, MSc. (thejefecomp)  
[thejefecomp@neartword.com](mailto:thejefecomp@neartword.com)



# Relacionamento entre Classes

## Nem as classes sobrevivem sem relacionar-se...

As classes, e seus respectivos objetos, também podem estabelecer relacionamentos entre si, os quais são fundamentais para o desenho de *software* orientado a objetos. Existem diferentes tipos de relacionamentos entre classes, a ser dois deles tipos especiais do relacionamento de **Associação** chamados de **Agregação** e **Composição** [OMG, 2017].

# Relacionamento de Associação

## Afinal, o que é o relacionamento de Associação? [OMG, 2017]...

O relacionamento de **Associação** estabelece uma relação entre classes, a qual é especificada pela presença de atributos do tipo de uma determinada classe em outra. De maneira geral, toda vez que uma classe define um atributo, cujo tipo é especificado por outra classe, tem-se configurado um relacionamento de **Associação** entre elas.

# Relacionamento de Associação

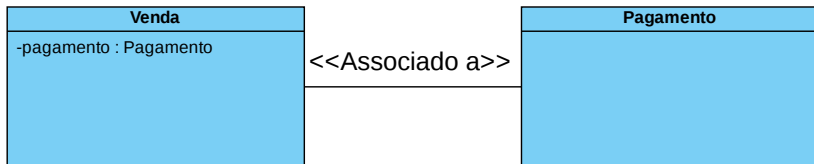
## Diferentes tipos e especialidades de Associação [OMG, 2017]...

Existem diferentes formas e especialidades de **Associação**, são elas:

- ▶ Associação sem navegação;
- ▶ Associação com navegação;
- ▶ Agregação\* ;
- ▶ Composição\* .

\* Os relacionamentos de **Agregação** e a **Composição**, por serem tipos especializados do relacionamento de **Associação**, também podem ser especificados com (ou sem) navegação.

# Relacionamento de Associação - Exemplo



Código-fonte em Java:

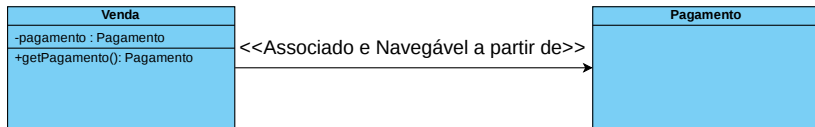
```
public class Venda{  
    private Pagamento pagamento;//Atributo a caracterizar a Associação  
    sem Navegação.
```

```
...
```

```
}
```

Relacionamento de **Associação sem Navegação**

# Relacionamento de Associação - Exemplo



Código-fonte em Java:

```
public class Venda{
    private Pagamento pagamento;//Atributo a caracterizar a Associação.
    public Pagamento getPagamento(){
        //Método a caracterizar a Navegação para Pagamento.
        return this.pagamento;
    }
    ...
}
```

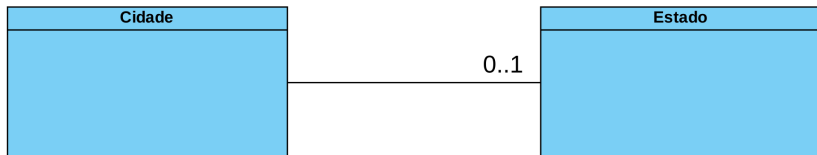
Relacionamento de **Associação com Navegação**

# Relacionamento de Associação

## Presença de cardinalidade (Opcional) [OMG, 2017]...

A presença de cardinalidade permite classificar o relacionamento, a especificar de forma visual quantas instâncias de uma determinada classe estão associadas à outra. Essa cardinalidade é representada no lado da relação que deseja-se, forçosamente, que seja caracterizado. Devido à presença da cardinalidade, as representações dos modelos podem conter (ou não) os atributos a caracterizar claramente o relacionamento.

# Relacionamento de Associação - Exemplo com Cardinalidade 0..1



Relacionamento de **Associação** com cardinalidade 0..1 [i.e pode existir até um (1) estado associado à uma dada cidade]

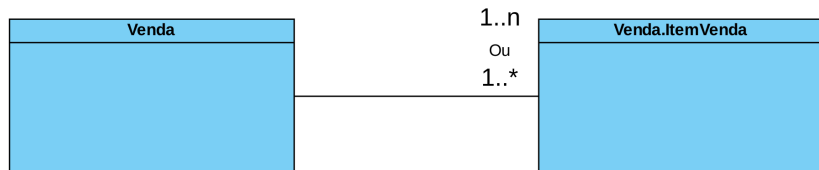


# Relacionamento de Associação - Exemplo com Cardinalidade 1



Relacionamento de **Associação** com cardinalidade 1 (i.e existe exatamente um (1) pagamento para cada Venda)

# Relacionamento de Associação - Exemplo com Cardinalidade 1..n ou 1..\*



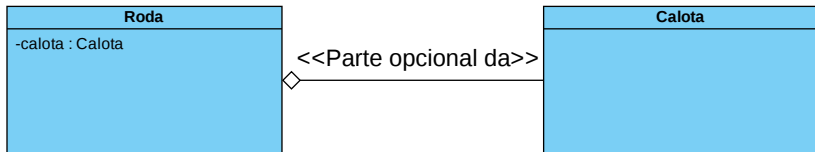
Relacionamento de **Associação** com cardinalidade 1..n ou 1..\* [i.e. deve existir ao menos um (1) item na venda, mas podem existir até n itens, onde o n pode ser inclusive um (1)]

# Relacionamento de Agregação

## Afinal, o que é o relacionamento de Agregação? [OMG, 2017]...

O relacionamento de **Agregação** é um tipo especializado do relacionamento de **Associação**, a especificar uma relação de “parte de” que é opcional. O modelo especificado pela classe que possui o atributo agregado existe sem a presença do referido atributo, a indicar que sua referência pode ser vazia (nula). Exemplo: Uma calota faz parte de uma roda, mas a roda existe (conceitualmente a falar) sem a calota.

# Relacionamento de Agregação - Exemplo



Código-fonte em Java:

```
public class Roda{
    private Calota calota; //Atributo a caracterizar a Agregação.
    ...
}
```

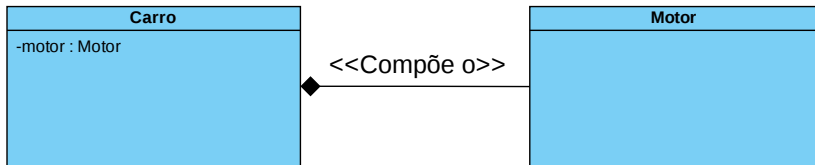
Relacionamento de **Agregação**

# Relacionamento de Composição

## Afinal, o que é o relacionamento de Composição? [OMG, 2017]...

O relacionamento de **Composição** é um tipo especializado do relacionamento de **Associação**, a especificar uma relação de “parte de” que é obrigatória. O modelo especificado pela classe que possui o atributo que compõe o modelo, existe somente com a presença do referido atributo, a indicar que sua referência não pode ser vazia (nula). Exemplo: Um motor faz parte de um carro, mas um carro não existe (conceitualmente a falar) sem a presença de um motor.

# Relacionamento de Composição - Exemplo



Código-fonte em Java:

```
public class Carro{
    private Motor motor;//Atributo a caracterizar a Composição.
    ...
}
```

Relacionamento de **Composição**

# Agregação versus Composição - Diferença

## Afinal, qual é a diferença entre os relacionamentos de Agregação e Composição?

A diferença é meramente conceitual. No relacionamento de **Agregação** o atributo a ser agregado não é parte obrigatória para definir o modelo, i.e., sua referência pode ser vazia (nula); enquanto que no relacionamento de **Composição** o atributo que compõe o modelo é obrigatório, i.e, sua referência não pode ser vazia (nula).

# Bibliografia



BOYARSKY, J. and SELIKOFF, S. **“Oracle Certified Associate Java SE 8 Programmer I: Study Guide”**. Sybex. Indianapolis, Indiana, USA. 2015.



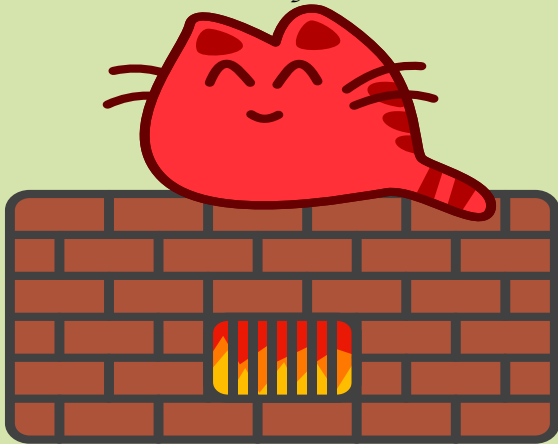
BOYARSKY, J. and SELIKOFF, S. **“Oracle Certified Associate Java SE 8 Programmer II: Study Guide”**. Sybex. Indianapolis, Indiana, USA. 2016.



OMG **“OMG Unified Modeling Language (OMG UML) - Version 2.5.1”**. Object Management Group (OMG).  
Dezembro, 2017. Disponível em:  
<https://www.omg.org/spec/UML/2.5.1/PDF>. Acesso em:  
21 Jun. 2021.



*That's it folks!*



*Thank you for your attention!*