

# Modelação e Design

## 14: Diagrama de Classes

Leonor Melo  
leonor@isec.pt

1

### Diagrama de classes

- Relações entre classes:
  - agregação partilhada
  - agregação composta
  - generalização
- Constraints

Leonor Melo

14 Diagramas de classe

2

2

## Agregação

- Relação do tipo "todo-parte"
- Binária
- Assimétrica
- Transitiva
- Pode ser
  - Partilhada
  - Composta

Leonor Melo

14 Diagramas de classe

3

3

## Agregação partilhada

- Agregação partilhada
  - Também conhecida por agregação
- Semanticamente não é clara a distinção entre
  - agregação partilhada
  - associação
- Considerada mais forte que a associação denota uma relação de "posse"

Leonor Melo

14 Diagramas de classe

4

4

## Agregação partilhada

- O objeto de uma das classes (o todo) "é dono" do objeto da outra classe (a parte)
  - Uma das componentes do todo é a parte
  - A parte pode ser partilhada por vários todos
  - A parte pode existir sem o todo
- em código muitas vezes traduz-se por um atributo que é uma referência para um objeto

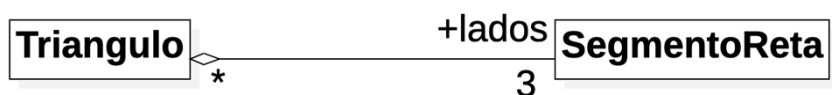
Leonor Melo

14 Diagramas de classe

5

5

## Agregação partilhada



- Um triângulo é composto por três lados
- Um segmento de reta pode pertencer a vários triângulos
  - Ou a nenhum

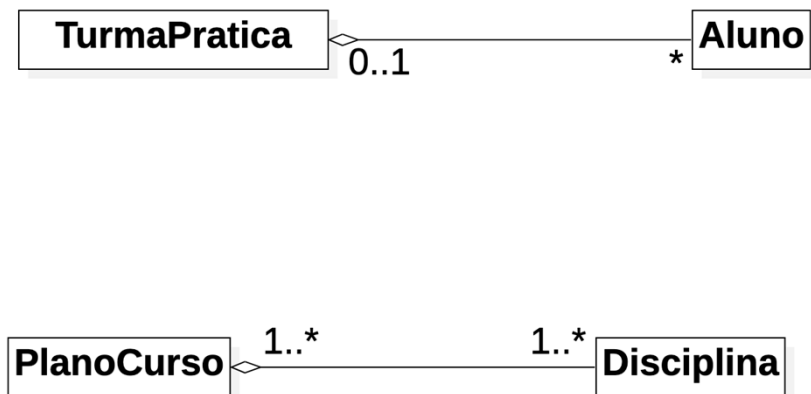
Leonor Melo

14 Diagramas de classe

6

6

## Agregação partilhada: outros exemplos



Leonor Melo

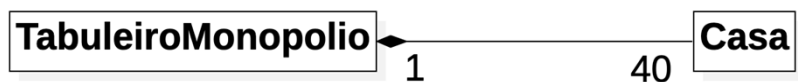
14 Diagramas de classe

7

7

## Agregação composta

- Agregação Composta:
  - também conhecida por composição
- Mais forte das agregações
  - Uma das componentes do todo é a parte
  - A parte pertence apenas a um todo
  - A parte não existe sem o todo



Leonor Melo

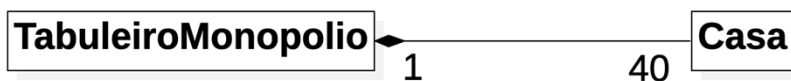
14 Diagramas de classe

8

8

## Agregação composta

- Em código muitas vezes traduz-se por um atributo que é um objeto
- Normalmente o todo é quem tem a responsabilidade de criar, destruir, e "propagar" as mensagens à parte



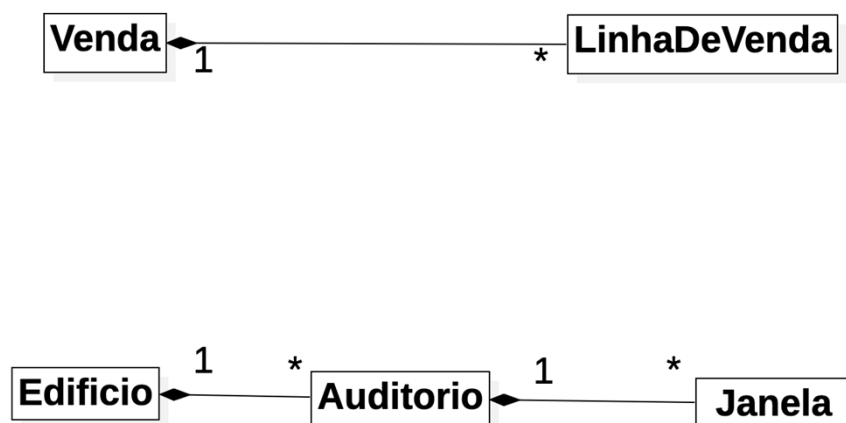
Leonor Melo

14 Diagramas de classe

9

9

## Agregação composta outros exemplo



Leonor Melo

14 Diagramas de classe

10

10

## Generalização

- Usada para descrever uma classe que é *um tipo* de outra classe
- Uma classe relaciona-se com outra
  - Tendo uma
    - associação,
    - agregação,
    - composição
  - Sendo uma
    - generalização
    - especialização

Leonor Melo

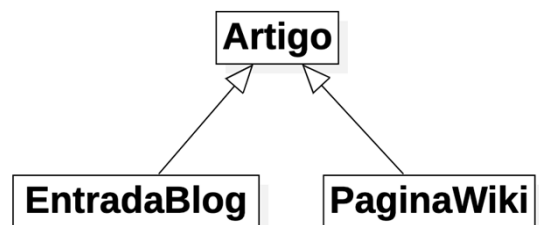
14 Diagramas de classe

11

11

## Generalização UML

- Classe mais geral
  - Classe mãe
  - Classe base
  - Superclasse
- Classe mais especializada
  - Classe filha
  - Classe derivada



Leonor Melo

14 Diagramas de classe

12

12

## Generalização

- Classe derivada
  - herda
    - Atributos
    - Operações
  - (que sejam visíveis)
  - Pode acrescentar
    - Atributos
    - Operações
  - que só façam sentido na classe derivada
- Generalização
  - Só faz sentido num sentido

Leonor Melo

14 Diagramas de classe

13

13

## Generalização e reutilização

- Generalização
  - Permite reutilização
  - Mas é o acoplamento mais forte
- Design deve favorecer acoplamento fraco
  - Usar generalização
    - se uma classe for uma especialização de outra
  - Não usar generalização
    - apenas para reutilizar algumas partes

Leonor Melo

14 Diagramas de classe

14

14

## Generalização múltipla

- Possível de representar em UML
  - Existe casos em que é útil
  - Algumas linguagens permitem
- Potencialmente problemática

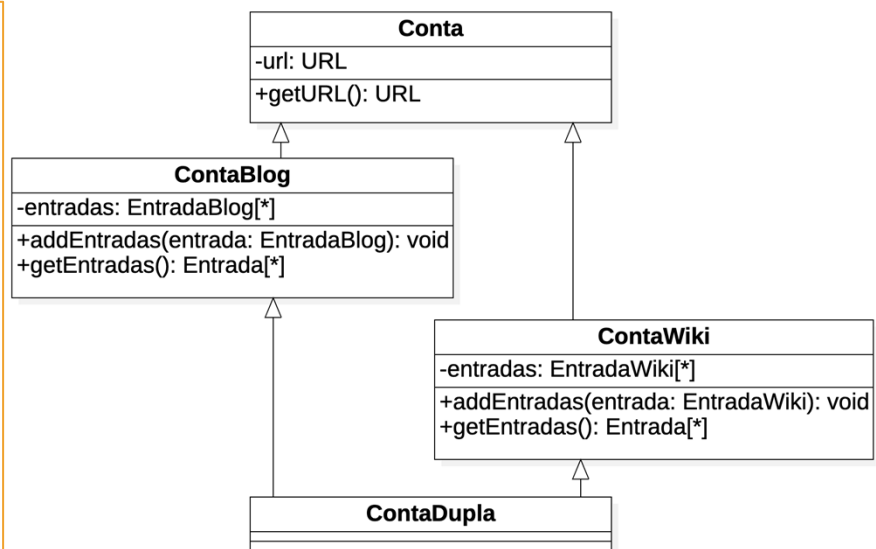
Leonor Melo

14 Diagramas de classe

15

15

## Problemas devido a Generalização múltipla?



Leonor Melo

14 Diagramas de classe

16

16



## Agregação, Composição ou Generalização ?

- Uma conta do banco tem um histórico de transações
- O animal de estimação tem um dono
- O animal de estimação tem uma cauda
- O animal de estimação é um cão
- O veículo tem 4 rodas (numa oficina de reparações); (num sistema aluguer carros)

Leonor Melo

14 Diagramas de classe

17

17

## Constraints (restrições)

- Condição ou restrição num elemento (classe ou associação) UML
- Usado sobretudo em diagramas de classes
- Indicam:
  - Uma condição específica que nunca se pode verificar
  - Que o valor de um atributo se baseia noutro
  - Que uma operação nunca pode deixar a classe num estado irregular
- Podem aplicar-se
  - a apenas um elemento
  - a mais de um elemento

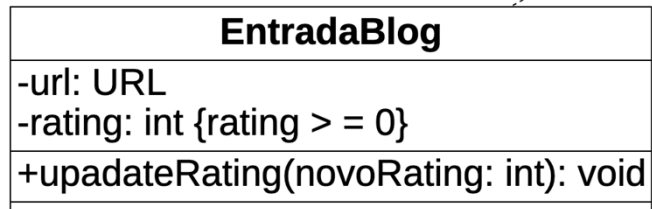
Leonor Melo

14 Diagramas de classe

18

18

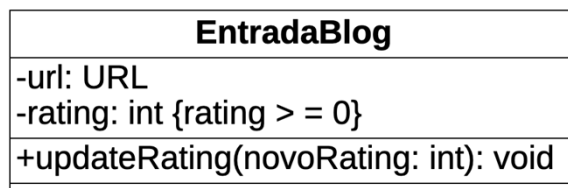
## Constraints invariantes



`self.url->notEmpty()`

- Invariante
  - Tem de ser sempre verdadeira
  - Aplica-se aos atributos da classe

## Constraints como pré-condições

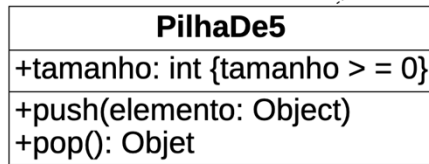


`context EntradaBlog::updateRating(novoRating: int):void`  
`pre: rating >= 0`

- Pré-condição
  - Verificada antes do método ser executado
  - Aplica-se aos métodos da classe

## Constraints como pós-condições

push(){post condition: novo tamanho = velho tamanho + 1 }



```

pop(){
  post condition: novo tamanho = velho tamanho -1
}
  
```

- Pós-condição
  - Verificada depois do método ser executado
  - Aplica-se aos métodos da classe

Leonor Melo

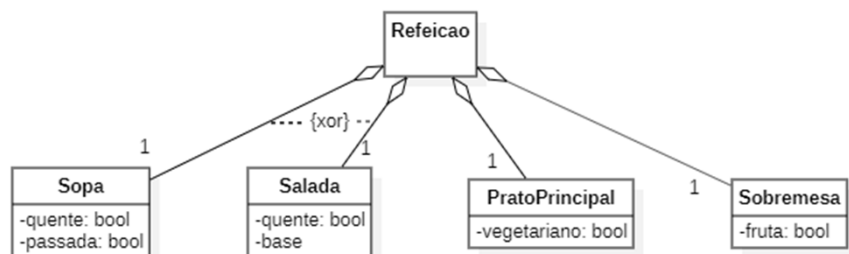
14 Diagramas de classe

21

21

## Constraints sobre agregações

- Uma refeição é composta por
  - sopa ou salada
  - prato principal
  - sobremesa



Leonor Melo

14 Diagramas de classe

22

22