

Programação I

Associação, Agregação e Composição

Samuel da Silva Feitosa

Aula 8

Introdução

- **Associação, agregação e composição** são conceitos de extrema importância dentro da **Programação e Modelagem** Orientada a Objetos.
- Na POO, um software é construído com inspiração em conceitos do mundo real, os quais podem ser relacionados.
 - Exemplo: Empresa e Funcionários.
- Classes e objetos são relacionados através de associações, agregações e composições.

Associação

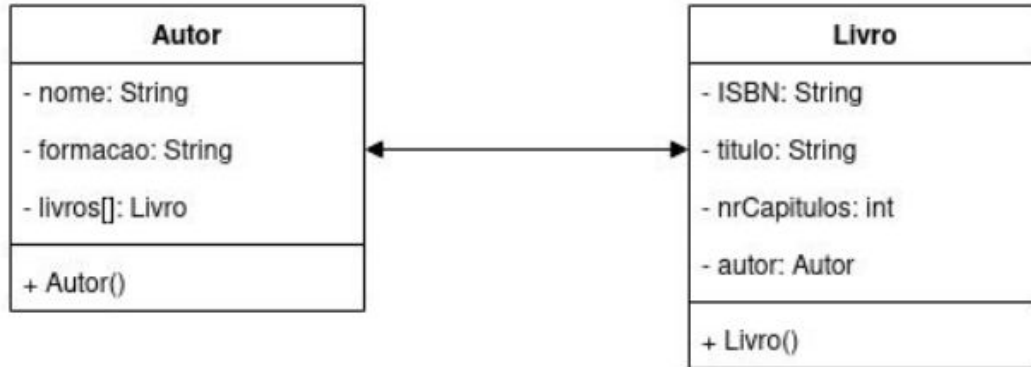
- É um **relacionamento** simples entre classes.
 - Por exemplo: **Autor** e **Livro**.
- Muitas das vezes é implementado de forma que uma classe torna-se **atributo** de outra classe.
 - A classe Autor pode ter um Livro (ou uma lista de livros) como atributo.
 - A classe Livro pode ter um Autor (ou uma lista de autores) como atributo.
- Outros exemplos:
 - Animal de estimação e seu dono, Empresa e seu endereço, etc.

Associação - Exemplo

- Classes Autor e Livro e seu diagrama UML.

```
public class Autor {  
    String nome;  
    String formacao;  
    Livro livros[];  
}
```

```
public class Livro {  
    String ISBN;  
    String titulo;  
    int nrCapitulos;  
    Autor autor;  
}
```



Agregação

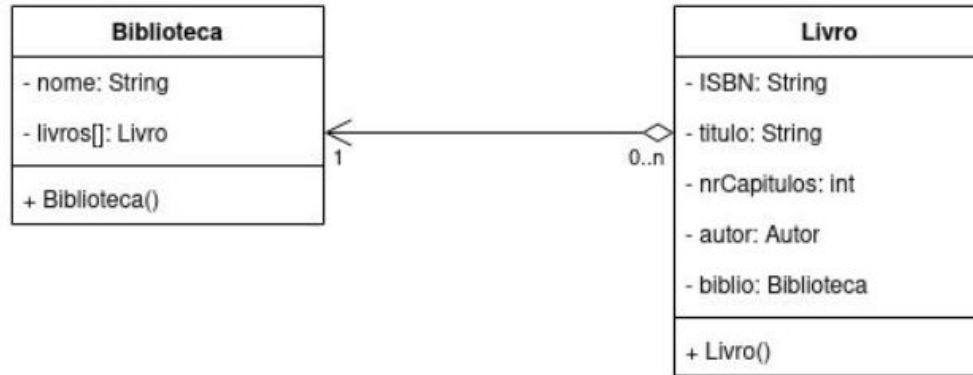
- Agregação é um tipo especial de associação onde o **elemento associado corresponde a uma parte** do elemento principal.
- As classes devem ter a característica de “**todo-parte**”, e a “parte” **pode estar contida em outros** elementos do “todo”.
 - Exemplo: Biblioteca e Livro.
- **Importante:** o conceito de Livro **pode existir sem** o conceito de Biblioteca, ou o mesmo Livro **pode estar** em diferentes bibliotecas.

Agregação - Exemplo

- Classes Biblioteca e Livro e seu diagrama UML.

```
public class Biblioteca {  
    String nome;  
    Livro livros[];  
}
```

```
public class Livro {  
    String ISBN;  
    String titulo;  
    int nrCapitulos;  
    Autor autor;  
    Biblioteca biblio;  
}
```



Composição

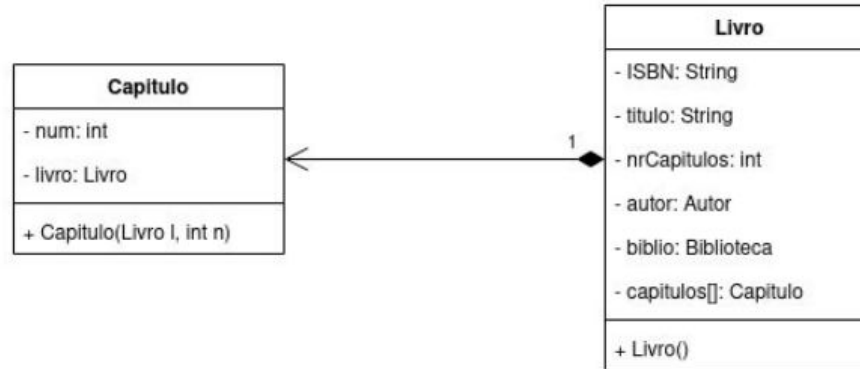
- Composição é um tipo especial de associação, onde as **“partes” pertencem somente a um único “todo”**.
- **Não faz sentido** pensar em um objeto da classe principal **sem os objetos que o compõem**.
- O **“todo” só existe** enquanto as **“partes” também existem**, e vice-versa.
 - Exemplo: Livro e Capítulo.
- Em geral, um capítulo deve estar em um livro.

Composição - Exemplo

- Classes Capítulo e Livro e seu diagrama UML.

```
public class Capítulo {  
    int num;  
    Livro livro;  
  
    Capítulo(Livro livro, int num) {  
        this.num = num;  
        this.livro = livro;  
    }  
}
```

```
public class Livro {  
    String ISBN;  
    String titulo;  
    int nrCapitulos;  
    Autor autor;  
    Biblioteca biblio;  
    Capítulo capitulos[];  
}
```

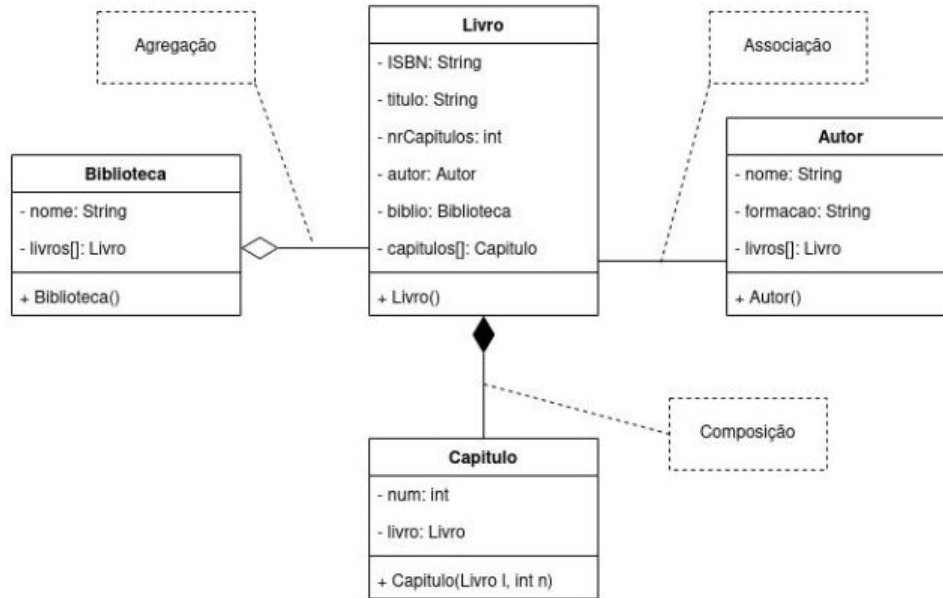


Na prática

- Percebam que o código usado para representar as associações, agregações e composições **são basicamente os mesmos**.
 - O objeto relacionado vira um atributo na classe.
- Em geral, estes relacionamentos são representados da **mesma forma** em linguagens de programação orientada a objetos.

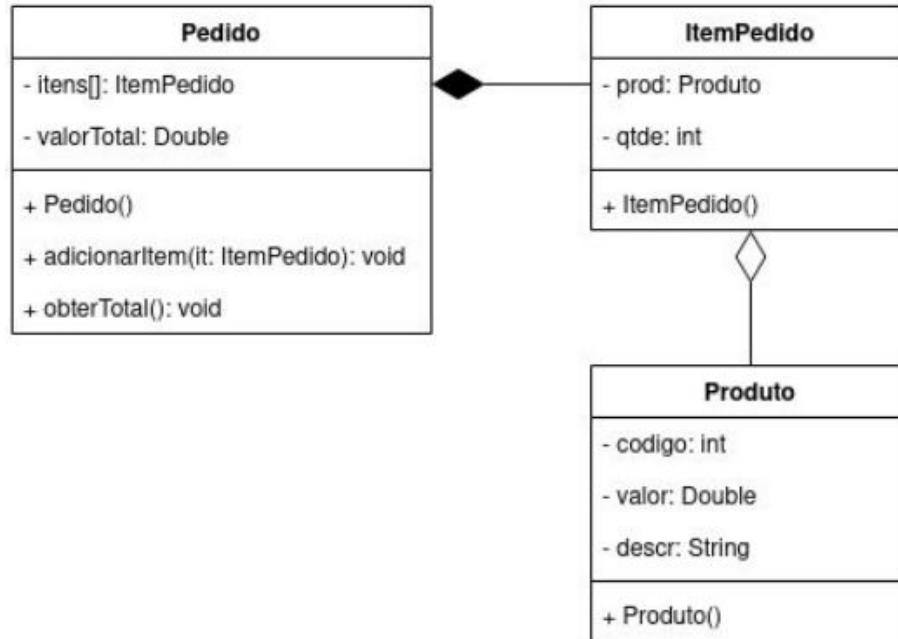
Representação UML

- Como vimos, em UML é possível representar de forma mais clara os relacionamentos.



Outro Exemplo - Pedido

- Diagrama UML para representar um sistema com Produtos e Pedidos.



Considerações Finais

- Conceitos importantes a respeito do relacionamento entre diferentes classes e objetos.
- Associação, agregação e composição são implementados de forma similar em linguagens de programação orientadas a objetos.
- É possível identificar estes relacionamentos através de diagramas UML.