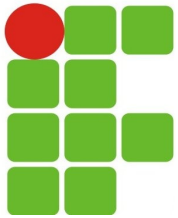


Relações entre objetos
(Agregação, Composição e Associação)

João Paulo Q. dos Santos
joao.queiroz@ifrn.edu.br

Relações

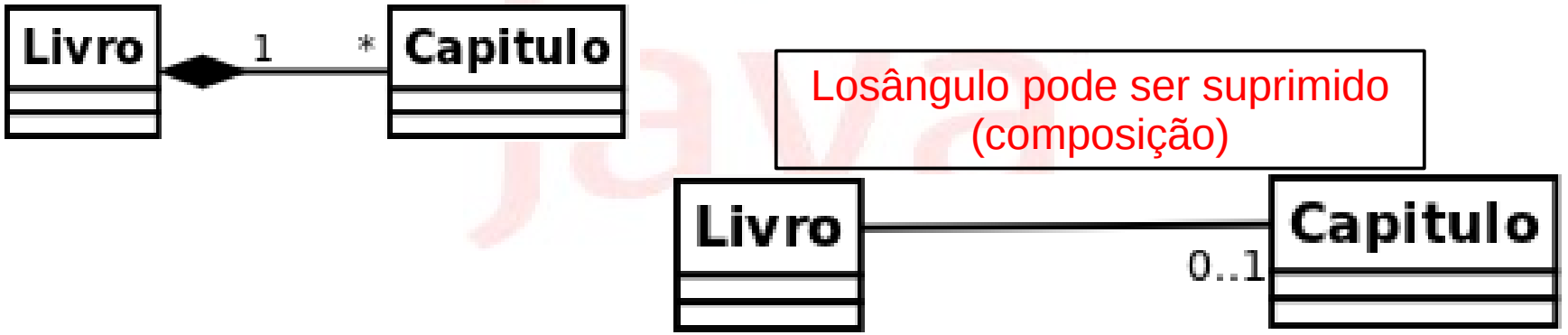
- Objetos não existem isolados:
  - São formados por outros objetos;
  - Objetos usam outros objetos;
  - Um programa OO possui vários objetos que interagem entre si;
  - Modelagem define quais objetos usamos em um programa.

Relações entre objetos

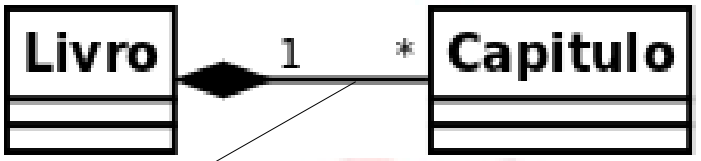
- Objetos possuem relacionamentos
  - Composição
    - Um objeto pode ser formado por outros objetos
      - Ex.: Casa, livro, jardim, agenda de contatos, etc.
  - Agregação
    - Um objeto pode conter outros objetos
      - Ex.: Carro (motor, pneu, porta)
  - Associação
    - Objetos podem usar outros objetos
      - Ex.: Trem usa estrada de ferro

Composição

- Um livro é composto de capítulos
  - Capítulo é parte essencial de livro
    - Se não existir capítulo, não existe livro
    - Capítulo não existe fora de livro
- Linha com losângulo preenchido na classe "dominante"
  - Livro é composto de 1 ou mais capítulos



Composição

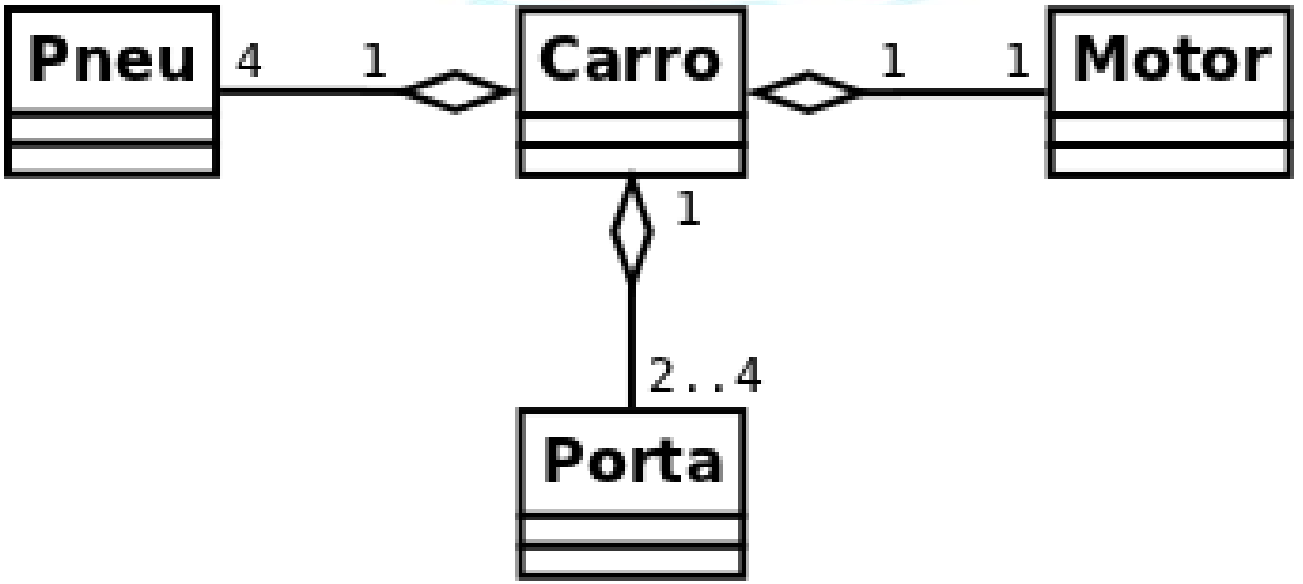



```java
public class Livro {

    private Capitulo[] capitulos;

    public Livro(int qtdCapitulo){

    capitulos = new Capitulo[qtdCapitulo];

    }
}
```

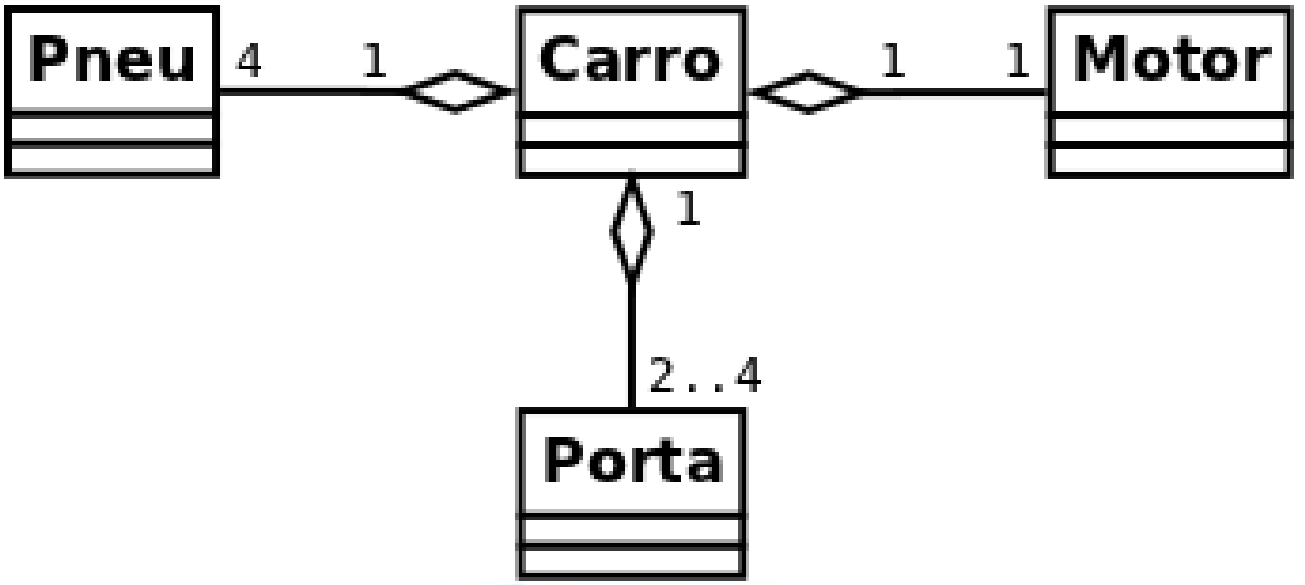
```java
public class Capitulo {
/* Definição da classe Capítulo */
}
```

Agregração

- Carro possui Pneu, Motor e Porta
  - Não são partes essenciais do carro
    - Retirando as portas um carro continua sendo um carro
    - Pneus/portas existem como objetos independentes
- Linha com losângulo vazio na classe “dominante”



Agregação




```
public class Carro {

    private Motor motor;
    private Porta portas[];
    private Pneu pneus[];
    /* ... */
}
```

```
public class Motor {
/* Definição da classe
Motor */
}
```

```
public class Porta {
/* Definição da classe
Porta */
}
```

```
public class Pneu {
/* Definição da classe
Pneu */
}
```

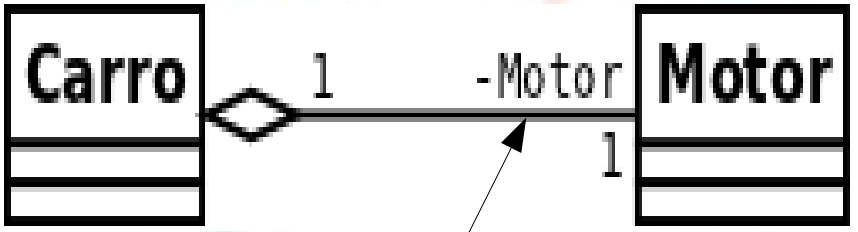
Associação

- Objetos que usam outros objetos
  - Podem ser implementados como atributos


```
public class Trem {

    private EstradaDeFerro estradaDeFerro;
    /* ... */
    public void definirEstrada(EstradaDeFerro estradaDeFerro){
        this.estradaDeFerro = estradaDeFerro;
    }
}
```

Nomes para atributos

- Pode-se especificar o nome do atributo
  - Obrigar existência do atributo
  - Carro tem um atributo privado motor do tipo Motor




```
public class Carro {
    private Motor motor;
    /* Definição da classe Motor */
}
```

Nomes para atributos

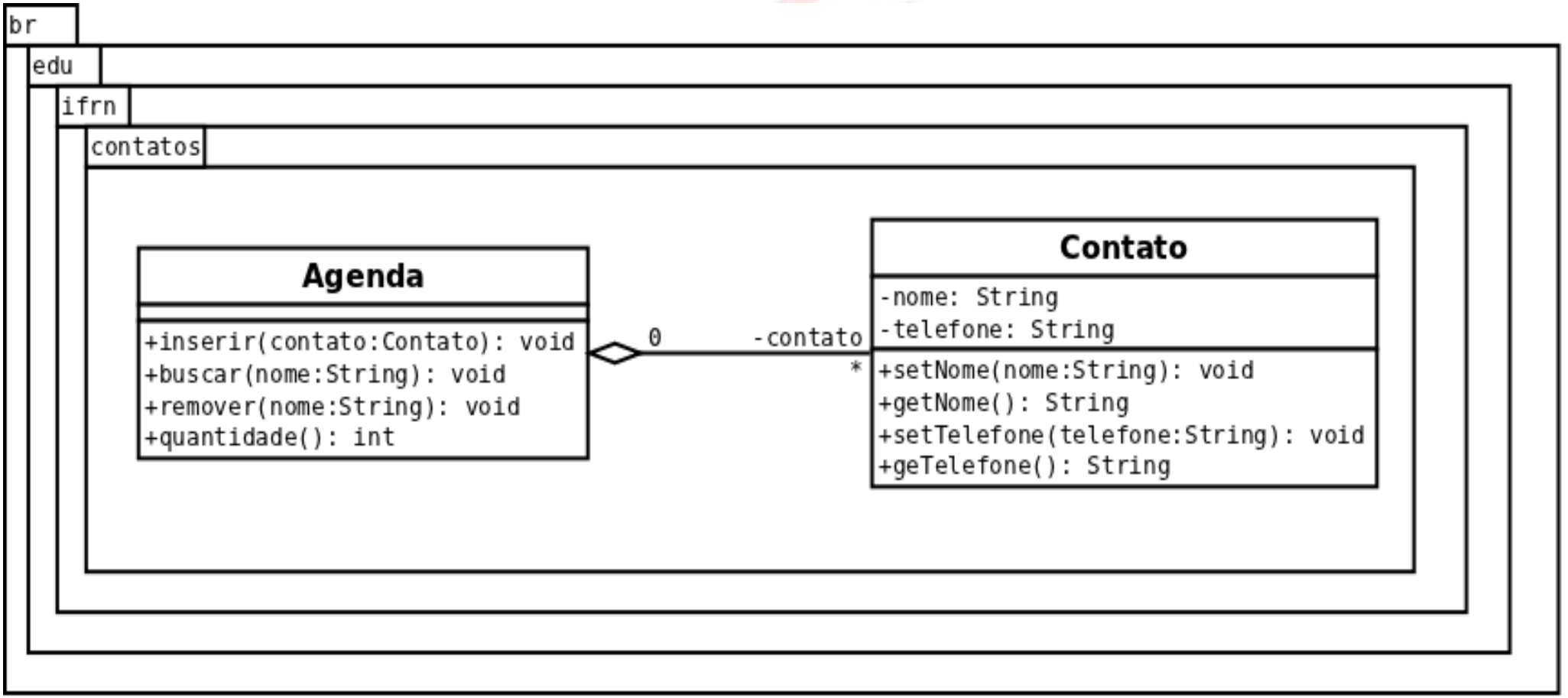
- Coleções
  - Atributos com multiplicidade * podem ser implementados por um array


```
public class Livro {

    private Capitulo[] capitulos;

    public Livro(int qtdCapitulo){

        capitulos = new Capitulo[qtdCapitulo];
    }
}
```

```
public class Capitulo {
/* Definição da classe Capítulo */
}
```

Exemplo

- Modelo de objetos de uma agenda de contatos



Dúvidas