

Dependência, Agregação e Composição

Profa. Karen Selbach Borges



Dependência



- É o relacionamento que ocorre com maior frequência dentro de um código Java.
- Dentro do pacote `java.lang` existem muitas classes que são frequentemente utilizadas nos códigos que criamos como, por exemplo:
 - `Object`
 - `String`
 - `Math`
 - `Enum`
 - Wrapper classes: `Integer`, `Long`, `Double`, `Float`, `Byte`, `Short`, `Boolean`, `Char`
 - `Exception`
 - `Error`

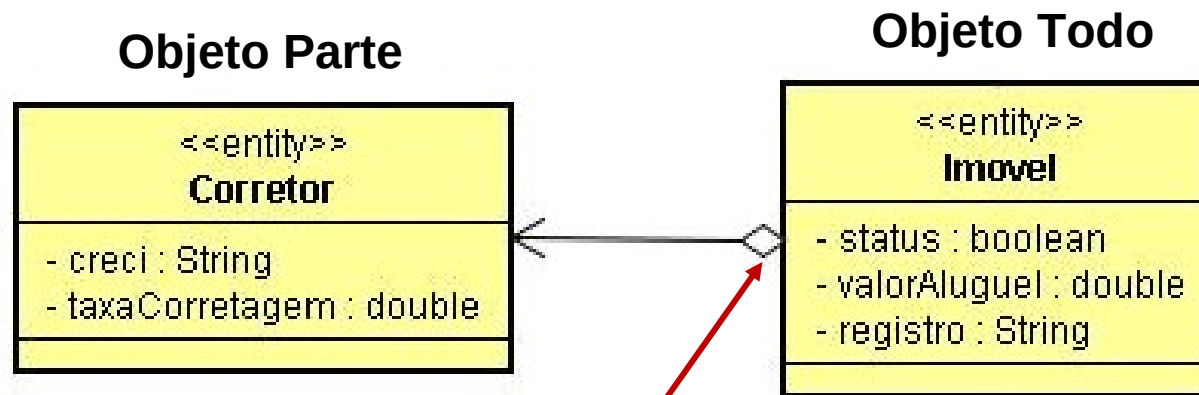


Agregação



Em um relacionamento de agregação os objetos podem existir independentemente uns dos outros.

Isso quer dizer que, se uma das partes for destruída, a outra continuará a existir.



Representação UML

Agregação



A agregação, na implementação Java, se caracteriza por:

- declaração de um objeto de outra classe (parte) como variável de instância do todo.
- a parte não compõe o conjunto de parâmetros do construtor do todo
- existe um método que adiciona a parte ao todo (por exemplo, um setter)



Exemplo em Java



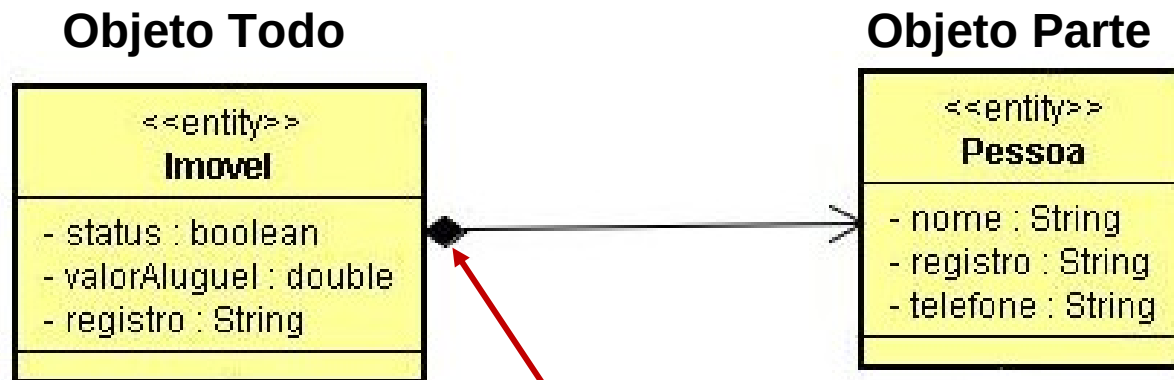
```
public class Imovel {  
    private Corretor corretor;  
    private String registro;  
    private double valorAluguel;  
    private boolean status;  
  
    public Imovel(String registro, double valorAluguel)  
    {  
        this.registro = registro;  
        this.valorAluguel = valorAluguel;  
    }  
  
    public void setCorretor(Corretor corretor) {  
        this.corretor = corretor;  
    }  
}
```



Composição



Em um relacionamento de composição os objetos não podem existir independentemente uns dos outros. Isso quer dizer que, se uma das partes for destruída, a outra será destruída também.



Representação UML



Composição



A composição, na implementação Java, se caracteriza por:

- declaração de um objeto de outra classe (parte) como variável de instância do todo.
- a parte compõe o conjunto de parâmetros do construtor do todo



Exemplo em Java



```
public class Imovel {  
  
    private Pessoa proprietario;  
    private String registro;  
    private double valorAluguel;  
    private boolean status;  
  
    public Imovel(String registro, Pessoa proprietario, double  
valorAluguel)  
    {  
        this.registro = registro;  
        this. proprietario = proprietario;  
        this.valorAluguel = valorAluguel;  
    }  
}
```

