



Relacionamentos

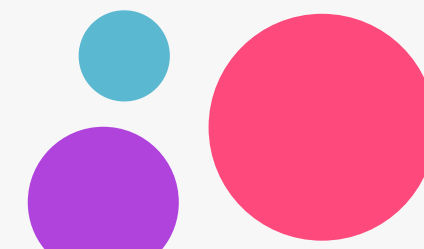
Aula 08 - Programação orientada a objetos



Professor Daniel Marques



INSTITUTO FEDERAL
Paraíba
Campus Campina Grande



1

Introdução



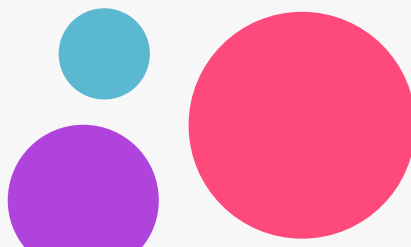
Unified Modeling Language (UML)

É a representação gráfica mais utilizada para modelagem de sistemas orientados a objetos



Unified
Modeling
Language

- Adotado como padrão internacional em 1997
- Define um conjunto padrão de notações gráfica



Unified Modeling Language (UML)

A versão 2.0 da UML oferece padrões de diagramas estruturais e comportamentais



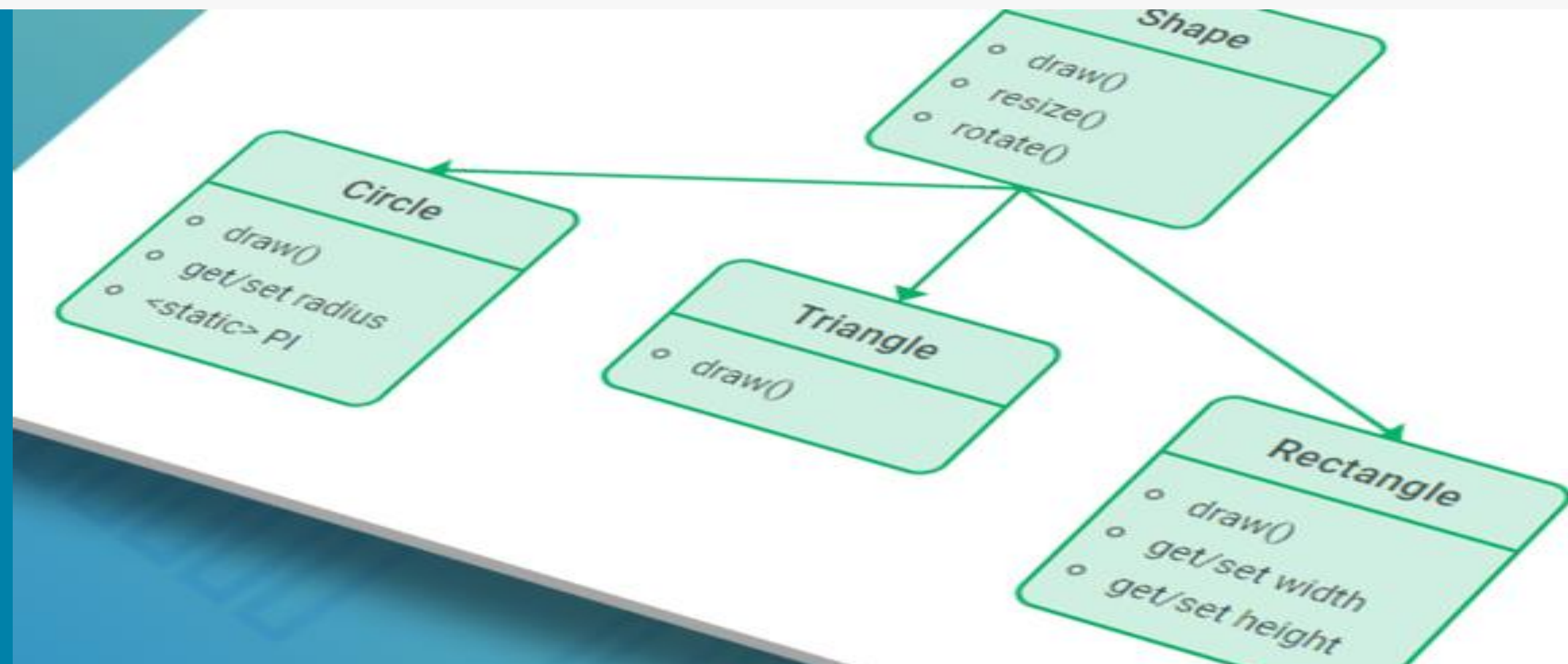
- Estamos interessados nos diagramas estruturais
- Especificamente nos Diagramas de Classes



Unified Modeling Language (UML)

Um diagrama de classes descreve a estrutura do sistema

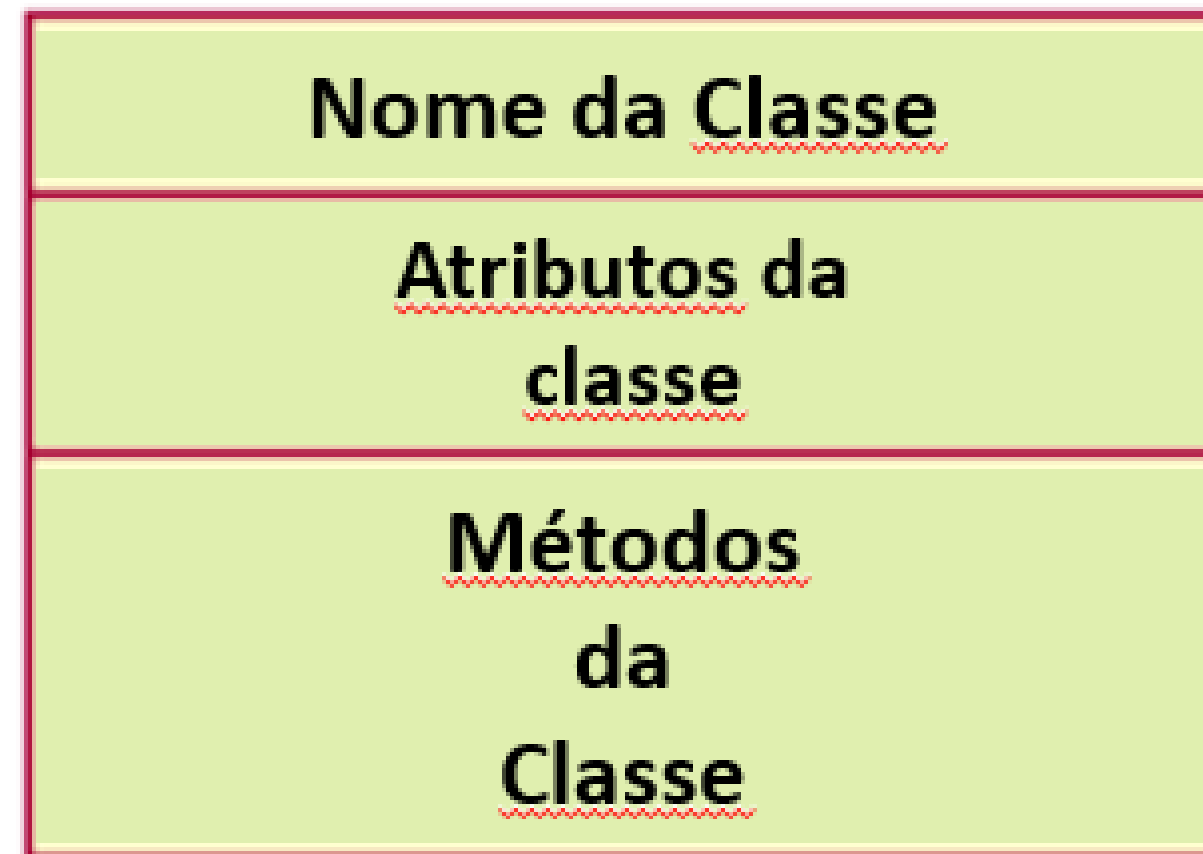
- Classes
- Atributos
- Métodos
- Relação entre classes



O elemento fundamental dos diagramas de classes é o componente gráfico que representa uma classe

Classe: notação UML

Uma classe em UML é representada como a figura ao lado



Obs: dependendo do software utilizado para modelagem UML, cores, formatos e opções mais detalhadas podem surgir.

Classe: notação UML

Visibilidade:

para especificar a visibilidade de um membro de uma classe (atributo ou método) são usados as seguintes notações

+ Público

- Privado

Protegido

Circulo
-raio:double -centro:Ponto
+area():double +circunferencia():double +setCentro(Ponto):void +setRaio(double):void



Software para modelar UML

Astah UML



Link para a versão gratuita de estudante

<https://astah.net/products/free-student-license/>



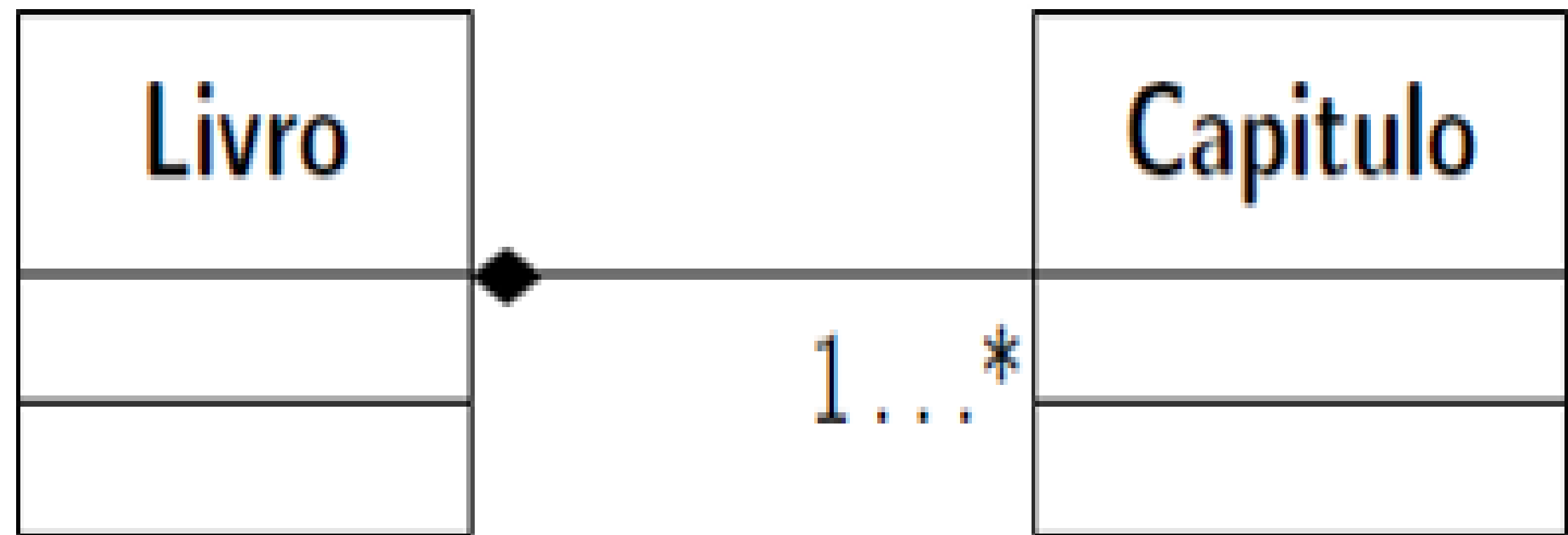
INSTITUTO FEDERAL
Paraíba
Campus Campina Grande

2

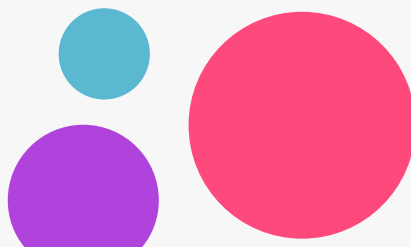
Diagrama de classe

Diagrama de classe (UML)

Em alguns diagramas, os dois últimos compartimentos são omitidos



- não são apresentados todos os atributos ou métodos
- apenas aqueles que são importantes para a finalidade do diagrama





Relacionamento entre classes

Além de descrever classes, a UML pode ser usada para descrever relacionamentos entre classes, tais como:

Esses relacionamentos são descritos por linhas conectando classes

- **Agregação / Associação**
- **Composição**
- **Herança**
- **Interface**





Relacionamento entre classes

Tipos de associações entre objetos de software



Simples

quando a classe possui atributos do tipo de outra classe.



Agregação

estabelecem um vínculo entre objetos.



Composição

relacionamento do tipo todo/parte com dependência de existência





Associação Simples

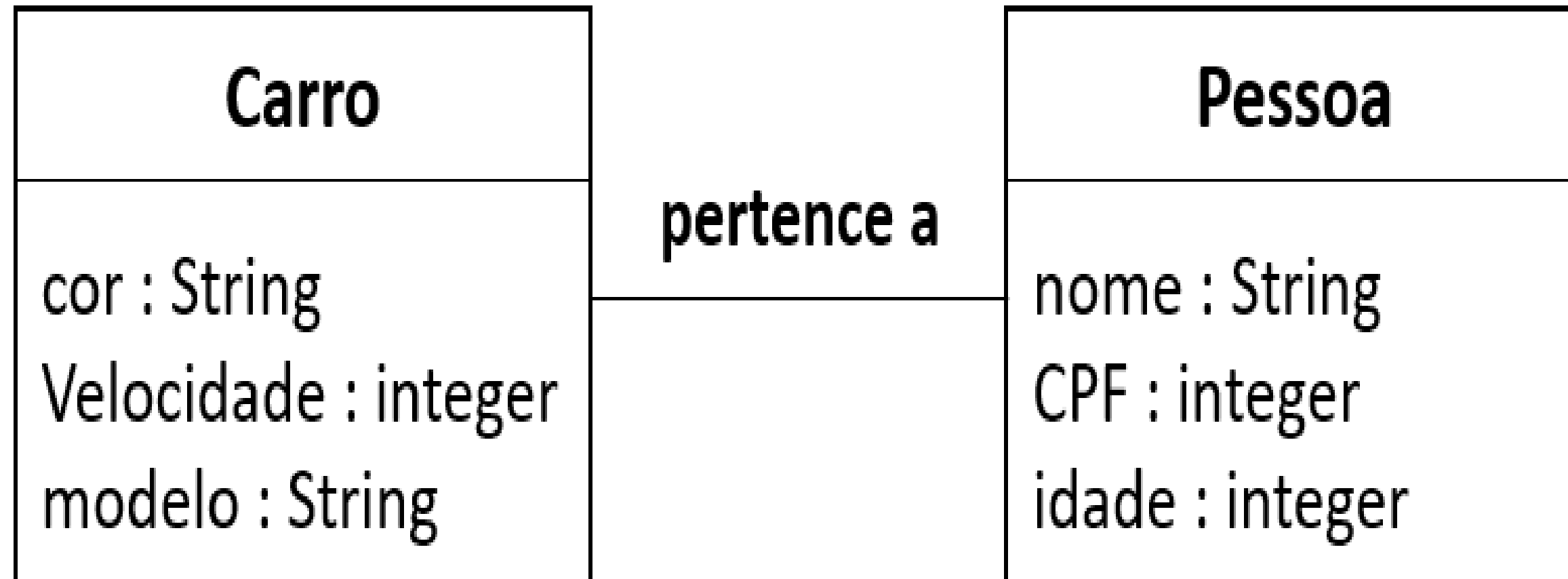
Ocorre quando uma classe possui um ou mais atributos do tipo de uma outra classe

```
class Carro {  
    Pessoa dono;  
    string modelo;  
}
```

```
class Pessoa {  
    string nome;  
    int idade;  
}
```

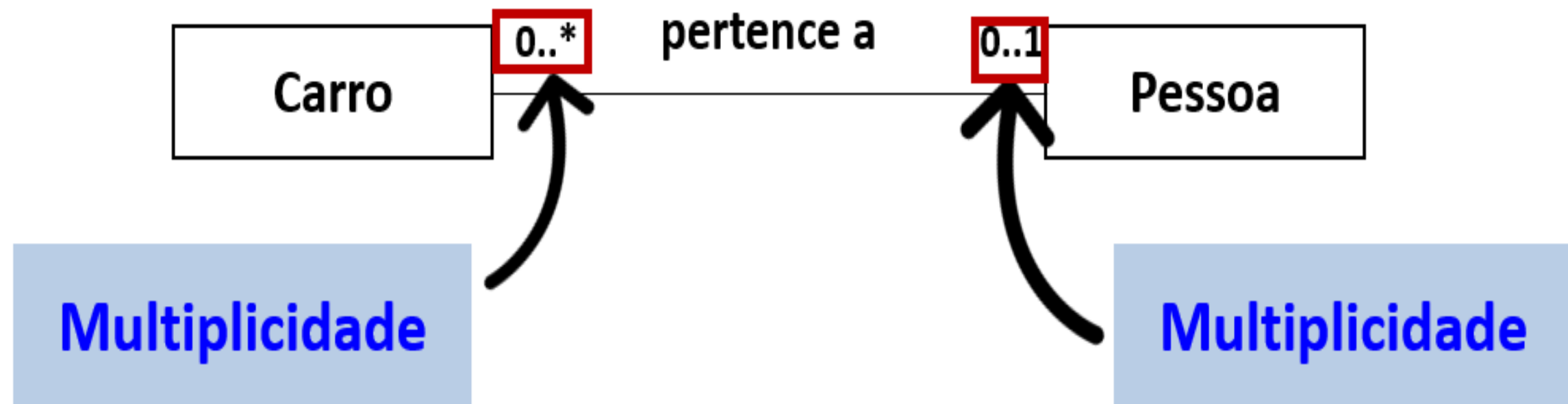
Associação Simples

Modela uma conexão
semântica entre elementos
UML



Associação Simples

Cada extremidade da uma linha define um relacionamento entre classe pode possuir um valor de multiplicidade





Associação Simples

- Pode ser um valor fixo: 1
- Pode ser um intervalo: [0...3]
- O * significa "vários"

0 ... 1	No máximo um
0 ... *	zero ou muitos, pode haver vários objetos envolvidos no relacionamento
1 ... *	Um ou muitos, pelo menos um objetos está envolvido



Associação por Agregação

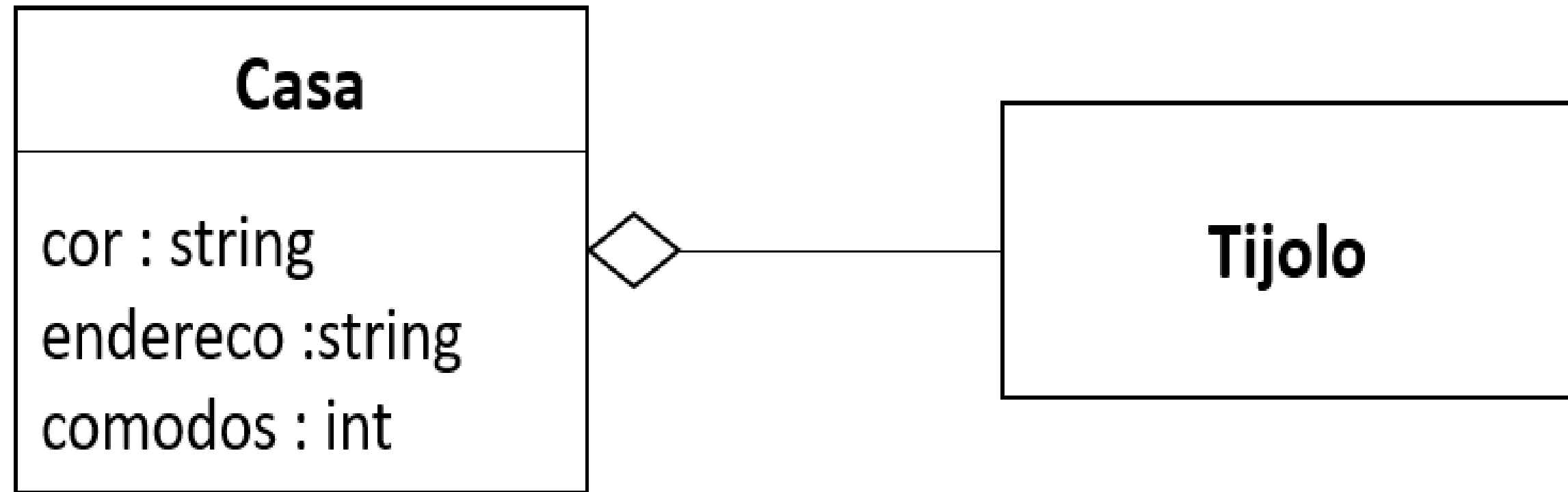
Ocorre quando uma classe (todo) pode possuir atributos do tipo de uma outra classe (partes), porém as classes são independentes uma das outras

```
class Carro {  
    Passageiro p[7];  
    string modelo;  
}
```

```
class Passageiro {  
    String nome;  
    int idade;  
}
```

Associação por Agregação

Modela um relacionamento
todo-parte entre elementos
UML



- Observação: A associação por agregação é simbolizada pelo losango branco.
- A classe que possui o losango branco “colado” é o todo (Casa), a parte é representada pela classe Tijolo



Associação por Composição

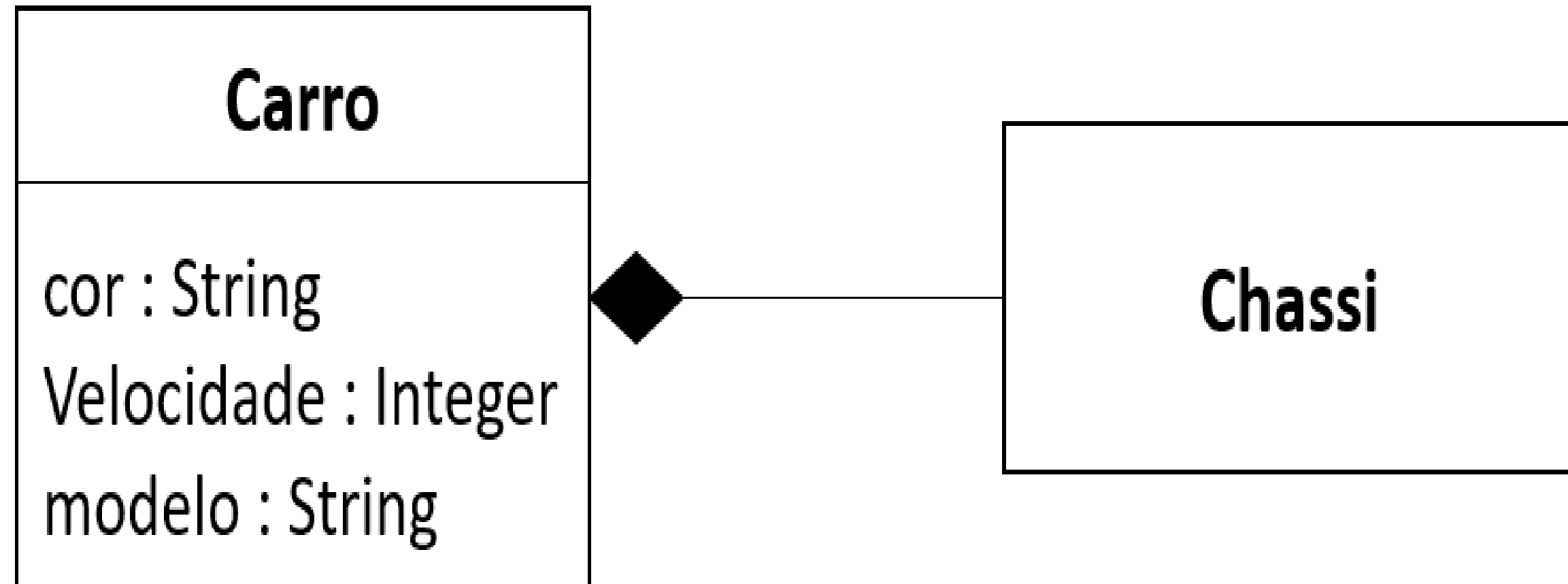
Ocorre quando uma classe (todo) deve possuir atributos do tipo de uma outra classe (partes), e a existência das partes depende da existência do todo.

```
class Carro {  
    string modelo;  
}
```

```
class DocCarro {  
    int numRegistro;  
    string localRegistro;  
    Carro automovel;  
}
```

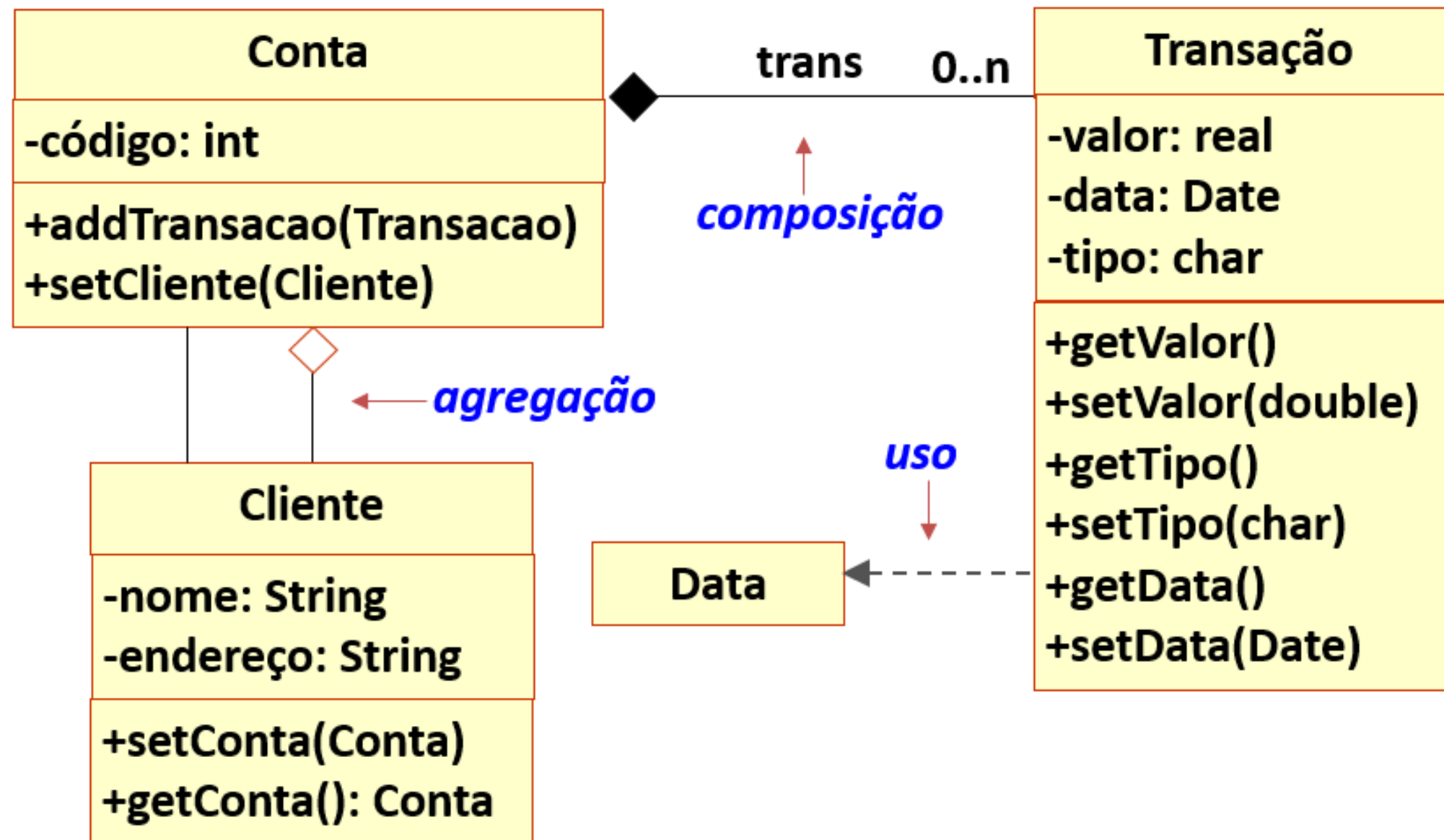
Associação por composição

Modela um relacionamento todo-parte, em que a parte não existe sem o todo.



Relacionamento entre classes e objetos

Notação UML para definir composições, agregações e uso:



Relacionamento entre classes e objetos

Trecho da implementação

```
class Conta {  
    private:  
        int codigo;  
        Cliente cliente;  
        Transacao trans[MAX];  
        int qTransacoes;  
    //...  
}
```

```
class Transacao {  
    private:  
        double valor;  
        char tipo;  
        Data data;  
    //...  
}
```

```
class Cliente {  
    private:  
        string nome;  
        Conta conta;  
        string endereço;  
    //...  
}
```

```
main() {  
    Conta cc5 = Conta();  
    Cliente cliente = Cliente();  
    Transacao credito = Transacao();  
  
    cc5.setCodigo(100);  
    cc5.setCliente(cliente);  
    cliente.setConta(cc5);  
    credito.setValor(100000000.0);  
    credito.setTipo('C');  
    credito.setData(Data());  
    cc5.addTransacao(credito);  
}
```

Concretiza a **agregação**
entre cliente e conta

Concretiza a **composição**
entre transação e conta

Relacionamento entre classes e objetos

Trecho da implementação na
aplicação principal



Alguma dúvida?

Não guardem dúvidas, perguntem

...



Referências

- 1 DA COSTA, Anderson Fabiano F. **Fundamentos de C++**. Instituto Federal da Paraíba. 2022.
- 2 Materiais de aula dos professores Guillermo Camara-Chavez, Tiago Maritan, Fred Guedes Pereira e Danielle Chaves.
- 3 DEITEL, **C++ Como Programar**, 5ª edição, Editora Prentice Hall, 2006
- 4
- 5

