Programação Orientada a Objetos I

CÁSSIO CAPUCHO PEÇANHA – 08

Implementação de relações

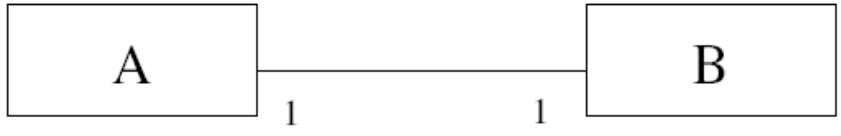
- Em OO os objetos se relacionam entre si por meio da implementação de relações, tais como:
  - Associação
  - Agregação
  - Composição
- A decisão de qual relacionamento será usado é tomada durante a análise do sistema.

Relação - Associação

- Uma associação representa que duas classes possuem uma ligação (link) entre elas, significando por exemplo que:
  - “A conhece um outro B",
  - “A tem um B”,
  - “B é usado por A”

Relação - Associação

- Relações de 1 para 1 bidirecional.




```
class A{
    private B b;

    public void setB(B aB){
        this.b = aB;
    }

    public B getB(){
        return this.b;
    }
}
```

```
class B{
    private A a;

    public void setA(A aA){
        this.a = aA;
    }

    public A getA(){
        return this.a;
    }
}
```

B “é usado por” A
e
A “é usado por” B

B “tem um” A
e
A “tem um” B

Relação - Associação

- Relações de 1 para 1 unidirecional.

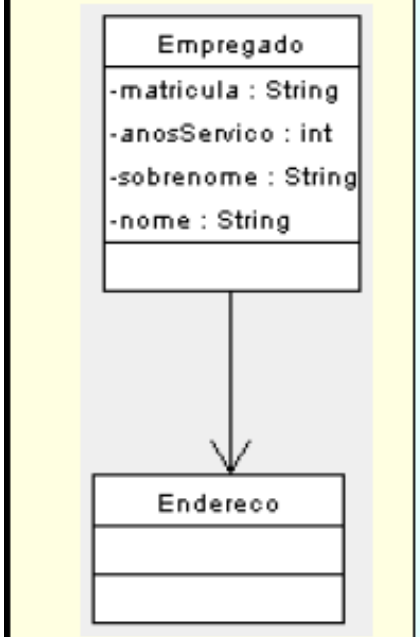
B “é usado por” A

A “tem um” B


```
class A{
    private B b;

    public void setB(B aB){
        this.b = aB;
    }

    public B getB(){
        return this.b;
    }
}
```

```
class B{

    /*...*/

}
```

Relação - Associação



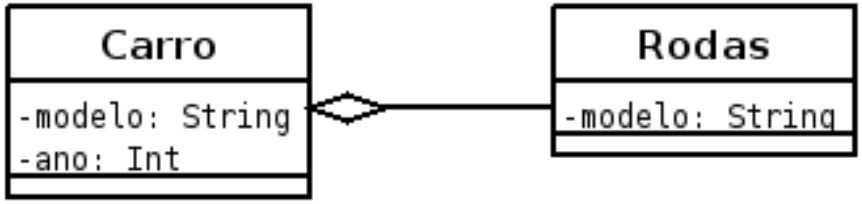

```
public class Empregado {
    private String matricula;
    private int anosServico;
    private String sobrenome;
    private String nome;
    private Endereco endereco;
  . . .
}
public class Endereco {
    . . .
}
```

Relação - Agregação

- A agregação é um caso particular da associação significando por exemplo que:
  - “B é parte de A”,
  - “A tem um B”,
  - “A é composto por B”
- A agregação estabelece uma relação **todo-parte** entre classes, sendo que **a parte pode existir sem o todo**.
- Embora as partes possam existir independentemente do todo, sua existência é basicamente para formar o todo,

Relação - Agregação

- Temos o objeto Carro que por sua vez faz referência ao objeto Rodas
- Porém o objeto Rodas pode existir mesmo que você destrua Carro



Roda faz parte de Carro.

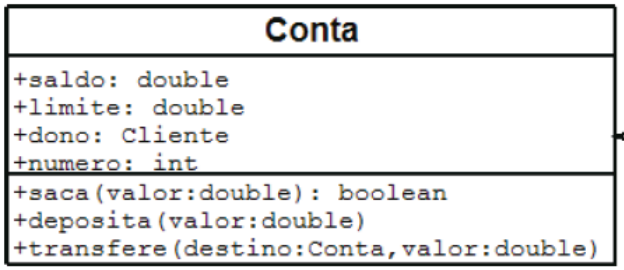
Carro tem Rodas

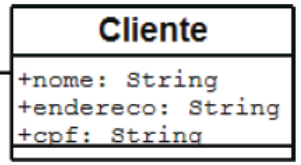
Relação - Agregação


```
public class A {
    private B b;
    public A( ){
    }
    public void setB( B b
){
        this.b = b;
    }
    public B getB( ) {
        return b;
    }
}
```

```
public class B {
    public B( ){
    }
}
```

Relação - Agregação






```
class Conta {
    int numero;
    double saldo;
    double limite;
    Cliente titular;
    // ..
}
```

```
class Cliente {
    String nome;
    String sobrenome;
    String cpf;
}
```

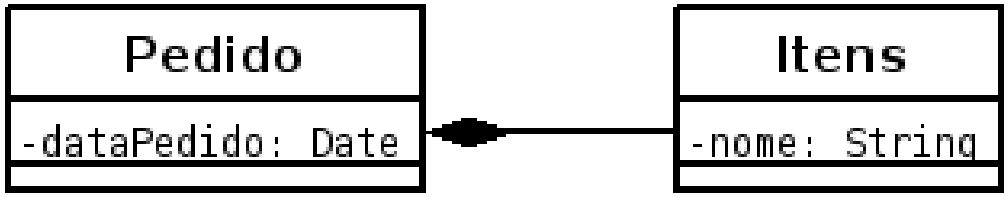
```
class Teste {
    public static void main(String[] args) {
        Conta minhaConta = new Conta();
        Cliente c = new Cliente();
        minhaConta.titular = c;
        // ...
    }
}
```

Relação - Composição

- A composição é um caso particular da agregação significando por exemplo que:
  - “B é parte essencial de A”,
  - “A tem um B”,
  - “A é composto por B”
- A composição estabelece uma relação **todo-parte** entre classes, sendo que **a parte NÃO existe sem o todo**.
- A diferença é que o todo CONTÉM as partes (e não referências para as partes). Quando o todo desaparece, todas as partes também desaparecem.
- As partes **NÃO** podem existir independentemente do todo.

Relação - Composição

- Temos o objeto Pedido que por sua vez faz referência ao objeto Itens.
- Portanto o objeto "Itens do Pedido" não faz sentido sem o objeto "Pedido".



Itens de Pedido faz parte de Pedido.

Pedido tem Itens

Relação - Composição

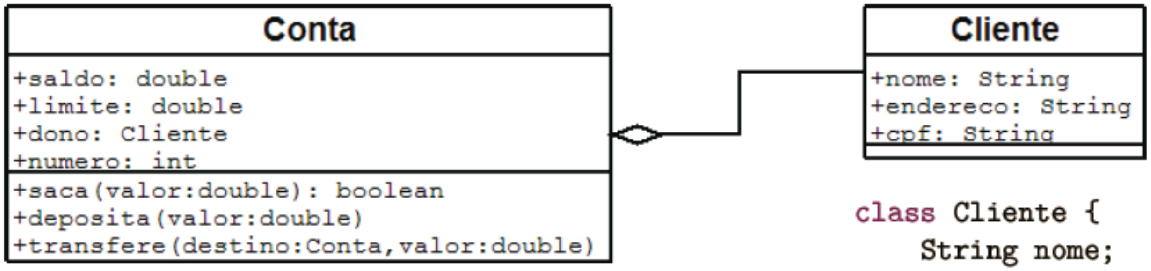

```
public class A {
    private B b;
    public A( ){
        b = new B();
    }
}
```

```
public class B {
    public B( ){
    }
}
```

B “é parte de” A

A “tem um” B

Relação - Composição

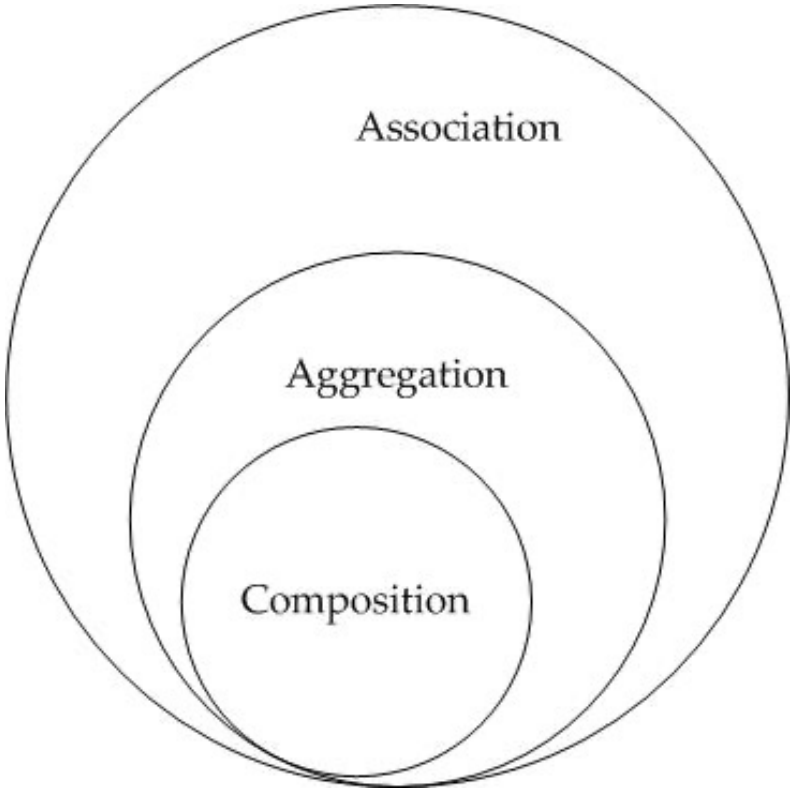



```
class Conta {
    int numero;
    double saldo;
    double limite;
    Cliente titular = new Cliente();

}
 // quando chamarem new Conta,
 //havera um new Cliente para ele.
```

```
class Cliente {
    String nome;
    String sobrenome;
    String cpf;
}
```

```
class Teste {
    public static void main(String[] args) {
        Conta minhaConta = new Conta();

        minhaConta.titular.nome = "paulo";
        // ...
    }
}
```

Composição x Agregação x Associação

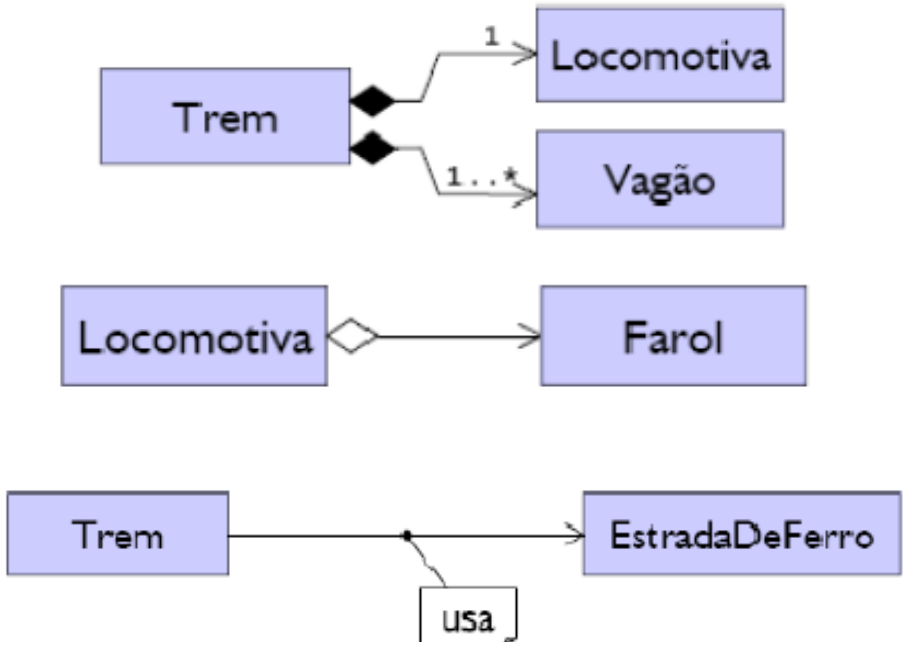


Composição x Agregação x Associação

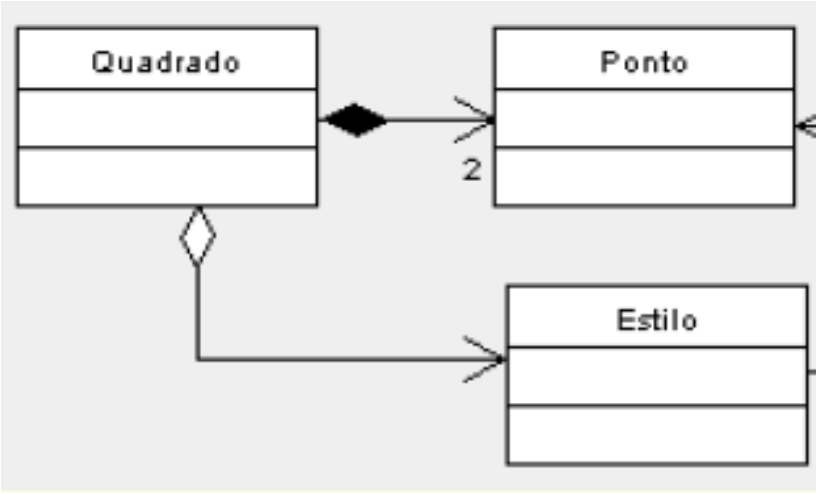
- Se NÃO HOUVER a relação todo-parte
  - ➢ASSOCIAÇÃO SIMPLES
- Se HOUVER a relação todo-parte, temos que ver se a parte pode existir sem o todo.
  - Se a parte existir sem o todo
    - ➢AGREGAÇÃO
  - Se a parte não existir sem o todo
    - ➢COMPOSIÇÃO

Composição x Agregação x Associação

- Um trem **não existe** sem a locomotiva e os vagões.
- Uma locomotiva **possui** um farol (mas não vai deixar de ser uma locomotiva se não o tiver)
- Um trem **usa** uma estrada de ferro (ela não faz parte do trem, mas ele depende dela)



Composição x Agregação x Associação




```
public class Quadrado {
   // p1 e p2 são composição - new
   // estilo é agregação - atribuição
   private Ponto p1, p2;
   private Estilo estilo;
   public Quadrado(int x1, int y1,
                   int x2, int y2, Estilo e){
         p1 = new Ponto(x1, y1);
         p2 = new Ponto (x2, y2);
         estilo = e;
   }
}
```