Programação Orientada a Objetos

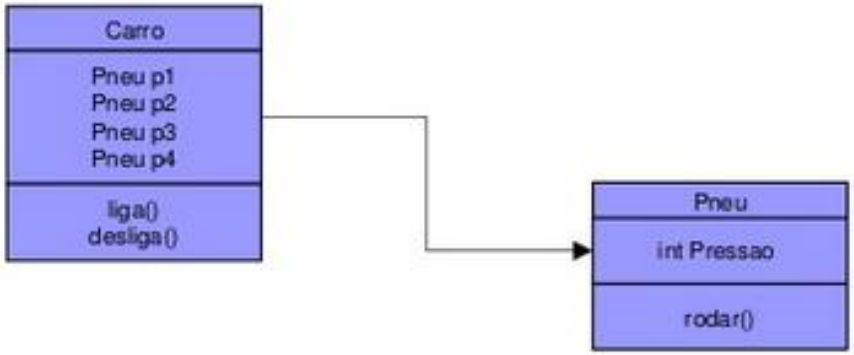
RELACIONAMENTO ENTRE CLASSES: ASSOCIAÇÃO

AGREGAÇÃO SIMPLES e AGREGAÇÃO COMPOSTA (COMPOSIÇÃO)

RELACIONAMENTO ENTRE CLASSES

Associação

- Associação ocorre quando uma classe possui atributos do tipo de outra classe.



Nota : Neste caso estamos dizendo que carro possui pneu (4 pneus)

FONTE: https://pt.slideshare.net/armandodaniel777/java-orientao-a-objetos-associacao-composicao-agregacao

RELACIONAMENTO ENTRE CLASSES

Associação

- A associação pode ser representada em Java da seguinte forma:

```
public class Pneu {
        int Pressao;

        void roda() {
                System.out.println("Pneu em movimento");
        }
}
```

```
public class Carro {
        Pneu p1;
        Pneu p2;
        Pneu p3;
        Pneu p4;

        void liga() {
                System.out.println("Carro ligado");
        }

        void desliga() {
                System.out.println("Carro desligado");
        }
}
```

FONTE: https://pt.slideshare.net/armandodaniel777/java-orientao-a-objetos-associacao-composicao-agregacao

RELACIONAMENTO ENTRE CLASSES

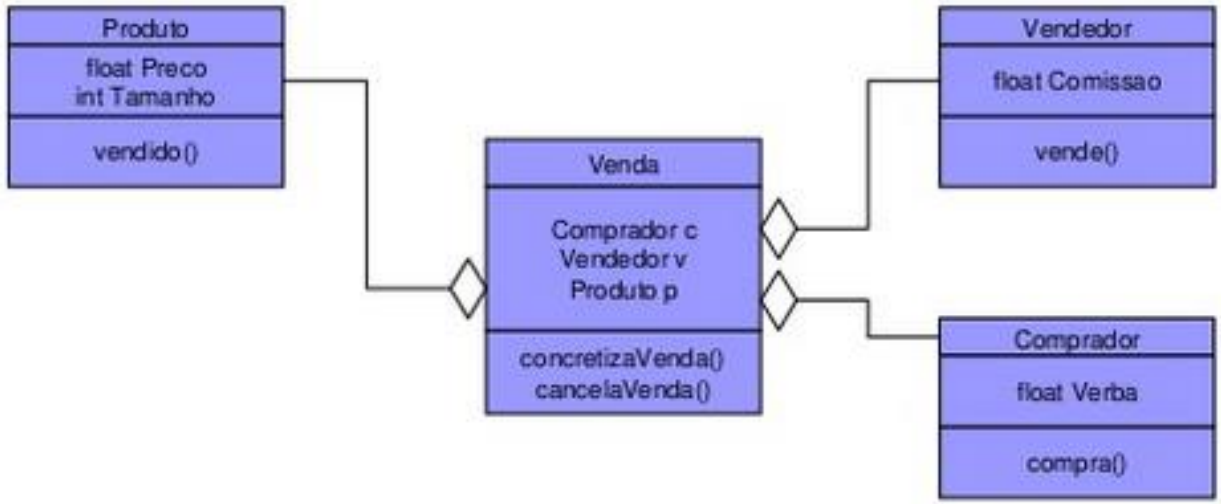
Agregação

- Ocorre quando uma classe usa outras classes em suas operações. As classes utilizadas participam da classe principal, mas a classe principal não contém estas classes utilizadas como sendo partes suas.

FONTE: https://pt.slideshare.net/armandodaniel777/java-orientao-a-objetos-associacao-composicao-agregacao

RELACIONAMENTO ENTRE CLASSES

Agregação



Nota : Neste caso **Venda** é o objeto definido como sendo o **todo.** E este objeto somente pode existir caso os demais objetos que o compõem também existam.

FONTE: https://pt.slideshare.net/armandodaniel777/java-orientao-a-objetos-associacao-composicao-agregacao

RELACIONAMENTO ENTRE CLASSES

Agregação

- A agregação pode ser representado da seguinte forma

```
public class Vendedor {
        float Comissao;

        void vende() {
                System.out.println("Vendido");
        }    }

public class Comprador {
        float Verba;

        void compra() {
                System.out.println("Comprado");
        }    }

public class Produto {
        float Preco;
        int Tamanho

        void vendido() {
                System.out.println("Vendido");
        }    }
```

```
public class Venda {
        Comprador c;
        Vendedor v;
        Produto p;

        void concretizaVenda() {
                System.out.println("Venda efetuada");
                c.Verba -= p.Preco;
                v.Comissao += p.Preco * 0.1f;
                p.vendido();
        }

        void cancelaVenda() {
                System.out.println("Venda cancelada");
        }    }
```

FONTE: https://pt.slideshare.net/armandodaniel777/java-orientao-a-objetos-associacao-composicao-agregacao

RELACIONAMENTO ENTRE CLASSES

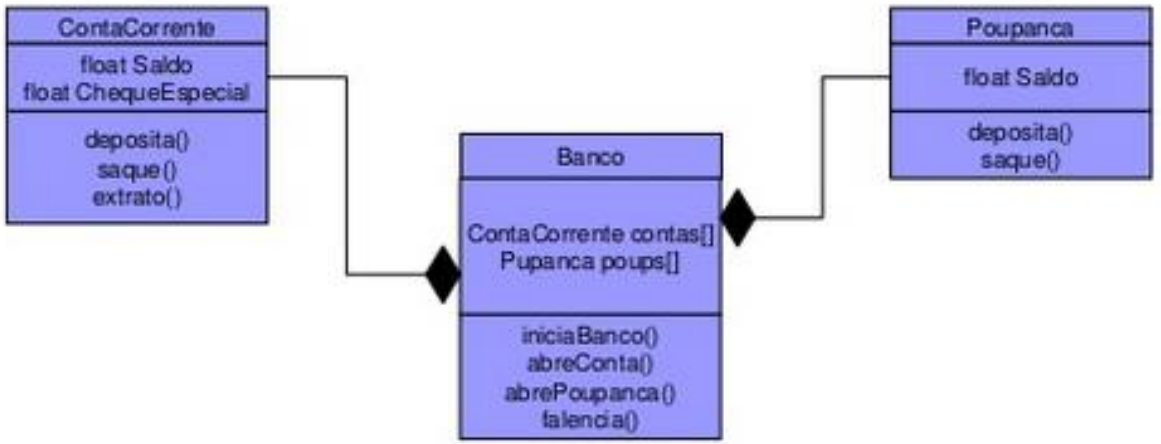
Composição

- Semelhante a agregação, a composição também é um conjunto onde há uma classe representando o **todo** e classes satélites funcionando como **partes**.
- Sua principal diferença ocorre que quando o objeto **todo** deixar de existir os seus objetos **partes** deverão deixar de existir também.

FONTE: https://pt.slideshare.net/armandodaniel777/java-orientao-a-objetos-associacao-composicao-agregacao

RELACIONAMENTO ENTRE CLASSES

Composição



Nota : No caso desta composição uma vez que o **Objeto** banco for destruído todas os objetos **Poupanca** e **ContaCorrente** deverão ser destruídos também.

FONTE: https://pt.slideshare.net/armandodaniel777/java-orientao-a-objetos-associacao-composicao-agregacao

RELACIONAMENTO ENTRE CLASSES

Composição

- A composição pode ser representado da seguinte forma:

```
public class Poupanca {
        float Saldo;

        void saque() {
                Saldo -= 10.0f;
                System.out.println("Novo Saldo →" + Saldo);
        }
        void deposito() {
                Saldo += 10.0f;
                System.out.println("Novo Saldo →" + Saldo);
        }    }
public class ContaCorrente {
        float Saldo;

        void saque() {
                Saldo -= 100.0f;
                System.out.println("Novo Saldo →" + Saldo);
        }
        void saque() {
                Saldo -= 100.0f;
                System.out.println("Novo Saldo →" + Saldo);
        }    }
```

```
public class Banco {
        Poupanca[] pops;
        ContaCorrente[] cc;
        int numConta, numPoupanca;
        void iniciaBanco() {
                pops = new Poupanca[100];
                cc = new ContaCorrente[100];
                numConta = 1;
                numPoupanca = 1;
        }
        void abreConta() {
                cc[ numConta ] = new ContaCorrente();
                numConta++;
        }
        void abrePoupanca() {
                pops[ numConta ] = new Poupanca();
                numPoupanca++;
        }
        void falencia() {
                for (int i = 0; i< 100; i++) {
                        pops[ i ] = null;
                        cc[ i ] = null;
                }
        }    }
```

FONTE: https://pt.slideshare.net/armandodaniel777/java-orientao-a-objetos-associacao-composicao-agregacao