

Agregação

É um tipo especial de associação onde tenta-se demonstrar que as informações de um objeto (chamado objeto-todo) precisam ser complementadas pelas informações contidas em um ou mais objetos de outra classe (chamados objetos-parte); conhecemos como todo/parte.

Time.java

```
public class Time {
    private int codigoTime;
    private String nomeTime;
    private Atleta atleta;

    public int getCodigoTime() {
        return codigoTime;
    }
    public void setCodigoTime(int codigoTime) {
        this.codigoTime = codigoTime;
    }

    public String getNomeTime() {
        return nomeTime;
    }
    public void setNomeTime(String nomeTime) {
        this.nomeTime = nomeTime;
    }

    public Atleta getAtleta() {
        return atleta;
    }
    public void setAtleta(Atleta atleta) {
        this.atleta = atleta;
    }
}
```

Main.java

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Atleta atleta = new Atleta();  
        atleta.inserirDados();  
  
        Time time = new Time();  
        time.setCodigoTime(55);  
        time.setNomeTime("Palmeiras");  
        time.setAtleta(atleta);  
  
        System.out.println("O time é: " + time.getNomeTime());  
        System.out.println("O nome do atleta é: " +  
time.getAtleta().getNome());  
        System.out.println("Repetindo nome do atleta: " +  
atleta.getNome());  
    }  
}
```

Composição

Pode-se dizer que composição é uma variação da agregação. Uma composição tenta representar também uma relação todo - parte. No entanto, na composição o objeto-pai (todo) é responsável por criar e destruir suas partes. Em uma composição um mesmo objeto-parte não pode se associar a mais de um objeto-pai.

Produto.java

```
public class Produto {

    private int codigo;
    private double preco;
    private String descricao;

    private double frete;
    private DataFabricacao dataFrab;

    public double getFrete() {
        return frete;
    }
    public void setFrete(double frete) {
        this.frete = frete;
    }
    public DataFabricacao getDataFrab() {
        return dataFrab;
    }
    public void setDataFrab(DataFabricacao dataFrab) {
        this.dataFrab = dataFrab;
    }

    public int getCodigo() {
        return codigo;
    }
    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }
    public double getPreco() {
        return preco;
    }
    public void setPreco(double preco) {
        this.preco = preco;
    }
    public String getDescricao() {
        return descricao;
    }
}
```

```

    }
    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }

    public double calcularFrete(int km) {
        double valor = km * 1.05;
        return valor;
    }
    public void calcularFrete(char regiao) {
        switch(regiao) {
            case 'I': System.out.println("Interior");
                this.frete = 100;
                break;
            case 'M': System.out.println("Metropolitana");
                this.frete = 0;
                break;
            case 'L': System.out.println("Litoral");
                this.frete = 20;
                break;
            default: System.out.println("Informe: (I) Interior (M)
Metropolitana (L) Litoral");
                break;
        }
    }
}

```

DataFabricacao.java

```

public class DataFabricacao {
    private int dia;
    private int mes;
    private int ano;

    public int getDia() {
        return dia;
    }
    public void setDia(int dia) {
        this.dia = dia;
    }
    public int getMes() {
        return mes;
    }
    public void setMes(int mes) {

```

```

        this.mes = mes;
    }
    public int getAno() {
        return ano;
    }
    public void setAno(int ano) {
        this.ano = ano;
    }

    public String dataCompleta() {
        return this.dia + "/" + this.mes + "/" + this.ano;
    }
}

```

TesteDataFabricacao.java

```

import javax.swing.JOptionPane;
public class TesteDataFabricacao {
    public static void main(String[] args) {

        DataFabricacao minhaData = new DataFabricacao();

        minhaData.setDia(Integer.parseInt(JOptionPane.showInputDialog("Dia:")));
        minhaData.setMes(Integer.parseInt(JOptionPane.showInputDialog("Mês:")));
        minhaData.setAno(Integer.parseInt(JOptionPane.showInputDialog("Ano:")));

        System.out.println(minhaData.dataCompleta());
    }
}

```

TesteProduto.java

```
import javax.swing.JOptionPane;
public class TesteProduto {
    public static void main(String[] args) {

        DataFabricacao dtFabr = new DataFabricacao();
        dtFabr.setDia(13);
        dtFabr.setMes(8);
        dtFabr.setAno(1986);

        Produto prod = new Produto();
        prod.setDescricao(JOptionPane.showInputDialog("Descrição do
Produto:"));
        prod.setDataFrab(dtFabr);

        JOptionPane.showMessageDialog(null, "O produto: " +
prod.getDescricao() +
        " foi fabricado em " +
prod.getDataFrab().dataCompleta());
    }
}
```