



Compartilhar o seu link com: [luciorocha @ professores.utfpr.edu.br](mailto:luciorocha@professores.utfpr.edu.br)

TURMA B

Nome e link:

Alexandre Calisto: [Aula 25](#)

Gabriel Takeshi e Plínio: [Cópia de Aula 25 - POCO4A - 10/11/2022 - Exercícios propostos](#)

Fernando Rafael: [Aula 25 - POCO4A - 10/11/2022 - Exercícios propostos](#)

João Pedro de Paula: [Cópia de Aula 25 - POCO4A - 10/11/2022 - Exercícios propostos](#)

Vinicius Letroche: [Aula 25 - POCO4A - 10/11/2022 - Exercícios propostos](#)

Andrei Fernandes Zani: [Cópia de Aula 25 - POCO4A - 10/11/2022 - Exercícios propostos](#)

Julio Farias: [Cópia de Aula 25 - POCO4A - 10/11/2022 - Exercícios propostos](#)

TURMA A

Nome e link:

Lucas Prado: [Cópia de Aula 25 - POCO4A - 10/11/2022 - Exercícios propostos](#)

Matheus H. A. Pereira: [Cópia de Aula 25 - POCO4A - 10/11/2022 - Exercícios propostos](#)

MARIA EDUARDA PEDROSO : [Aula 25 - POCO4A - 10/11/2022 - Exercícios propostos](#)

Vitor Batista: [Cópia de Aula 25 - POCO4A - 10/11/2022 - Exercícios propostos](#)

Guilherme Conceição Ramalho:

[Cópia de Aula 25 - POCO4A - 10/11/2022 - Exercícios propostos](#)

Lucas Ribeiro Penna Maroja: [Aula 25 - POCO4A - 10/11/2022](#)

Gabriel Candelária: [Cópia de Aula 25 - POCO4A - 10/11/2022 - Exercícios propostos](#)

Camila Costa: [Cópia de Aula 25 - POCO4A - 10/11/2022 - Exercícios propostos](#)

Michael Pariz: [Cópia de Aula 25 - POCO4A - 10/11/2022 - Exercícios propostos](#)

Erik Noda: [Aula25](#)

Ruan Perondi Urbanjos: [Copy of Aula 25 - POCO4A - 10/11/2022 - Exercícios propostos](#)

Rodrigo Leandro Benedito:

[Cópia de Aula 25 - POCO4A - 10/11/2022 - Exercícios propostos](#)

Felipe Antonio Magro: [Cópia de Aula 25 - POCO4A - 10/11/2022 - Exercícios propostos](#)

Mariana Pedroso Naves: [Aula 25 - POO](#)

Leonardo G. Fagote: [Padrão de Projeto](#)

William Eizo Hatakeyama:

[Cópia de Aula 25 - POCO4A - 10/11/2022 - Exercícios propostos](#)

Gustavo Nunes: [Cópia de Aula 25 - POCO4A - 10/11/2022 - Exercícios propostos](#)

Gustavo Naoki Jodai Kurozawa: [Aula 25 - POCO4A - 10/11/2022](#)

1) Padrão de Projeto: Facade (Fachada).

Motivação: Simplificar a chamada de vários métodos.

```

public class Principal {

    public abstract class Facade {
        public abstract void iniciar();
    }

    public class Supermercado extends Facade {
        List <Integer> listaCaixas;
        List <Integer> listaVendas;
        List <Integer> listaDespesas;

        public void iniciar(){

            iniciarCaixas();
            iniciarVendas();
            iniciarDespesas();
        } //fim iniciar

        private void iniciarCaixas(){ this.listaCaixas = new ArrayList<>(); }
        private void iniciarVendas(){ this.listaVendas = new ArrayList<>(); }
        private void iniciarDespesas(){ this.listaDespesas = new ArrayList<>(); }

    } //fim classe

    public Principal(){

        Facade supermercado = new Supermercado();
        supermercado.iniciar();
        Facade comercioEletronico = new ComercioEletronico();
        comercioEletronico.iniciar();

    }

    public static void main(String [] args){
        new Principal();

    }
}

```

- 2) (Online) Exercício: Acesse o link da atividade (Aula25prog1):
<https://codeboard.io/projects/357605>

3) Padrão de Projeto: Delegação.

Motivação: o objeto poderá mudar de classe com o uso de polimorfismo, ou seja, o objeto da subclasse é tratado como um objeto de sua superclasse através de sobrescrita de métodos. Delegação é implementada pela composição de objetos.

```
public class Principal {
    public abstract class Fruta {
        public abstract String toString();
    }
    public class Maca extends Fruta {
        public String toString(){ return this.getClass().getSimpleName(); }
    }
    public class Abacate extends Fruta {
        public String toString(){ return this.getClass().getSimpleName(); }
    }

    public class Prateleira {
        private Fruta fruta;
        public Prateleira(Fruta fruta) {
            setFruta(fruta);
        }
        public void setFruta(Fruta fruta) {
            this.fruta = fruta;
        }
        public String toString(){ return this.fruta.toString(); }
    }

    public Principal(){

        Prateleira prateleira = new Prateleira( new Maca() );
        System.out.println( prateleira );

        Prateleira prateleira2 = new Prateleira( new Abacate() );
        System.out.println( prateleira2 );

        prateleira.setFruta( new Abacate() );
        System.out.println( prateleira );
    }

    public static void main(String [] args ){

        new Principal();
    }
}
```

```
}
```

- 4) (Online) Exercício: Acesse o link da atividade (Aula25prog2): <https://codeboard.io/projects/357606>

- 5) Padrão de Projeto: Interface com Delegação.

Motivação: o objeto poderá mudar de classe com o uso de polimorfismo, ou seja, o objeto da subclasse é tratado como um objeto de sua superclasse. Interface com Delegação é implementada pela composição com uma interface.

```
public class Principal {
    public interface IFruta {
        public abstract String toString();
    }
    public class Maca implements IFruta {
        public String toString(){ return this.getClass().getSimpleName(); }
    }
    public class Abacate implements IFruta {
        public String toString(){ return this.getClass().getSimpleName(); }
    }

    public class Prateleira {
        private IFruta interfaceFruta;
        public Prateleira(IFruta interfaceFruta) {
            setFruta(interfaceFruta);
        }
        public void setFruta(IFruta interfaceFruta) {
            this.interfaceFruta = interfaceFruta;
        }
        public String toString(){ return this.interfaceFruta.toString(); }
    }

    public Principal(){

        Prateleira prateleira = new Prateleira( new Maca() );
```

```
        System.out.println( prateleira );

        Prateleira prateleira2 = new Prateleira( new Abacate() );
        System.out.println( prateleira2 );

        prateleira.setFruta( new Abacate() );
        System.out.println( prateleira );
    }

    public static void main(String [] args ){

        new Principal();
    }
}
```

- 6) (Online) Exercício: Acesse o link da atividade (Aula25prog3):
<https://codeboard.io/projects/357610>

//Exemplo 5.1: Padrão de Projeto: Interface com Delegação (com classes não-relacionadas). Relacionar: Bicicleta, Carro, Predio

```
public class Principal {

    public interface IEmissaoCarbono {
        public abstract float getEmissaoCarbono();
    }

    public class Bicicleta implements IEmissaoCarbono {

        public float getEmissaoCarbono(){
            return 0.0f;
        }
    }

    public class Carro implements IEmissaoCarbono {

        public float getEmissaoCarbono(){
            return 50.0f;
        }
    }

    public class Predio implements IEmissaoCarbono {

        public float getEmissaoCarbono(){
            return 10.0f;
        }
    }

    public class Poluente {
        private IEmissaoCarbono poluente;
        public Poluente(IEmissaoCarbono poluente){
            setPoluente(poluente);
        }
        public void setPoluente(IEmissaoCarbono poluente){
            this.poluente = poluente;
        }
        public String toString(){
            return this.poluente.getEmissaoCarbono();
        }
    }

    public Principal(){
        Poluente poluente = new Poluente( new Carro() );
        System.out.println( poluente );

        poluente.setPoluente( new Bicicleta() );
        System.out.println( poluente );
    }
}
```

```
}  
  
public static void main(String [] args){  
    new Principal();  
  
}  
}
```

- 7) (Online) Exercício: Acesse o link da atividade (Aula25prog4):
<https://codeboard.io/projects/357611>

- 8) (Online) Exercício: Acesse o link da atividade (Aula25prog5):
<https://codeboard.io/projects/357613>