












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

Nome:

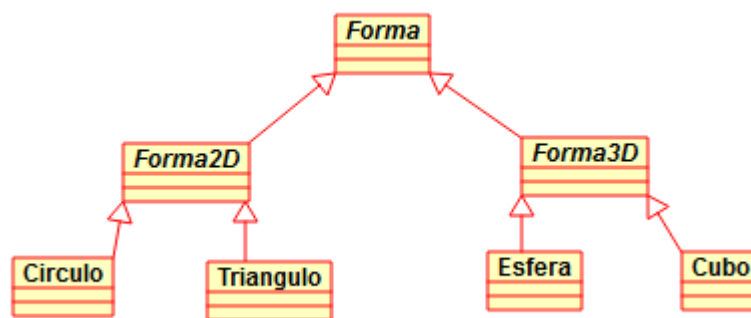
Link:

Luis Henrique Mendes <  Cópia de Aula 14 - POCO4A - Exercícios propostos >
Thiago Cristovão de Souza <  Cópia de Aula 14 - POCO4A - Exercícios propostos >
Filipe Augusto Parreira Almeida  Cópia de Aula 14 - POCO4A - Exercícios propostos
Isabella Melo Almeida  Cópia de Aula 14 - POCO4A - Exercícios propostos
Angélica B. G. Luciano  Cópia de Aula 14 - POCO4A - Exercícios propostos
Rafael Kendy Naramoto Lopes <  Cópia de Aula 14 - POCO4A - Exercícios propostos >
Matheus Mazali Maeda and Alexandre Aparecido <  Exercícios >
Deivid da Silva Galvão<  Cópia de Aula 14 - POCO4A - Exercícios propostos >
João Vitor N. Yoshida <  Cópia de Aula 14 - POCO4A - Exercícios propostos >
Thales Alves <  Aula 14 - POCO4A - Exercícios propostos >
Thales Kohki Hasegawa <  Cópia de Aula 14 - POCO4A - Exercícios propostos >
Daniel Martins de Carvalho <  Cópia de Aula 14 - POCO4A - Exercícios propostos >
Julio Farias<  Cópia de Aula 14 - POCO4A - Exercícios propostos >
Gabriel Reis Macedo <  Cópia de Aula 14 - POCO4A - Exercícios propostos >
Marcos Tadao Shoji<  Cópia de Aula 14 - POCO4A - Exercícios propostos >
mabyly <  Cópia de Aula 14 - POCO4A - Exercícios propostos >
Matheus Hirata <  Cópia de Aula 14 - POCO4A - Exercícios propostos >
Rodrigo Leandro Benedito  Cópia de Aula 14 - POCO4A - Exercícios propostos
Vitor Luiz de Castro Viana:  Cópia de Aula 14 - POCO4A - Exercícios propostos
Thiago Tieghi: <  Cópia de Aula 14 - POCO4A - Exercícios propostos >
Felipinho Lorusso: < [Arquivo Felipe Lorusso](#) >
Pedro Reis Aspeta :  Cópia de Aula 14 - POCO4A - Exercícios propostos
Felipe Antonio Magro:  Cópia de Aula 14 - POCO4A - Exercícios propostos
Carlos Eduardo Ribeiro:  Cópia de Aula 14 - POCO4A - Exercícios propostos
Victor Ramos  Cópia de Aula 14 - POCO4A - Exercícios propostos
Lucas Viana  Cópia de Aula 14 - POCO4A - Exercícios propostos

1) (Online) Acesse o link e realize as atividades propostas:

<https://codeboard.io/projects/346997>

2) Observe a Figura a seguir:



```

import java.util.ArrayList;

public class Principal {

    public abstract class Forma {
        protected float area;
        /* public Forma(float area){
            this.area = area;
        }*/
        //Deve ser sobrescrito (Override) pelas subclasses
        public String toString(){ return this.getClass().getSimpleName(); }
    }

    public abstract class Forma2D extends Forma {
        public abstract float getArea();
        /*public Forma2D(float area){
            super( area );
        }*/
    }

    public abstract class Forma3D extends Forma {
        protected double volume;
        public abstract double getArea();
        public abstract double getVolume();
        /*public Forma3D(float area, float volume){
            //super( area );
            this.volume = volume;
        }*/
    }

    public final class Circulo extends Forma2D {
        private float raio;
        public Circulo(float raio){
            // super( getArea() );
            this.raio = raio;
        }
    }
}

```

```

    }
    public float getArea(){
        return 3.14f * raio*raio;
    }
}

public final class Triangulo extends Forma2D {
    private float base;
    private float altura;
    public Triangulo(float base, float altura){
        //super( getArea() );
        this.base = base;
        this.altura = altura;
    }
    public float getArea(){
        return (this.base * this.altura )/2;
    }
}

public final class Esfera extends Forma3D {
    private double raio;
    public Esfera( double raio ){
        //super( getArea(), getVolume() );
        this.raio = raio;
    }
    public double getArea(){
        return (double)4*3.14*raio*raio;
    }
    public double getVolume(){
        double raio = this.raio;
        return 4/3*3.14f*Math.pow(raio,3f);
    }
}

public final class Cubo extends Forma3D {
    private double lado;
    public Cubo(double lado){
        //super( getArea(), getVolume() );
        this.lado = lado;
    }
    public double getArea(){
        return 6*Math.pow(this.lado,2);
    }
    public double getVolume(){
        return Math.pow(this.lado, 3);
    }
}

```

```

public void iniciar(){

    ArrayList<Forma> lista = new ArrayList<>();
    Circulo c1 = new Circulo(123.0f);
    lista.add( c1 );

    Triangulo t1 = new Triangulo(123.0f, 2.0f);
    lista.add( t1 );

    //A= 4*Pi*r^2
    //V= 4/3*Pi*r^3
    Esfera e1 = new Esfera( 123.0 );
    lista.add( e1 );

    //A = 6*lado^2
    //V = lado^3
    Cubo cubo1 = new Cubo(123.0);
    lista.add( cubo1 );

    for( Forma forma : lista )
        System.out.println( forma );
    }

    public static void main(String [ ] args){
        Principal principal = new Principal();
        principal.iniciar();
    }
}

```

- a) Implemente a hierarquia de Classes mostrada na Figura. Apenas as Classes folha são Classes concretas, as demais são classes abstratas.
- b) A Classe Forma2D deve conter o método getArea.
- c) A Classe Forma3D deve conter os métodos getArea e getVolume.
- d) Crie uma Classe Principal que tenha um vetor de Formas com objetos de cada Classe concreta.
- e) O programa deve imprimir o tipo de cada objeto instanciado.

f) Modifique o exercício para incluir todas as classes em uma classe interna.

3) Faça a implementação Orientada a Objetos do problema enunciado a seguir:

- a) Crie 3 (três) classes não relacionadas por herança: Construção, Carro e Bicicleta.
- b) Dê a cada Classe atributos e comportamentos únicos que não estão presentes em outras classes.
- c) Crie a Interface EmissaoCarbono com um método getEmissaoCarbono.
- d) Cada Classe deve implementar a Interface EmissaoCarbono.
- e) invoque o método getEmissaoCarbono de cada objeto.

```
import java.util.ArrayList;

public class Principal {

    public interface EmissaoCarbono {

        public abstract float getEmissaoCarbono();
        public abstract String toString();
    }

    public class Construção implements EmissaoCarbono {
        private float tamanho;
        public float getTamanho(){ return this.tamanho; }

        public String toString(){
            return this.getClass().getSimpleName() +
                " EmissaoCarbono:" + getEmissaoCarbono();
        }

        @Override
        public float getEmissaoCarbono() {
            return 0.0f;
        }
    }
}
```

```

}
public class Carro implements EmissaoCarbono {
    private String marca;
    public String getMarca(){ return this.marca; }
    public String toString(){
        return this.getClass().getSimpleName() +
            " EmissaoCarbono:" + getEmissaoCarbono();
    }
    @Override
    public float getEmissaoCarbono() {
        return 123.0f;
    }
}

public class Bicicleta implements EmissaoCarbono {
    private String cor;
    public String getCor(){ return this.cor; }
    public String toString(){
        return this.getClass().getSimpleName() +
            " EmissaoCarbono:" + getEmissaoCarbono();
    }
    @Override
    public float getEmissaoCarbono() {
        return 0.0f;
    }
}

}

    public Principal(){

        ArrayList<EmissaoCarbono> lista = new ArrayList<>();
        Construc o construc o = new Construc o();
        lista.add( construc o );

        Carro carro = new Carro();
        lista.add( carro );

        Bicicleta bicicleta = new Bicicleta();
        lista.add( bicicleta );

        for( EmissaoCarbono objeto : lista )
            System.out.println( objeto );

    }

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

```

}
