

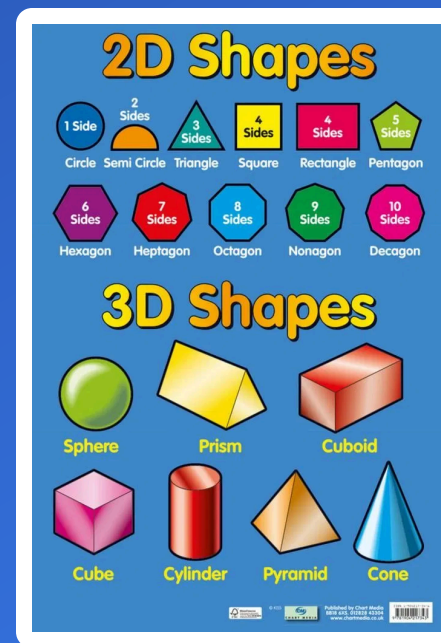
Resolução da Questão 3

Formas Geométricas

Enunciado:

Crie uma interface chamada **FormaGeometrica** com os seguintes métodos:

- Método **calcularArea()** que retorna a área da forma geométrica.
- Método **calcularPerimetro()** que retorna o perímetro da forma (para formas 2D).
- Método **calcularVolume()** que retorna o volume da forma (para formas 3D).
- Implemente as classes **Retangulo** (2D) e **Caixa** (3D).
- Implemente as classes **Circulo** (2D) e **Cilindro** (3D).
- Crie uma classe de teste para verificar o funcionamento das implementações.



Estrutura da Interface e Classes

Hierarquia e implementação da interface FormaGeometrica

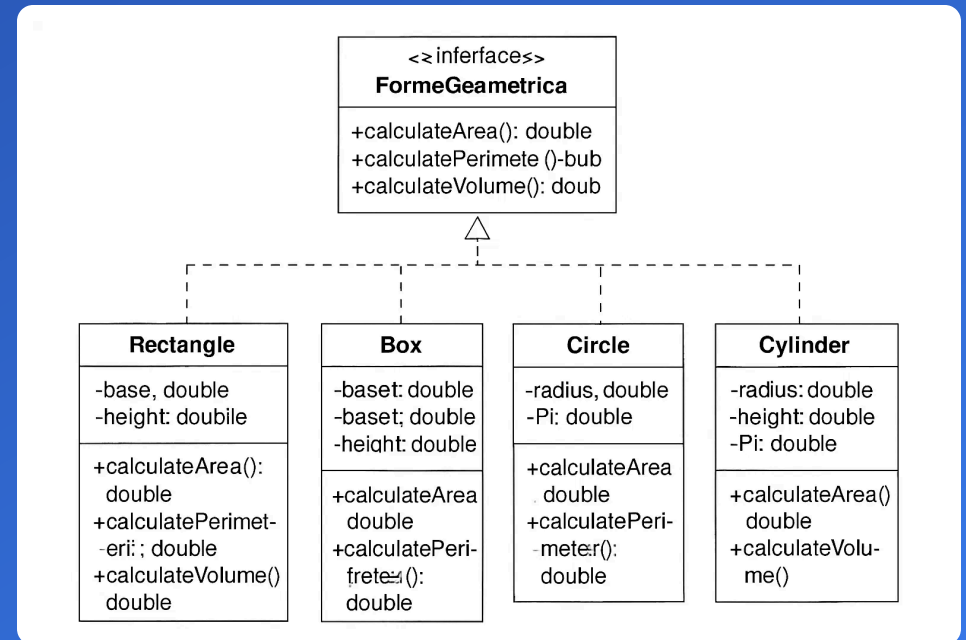
Interface FormaGeometrica:

```
public interface FormaGeometrica {  
    double calcularArea();  
    double calcularPerimetro(); // Para 2D  
    double calcularVolume();    // Para 3D  
}
```

Estrutura do Projeto:

- **Interface:** FormaGeometrica
- **Formas 2D:** Retangulo, Circulo
- **Formas 3D:** Caixa, Cilindro

Todas as classes implementam a interface FormaGeometrica, mas as formas 2D retornam 0 para volume, e as formas 3D retornam 0 para perímetro.



Classes Retangulo e Caixa

Implementação das formas retangulares 2D e 3D

Classe Retangulo (2D):

```
public class Retangulo implements FormaGeometrica {
    private double base;
    private double altura;

    public Retangulo(double base, double altura) {
        this.base = base;
        this.altura = altura;
    }

    @Override
    public double calcularArea() {
        return base * altura;
    }

    @Override
    public double calcularPerimetro() {
        return 2 * base + 2 * altura;
    }

    @Override
    public double calcularVolume() {
        return 0; // Retângulo é 2D, não tem volume
    }
}
```

Classe Caixa (3D):

```
public class Caixa implements FormaGeometrica {
    private double base1;
    private double base2;
    private double altura;

    public Caixa(double base1, double base2, double altura) {
        this.base1 = base1;
        this.base2 = base2;
        this.altura = altura;
    }

    @Override
    public double calcularArea() {
        return 2 * (base1 * base2 + base1 * altura +
                    base2 * altura);
    }

    @Override
    public double calcularPerimetro() {
        return 0; // Caixa é 3D, não tem perímetro 2D
    }

    @Override
    public double calcularVolume() {
        return base1 * base2 * altura;
    }
}
```

