



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PIAUÍ

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
MÓDULO IV - ENGENHARIA DE SOFTWARE III
PROF.: Ely Miranda
ALUNO: Lucas Gomes de Oliveira (20191ADS0185)

01. Imagine que você deve modelar várias figuras geométricas em TypeScript e que cada uma tem sua forma específica de calcular área e perímetro. Proponha e implemente uma hierarquia de classes usando uma classe abstrata chamada **FiguraGeometrica** e outras concretas: **Quadrado**, **Triangulo**, etc.

```
interface FiguraGeometrica {  
    valor: number;  
    quantidade: number;  
    //calculaArea(): void;  
    calculaPerimetro(): void;  
}  
  
class Figura implements FiguraGeometrica {  
    //const objeto: FiguraGeometrica;  
  
    valor: number;  
    quantidade: number;  
  
    constructor(x: number, y: number) {  
        this.valor = x;  
        this.quantidade = y;  
    }  
  
    calculaPerimetro() {  
        console.log(`Valor do perimetro => ${this.valor *  
this.quantidade}`)  
    }  
}
```

```
const Quadrado: Figura = new Figura(12, 4);
Quadrado.calculaPerimetro()
```

05

```
interface Tributavel {
    calculaTributo(): number;
}

class Conta {
    private _nome: string;
    private _saldo: number;

    constructor(x: string, y: number) {
        this._nome = x;
        this._saldo = y;
    }

    get getNome(): string {
        return this._nome;
    }

    set setNome(novoNome: string) {
        this._nome = novoNome;
    }

    get getSaldo(): number {
        return this._saldo;
    }

    set setSaldo(novoSaldo: number) {
        this._saldo = novoSaldo;
    }
}

class ContaCorrente extends Conta implements Tributavel {
    constructor(nome: string, saldo: number) {
        super(nome, saldo);
    }

    calculaTributo() {
```

```

        return this.getSaldo * 0.1;
    }
}

class SeguroDeVida implements Tributavel {
    calculaTributo() {
        return 50;
    }
}

class AuditoriaInterna {
    public lista: Array<Tributavel> = [];
    public tributavel = {} as Tributavel;

    adicionar(tributavel: Tributavel): void {
        this.lista.push(tributavel);
    }

    calcularTributos(): number {
        let valor: number = 0;

        for (let i: number = 0; i < this.lista.length; i++) {
            valor = valor + this.tributavel.calculaTributo();
        }

        return valor;
    }
}

const contaLucas: ContaCorrente = new ContaCorrente("Lucas Gomes de Oliveira", 790);
const contaClara: ContaCorrente = new ContaCorrente("Clara Beatriz", 125);
const contaAna: ContaCorrente = new ContaCorrente("Ana", 16287);

const seguro1: SeguroDeVida = new SeguroDeVida();
const auditoria = new AuditoriaInterna();

auditoria.adicionar(contaLucas);
auditoria.adicionar(seguro1);

console.log(contaLucas.getSaldo);
console.log(seguro1.calculaTributo());

```

