



UEPB

DISCIPLINA **Métodos Avançados de Programação**

PERÍODO: **2021.2**

PROFESSORA: **Sabrina de F. Souto.**

DATA: ____/____/____

ALUNO(A): _____ Nota: _____

Laboratório: Baixo Acoplamento e Alta Coesão

1. Modifique o código fornecido para deixá-lo coeso e com um baixo nível de acoplamento.

Sugestões pré-laboratório:

- Entenda o código da classe FigurasGeometricas e BrincandoComAsFigurasGeometricas;
- Identifique falhas no encapsulamento;
- Identifique falhas na distribuição dos conceitos utilizados;
- Identifique responsabilidades semelhantes entre os elementos manipulados.

Sugestões para resolução do laboratório:

- Melhore o tratamento os erros;
- Melhore o encapsulamento dos dados;
- Melhore a coesão dos dados;
- Melhore o acoplamento dos dados;
- Melhore o tratamento para classes com comportamento semelhantes.

Itens avaliados:

- 30% - coesão
- 30% - acoplamento
- 20% - adequação da classe main
- 20% - testes

Classe

```
public class FigurasGeometricas {  
  
    public static final int R = -1;  
    public static final int Q = 0;  
    public static final int C = 1;  
    public static final int T = 2;  
  
    private int alturaRetangulo;  
    private int larguraRetangulo;  
    private int larguraQuadrado;  
    private int raioCirculo;  
  
    public FigurasGeometricas(int altura, int largura, int lado, int  
    raio){  
        this.alturaRetangulo = altura;  
        this.larguraRetangulo = largura;  
        this.larguraQuadrado = lado;  
        this.raioCirculo = raio;  
    }  
}
```

```

}

public void a(int tipoDaFigura){
    switch (tipoDaFigura) {
        case R:
            System.out.println("A área deste retângulo é: " +
                alturaRetangulo * larguraRetangulo);
            break;
        case Q:
            System.out.println("A área deste quadrado é: " + 2 *
                larguraQuadrado);
            break;
        case C:
            System.out.println("A área deste círculo é: " + (2 *
                3.14 * raioCirculo));
            break;
        case T:
            System.out.println("Todas as áreas são: " +
                (alturaRetangulo * larguraRetangulo) + " " + (2 *
                larguraQuadrado) + " e " + (2 * 3.14 * raioCirculo) +
                " retângulo, quadrado e círculo, respectivamente");
            break;
        default:
            System.out.println("O valor que vc forneceu não
                representa nenhuma figura geométrica conhecida!");
            break;
    }
}

public int p(int tipoDaFigura){

    int perimetro;

    if (tipoDaFigura == -1){
        perimetro = (2 * alturaRetangulo) + (2 *
            larguraRetangulo);
        System.out.println("O perímetro do retângulo é: " +
            perimetro);
    }else if (tipoDaFigura == 0){
        perimetro = 4 * larguraQuadrado;
        System.out.println("O perímetro do quadrado é: " +
            perimetro);
    }else if (tipoDaFigura == 1){
        perimetro = (int) (3.14 * raioCirculo);
        System.out.println("O perímetro do círculo é: " +
            perimetro);
    }else if (tipoDaFigura == 2){
        perimetro = -234;
    }else{
        perimetro = -2658;
    }
    return perimetro;
}

public String toStringDaFigura(int tipoDaFigura){
    String toString;
    switch (tipoDaFigura) {

```

```

        case R:
            toString = "O retângulo criado tem altura de tamanho: "
                + alturaRetangulo + " e largura de tamanho: " +
                larguraRetangulo;
            break;
        case Q:
            toString = "O quadrado criado tem lados de tamanho: "
                + larguraQuadrado;
            break;
        case C:
            toString = "O círculo criado tem raio de tamanho: " +
                raioCirculo;
            break;
        case T:
            toString = "O retângulo criado tem altura: " +
                alturaRetangulo + " e largura de: " +
                larguraRetangulo + " O quadrado criado tem lados de: "
                + larguraQuadrado + " O círculo criado tem raio de: "
                + raioCirculo;
            break;

        default:
            toString = null;
            break;
    }
    return toString;
}
}

```

Main

```

public class BrincandoComAsFigurasGeometricas {

    public static void main(String[] args) {

        FigurasGeometricas fig = new FigurasGeometricas(10, 5, 7,
            25);

        int per = fig.p(FigurasGeometricas.R);
        if (per == -234 || per == -2658) {
            System.out.println("ERRO");
        }

        per = fig.p(FigurasGeometricas.C);
        if (per == -234 || per == -2658) {
            System.out.println("ERRO");
        }

        per = fig.p(-2);
        if (per == -234 || per == -2658) {
            System.out.println("ERRO");
        }

        per = fig.p(0);
    }
}

```

```
    if (per == -234 || per == -2658) {  
        System.out.println("ERRO");  
    }  
  
    fig.a(0);  
    fig.a(-1);  
    fig.a(2);  
    fig.a(1);  
    fig.a(-2);  
  
    String inf = fig.toStringDaFigura(-1);  
    inf = fig.toStringDaFigura(1);  
    System.out.println(inf);  
    inf = fig.toStringDaFigura(0);  
    System.out.println(inf);  
    inf = fig.toStringDaFigura(4);  
    System.out.println(inf);  
    inf = fig.toStringDaFigura(2);  
    System.out.println(inf);  
}  
  
}
```