

AVALIAÇÃO 2º ANO - 2ª ETAPA

ansiosocraft@gmail.com [Alternar conta](#)



Rascunho salvo.

***Obrigatório**

Questões

Avaliação referente aos conhecimentos de de LP e POO

Pontos por questão: 5 pontos

Pontuação total: 50 pontos



Observe a imagem abaixo e marque a opção FALSA: *

```
internal class ClasseAtividade
{
    private double valor;

    0 referências
    public ClasseAtividade(double valor)
    {
        this.valor = valor;
    }

    1 referência
    public double CalcularValores(int x, int b)
    {
        return x * b * valor;
    }

    0 referências
    public double CalcularValores2(int k)
    {
        return k * valor * CalcularValores(k, k);
    }
}
```

- ☒ Todos os métodos que constam na classe acima tem algum tipo retorno
- ☐ Trata-se de uma classe que pode ser instanciada(viar objeto) normalmente.
- ☐ A classe possui apenas um atributo
- ☐ A classe tem dois métodos com retorno do tipo Double
- ☐ A classe está implementada de forma correta



MARQUE A OPÇÃO VERDADEIRA: *

```
internal class FigurasGeometricas
{
    private double ladoA;
    private double ladoB;
    private double area;

    0 referências
    public double LadoA { get { return ladoA; } set { ladoA = value; } }
    0 referências
    public double LadoB { get { return ladoB; } set { ladoB = value; } }
    0 referências
    public double Area { get { return area; } set { area = value; } }

    0 referências
    public void CalcularAreaDoLosango()
    {
        area = (ladoA*ladoB) / 2;
    }

    0 referências
    public void CalcularAreaRetangulo()
    {
        area = (ladoA*ladoB);
    }
}

1 using ConsoleAppProva.RegrasDeNegocio;
2 FigurasGeometricas[] listaFig = new FigurasGeometricas[50];
3 FigurasGeometricas obj1 = new FigurasGeometricas();
4 FigurasGeometricas obj2 = new FigurasGeometricas();
5 FigurasGeometricas obj3 = new FigurasGeometricas();
6 FigurasGeometricas obj4 = new FigurasGeometricas();
7
8 obj1.LadoA = 7;
9 obj1.LadoB = 8;
10 obj1.CalcularAreaDoLosango();
11
12 obj3.LadoA = 7;
13 obj3.LadoB = 8;
14 obj3.CalcularAreaRetangulo();
15
16 obj4.LadoA = 5;
17 obj4.LadoB = obj1.Area;
18 obj4.CalcularAreaDoLosango();
19
20 listaFig[3] = obj3;
21 listaFig[16] = obj1;
22 listaFig[4] = obj4;
23 listaFig[47] = obj2;
```

- ☐ Podemos afirmar que não é possível calcular a área de duas figuras geométricas
- ☐ A classe possui apenas três atributos do tipo "int"
- ☐ Foram criados 5 objetos e um vetor para registrar esses objetos
- ☐ Considerando os comandos até a linha 23, é possível concluir que nenhum objeto ficou vazio
- ☒ É possível registrar mais de 49 objetos na lista que foi criada



Observe a Imagem abaixo e responda: *

```
public static class ClasseAtividade2
{
    0 referências
    public static double Calcular(int x, int y)
    {
        int k = x * y;
        return k;
    }

    0 referências
    public static double Calcular(double x, int y)
    {
        double k = x * y;
        return k;
    }
}
```

```
// Class Program
using Prova2;
double x = ClasseAtividade2.Calcular(0.5, 10);
```

- ☐ Não será possível calcular porque o programador não instanciou a classe (ClasseAtividade2 at = new ClasseAtividade2())
- ☐ Será possível calcular, e será armazenado 50 no valor de "x".
- ☒ Será possível calcular, e será armazenado 5 no valor de "x".
- ☐ Não será possível calcular porque o programador escolheu o primeiro método, onde variável "x" é do tipo inteiro
- ☐ Não será possível calcular porque o programador não definiu qual dos métodos ele vai utilizar



Análise a imagem abaixo: *

- ☐
- ☐
- ☒
- ☐
- ☐

Observe os dois trechos de código abaixo e marque a opção verdadeira: *

- ☒
- ☐
- ☐
- ☐
- ☐

mente o `CalcularDesconto()` está implementado incorretamente.

O "return" no método `CalcularJuros()` está sendo utilizado de forma incorreta



Analise a imagem abaixo: *

```

internal class FigurasGeometricas
{
    private double ladoA;
    private double ladoB;
    private double area;

    O referências
    public double LadoA { get { return ladoA; } set { ladoA = value; } }
    O referências
    public double LadoB { get { return ladoB; } set { ladoB = value; } }
    O referências
    public double Area { get { return area; } set { area = value; } }

    O referências
    public void CalcularAreaDoLosango()
    {
        area = (ladoA*ladoB) / 2;
    }

    O referências
    public void CalcularAreaRetangulo()
    {
        area = (ladoA*ladoB);
    }
}

1 using ConsoleAppProva.RegasDeNegocio;
2 FigurasGeometricas[] listaFig = new FigurasGeometricas[50];
3 FigurasGeometricas obj1 = new FigurasGeometricas();
4 FigurasGeometricas obj2 = new FigurasGeometricas();
5 FigurasGeometricas obj3 = new FigurasGeometricas();
6 FigurasGeometricas obj4 = new FigurasGeometricas();
7
8 obj1.LadoA = 7;
9 obj1.LadoB = 8;
10 obj1.CalcularAreaDoLosango();
11
12 obj3.LadoA = 7;
13 obj3.LadoB = 8;
14 obj3.CalcularAreaRetangulo();
15
16 obj4.LadoA = 5;
17 obj4.LadoB = obj1.Area;
18 obj4.CalcularAreaDoLosango();
19
20 listaFig[3] = obj3;
21 listaFig[15] = obj1;
22 listaFig[4] = obj4;
23 listaFig[47] = obj2;

```

Se na linha 24 e 25 e for inserido os comandos abaixo, qual será o resultado exibido na tela:

```

Console.WriteLine("R1: " + listaFig[4].Area*2);
Console.WriteLine("R2: " + listaFig[15].LadoB*5);

```

- ☐ R1: 140 e R2: 56
- ☐ R1: 140 e R2: 8
- ☐ R1: 140 e R2: 180
- ☒ R1: 140 e R2: 80
- ☐ R1: 102 e R2: 16

Na orientação a objetos, o conceito de encapsulamento corresponde à propriedade de: *

- ☐ usar variáveis e constantes do tipo inteiro nos métodos das classes implementadas.
- ☐ ter um conjunto de objetos com a mesma classe.
- ☐ receber, por uma classe, uma mensagem sem parâmetros.
- ☐ utilizar estruturas de matrizes quadradas nos programas desenvolvidos.
- ☒ esconder ou ocultar detalhes da implementação de uma dada classe de outras classes.




```
internal class FigurasGeometricas
{
    private double ladoA;
    private double ladoB;
    private double area;

    0 referências
    public double LadoA { get { return ladoA; } set { ladoA = value; } }
    0 referências
    public double LadoB { get { return ladoB; } set { ladoB = value; } }
    0 referências
    public double Area { get { return area; } set { area = value; } }

    0 referências
    public void CalcularAreaDoLosango()
    {
        area = (ladoA*ladoB) / 2;
    }

    0 referências
    public void CalcularAreaRetangulo()
    {
        area = (ladoA*ladoB);
    }
}
```

```
using ConsoleAppProva.RegrasDeNegocio;

FigurasGeometricas retangulo = new FigurasGeom
Console.Write("base....: ");
retangulo.LadoA = Convert.ToDouble(Console.Reac
Console.Write("altura....: ");
retangulo.LadoA= Convert.ToDouble(Console.ReadL
Console.WriteLine("Área do Retangulo: " + retang
```

O programa não tem nenhum erro de i

O programa possui um erro, e para fun
processamento.

Da forma que foi implementado, o pro