





TOPICOS DA AULA

PROGRAMAÇÃO ORIENTADA À OBJETOS

APLICAÇÃO DE MÉTODOS EM CLASSES

USO DE COMENTÁRIOS

O USO DO UML

EXERCÍCIOS



PROBLEMA EXEMPLO

CRIE UM PROGRAMA PARA LER AS MEDIDAS DOS LADOS DE DOIS TRIÂNGULOS X E Y (MEDIDAS VÁLIDAS). EM SEGUIDA, MOSTRE O VALOR DAS ÁREAS DOS DOIS TRIÂNGULOS E MOSTRAR QUAL DOS DOIS POSSUI A MAIOR ÁREA.

UTILIZE A FÓRMULA A SEGUIR PARA CALCULAR A ÁREA DO TRIÂNGULO E SEUS RESPECTIVOS LADOS A, B, C:

EXEMPLO

ENTRE COM AS MEDIDAS DO TRIANGULO X:

3.00

4.00

5.00

ENTRE COM AS MEDIDAS DO TRIANGULO Y:

7.50

4.50

4.02

AREA DE X = 6.00

AREA DE Y = 7.5638

MAIOR AREA = Y



ESTRUTURADO

```
using System;
using System.Globalization;

namespace Course{
    public class Program{
        public static void Main(string[] args){
            double xA, xB, xC, yA, yB, yC;
            Console.WriteLine("Entre com as medidas do triângulo X:");
            xA = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            xB = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            xC = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            Console.WriteLine("Entre com as medidas do triângulo Y:");
            yA = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            yB = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            yC = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

            double p = (xA + xB + xC) / 2.0;
            double areaX = Math.Sqrt(p * (p - xA) * (p - xB) * (p - xC));

            p = (yA + yB + yC) / 2.0;
            double areaY = Math.Sqrt(p * (p - yA) * (p - yB) * (p - yC));
            Console.WriteLine("Área de X = " + areaX.ToString("F4", CultureInfo.InvariantCulture));
            Console.WriteLine("Área de Y = " + areaY.ToString("F4", CultureInfo.InvariantCulture));

            if (areaX > areaY) {
```



ESTRUTURADO

```
if (areaX > areaY) {  
    Console.WriteLine("Maior área: X");  
}  
else {  
    Console.WriteLine("Maior área: Y");  
}  
}  
}
```



ORIENTADO A OBJETOS

```
namespace Course  
class Triangulo{
```

```
    public double A;  
    public double B;  
    public double C;
```

```
}
```



ORIENTADO A OBJETOS

```
namespace Course
{
    public class Program
    {
        public static void Main(string[] args)
        {
            Triangulo x, y;

            x = new Triangulo();
            y = new Triangulo();

            Console.WriteLine("Entre com as medidas do triângulo X:");
            x.A = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            x.B = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            x.C = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

            Console.WriteLine("Entre com as medidas do triângulo Y:");
            y.A = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            y.B = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            y.C = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

            double p = (x.A + x.B + x.C) / 2.0;
            double areaX = Math.Sqrt(p * (p - x.A) * (p - x.B) * (p - x.C));
```




ORIENTADO A OBJETOS

```
double p = (x.A + x.B + x.C) / 2.0;  
double areaX = Math.Sqrt(p * (p - x.A) * (p - x.B) * (p - x.C));
```

```
p = (y.A + y.B + y.C) / 2.0;  
double areaY = Math.Sqrt(p * (p - y.A) * (p - y.B) * (p - y.C));
```

```
Console.WriteLine("Área de X = " + areaX.ToString("F4", CultureInfo.InvariantCulture));  
Console.WriteLine("Área de Y = " + areaY.ToString("F4", CultureInfo.InvariantCulture));
```

```
if (areaX > areaY)  
{  
    Console.WriteLine("Maior área: X");  
}  
else  
{  
    Console.WriteLine("Maior área: Y");  
}  
  
}  
  
}
```



INSERINDO UM MÉTODO

```
namespace Course  
class Triangulo{
```

```
    public double A;  
    public double B;  
    public double C;
```

```
    public double Area () {
```

```
        double p = (A + B + C) / 2;
```

```
    }
```

```
}
```



INSERINDO UM MÉTODO

```
using System;
namespace Course
class Triangulo{

    public double A;
    public double B;
    public double C;

    public double Area () {

        double p = (A + B + C) / 2;
        double raiz = Math.Sqrt(p * (p - A) * (p - B) * (p - C));
        return raiz;

    }

}
```



ORIENTADO A OBJETOS

```
namespace Course
{
    public class Program
    {
        public static void Main(string[] args)
        {
            Triangulo x, y;
```

```
            x = new Triangulo();
            y = new Triangulo();
```

```
            Console.WriteLine("Entre com as medidas do triângulo X:");
            x.A = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            x.B = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            x.C = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
```

```
            Console.WriteLine("Entre com as medidas do triângulo Y:");
            y.A = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            y.B = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            y.C = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
```



ORIENTADO A OBJETOS

```
double areaX = x.Area();
```

```
double areaY = y.Area();
```

```
Console.WriteLine("Área de X = " + areaX.ToString("F4", CultureInfo.InvariantCulture));
```

```
Console.WriteLine("Área de Y = " + areaY.ToString("F4", CultureInfo.InvariantCulture));
```

```
if (areaX > areaY)
```

```
{  
    Console.WriteLine("Maior área: X");
```

```
}
```

```
else
```

```
{  
    Console.WriteLine("Maior área: Y");
```

```
}
```

```
}
```

```
}
```

```
}
```



INSERINDO UM MÉTODO

//importação de dependências em um projeto

using System;

//nome do projeto

namespace Course

//nome da classe

class Triangulo{

 //atributos da classe

 //prefixo public são utilizados em métodos e atributos para ser usados em outros arquivos

 public double A;

 public double B;

 public double C;

 // o valor double é o tipo de dado que retorna, caso o método não retorne nada se coloca return void

 // nome de método começa com letras maiúsculas e acompanhados de parênteses

 // caso o não tenha nada entre os parênteses de um método é porque não existe mais parâmetros

 public double Area () {

 // corpo do método

 double p = (A + B + C) / 2;

 double raiz = Math.Sqrt(p * (p - A) * (p - B) * (p - C));

 return raiz;

 }

EXERCÍCIO

REFAÇA O EXERCÍCIO DE EXEMPLO E ENVIE
PARA UM REPOSITÓRIO CRIADO NO GITHUB.

NOS DOIS EXERCÍCIOS ANTERIORES, ADICIONE
O MÉTODO DE CÁLCULO DENTRO DA CLASSE
CRIADA.



USO DO UML

O UML é um acrônimo para a expressão Unified Modeling Language e se define em uma série de artefatos que nos ajuda a modelar e documentar os sistemas orientados a objetos que desenvolvemos.

Ator	Caso de Uso	Comunicação
		



USO DO UML

Triangulo

- A : double
- B : double
- C : double

+ Area (); double

EXERCÍCIOS

COM O ENTENDIMENTO SOBRE O UML, UTILIZE O MODELO PARA REALIZAR O PRIMEIRO EXERCÍCIO INDIVIDUAL DADO (PACMAN) E O EXERCÍCIO EM GRUPO (JOGO A ESCOLHA).

UTILIZE O SBC_TEMPLATE PARA ENTREGA DA TAREFA



EXERCÍCIOS

COM BASE NAS DESCRIÇÕES DE TELA A SEGUIR E NO UML DESENVOLVA O PROGRAMA.

UML

Produto

- Nome: string
- Preço: double
- Quantidade: int

- + ValorTotalEstoque (): double
- + AdicionarProdutos (quantidade: int) : void
- + RemoverProdutos (quantidade: int): void



EXERCÍCIOS

ENTRE COM OS PRODUTOS:

NOME: **PS4**

PREÇO: **2000.00**

QUANTIDADE EM ESTOQUE: **10**

DADOS DO PRODUTO: **TV, R\$ 2000.00, 10**
UNIDADES, TOTAL: R\$ 20000.00

DIGITE O NÚMERO A SER ADICIONADO NO
ESTOQUE: **5**

DADOS ATUALIZADOS: **TV, R\$ 2000.00, 15**
UNIDADES, TOTAL: 30000.00

DIGITE O NÚMERO A SER REMOVIDO DO
ESTOQUE: **3**

DADOS ATUALIZADOS: **TV, R\$ 2000.00, 12**
UNIDADES, TOTAL: 24000.00



Perguntas?



MUITO OBRIGADO!!!



daniel.ohata@facens.br



Referências Bibliográficas

- DEITEL, Harvey M. et al. **C# como programar**. São Paulo, SP: Pearson Makron Books, 2003. 1153 p.
- GALUPPO, Fabio; MATHEUS, Vanclei; SANTOS, Wallace. **Desenvolvendo com C#**. Porto Alegre, RS: Bookman, 2004. 496 p.
- MARTIN, James; ODELL, James J. **Análise e projeto orientados a objeto**. São Paulo, SP: Makron Books, 1996. 639 p. ISBN 8534604266.
- LARMAN, Craig. **Utilizando UML e padrões: uma introdução a análise e ao projeto orientados a objetos e ao processo unificado**. 2. ed. Porto Alegre, RS: Bookman, 2004.
- MEDEIROS, Ernani Sales de. **Desenvolvendo software com UML 2.0: definitivo**. São Paulo, SP: Pearson Makron Books, 2006. 264 p.
- ALBAHARI, Joseph; ALBAHARI, Ben. **C# 5.0: in a nutshell: the definitive reference**. United States: O'Reilly, 2012
- SANTOS, Rafael. **Introdução à programação orientada a objetos usando Java**. Rio de Janeiro: Campus, 2003. 326 p.
- STEPHENS, Rod. **C# 5.0: programmer's reference**. United States: John Wiley, 2014