

Curso C# Completo

Programação Orientada a Objetos + Projetos

Capítulo: Classes, atributos, métodos, membros estáticos

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Resolvendo um problema sem orientação a objetos

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Problema exemplo

Fazer um programa para ler as medidas dos lados de dois triângulos X e Y (suponha medidas válidas). Em seguida, mostrar o valor das áreas dos dois triângulos e dizer qual dos dois triângulos possui a maior área.

A fórmula para calcular a área de um triângulo a partir das medidas de seus lados a, b e c é a seguinte (fórmula de Heron):

$$area = \sqrt{p(p-a)(p-b)(p-c)} \quad \text{onde} \quad p = \frac{a+b+c}{2}$$

Exemplo:

```
Entre com as medidas do triângulo X:
3.00
4.00
5.00
Entre com as medidas do triângulo Y:
7.50
4.50
4.02
Área de X = 6.0000
Área de Y = 7.5638
Maior área: Y
```

```
using System;
using System.Globalization;

namespace Course {
    class Program {
        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) {
                Console.WriteLine("Maior área: X");
            }
            else {
                Console.WriteLine("Maior área: Y");
            }
        }
    }
}
```

Criando uma classe com três atributos para representar melhor o triângulo

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Discussão

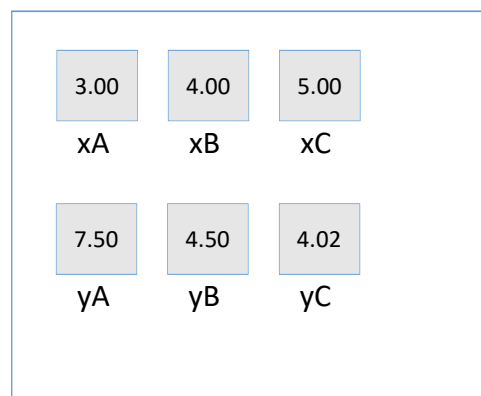
Triângulo é uma entidade com três atributos: a, b, c.

Estamos usando três variáveis distintas para representar cada triângulo:

```
double xA, xB, xC, yA, yB, yC;
```

Para melhorar isso, vamos usar uma CLASSE para representar um triângulo.

Memória:

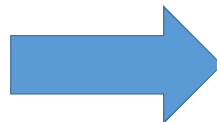


Classe

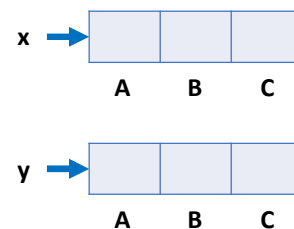
- É um tipo estruturado que pode conter (membros):
 - Atributos (dados / campos)
 - Métodos (funções / operações)
- A classe também pode prover muitos outros recursos, tais como:
 - Construtores
 - Sobrecarga
 - Encapsulamento
 - Herança
 - Polimorfismo
- Exemplos:
 - Entidades: Produto, Cliente, Triangulo
 - Serviços: ProdutoService, ClienteService, EmailService, StorageService
 - Controladores: ProdutoController, ClienteController
 - Utilitários: Calculadora, Compactador
 - Outros (views, repositórios, gerenciadores, etc.)

```
namespace Course {  
    class Triangulo {  
  
        public double A;  
        public double B;  
        public double C;  
  
    }  
}
```

```
double xA, xB, xC, yA, yB, yC;
```



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

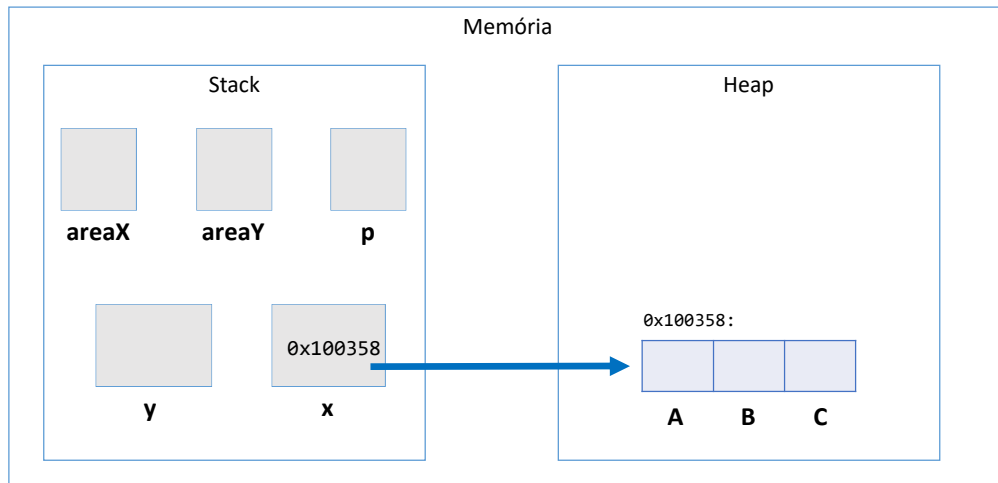


Instanciação

(alocação dinâmica de memória)

```
double areaX, areaY, p;  
Triangulo x, y;
```

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

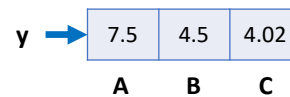
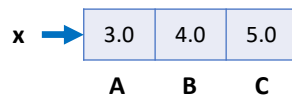


```
using System;  
using System.Globalization;  
  
namespace Course {  
    class Program {  
        static void Main(string[] args) {  
  
            Triangulo x, y;  
  
            x = new Triangulo();  
            y = new Triangulo();  
  
            Console.WriteLine("Entre com as medidas 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));  
  
            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");  
            }  
        }  
    }  
}
```

Classes, objetos, atributos

- Classe: é a definição do tipo
- Objetos: são instâncias da classe

```
namespace Course {  
    class Triangulo {  
        public double A;  
        public double B;  
        public double C;  
    }  
}
```



Primeiros exercícios (classes, objetos e atributos)

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Favor fazer os exercícios do arquivo "**primeiros-exercícios.pdf**" deste capítulo.

Criando um método para
obtermos os benefícios de
reaproveitamento e delegação

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

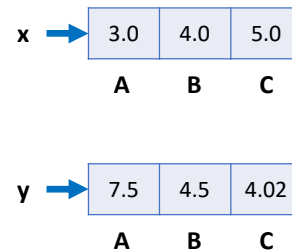
Discussão

Com o uso de CLASSE, agora nós temos uma variável composta do tipo "Triangulo" para representar cada triângulo:

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

Agora vamos melhorar nossa CLASSE, acrescentando nela um MÉTODO para calcular a área.

Memória:




```

using System;

namespace Course {
    class Triangulo {

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

        public double Area() {
            double p = (A + B + C) / 2.0;
            return Math.Sqrt(p * (p - A) * (p - B) * (p - C));
        }
    }
}

```

```

using System;
using System.Globalization;

namespace Course {
    class Program {
        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);

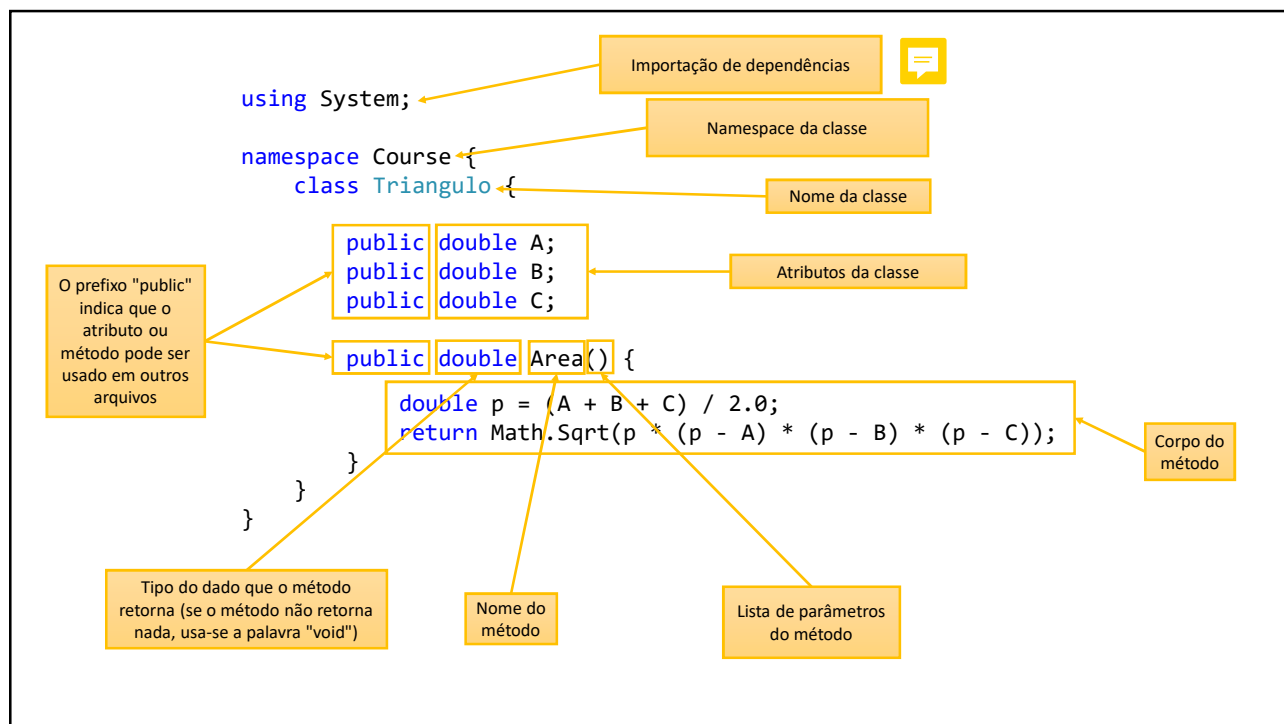
            double areaX = x.Area();

            double areaY = y.Area();

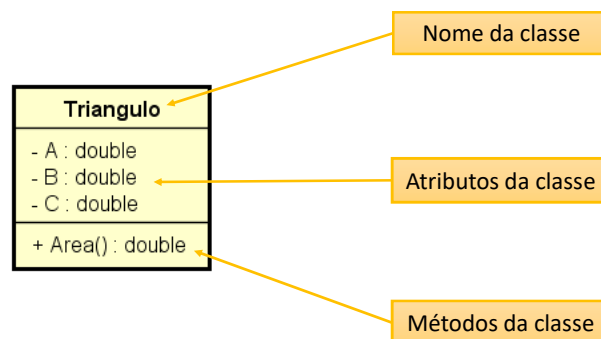
            Console.WriteLine("Área de X = " + areaX.ToString("F4", CultureInfo.InvariantCulture));
            Console.WriteLine("Área de Y = " + areaY.ToString("F4", CultureInfo.InvariantCulture));

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

```



Projeto da classe (UML)



Discussão

Quais são os benefícios de se calcular a área de um triângulo por meio de um MÉTODO dentro da CLASSE Triangulo?

1) Reaproveitamento de código: nós eliminamos o código repetido (cálculo das áreas dos triângulos x e y) no programa principal.

2) Delegação de responsabilidades: quem deve ser responsável por saber como calcular a área de um triângulo é o próprio triângulo. A lógica do cálculo da área não deve estar em outro lugar.

Começando a resolver um segundo problema exemplo

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Problema:

Fazer um programa para ler os dados de um produto em estoque (nome, preço e quantidade no estoque). Em seguida:

- Mostrar os dados do produto (nome, preço, quantidade no estoque, valor total no estoque)
- Realizar uma entrada no estoque e mostrar novamente os dados do produto
- Realizar uma saída no estoque e mostrar novamente os dados do produto

Para resolver este problema, você deve criar uma CLASSE conforme projeto ao lado:

(veja exemplo na próxima página)

Produto
- Nome : string - Preço : double - Quantidade : int
+ ValorTotalEmEstoque() : double + AdicionarProdutos(quantidade : int) : void + RemoverProdutos(quantidade : int) : void

Exemplo:

Entre os dados do produto:

Nome: **TV**

Preço: **900.00**

Quantidade no estoque: **10**

Dados do produto: TV, \$ 900.00, 10 unidades, Total: \$ 9000.00

Digite o número de produtos a ser adicionado ao estoque: **5**

Dados atualizados: TV, \$ 900.00, 15 unidades, Total: \$ 13500.00

Digite o número de produtos a ser removido do estoque: **3**

Dados atualizados: TV, \$ 900.00, 12 unidades, Total: \$ 10800.00

Produto
- Nome : string - Preço : double - Quantidade : int
+ ValorTotalEmEstoque() : double + AdicionarProdutos(quantidade : int) : void + RemoverProdutos(quantidade : int) : void

Object e ToString

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Discussão

- Toda classe em C# é uma subclasse da classe Object
- Object possui os seguintes métodos:
 - GetType - retorna o tipo do objeto
 - Equals - compara se o objeto é igual a outro
 - GetHashCode - retorna um código hash do objeto
 - ToString - converte o objeto para string
- Demo

```
using System.Globalization;

namespace Course {
    class Produto {

        public string Nome;
        public double Preco;
        public int Quantidade;

        public double ValorTotalEmEstoque() {
            return Preco * Quantidade;
        }

        public override string ToString() {
            return Nome
                + ", $ "
                + Preco.ToString("F2", CultureInfo.InvariantCulture)
                + ", "
                + Quantidade
                + " unidades, Total: $ "
                + ValorTotalEmEstoque().ToString("F2", CultureInfo.InvariantCulture);
        }
    }
}
```

Finalizando o programa

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

```

using System.Globalization;

namespace Course {
    class Produto {

        public string Nome;
        public double Preco;
        public int Quantidade;

        public double ValorTotalEmEstoque() {
            return Preco * Quantidade;
        }

        public void AdicionarProdutos(int quantidade) {
            Quantidade += quantidade;
        }

        public void RemoverProdutos(int quantidade) {
            Quantidade -= quantidade;
        }

        public override string ToString() {
            return Nome
                + ", $ "
                + Preco.ToString("F2", CultureInfo.InvariantCulture)
                + ", "
                + Quantidade
                + " unidades, Total: $ "
                + ValorTotalEmEstoque().ToString("F2", CultureInfo.InvariantCulture);
        }
    }
}

```

```

using System;
using System.Globalization;

namespace Course {
    class Program {
        static void Main(string[] args) {

            Produto p = new Produto();

            Console.WriteLine("Entre os dados do produto:");
            Console.Write("Nome: ");
            p.Nome = Console.ReadLine();
            Console.Write("Preço: ");
            p.Preco = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            Console.Write("Quantidade no estoque: ");
            p.Quantidade = int.Parse(Console.ReadLine());

            Console.WriteLine();
            Console.WriteLine("Dados do produto: " + p);

            Console.WriteLine();
            Console.Write("Digite o número de produtos a ser adicionado ao estoque: ");
            int qte = int.Parse(Console.ReadLine());
            p.AdicionarProdutos(qte);

            Console.WriteLine();
            Console.WriteLine("Dados atualizados: " + p);

            Console.WriteLine();
            Console.Write("Digite o número de produtos a ser removido do estoque: ");
            qte = int.Parse(Console.ReadLine());
            p.RemoverProdutos(qte);

            Console.WriteLine();
            Console.WriteLine("Dados atualizados: " + p);
        }
    }
}

```

Exercícios de fixação

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Exercício 1

Fazer um programa para ler os valores da largura e altura de um retângulo. Em seguida, mostrar na tela o valor de sua área, perímetro e diagonal. Usar uma classe como mostrado no projeto ao lado.

Retangulo
- Largura : double - Altura : double
+ Area() : double + Perimetro() : double + Diagonal() : double

Exemplo:

Entre a largura e altura do retângulo:

3.00

4.00

AREA = 12.00

PERÍMETRO = 14.00

DIAGONAL = 5.00

Correção: arquivo **correcao-exercicios-fixacao.pdf**

Exercício 2

Fazer um programa para ler os dados de um funcionário (nome, salário bruto e imposto). Em seguida, mostrar os dados do funcionário (nome e salário líquido). Em seguida, aumentar o salário do funcionário com base em uma porcentagem dada (somente o salário bruto é afetado pela porcentagem) e mostrar novamente os dados do funcionário. Use a classe projetada abaixo.

Exemplo:

Nome: **Joao Silva**
Salário bruto: **6000.00**
Imposto: **1000.00**

Funcionário: Joao Silva, \$ 5000.00

Digite a porcentagem para aumentar o salário: **10.0**

Dados atualizados: Joao Silva, \$ 5600.00

Funcionario
- Nome : string - SalarioBruto : double - Imposto : double
+ SalarioLiquido() : double + AumentarSalario(porcentagem : double) : void

Correção: arquivo **correcao-exercicios-fixacao.pdf**

Exercício 3

Fazer um programa para ler o nome de um aluno e as três notas que ele obteve nos três trimestres do ano (primeiro trimestre vale 30 e o segundo e terceiro valem 35 cada). Ao final, mostrar qual a nota final do aluno no ano. Dizer também se o aluno está APROVADO ou REPROVADO e, em caso negativo, quantos pontos faltam para o aluno obter o mínimo para ser aprovado (que é 60 pontos). Você deve criar uma classe **Aluno** para resolver este problema.

Exemplo 1:

Nome do aluno: **Alex Green**
Digite as três notas do aluno:
27.00
31.00
32.00
NOTA FINAL = 90.00
APROVADO

Exemplo 2:

Nome do aluno: **Alex Green**
Digite as três notas do aluno:
17.00
20.00
15.00
NOTA FINAL = 52.00
REPROVADO
FALTARAM 8.00 PONTOS

Correção: arquivo **correcao-exercicios-fixacao.pdf**

Membros estáticos - PARTE 1

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

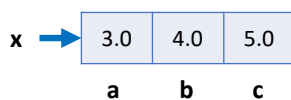
Produto
- Nome : string - Preço : double - Quantidade : int
+ ValorTotalEmEstoque() : double + AdicionarProdutos(quantity : int) : void + RemoverProdutos(quantity : int) : void

membros
=
atributos e métodos

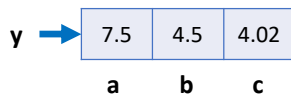
Membros estáticos

- Também chamados membros de classe
 - Em oposição a membros e instância
- São membros que fazem sentido independentemente de objetos. Não precisam de objeto para serem chamados. São chamados a partir do próprio nome da classe.
- Aplicações comuns:
 - Classes utilitárias → **Math.Sqrt(double)**
 - Declaração de constantes
- Uma classe que possui somente membros estáticos, pode ser uma classe estática também. Esta classe não poderá ser instanciada.

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



x.Area() → 6.0



y.Area() → 7.5638

Problema exemplo

Fazer um programa para ler um valor numérico qualquer, e daí mostrar quanto seria o valor de uma circunferência e do volume de uma esfera para um raio daquele valor. Informar também o valor de PI com duas casas decimais.

Exemplo:

```
Entre o valor do raio: 3.0  
Circunferência: 18.84  
Volume: 113.04  
Valor de PI: 3.14
```

Checklist

- Versão 1: métodos na própria classe do programa
 - Nota: dentro de um método estático você não pode chamar membros de instância da mesma classe.
- Versão 2: classe Calculadora com membros de instância
- Versão 3: classe Calculadora com método estático

```

using System;
using System.Globalization;

namespace Course {
    class Program {

        static double Pi = 3.14;

        static void Main(string[] args) {

            Console.Write("Entre o valor do raio: ");
            double raio = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

            double circ = Circunferencia(raio);
            double volume = Volume(raio);

            Console.WriteLine("Circunferência: " + circ.ToString("F2", CultureInfo.InvariantCulture));
            Console.WriteLine("Volume: " + volume.ToString("F2", CultureInfo.InvariantCulture));
            Console.WriteLine("Valor de PI: " + Pi.ToString("F2", CultureInfo.InvariantCulture));
        }

        static double Circunferencia(double r) {
            return 2.0 * Pi * r;
        }

        static double Volume(double r) {
            return 4.0 / 3.0 * Pi * r * r * r;
        }
    }
}

```

VERSÃO 1

Membros estáticos - PARTE 2

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

VERSÃO 2

Discussão

- ```
Calculadora calc1 = new Calculadora();
Calculadora calc2 = new Calculadora();
```
- calc1** → **3.14**  
**Pi**
- calc1.Pi**  
3.14  
**calc1.Circunferencia(3.0)**  
18.85
- calc2** → **3.14**  
**Pi**
- calc2.Pi**  
3.14  
**calc2.Circunferencia(3.0)**  
18.85

```

namespace Course {
 class Calculadora {

 public static double Pi = 3.14;

 public static double Circunferencia(double r) {
 return 2.0 * Pi * r;
 }

 public static double Volume(double r) {
 return 4.0 / 3.0 * Pi * r * r * r;
 }
 }
}

```

**VERSÃO 3**

```

Console.Write("Entre o valor do raio: ");
double raio = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

double circ = Calculadora.Circunferencia(raio);
double volume = Calculadora.Volume(raio);

Console.WriteLine("Circunferência: " + circ.ToString("F2", CultureInfo.InvariantCulture));
Console.WriteLine("Volume: " + volume.ToString("F2", CultureInfo.InvariantCulture));
Console.WriteLine("Valor de PI: " + Calculadora.Pi.ToString("F2",
 CultureInfo.InvariantCulture));

```

## Exercício de fixação (membros estáticos)

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

## Exercício de fixação

Faça um programa para ler a cotação do dólar, e depois um valor em dólares a ser comprado por uma pessoa em reais. Informar quantos reais a pessoa vai pagar pelos dólares, considerando ainda que a pessoa terá que pagar 6% de IOF sobre o valor em dólar. Criar uma classe **ConversorDeMoeda** para ser responsável pelos cálculos.

### Exemplo:

Qual é a cotação do dólar? **3.10**  
Quantos dólares você vai comprar? **200.00**  
Valor a ser pago em reais = 657.20

(correção na próxima página)

```
using System;
using System.Globalization;

namespace Course {
 class Program {
 static void Main(string[] args) {

 Console.WriteLine("Qual é a cotação do dólar? ");
 double cotacao = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

 Console.WriteLine("Quantos dólares você vai comprar? ");
 double quantia = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

 double result = ConversorDeMoeda.DolarParaReal(quantia, cotacao);

 Console.WriteLine("Valor a ser pago em reais = " + result.ToString("F2", CultureInfo.InvariantCulture));

 }
 }
}
```

```
namespace Course {
 class ConversorDeMoeda {

 public static double Iof = 6.0;

 public static double DolarParaReal(double quantia, double cotacao) {
 double total = quantia * cotacao;
 return total + total * Iof / 100.0;
 }
 }
}
```