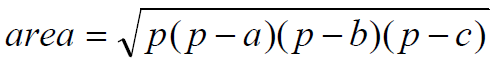
# Override, ToString e void – Aula dia 16/10/2020

# Triângulo

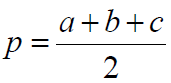
Usando um problema para continuarmos os estudos sobre Classe:

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.

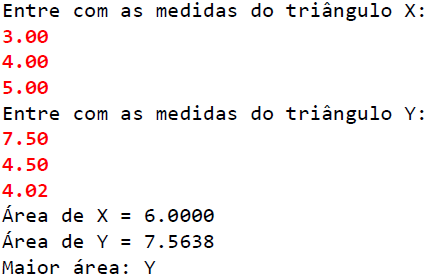
A fórmula para calcular a área de um triângulo a partir das medidas de seus lados a, b e c é a seguinte (fórmula de Heron):



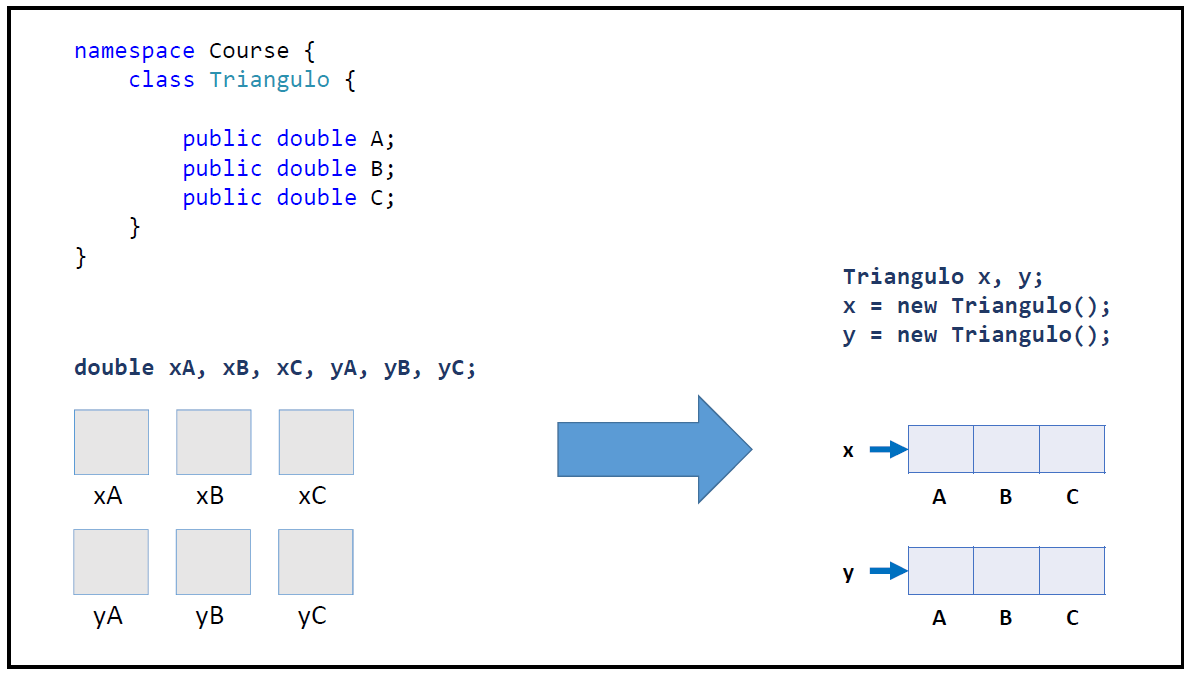
Onde:



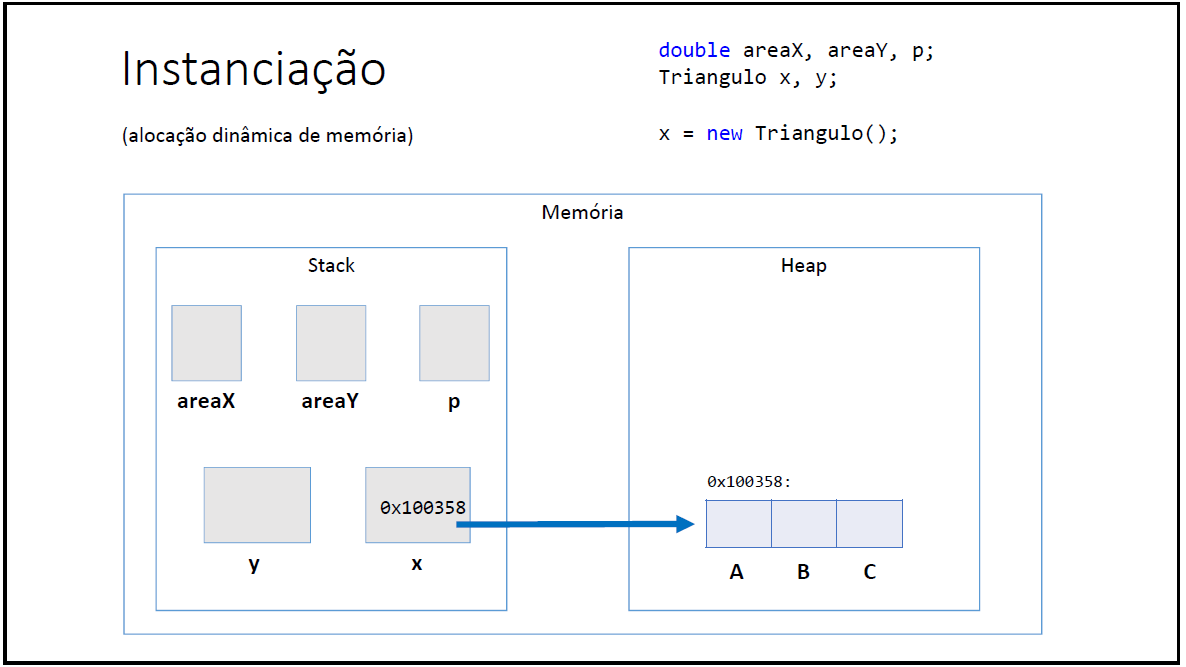
Exemplo:

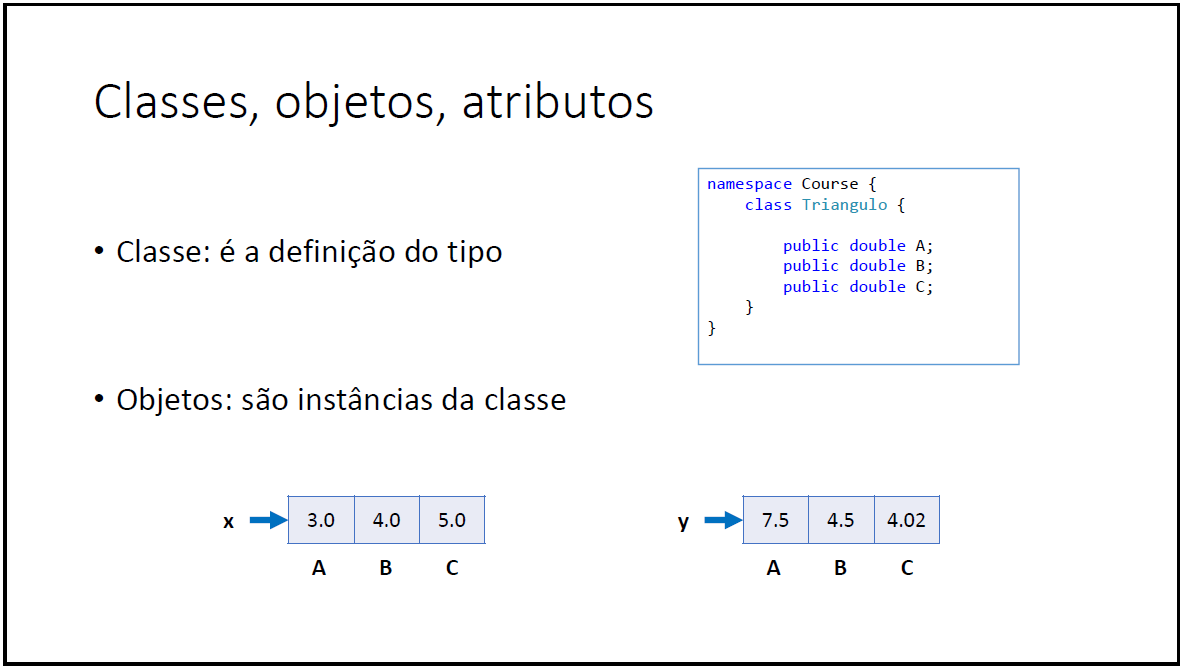


Uma forma para se fazer a classe Triangulo:

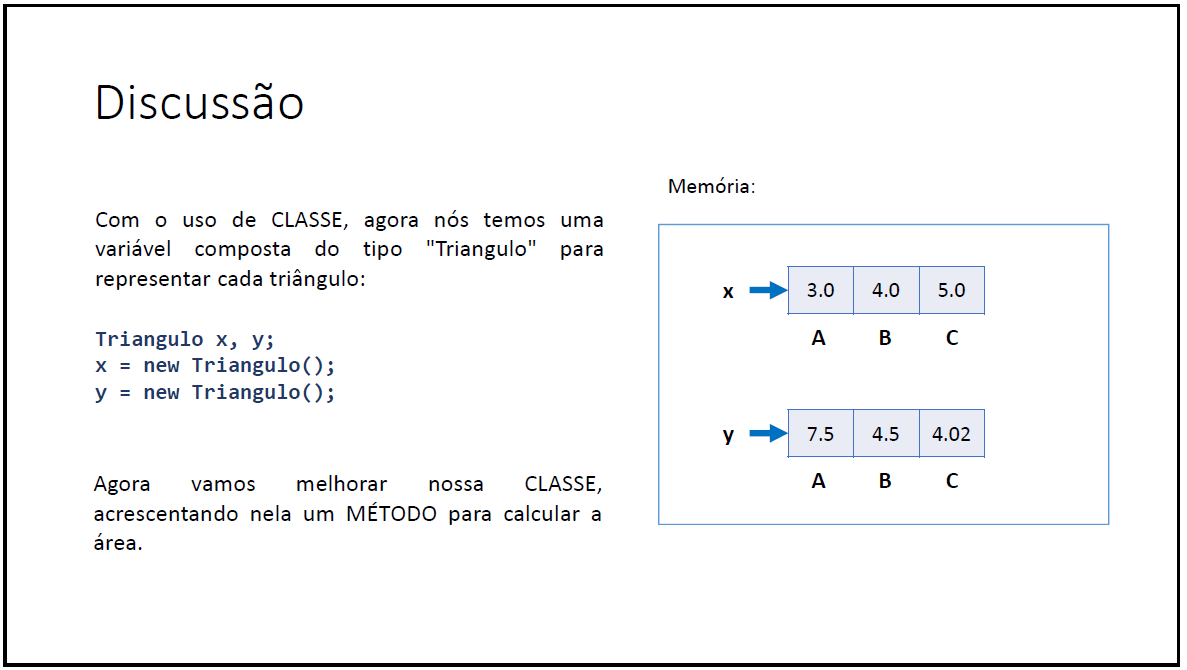


Instanciação:

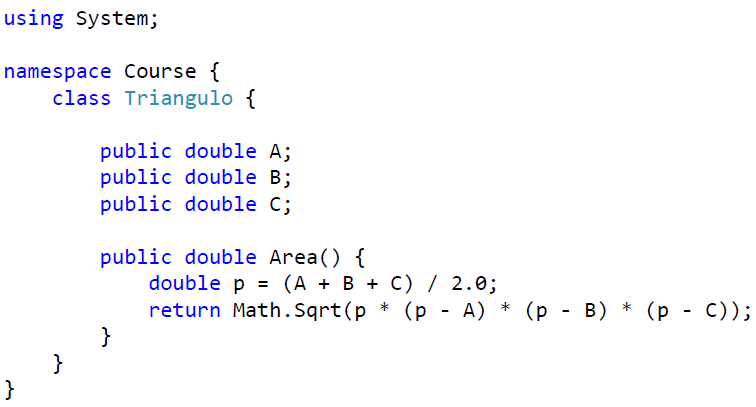




Método



Criando o método Area() dentro da classe Triangulo



Código da classe:

using System;

namespace Problema1\_Triangulo

{

class Triangulo

{

public double A;

public double B;

public double C;

public double Area()

{

double p = (A + B + C) / 2.0;

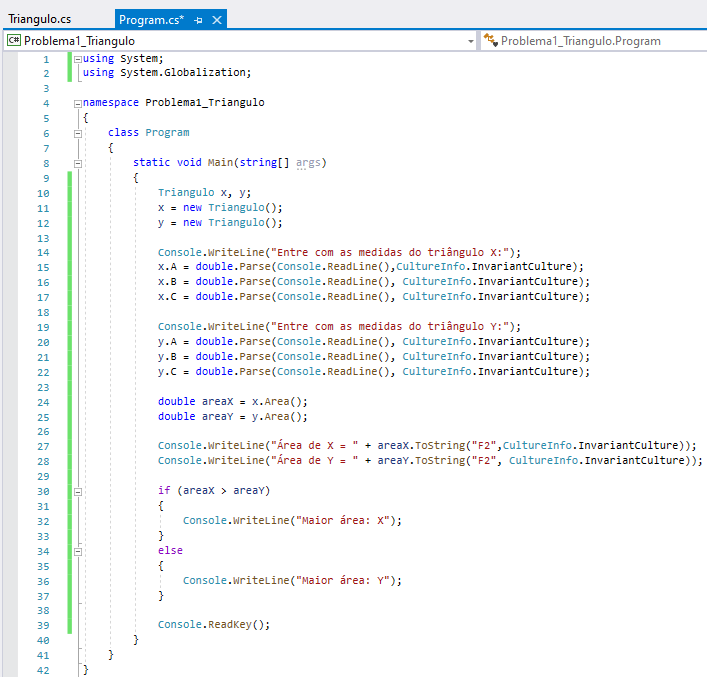
return Math.Sqrt(p \*( p - A) \* (p - B) \* (p - C));

}

}

}

Programa principal:



Código principal:

using System;

using System.Globalization;

namespace Problema1\_Triangulo

{

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("F2",CultureInfo.InvariantCulture));

Console.WriteLine("Área de Y = " + areaY.ToString("F2", CultureInfo.InvariantCulture));

if (areaX > areaY)

{

Console.WriteLine("Maior área: X");

}

else

{

Console.WriteLine("Maior área: Y");

}

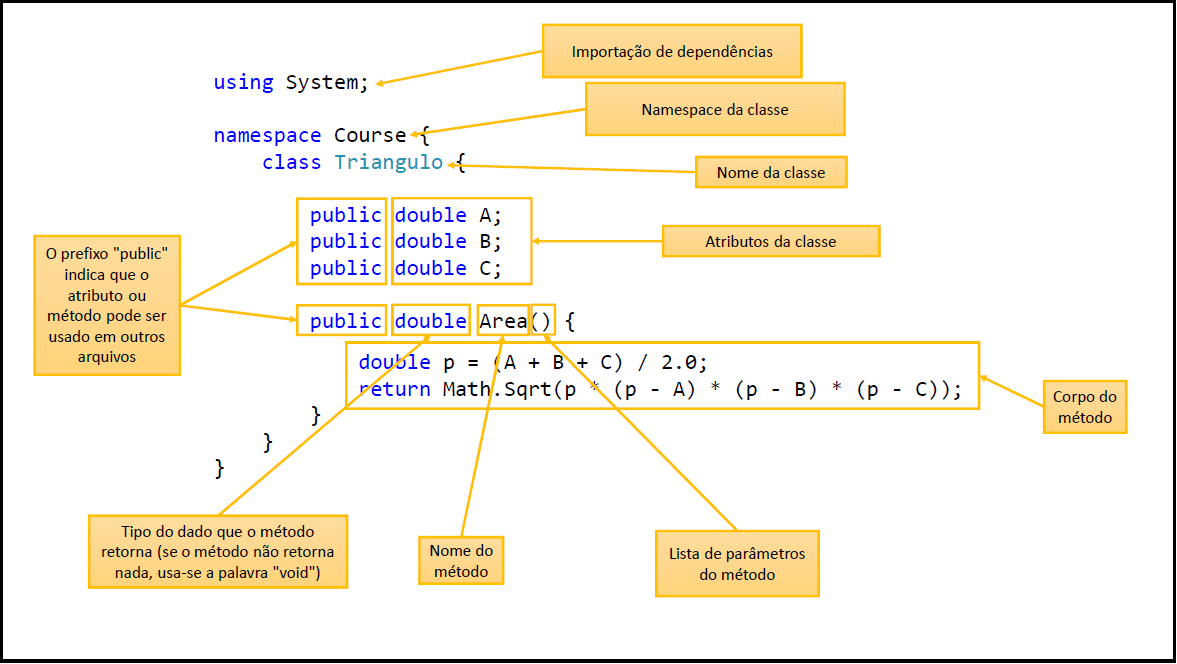
Console.ReadKey();

}

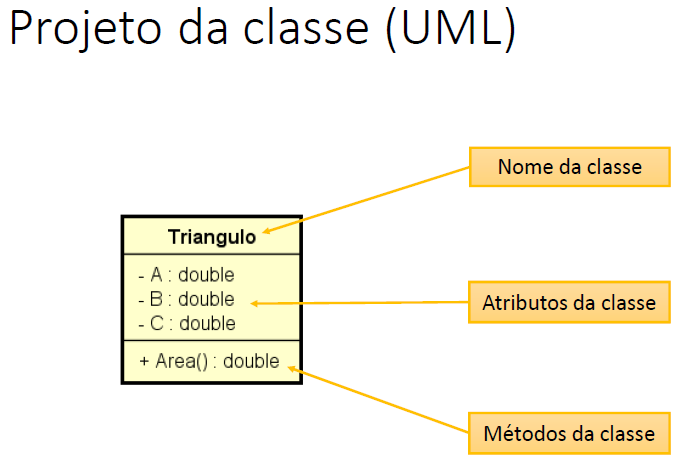
}

}

Definições:



O desenho em UML



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.

# Object e ToString

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

# Hoje veremos o ToString

Usando um problema para continuarmos os estudos sobre Classe:

# Estoque

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

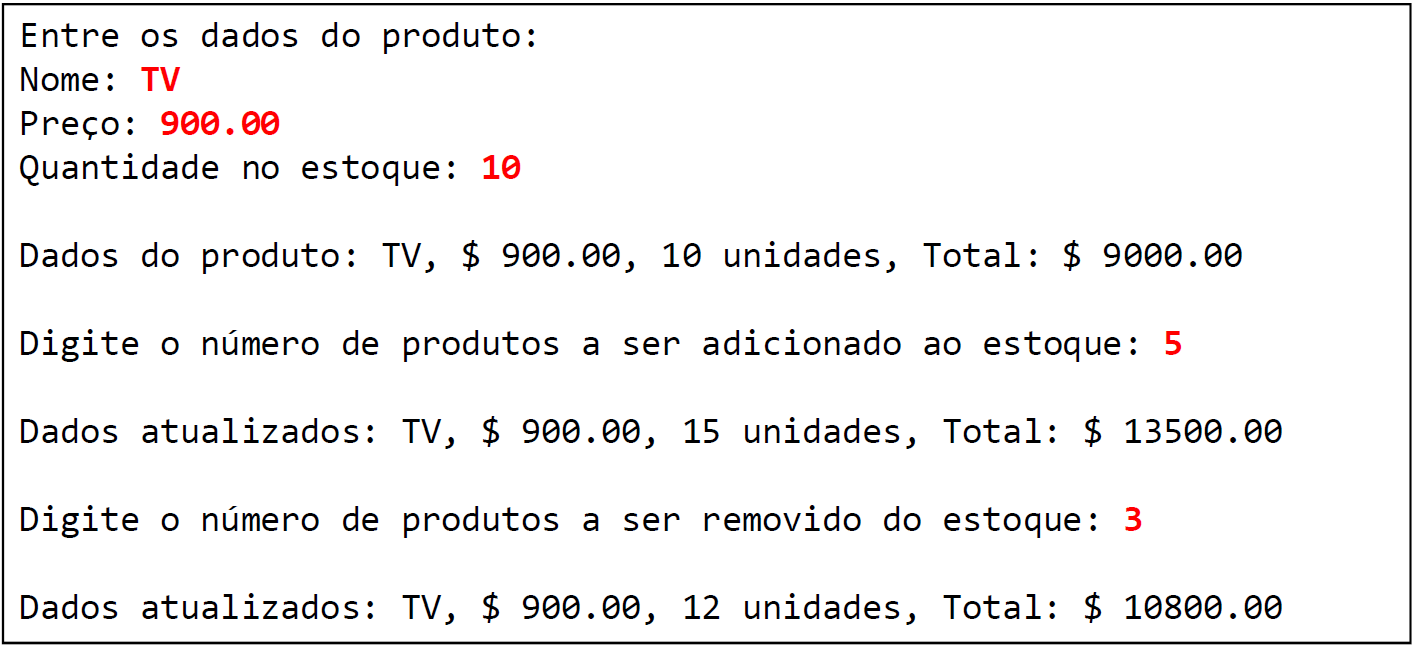
Em seguida:

* 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, vamos criar uma CLASSE conforme projeto descrito no diagrama ao lado.



Exemplo:



Ao invés de deixar fazer uma configuração de ToString por resultado. Vamos colocar isso também na Classe:

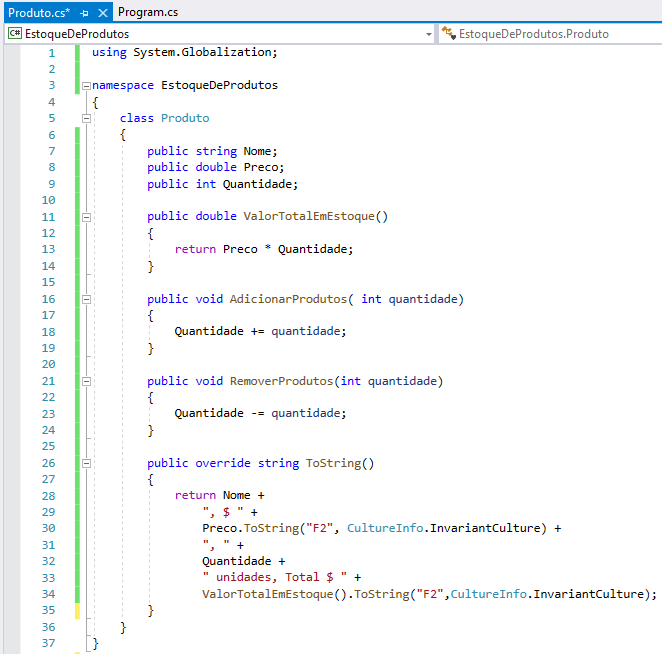
Sobreposição

Na classe escreva ( override To ) e pressione a tecla Tab ou Enter para completar a assinatura.

Override – Quer dizer que essa operação veio de outra classe.

Void – Vazio: Quer dizer que o C# vai apenas atualizar um valor

A classe Produto:



Código da classe Produto:

using System.Globalization;

namespace EstoqueDeProdutos

{

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 $ " +

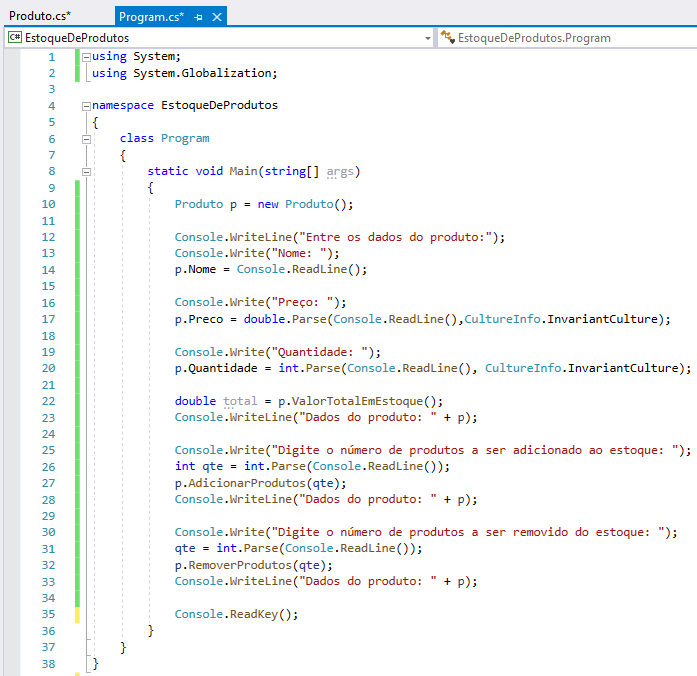
ValorTotalEmEstoque().ToString("F2",CultureInfo.InvariantCulture);

}

}

}

O Programa principal:



O código do programa principal:

using System;

using System.Globalization;

namespace EstoqueDeProdutos

{

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: ");

p.Quantidade = int.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

double total = p.ValorTotalEmEstoque();

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

Console.Write("Digite o número de produtos a ser adicionado ao estoque: ");

int qte = int.Parse(Console.ReadLine());

p.AdicionarProdutos(qte);

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

Console.Write("Digite o número de produtos a ser removido do estoque: ");

qte = int.Parse(Console.ReadLine());

p.RemoverProdutos(qte);

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

Console.ReadKey();

}

}

}