

Linguagem de Programação Visual Studio (C#)

Professor Antonio Marcos Alvarez

Proposta

Desenvolver um programa que leia (entrada via usuário) as informações relacionadas de um determinado produto:

- Descrição;
- Preço (unitário);
- Quantidade.

Em seguida apresente os detalhes deste produto:

Descrição, Quantidade, Valor e Valor (Total) em Estoque do Produto.

Modelo Exemplo

```
C:\Users\User\source\repos\EX_Classe_Object_ToString\EX_Classe_Object_ToString\bin\Debug\netcoreapp3.1\EX_Classe_Object_ToString.exe
Lançar Produtos
=====
Descrição...:TV
Preço.....:1000
Quantidade.:15
Detalhes do Produto: TV, Quantidade: 15, Valor (Unitário) R$ 1000,00, Valor Total R$ 15000,00
```

Exercícios simples para treinar uso de Classe
(Atributos e Métodos)

Passo a Passo...

Produto
- Descricao : string
- Preco : double
- Quantidade : int
+ ValorEmEstoque() : double

Object.ToString Método (System)

“Object.ToString” é o método de formatação principal no .NET Framework. Ele converte um objeto em sua representação de cadeia de caracteres para que ele seja adequado para exibição.

As implementações padrão do Object.ToString método retornam o nome totalmente qualificado do tipo do objeto.

Resumindo, Retorna uma cadeia de caracteres que representa o objeto atual (Determinado).

Reflexão (Análise)

- 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

A principio vamos analisar a Classe Objeto, como uma classe “genérica”, e podemos dizer que toda classe é um Objeto (Object).

Usando o Método “ToString”

Para isso vamos usar um recurso chamado “sobreposição” (ação de colocar (algo) por cima, reescrever, aposição, que se coloca acima, ou acrescenta ou altera).

Para que isso seja possível em nosso exemplo, vamos usar o métodos “override” (usado para estender ou modificar a implementação “abstrata” ou “virtual” de um método, propriedade, indexador ou evento herdado).

O “override”, possibilita reescrever na classe filha os métodos implementados previamente na classe pai, ou seja, uma classe filha pode redefinir métodos herdados de suas descendentes, mantendo o nome e a assinatura.

Como proceder?

Na área de edição do código (Classe “Produto”, desenvolvida no C#), vamos digitar:
... Como se fossemos inserir um novo Método ...

`override ToString` (No auto completar pressione [Enter] ou [Tab])...

Exemplo:

`override to`

 `ToString()`

`string? object.ToString()`

Returns a string that represents the current object.

```
public override string ToString()  
{  
    return base.ToString();  
}
```

Codificação (EX_Classe_Object_ToString) – Produto.cs

```
class Produto {  
    public string Descricao;  
    public double Preco;  
    public int Quantidade;  
  
    public double ValorEmEstoque() {  
        return Preco * Quantidade;  
    }  
    public override string ToString() {  
        return Descricao +  
            ", Quantidade: " + Quantidade +  
            ", Valor (Unitário) R$ " + Preco.ToString("F2") +  
            ", Valor Total R$ "+ ValorEmEstoque().ToString("F2");  
    }  
}
```


Codificação (EX_Classe_Object_ToString) – Program.cs

```
//Declarar e Instanciar a Variável - Classe "Produto"
Produto p = new Produto();
Console.Clear();
Console.WriteLine("Lançar Produtos");
Console.WriteLine("=====");
Console.Write("Descrição..:");
p.Descricao = Console.ReadLine();
Console.Write("Preço.....:");
p.Preco = double.Parse(Console.ReadLine());
Console.Write("Quantidade.:");
p.Quantidade = int.Parse(Console.ReadLine());
//Para Aprensenytar a Classe "Produto"...
//Console.WriteLine("Detalhes do Produto: "+ p.ToString());
//Não é necessário colocar o método ToString(), pois o C# já associa automaticamente ...
Console.WriteLine("Detalhes do Produto: " + p);
Console.ReadLine();
```

Implementação de Classe

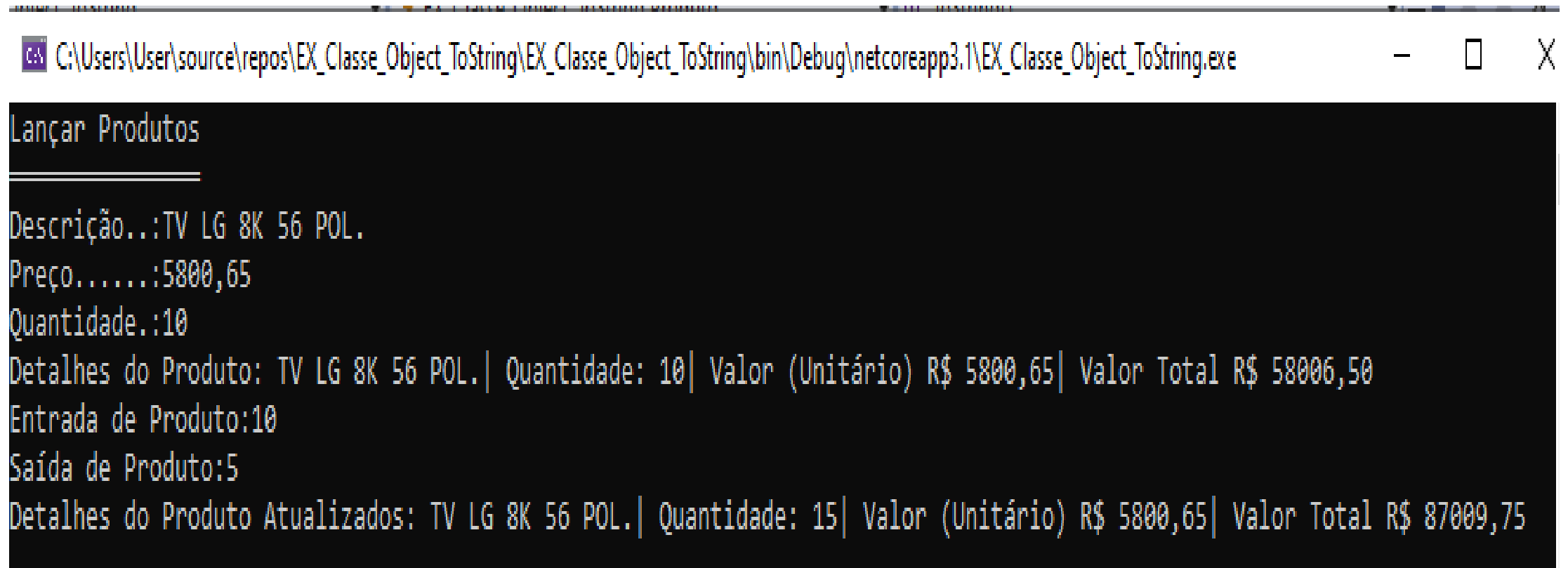
Produto
- Descricao : string - Preco : double - Quantidade : int
+ ValorEmEstoque() : double + AdicionarEstoque() : void + RemoverEstoque() : void

Agora vamos implementar dois métodos conforme diagrama de classe acima, onde:

- ***AdicionarEstoque()***, irá incrementar a quantidade de produtos no estoque;
- ***RemoverEstoque()***, irá decremenmtar a quantidade de produtos no estoque.

Após o sistema irá apresentar “Estoque do Produto Atualizado”...

Modelo (Console)



```
CS C:\Users\User\source\repos\EX_Classe_Object_ToString\EX_Classe_Object_ToString\bin\Debug\netcoreapp3.1\EX_Classe_Object_ToString.exe
Lançar Produtos
=====
Descrição...:TV LG 8K 56 POL.
Preço.....:5800,65
Quantidade.:10
Detalhes do Produto: TV LG 8K 56 POL. | Quantidade: 10 | Valor (Unitário) R$ 5800,65 | Valor Total R$ 58006,50
Entrada de Produto:10
Saída de Produto:5
Detalhes do Produto Atualizados: TV LG 8K 56 POL. | Quantidade: 15 | Valor (Unitário) R$ 5800,65 | Valor Total R$ 87009,75
```

Observe que trocamos os separadores “,” por “|”, apenas por uma melhor apresentação, somente alteramos o `Object.ToString` da nossa Classe “`Produto`”...

Implementação dos Métodos na Classe “Produtos”

```
public void AdicionarEstoque(int Qtde) {  
    Quantidade += Qtde;  
}
```

```
public void RemoverEstoque(int Qtde) {  
    Quantidade -= Qtde;  
}
```

Implementação no Código (Program.cs)...

```
Console.Write("Entrada de Produto:");
```

```
q = int.Parse(Console.ReadLine());
```

```
p.AdicionarEstoque(q);
```

```
Console.Write("Saída de Produto...:");
```

```
q = int.Parse(Console.ReadLine());
```

```
p.RemoverEstoque(q);
```

```
Console.WriteLine("Detalhes do Produto Atualizados: " + p);
```

Lembre-se

Esta atividade deverá estar relacionado ao Roteiro de Boas práticas de estudo, além de fazer parte do material de entrega como trabalho (avaliação)...

Bons estudos...