



Atributos e métodos Static

Prof. Gustavo Dibbern Piva



Static definição

- Atributos e/ou métodos static são atributos ou métodos que pertencem à classe e não a suas instâncias.

Atributos Static

- Um atributo static é criado em uma área de memória compartilhada entre todas as instâncias da classe que o possui, sendo possível que, todas as suas instâncias “vejam” o mesmo atributo(valor).



Método Static

- Método Static pode ser utilizado sem que a classe precise ser instanciada.



Observações

- Não existem Classes static.
- Não podemos utilizar atributos não static dentro de um método static pois os atributos dependem da classe estar instanciada para existir.
- Podemos criar, normalmente, variáveis e objetos dentro de métodos static.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace prjExemploStatic
{
    public class Cliente
    {
        private String nome;

        public String Nome
        {
            get { return nome; }
            set { nome = value; }
        }
        private static int nroClientes;

        public static int NroClientes
        {
            get { return Cliente.nroClientes; }
        }

        public Cliente(String nome)
        {
            this.nome = nome;
            nroClientes++;
        }
        ~Cliente()
        {
            nroClientes--;
        }
    }
}
```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace prjExemploStatic
{
    public partial class frmExemploStatic : Form
    {
        private Cliente c1, c2, c3, c4, c5;
        public frmExemploStatic()
        {
            InitializeComponent();
        }

        private void btnCria_Click(object sender, EventArgs e)
        {
            switch (Cliente.NroClientes)
            {
                case 0:
                    c1 = new Cliente("Jão");
                    break;
                case 1:
                    c2 = new Cliente("Jão");
                    break;
                case 2:
                    c3 = new Cliente("Jão");
                    break;
                case 3:
                    c4 = new Cliente("Jão");
                    break;
                case 4:
                    c5 = new Cliente("Jão");
                    break;
            }
        }
    }
}

```

```

        txtNroClientes.Text = Cliente.NroClientes.ToString();
    }

    private void btnApaga_Click(object sender, EventArgs e)
    {
        switch (Cliente.NroClientes)
        {
            case 1:
                c1 = null;
                GC.Collect();
                break;
            case 2:
                c2 = null;
                GC.Collect();
                break;
            case 3:
                c3 = null;
                GC.Collect();
                break;
            case 4:
                c4 = null;
                GC.Collect();
                break;
            case 5:
                c5 = null;
                GC.Collect();
                break;
        }
        txtNroClientes.Text = Cliente.NroClientes.ToString();
    }
}

```