Fundamentos CSharp

Introducción

# Variables

1. using System;
2. namespace FundamentosCSharp
3. {
4. internal class Program
5. {
6. static void Main(string[] args)
7. {
8. byte numero = 255; // numeros hasta 255
9. sbyte numero2 = 127; // negativos positivos hasta 127
10. int numero3 = 1; // negativos/positivos
11. uint numero4 = 2;// solo positivos
12. long numero5 = 189;
13. float numero6 = 189.1f; // 4 bytes
14. double numero7 = 189.1d; // 8 bytes
15. decimal numero8 = 189.1m; // 16 bytes
16. char caracter = 'a'; // caracteres
17. string cadena = "hola";// cadena caracteres
18. DateTime date = DateTime.Now; // Objeto \_ fecha
19. bool siOno = true; // booleano
20. //int numero9 = new int();
21. int? numero9 = null; // lo haces nullable
22. var nombre = "German"; // no es igual a JavaScript
23. //nombre = 1; // Da error, este var no es como JAvaScript, este nombre va a ser para siempre un String
24. var limite = 50; // var de tipo int
25. var persona = new { nombre = "German", apellido = "Iglesias Ramos" }; // objeto anónimo
26. Console.WriteLine(numero9.ToString());
27. Console.WriteLine(persona.nombre);
29. }
30. }
31. }

# Clases y objetos

Una captura de pantalla de una computadora

Descripción generada automáticamente

**Bebida.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FundamentosCSharp.Models // namespace : grupo

{

internal class Bebida // internal (desde el mismo dll)

{

public string Nombre { get; set; } // public o private o protected

public int Cantidad { get; set; }

public Bebida(string Nombre, int Cantidad)

{

this.Nombre = Nombre;

this.Cantidad = Cantidad;

}

public void Beberse(int CuantoBebio)

{

this.Cantidad -= CuantoBebio;

}

}

}

**Cerveza.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

/// HERENCIA

namespace FundamentosCSharp.Models

{

internal class Cerveza: Bebida

{

/// <summary>

/// Constructor base -> hereda del padre

/// </summary>

public Cerveza() : base("Cerveza", 500)

{

}

/// <summary>

/// Constructor con parametros opcionales (Nota: al final siempre)

/// </summary>

/// <param name="Cantidad"></param>

/// <param name="Nombre"></param>

public Cerveza(int Cantidad=1, string Nombre="Cerveza") : base(Nombre, Cantidad)

{

}

}

}

**Program.cs**

using FundamentosCSharp.Models;

using System;

namespace FundamentosCSharp

{

internal class Program

{

static void Main(string[] args)

{

Bebida bebida = new Bebida("Coca Cola", 1000);

bebida.Beberse(500);

Console.WriteLine(bebida.Cantidad);

Cerveza cerveza = new Cerveza(); // creacion de objetos

cerveza.Beberse(200);

Console.WriteLine(cerveza.Cantidad);

Cerveza mahou = new Cerveza(500);

mahou.Beberse(10);

Console.WriteLine(mahou.Cantidad);

}

}

}

# Arreglos y Listas

using FundamentosCSharp.Models;

using System;

using System.Collections.Generic;

namespace FundamentosCSharp

{

internal class Program

{

static void Main(string[] args)

{

// array -> se le da una dimension, mas rapido que una lista, orientado a una coleccion

int[] numeros = new int[10] { 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 }; // coleccion de numero

int numero = numeros[0]; // acceder al primer elemento

for (int i = 0; i < numeros.Length; i++) // recorrer 'for'

{

Console.WriteLine("iteracion: " + i + " - " + numeros[i]);

}

foreach (var n in numeros) // recorrer 'foreach'

{

Console.WriteLine(n);

}

Console.WriteLine(" +++++++++++++++++++++ ");

// lista -> suele implementar una interfaz, dinamica (longitud varia)

List<int> lista = new List<int>() { 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 };

lista.Add(1);

lista.Add(2);

lista.Add(3);

lista.Remove(2);

foreach (var nu in lista)

{

Console.WriteLine("elemento: " + nu);

}

Console.WriteLine(" +++++++++++++++++++++ ");

Console.ReadLine();

List<Cerveza> cervezas = new List<Cerveza>() { new Cerveza(10, "Cerveza premium") };

cervezas.Add(new Cerveza(500));

Cerveza estrella = new Cerveza(200, "Cerveza de trigo");

cervezas.Add(estrella);

foreach (var cer in cervezas)

{

Console.WriteLine("cerveza: " + cer.Nombre);

}

// cola

Queue<int> cola = new Queue<int>();

// pila

}

}

}

# Interfaces