INTRODUCCIÓN A C#

Juanra Collado

TIPOS DE DATOS

- int i = 0;
- decimal x = 0.0m;
- float f = 0.0f;
- double d = 0.0D;
- string cadena = “hello world”;
- bool flag= true;
- DateTime date = DateTime.MinValue;

CONVERSIONES DE TIPO

- https://docs.microsoft.com/es-es/dotnet/standard/base-types/conversion-tables
- Convert
  - Convert.ToString(myIntValue);
  - Convert.ToInt32(myStringValue);
  - Convert.ToBoolean(myDoubleValue)
  - Muchos más → https://docs.microsoft.com/en-us/dotnet/api/system.convert?view=netframework-4.7.2
- .ToString();
- (int)myStringValue
- ...

ARRAYS

- Declaración de arrays

```
int[] array = new int[5];
```

- Inicialización de arrays

```
int[] array2 = { 1, 3, 5, 7, 9 };
string[] weekDays2 = { "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" };
```

CONDICIONALES

- if - else

```
bool condition = true;

if (condition)
{
    Console.WriteLine("The variable is set to true.");
}
else
{
    Console.WriteLine("The variable is set to false.");
}
```

- switch

```
int caseSwitch = 1;
switch (caseSwitch)
{
    case 1:
        Console.WriteLine("Case 1");
        break;
    case 2:
        Console.WriteLine("Case 2");
        break;
    default:
        Console.WriteLine("Default case");
        break;
}
```

ITERACIÓN

- do

```
int n = 0;
do
{
    Console.WriteLine(n);
    n++;
} while (n < 5);
```

- for

```
for (int i = 0; i < 5; i++)
{
    Console.WriteLine(i);
}
```

ITERACIÓN

- foreach - in

```
var fibNumbers = new List<int> { 0, 1, 1, 2, 3, 5, 8, 13 };
int count = 0;
foreach (int element in fibNumbers)
{
    count++;
    Console.WriteLine($"Element #{count}: {element}");
}
Console.WriteLine($"Number of elements: {count}");
```

- while

```
int n = 0;
while (n < 5)
{
    Console.WriteLine(n);
    n++;
}
```

FUNCIONES / MÉTODOS

- Declaración y uso (Fijaros en la mayúscula al inicio)

```
int Suma(int x, int y)
{
    return x + y;
}
```

```
Suma(1, 2);
```

- Sobrecarga

```
int Suma(int x, int y)
{
    return x + y;
}
int Suma(int x, int y, int z)
{
    return x + y + z;
}
int Suma(params int[] numeros)
{
    int Sumatoria = 0;
    foreach(int c in numeros)
        Sumatoria += c;
    return Sumatoria;
}

Suma(1, 2); // Llamará al primer método.
Suma(1, 2, 3); // Llamará al segundo método.
Suma(1, 2, 3, 4, 5, 6) // Llamará al tercer método.
```

CLASES

- Creación

```
public class Customer
{
   // Fields, properties, methods and events go here...
}
```

```
Customer object1 = new Customer();
```

- Herencia

```
public class Manager : Employee
{
    // Employee fields, properties, methods and events are inherited
    // New Manager fields, properties, methods and events go here...
}
```

CLASES

- Propiedades - getters y setters

```
public class Fecha
{
    private int mes = 2;

    public int Mes
    {
        get
        {
            return mes;
        }
        set
        {
            if ((value > 0) && (value < 13))
            {
                mes = value;
            }
        }
    }
}
```

```
public int Mes { get; set; }
```

CLASES

- Propiedades - getters y setters

```
public class Carrots
{
    public string Name { get; set; }
}
```

```
public class Carrots
{
    public string Name { get; private set; }
}
```

```
var c = new Carrots();
c.Name = "This is a test!";
Console.WriteLine(c.Name); //outputs "This is a test!"
```

EXCEPCIONES

```
try
{
    string s = null;
    ProcessString(s);
}
// Most specific:
catch (ArgumentNullException e)
{
    Console.WriteLine("{0} First exception caught.", e);
}
// Least specific:
catch (Exception e)
{
    Console.WriteLine("{0} Second exception caught.", e);
}
```

MUCHO MÁS

- Arrays multimensionales
- Interfaces
- Polimorfismo
- ArrayList, HashMap, List
- Genéricos
- ...