ILERNA Online

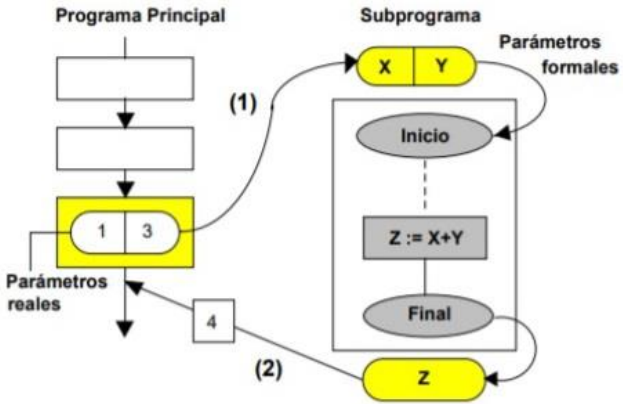
Videotutoría 9 (UF2): Paso de parámetros

Módulo 03A: Programación

RESUMEN

Parámetro real: Cuando utilizamos la función Par en el programa principal, vemos que la llamada la hacemos con una variable. Este parámetro es el que denominamos parámetro actual.

Parámetro formal: Como podéis ver en el ejemplo, hemos definido e implementado una función Par. Esta función tiene un parámetro formal que como hemos dicho anteriormente solo tiene validez dentro de la función. Si es función ponemos el tipo de esta y si es procedimiento el tipo void, que es el tipo vacío.



Public

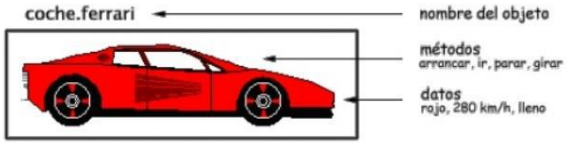
Puede obtener acceso al tipo o miembro cualquier otro código del mismo ensamblado o de otro ensamblado que haga referencia a éste.

Private

Solamente el código de la misma clase o estructura puede acceder al tipo o miembro

- **Clase**: *Coche*

- **Objeto**: *Ferrari*



ILERNA Online

Paso por valor

Cuando ejecutamos una función que tiene parámetros pasados por valor, se realiza una copia del parámetro que se ha pasado, es decir, que todas las modificaciones y/o cambios que se realicen se están haciendo en esta copia que se ha creado. El original no se modifica, de manera que no se altera su valor en la función.

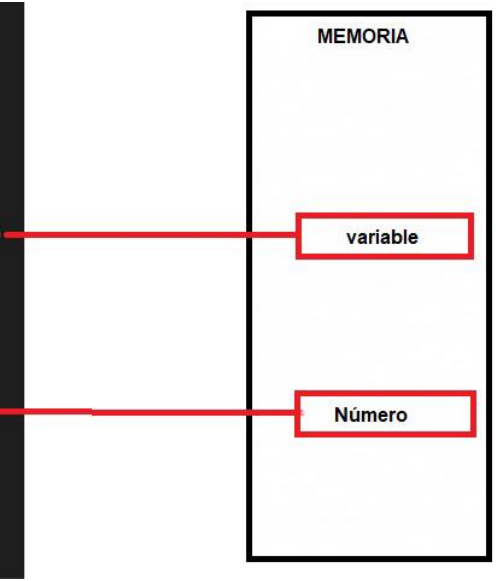
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp36
{
    class Program
    {
        static void MiMetodo(int variable)
        {
            variable = variable + 5;
        }

        static void Main(string[] args)
        {

            int numero = 1;
            MiMetodo(numero);
            Console.WriteLine(numero);
            Console.ReadKey();

        }
    }
}
```



ILERNA Online

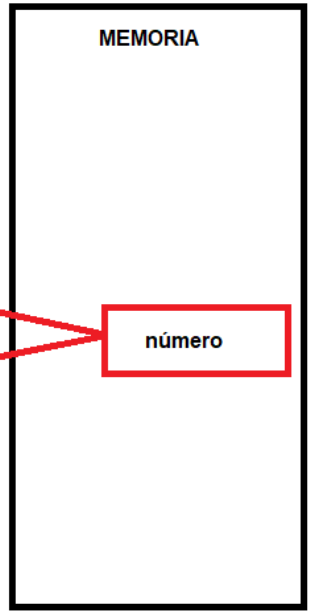
Paso por referencia

ref

out es lo msmo que ref, pero es necesario inicializar la variable dentro del método cosa que con ref no hace falta.

Sin embargo, cuando ejecutamos una función que tiene parámetros pasados por referencia, todas aquellas modificaciones que se realicen en la función van a afectar a sus parámetros, ya que se trabaja con los originales.

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   namespace ConsoleApp36
8   {
9       class Program
10      {
11          static void MiMetodo(ref int variable)
12          {
13              variable = variable + 5;
14          }
15
16          static void Main(string[] args)
17          {
18
19              int numero = 1;
20              MiMetodo(ref numero);
21              Console.WriteLine(numero);
22              Console.ReadKey();
23
24          }
25      }
26  }
```



ILERNA Online

Librerías

Concepto de librerías

- Cuando hablamos de librerías nos referimos a archivos que nos permiten llevar a cabo diferentes acciones y tareas sin necesidad de que el programador se preocupe de cómo están desarrolladas, solo debe entender cómo utilizarlas.

- Las librerías en C# permiten hacer nuestros programas más modulares y reutilizables, facilitando además crear programas con funcionalidades bastante complejas.

- double calcualo = Math.Pow(10, 2);

Podemos usarlo igualándolo a una variable para calcular una potencia.

Recursividad

Definir la recursividad como la llamada de una función a sí misma hasta que cumpla una determinada condición de salida**.**

La recursividad tiene la siguiente estructura:

• Un **caso base** que permita la finalización del programa.

• **Casos recursivos**, que son los que se van a encargar de que la función vuelva a ejecutarse, pero acercándose cada vez más al caso base.

Ejemplo

El factorial de un número n se calcula:

Debemos saber qué es el factorial de un número. Por ejemplo:

3!=3*2*1
4!=4*3*2*1
5!=5*4*3*2*1 = 5 x 4 x 3 x 2 x 1 = 120

Caso base: 1! = 1; 0! = 1

De modo que nos quedaría de la siguiente forma:

Caso base -> Si n=1, factorial(1) devuelve 1.
Caso genérico -> Si n>1, factorial(n) = n*factorial(n-1).

Ejemplo

```
static int factorial(int n)
{
if (n>1)

return n*factorial(n-1);

Else

return 1;
}
```

Preguntas UF2

El factorial de un número se puede calcular de forma recursiva. ¿Qué es la recursividad?

a) Un método exclusivo de C#
b) Una función recursiva es aquella que se "llama a ella misma”
c) No se puede aplicar en la programación orientada a objetos
d) Es un tipo de dato

Una ventaja de la programación modular es la reutilización de código
a) Verdadero
b) Falso

Preguntas UF2

El factorial de un número se puede calcular de forma recursiva. ¿Qué es la recursividad?

a) Un método exclusivo de C#
b) Una función recursiva es aquella que se "llama a ella misma”
c) No se puede aplicar en la programación orientada a objetos
d) Es un tipo de dato

Una ventaja de la programación modular es la reutilización de código
a) Verdadero
b) Falso

Preguntas UF2

Cuando instanciamos una clase es crear un objeto y por lo tanto darle todos los métodos y todos los atributos (variables).

De las siguientes afirmaciones referidas a los métodos, señala cuál es la correcta:

a. Los atributos de instancia junto con los métodos de instancia reciben el nombre de miembros de clase.
b. Cualquier método puede no devolver un valor, en cuyo caso se indica sin utilizar ninguna palabra reservada.
c. La lista de parámetros de un método debe coincidir con la lista de argumentos con los que es llamado.
d. Todas son correctas.

Preguntas UF2

De las siguientes afirmaciones referidas a los métodos, señala cuál es la correcta:

a. Los atributos de instancia junto con los métodos de instancia reciben el nombre de miembros de clase.
b. Cualquier método puede no devolver un valor, en cuyo caso se indica sin utilizar ninguna palabra reservada.
c. La lista de parámetros de un método debe coincidir con la lista de argumentos con los que es llamado.*
d. Todas son correctas.

* Efectivamente, la respuesta sería la c porque no sería correcto no utilizar ninguna palabra reservada.

Preguntas UF2

Una variable local almacena un valor temporal y se declara dentro de

a. Una clase.
b. Un método.
c. Un tipo de datos.
d. Un bloque de código entre corchetes.

Preguntas UF2

Una variable local almacena un valor temporal y se declara dentro de

a. Una clase.
b. Un método.
c. Un tipo de datos.
d. Un bloque de código entre corchetes.

Preguntas UF2

¿Es posible utilizar una return en cualquier punto de un método, con lo que éste finalizará en el lugar donde se encuentre dicho return.?

a. No, siempre debe ir al final del método.
b. Sí y hará que éste finalice en el lugar donde se encuentre el return.
c. Sí y podemos añadir tantos return como necesitemos.
d. No, return sólo se incluirá en aquellos métodos que devuelven void.

Preguntas UF2

¿Es posible utilizar una return en cualquier punto de un método, con lo que éste finalizará en el lugar donde se encuentre dicho return.?

a. No, siempre debe ir al final del método.
b. Sí y hará que éste finalice en el lugar donde se encuentre el return.
c. Sí y podemos añadir tantos return como necesitemos.
d. No, return sólo se incluirá en aquellos métodos que devuelven void.

Preguntas UF2

La programación modular consiste en…

a. un número de funciones mayor que de procedimientos
b. Crear constantes en el código
c. Realizar un número de procedimientos mayor que el de funciones
d. Dividir el problema original en diversos subproblemas

Preguntas UF2

La programación modular consiste en…

a. un número de funciones mayor que de procedimientos
b. Crear constantes en el código
c. Realizar un número de procedimientos mayor que el de funciones
d. Dividir el problema original en diversos subproblemas

Preguntas UF2

```
class funcexer11
{
static void Main()
{
        decimal f;
        Console.Write("--------------\n");
        Console.Write("--------------\n");
        Console.Write("Input a number : ");
        int num= Convert.ToInt32(Console.ReadLine());
        f = Funcion(num);
        Console.WriteLine("The result of {0}! is  {1}", num, f);
}
static decimal Funcion(int n1)
    {
      if (n1 == 0)
        {
         return 1;
        }
       else
        {
          return n1 * Funcion(n1 - 1);
        }
    }
}
```

¿Es correcto el código?

a. No. No existen funciones de tipo decimal
b. No porque hay dos returns
c. No. No existe la sentencia Convert
d. Sí. El código es correcto

ILERNA Online

Preguntas UF2

```
class funcexer11
{
static void Main()
{
        decimal f;
        Console.Write("--------------\n");
        Console.Write("--------------\n");
        Console.Write("Input a number : ");
        int num= Convert.ToInt32(Console.ReadLine());
        f = Funcion(num);
        Console.WriteLine("The result of {0}! is  {1}", num, f);
}
static decimal Funcion(int n1)
    {
      if (n1 == 0)
        {
         return 1;
        }
       else
        {
          return n1 * Funcion(n1 - 1);
        }
    }
}
```

¿Es correcto el código?

a. No. No existen funciones de tipo decimal
b. No porque hay dos returns
c. No. No existe la sentencia Convert
d. Sí. El código es correcto

ILERNA Online

Preguntas UF2

Cuando hablamos de librerías nos referimos a...

a) A un tipo de función
b) A un tipo de argumento que pasa el main()
c) A un conjunto de parámetros pasados por valor
d) Archivos que nos permiten realizar diferentes acciones sin necesidad que el programador se preocupe de su desarrollo

Preguntas UF2

Cuando hablamos de librerías nos referimos a…

a) A un tipo de función
b) A un tipo de argumento que pasa el main()
c) A un conjunto de parámetros pasados por valor
d) Archivos que nos permiten realizar diferentes acciones sin necesidad que el programador se preocupe de su desarrollo

Preguntas UF2

Cuando hablamos de librerías nos referimos a…

a) A un tipo de función
b) A un tipo de argumento que pasa el main()
c) A un conjunto de parámetros pasados por valor
d) Archivos que nos permiten realizar diferentes acciones sin necesidad que el programador se preocupe de su desarrollo

Preguntas UF2

```
1  using System;
2  public class miFuncion
3  {
4  public static bool EstaEsMiFuncion(int n)
5  {
6  for (int i = 2; i < n; i++)
7  if (n %i == 0)
8  return false;
9
10 return true;
11 }
12
13 public static void Main()
14 {
15 if (EstaEsMiFuncion(127))
16 Console.WriteLine("Salida por pantalla1");
17 else
18 Console.WriteLine("Salida por pantalla2");
19 }
20 }
```

¿Es correcto este código al tener dos return la función?

a) Sí porque el return de la línea 8, está dentro del bucle if y el return de la línea 10, fuera del bucle if

b) No. No puede tener dos return en una función

c) No. Los dos return deberían tener la misma salida (true o false)

d) Ninguna respuesta es correcta

ILERNA Online

Preguntas UF2

```
1  using System;
2  public class miFuncion
3  {
4  public static bool EstaEsMiFuncion(int n)
5  {
6  for (int i = 2; i < n; i++)
7  if (n %i == 0)
8  return false;
9
10 return true;
11 }
12
13 public static void Main()
14 {
15 if (EstaEsMiFuncion(127))
16 Console.WriteLine("Salida por pantalla1");
17 else
18 Console.WriteLine("Salida por pantalla2");
19 }
20 }
```

¿Es correcto este código al tener dos return la función?

Al no haber corchetes es una única línea de código:

a) Sí porque el return de la línea 8, está dentro del bucle if y el return de la línea 10, fuera del bucle if

b) No. No puede tener dos return en una función

c) No. Los dos return deberían tener la misma salida (true o false)

d) Ninguna respuesta es correcta

ILERNA Online

Preguntas UF2

```
1  using System;
2  public class miFuncion
3  {
4  public static bool EstaEsMiFuncion(int n)
5  {
6  for (int i = 2; i < n; i++)
7  if (n %i == 0)
8  return false;
9
10 return true;
11 }
12
13 public static void Main()
14 {
15 if (EstaEsMiFuncion(127))
16 Console.WriteLine("Salida por pantalla1");
17 else
18 Console.WriteLine("Salida por pantalla2");
19 }
20 }
```

La función “EstaEsMiFunción”

a) Es de tipo int
b) Es de tipo double
c) Es de tipo booleano
d) Es de tipo void

ILERNA Online

Preguntas UF2

```
1  using System;
2  public class miFuncion
3  {
4  public static bool EstaEsMiFuncion(int n)
5  {
6  for (int i = 2; i < n; i++)
7  if (n %i == 0)
8  return false;
9
10 return true;
11 }
12
13 public static void Main()
14 {
15 if (EstaEsMiFuncion(127))
16 Console.WriteLine("Salida por pantalla1");
17 else
18 Console.WriteLine("Salida por pantalla2");
19 }
20 }
```

La función “EstaEsMiFunción”

a) Es de tipo int
b) Es de tipo double
c) Es de tipo booleano
d) Es de tipo void

ILERNA Online

Preguntas UF2

```
1  using System;
2  public class miFuncion
3  {
4  public static bool EstaEsMiFuncion(int n)
5  {
6  for (int i = 2; i < n; i++)
7  if (n %i == 0)
8  return false;
9
10 return true;
11 }
12
13 public static void Main()
14 {
15 if (EstaEsMiFuncion(127))
16 Console.WriteLine("Salida por pantalla1");
17 else
18 Console.WriteLine("Salida por pantalla2");
19 }
20 }
```

La línea 7:

a) N es un entero del cual se hace el tanto por ciento que indique i (p.ej: n=8 i=10%)

b) Se realiza una división entre las variables n e i comprobando el resto de esa división

c) Se realiza una división entre las variables n e i comprobando el cociente de esa división

d) Ninguna respuesta es correcta

ILERNA Online

Preguntas UF2

```
1  using System;
2  public class miFuncion
3  {
4  public static bool EstaEsMiFuncion(int n)
5  {
6  for (int i = 2; i < n; i++)
7  if (n %i == 0)
8  return false;
9
10 return true;
11 }
12
13 public static void Main()
14 {
15 if (EstaEsMiFuncion(127))
16 Console.WriteLine("Salida por pantalla1");
17 else
18 Console.WriteLine("Salida por pantalla2");
19 }
20 }
```

La línea 7:

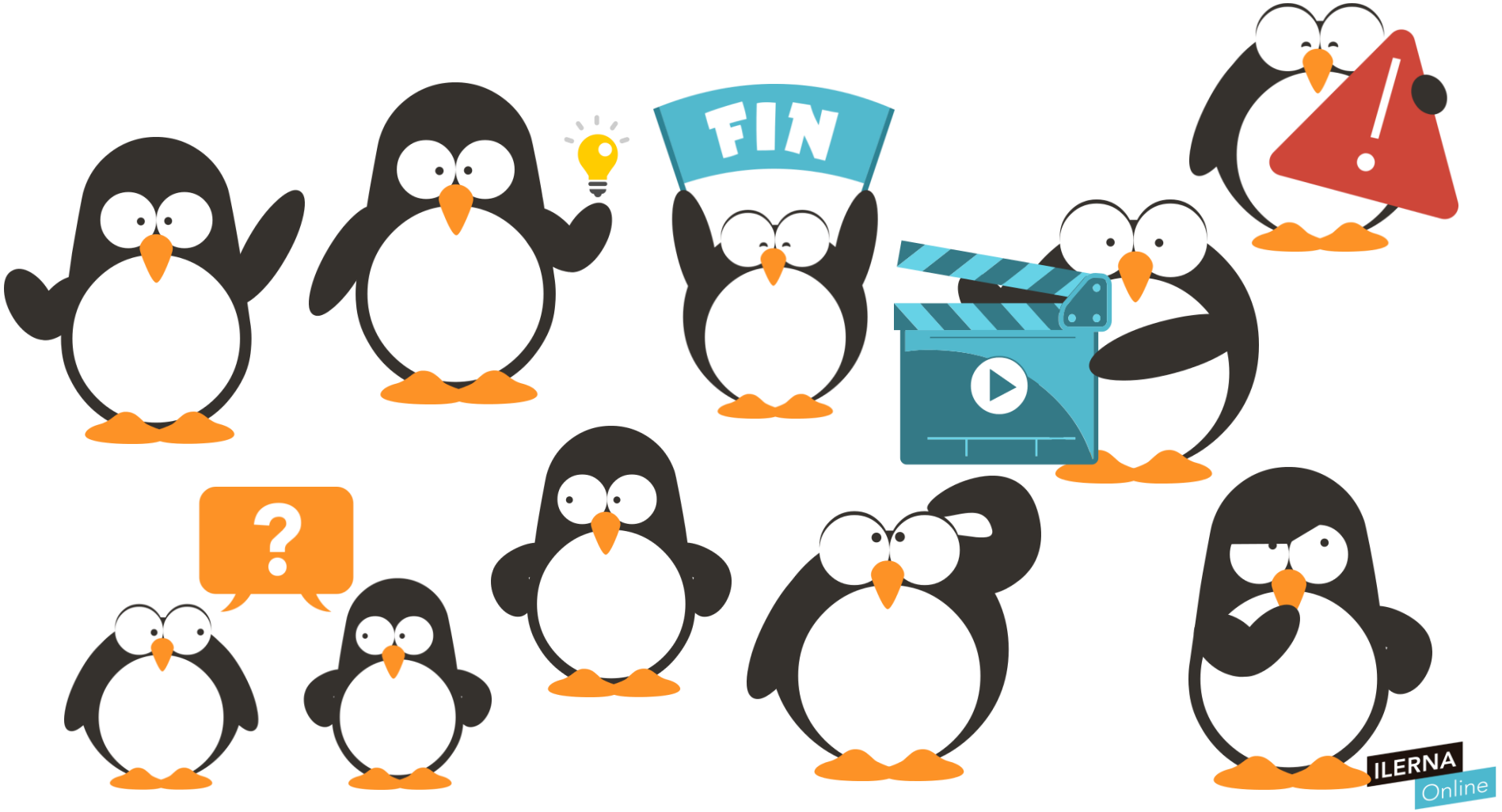
a) N es un entero del cual se hace el tanto por ciento que indique i (p.ej: n=8 i=10%)

b) Se realiza una división entre las variables n e i comprobando el resto de esa división

c) Se realiza una división entre las variables n e i comprobando el cociente de esa división

d) Ninguna respuesta es correcta

ILERNA Online


FIN
?
ILERNA
Online