



PROGRAMACIÓN EN C# .NET

Módulo 5 .- Subrutinas y Métodos

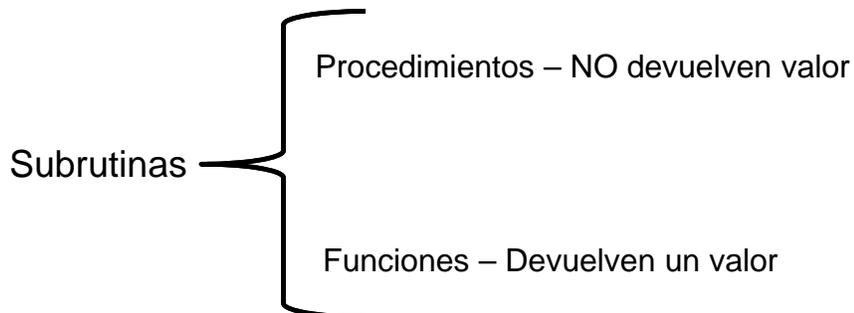
Ing. Bruno López Takeyas
Instituto Tecnológico de Nuevo Laredo

0

1

MÉTODOS

- En C# las subrutinas se conocen como métodos y se codifican como parte de una clase



2

MODIFICADORES DE ACCESO

Modificador de acceso	Descripción
public	Es accesible desde fuera de la clase
protected	No es visible desde fuera de la clase y se puede acceder sólo desde las clases derivadas
private	No se pueden acceder desde fuera del ámbito de la clase que lo define. Por lo tanto, ni siquiera las clases derivadas tienen acceso
internal	Sólo es visible desde la unidad de compilación actual. Este modificador de acceso crea un híbrido de acceso public y protected dependiendo de dónde resida el código

3

MIEMBROS ESTÁTICOS Y MIEMBROS DE INSTANCIA

- **Miembro estático (static):** Sólo se crea una copia del miembro de la clase. Se crea cuando se carga la aplicación que contiene la clase y existe mientras se ejecute la aplicación
- **Miembro de instancia:** Se crea por default. Se crea una copia para cada instancia de la clase.

4

SINTAXIS DE SUBROUTINAS

```
< tipoValorDevuelto > < nombreMétodo > (< parámetros >)  
{  
    < cuerpo >  
}
```

void significa que NO devuelve valor (procedimiento)

■ Ejemplo:

```
void Saludo( )  
{  
    Console.WriteLine("Hola");  
}
```

5

LLAMADAS DE SUBROUTINAS

```
class Program  
{  
    static void Main(string[] args)  
    {  
        Subrutina(); // Llamada  
    }  
  
    static void Subrutina( )  
    {  
        . . . // Codificación  
    }  
}
```

6

LLAMADAS DE PROCEDIMIENTOS

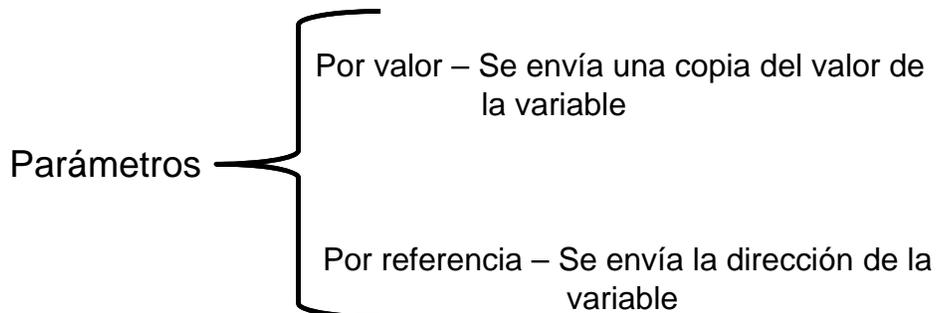
```
class Program
{
    static void Main(string[] args)
    {
        Procedimiento(); // Llamada
    }

    static void Procedimiento( )
    {
        Console.WriteLine("Tec Laredo");
        return();// Fin del Procedimiento
    }
}
```

El Procedimiento NO devuelve valor

7

ENVÍO DE PARÁMETROS O ARGUMENTOS



8

ENVÍO DE PARÁMETROS POR VALOR

```
class Program
{
    static void Main(string[] args)
    {
        int x=10;
        Subrutina( x ); // Se envia el valor de x
        Console.WriteLine("x="+x.ToString()); // x=10
    }
    static void Subrutina(int y)
    {
        y+=5;
        Console.WriteLine("y="+y.ToString()); // y=15
    }
}
```

9

ENVÍO DE PARÁMETROS POR REFERENCIA

```
static void Main(string[] args)
{
    int x = 10;
    Rutina(ref x); // Se envia la referencia de x
    Console.WriteLine("x=" + x); // x=15
    Console.ReadKey();
}
static void Rutina(ref int y)
{
    y += 5;
    Console.WriteLine("\n\ny=" + y); // y=15
}
```

10

LLAMADAS DE FUNCIONES

```
static void Main(string[] args)
{
    double Radio = 10, Area;
    Area = Funcion(Radio);
    Console.WriteLine("Area=" + Area);
    Console.ReadKey();
}
```

Variable receptora

Argumento enviado a la Función

```
static double Funcion(double r)
{
    return (Math.PI * r * r);
}
```

Valor devuelto por la Función