Sobrecarga de Métodos

* Los métodos no pueden tener el mismo nombre que otros elementos en una misma clase (atributos, propiedades, etc.).
* Sin embargo, dos o más métodos en una clase sí pueden compartir el mismo nombre.
* A esto se le da el nombre de **sobrecarga**.

Métodos Sobrecargados

* Los métodos se sobrecargan cambiando el número, el tipo y el orden de los parámetros (se cambia la firma del método).
* El compilador de C# distingue métodos sobrecargados comparando las listas de parámetros.

Ejemplo – Cantidad Parámetros

**class Sobrecarga {**

**static int Sumar (int a, int b) { //método con dos parámetros**

**return a + b;**

**}**

**static int Sumar (int a, int b, int c) { //método con tres parámetros**

**return a + b + c;**

**}**

**static void Main( ) {**

**Console.WriteLine(Sumar(1, 2) + Sumar(1, 2, 3));**

**}**

**}**

Ejemplo – Tipo de Parámetros

**class Sobrecarga {**

**static int Sumar (int a, int b) { //método con dos parámetros**

**return a + b; //de tipo int**

**}**

**static int Sumar (float a, float b) { //método con dos parámetros**

**return (int) (a + b); //de tipo float**

**}**

**static void Main( ) {**

**Console.WriteLine(Sumar(1, 2) + Sumar(1.0, 2.0));**

**}**

**}**

Firmas de Métodos

-Las firmas de los métodos deben ser únicas dentro de una clase.

-Forman la definición de la firma de un método:

* + Nombre del método.
  + Tipo de parámetros.
  + Cantidad de parámetros.
  + Modificador de parámetro (**out o ref**)

-No afectan la firma de un método:

* + Nombres de parámetros.
  + Tipo de retorno del método.

Ejemplo

**class Sobrecarga**

**{**

**static int Metodo1 (int n)**

**{ … } //Error misma Firma**

**static string Metodo1 (int n)**

**{ … }**

**static int Metodo2(int n)**

**{ … } //Error misma Firma**

**static int Metodo2(int x)**

**{ … }**

**}**

Conviene el Uso...

* Si hay métodos similares que requieren parámetros diferentes.
* Si se quiere añadir funcionalidad al código existente.

No hay que abusar ya que... Son difíciles de depurar.

* Son difíciles de mantener.

Sobrecarga de Constructores

* Al igual que los métodos, los constructores también se pueden sobrecargar.
* Las normas para hacerlo son las mismas.
* Se suele hacer cuando se quiere dar la posibilidad de instanciar objetos de formas diferentes.

**class Sobrecarga {**

**public Sobrecarga()**

**{ … }**

**public Sobrecarga(int n)**

**{ … }**

**public Sobrecarga(int n, int x)**

**{ … }**

**public Sobrecarga(string mensaje)**

**{ … }**

**}**