Introducción a la programación con C#

Métodos

Un poco de repaso...

- ¿Qué nos permiten hacer las **collection** en .NET?
- ¿Cuál es la diferencia entre **List** y **Dictionary**?
- En **List<T>**... ¿Qué vendría a ser **<T>**?
- Nombrar algunas de las principales operaciones que podemos hacer con un **List**

¿Qué vamos a ver hoy?

- Métodos, sobrecarga de métodos
- Constructores
- Pasaje o copia por valor y por referencia
- Pasaje de valor por **ref** o **out**
- Métodos estáticos, ¿para qué nos sirven?

Entonces... ¿Qué es un método?

Un método es una **subrutina** cuyo código está definido en una **clase.**
Puede pertenecer tanto a una **clase**, como es el caso de los métodos de clase o estáticos, como a un **objeto**, como es el caso de los métodos de instancia.

Se presenta como un **subalgoritmo** que forma parte de un **algoritmo principal**, el cual permite resolver una tarea específica.

¿Qué es la firma del método?

Es la combinación del nombre y los tipos de los parámetros que recibe.

```
List<Chair> GetChairsInStockByColor(string color){
}

List<Chair> GetChairsInStockByColor(string color, int code){
}

List<Chair> GetChairsInStockByColorAndCode(string color, int code){
}
```

Hablamos de sobre carga de método cuando..

Creamos varios métodos con el mismo nombre pero con diferente lista de tipos de parámetros.

```
List<Chair> GetChairsInStock(){
}

List<Chair> GetChairsInStockByColor(string color){
}

List<Chair> GetChairsInStockByColor(string color, int code){
}
```

¿Se puede hacer sobre carga del constructor?

```
public class Chair{

    public Chair(){
    }

    public Chair(string color){
    }

    public Chair(int code){
    }
}
```

Copia por referencia y por valor

En C#, los argumentos se pueden pasar a parámetros por valor o por referencia.

El paso de parámetros por referencia permite a los miembros de funciones, métodos, propiedades, indexadores, operadores y constructores cambiar el valor de los parámetros y hacer que ese cambio persista en el entorno de la llamada.

Para pasar un parámetro por referencia con la intención de cambiar el valor, use la palabra clave **ref** o **out**.

Para pasar un parámetro por referencia con la intención de evitar la copia pero no modificar el valor, use el modificador **in**.

Documentación oficial

¡¡A trabajar!!