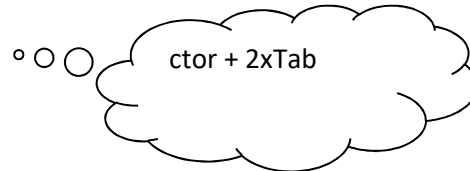


## Teoria

### Costruttori

Quando si vuole istanziare una variabile di classe e si desidera inizializzare alcune variabile della stessa, si definisce un metodo che verrà effettuato automaticamente all'atto della creazione. Questo metodo particolare è definito normalmente pubblico e ha il nome della classe stessa. Per convenzione nostra, metteremo questo metodo nella region costruttori.

```
#region =03=== costruttori =====
public Clown()
{
    Altezza = 2;
    Nome = "Dimitri";
}
public Clown(string nome, double altezza)
{
    Nome = nome;
    Altezza = altezza;
}
#endregion
```



Oltre a definire costruttori di default (non obbligatori), si possono aggiungere altri costruttori con parametri. Qualora fosse richiesto di far eseguire un costruttore particolare, è possibile utilizzando la seguente forma:

```
public Clown() : this("Sconosciuto", 2)
{
}
```

Con questo esempio, quando si usa il costruttore di default, viene automaticamente richiamato il costruttore che richiede 2 parametri passando dei valori di default (identico al primo caso).

In C# è possibile definire i membri della classe utilizzando un'ulteriore forma dove i valori vengono passati all'atto di costruzione con le parentesi graffe e poi assegnando i valori (VS propone automaticamente i possibili membri).

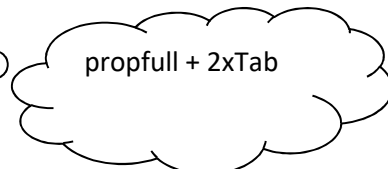
```
Clown buffo = new Clown() { Nome = "Buffo", Altezza = 1.71 };
```

### Proprietà

A differenza di altri linguaggi, viene suggerito l'uso di proprietà al posto di variabili pubbliche. Al posto di utilizzare una variabile, viene proposto l'uso di una variabile privata (scritta in minuscolo) e poi, in contemporanea, il costrutto della proprietà che congloba sia un setter che un getter, con la possibilità di definire anche se in sola lettura/scrittura. Queste proprietà vengono inserite nella region membri e proprietà (per nostra convenzione). Un ulteriore vantaggio è la possibilità di definire dei controlli nel setter in modo che il dato venga sempre convalidato senza che l'utilizzatore della classe se ne debba preoccupare.

```
#region =02=== membri & proprietà =====
//public double Peso;
private double _peso;
public double Peso
{
    get { return _peso; }
    set { _peso = value; }
}

// propfull + 2xTab (e spostamento ancora con Tab)
private string _nome;
public string Nome
{
    get { return _nome; }
```



```
set
{
    if (string.IsNullOrEmpty(value))
        value = "<sconosciuto>";
    _nome = value;
}

private double _altezza;
public double Altezza
{
    get { return _altezza; }
    set
    {
        if (value < 0)
            value = 1;
        _altezza = value;
    }
}
#endregion
```

