

Uso delle funzioni Single e First di LINQ

La libreria LINQ prevede anche delle funzioni che restituiscono un singolo oggetto prelevato dalla collezione di partenza. Si tratta delle funzioni Single e First.

La funzione **Single** si aspetta di trovare **esattamente un oggetto** che soddisfa la condizione impostata, e genera un **errore runtime** nel caso in cui ci sia più di un oggetto che soddisfa tale condizione.

La logica è quella di usare come criterio di ricerca un attributo identificativo dell'oggetto stesso, come potrebbero essere il numero di targa di un veicolo oppure il numero di matricola di uno studente.

La funzione **First**, invece, restituisce **il primo oggetto** della collezione che verifica la condizione impostata.

Entrambe le suddette funzioni Single() e First() producono un **errore runtime** nel caso in cui non ci sia **nessun oggetto** che soddisfa al condizione impostata.

Esempi con una lista di persone:

```
List<Persona> lista = ...;
// si suppone che la lista non sia vuota! altrimenti c'è un runtime error
Persona p = lista.Single(x=> x.Nome=="luigi neri"); // si suppone che Nome sia univoco
Persona p = lista.First(x=> x.Età>=18); // si suppone che ci sia almeno un maggiorenne
```

Per evitare errori runtime conviene innanzitutto controllare se l'oggetto cercato esiste:

```
Persona p = null;
if (lista.Any(x => x.Nome == "luigi neri"))
{
    p = lista.Single(x=>x.Nome=="luigi neri");
}
```

In alternativa a quanto appena mostrato, si possono utilizzare le analoghe funzioni **SingleOrDefault** e **FirstOrDefault**, che evitano di incorrere in un errore runtime nel caso in cui non ci siano elementi che soddisfano la condizione richiesta.

- **SingleOrDefault** ritorna null se non c'è nessuno, da comunque errore se ce n'è più di uno
- **FirstOrDefault** ritorna null se non c'è nessuno

Esempio: Aumentare lo stipendio del Professore di nome Bandiera contenuto in una lista di Persone, sapendo che Professore è un tipo che deriva da Persona.

```
List<Persona> elenco = .... // creazione e caricamento dell'elenco di persone
Professore p = elenco.OfType<Professore>().SingleOrDefault(x=>x.Nome == "Bandiera");
if (p != null)
{
    p.AumentaStipendio(1000);
}
```

La funzione `OfType<Professore>()` oltre a selezionare il sottoinsieme dei professori contenuti nella lista, effettua anche la conversione formale del risultato al tipo `Professore` (casting).