BCSCHOOL

Empowering Digital Skills For The Jobs Of The Future

by

BC Soft
consulenza sistemi informativi

AiRED
Artificial Intelligence Ricerca e Sviluppo

Academy .NET

Liste

Sommario

- List <T> - Panoramica
- List<T> Data Structure
- Leggere le liste
- Stampare le liste
- Ordinare le liste

List <T> - Panoramica

List<T> contiene un elenco di elementi di qualsiasi tipo **T**:

```
var names = new List<string>(); // Create a list of strings
names.Add("Peter");
names.Add("Maria");
names.Add("George");
foreach (var name in names)
  Console.WriteLine(name);
names.Remove("Maria");
Console.WriteLine(
  String.Join(", ", names));
```

```
var nums = new List<int> {
  10, 20, 30, 40, 50, 60};
nums.RemoveAt(2);
nums.Add(100);
nums.Insert(0, -100);
Console.WriteLine(
  String.Join(", ", nums));
```

List<T> Data Structure

- **List <T>** contiene un elenco di elementi (come array, ma estendibile)
- Fornisce operazioni per aggiungere / inserire / rimuovere / trovare elementi:

  Add (element): aggiunge un elemento a **List<T>**

  Count - conteggio di elementi in **List<T>**

  Remove (element) - rimuove un elemento (restituisce **true/false**)

  RemoveAt (index) - rimuove l'elemento nel corrispettivo indice

  Insert (index, element): inserisce un elemento in una determinata posizione

  Contains (element): determina se un elemento è nell'elenco

  Sort (): ordina l'array in ordine crescente

Lettura liste dalla Console

Lettura da Console della lunghezza della lista:

```
int n = int.Parse(Console.ReadLine());
```

Creazione di un elenco di dimensione n e lettura dei suoi elementi:

```
List<int> list = new List<int>();

for (int i = 0; i < n; i++)
{
  list.Add(int.Parse(Console.ReadLine()));
}
```

Leggere valori di una Lista da singola riga

Le **liste** possono essere lette da una singola riga di valori separati da spazi:

```
2 8 30 25 40 72 -2 44 56
```

```
string values = Console.ReadLine();
List<string> items = values.Split(' ')
.ToList();
List<int> nums = new List<int>();
for (int i = 0; i < items.Count; i++)
  nums.Add(int.Parse(items[i]));
```

Converte la collezione in Lista

```
var items = Console.ReadLine().Split(' ')

.Select(int.Parse).ToList();
```

In una sola volta

Stampare lista in Console

- Stampa di un elenco utilizzando un **ciclo for**:

```
List<string> list = new List<string>() {
  "one", "two", "three", "four", "five", "six"};
for (int index = 0; index < list.Count; index++)
  Console.WriteLine("arr[{0}] = {1}", index, list[index]);
```

- Stampa di un elenco utilizzando **String.Join (...)**:

```
List<string> list = new List<string>() {
  "one", "two", "three", "four", "five", "six"};
Console.WriteLine(String.Join("; ", list));
```

Ordinare Liste

- **Ordinare** una lista == riordinare i suoi elementi in modo incrementale
- Gli elementi dell'elenco dovrebbero essere comparabili, ad es. numeri, stringhe, date, ...

```
var names = new List<string>() {"Nakov", "Angel",
  "Ivan", "Atanas", "Boris" };
names.Sort();
Console.WriteLine(string.Join(", ", names));
// Angel, Atanas, Boris, Ivan, Nakov
names.Sort(); names.Reverse();
Console.WriteLine(string.Join(", ", names));
// Nakov, Ivan, Boris, Atanas, Angel
```

Ordinamento **ascendente**

Ordinamento **discendente**

Domande & approfondimenti

Academy .NET