List

* List sınıfı System.Collections.Generic’de yer almaktadır.
* List sınıfı IList<T>, ICollection<T>, IEnumerable<T>, IList, ICollection, IEnumerable arayüzlerini uygulamaktadır.
* List nesnesi hazırlanırken kapasite belirilmediğinde 0 olduğu varsayılmaktadır. İlk eleman eklendiğinde kapasite 4 olur. Beşinci eleman eklenmek istendiğinde mevcut kapasite yetmeyeceği için otomatik olarak 8 yapılır.
* List nesnesinin başlangıç kapasitesi yapıcı metod aracılığı ile belirtilebilir.

Example

//Kapasite 16 olur. Eleman sayısı 16'yı geçerse kapasite 32 yapılır.

List<int> numbers = new List<int>(16);

* IEnumerable arayüzünü uygulamış sınıflardan yola çıkılarak hazırlanan nesneler List koleksiyonunun yapıcı metoduna parametre olarak verilebilmektedir.

Example

List<int> liste1 = new List<int>(8);

List<int> liste2 = new List<int>(liste1);

Example

int[] arr = new int[3];

arr[0] = 0;

arr[1] = 1;

arr[2] = 2;

List<int> liste2 = new List<int>(arr);

* List nesnesine eleman ekleme işlemi Add() metodu ile yapılmaktadır.

Example

List<int> numbers = new List<int>();

numbers.Add(1);

numbers.Add(2);

numbers.Add(3);

* List koleksiyonunun tipini object olarak seçmeniz halinde performans ve işlev olarak ArrayList’den bir farkı kalmaz.
* List koleksiyonu hazırlanırken ilk değer ataması yapılabilmektedir.

Example

List<int> numbers = new List<int>() { 1, 2, 3 };

* List koleksiyonuna eleman eklerken ArrayList’de olduğu gibi Insert() ve InsertRange() metodlarını kullanabilirsiniz.
* List koleksiyonu içerisinde arama yaparken IndexOf() ve LastIndexOf() metodlarını kullanabilirsiniz.
* List koleksiyonundan eleman silinirken Remove(), RemoveAt() ve RemoveRange() metodları kullanılabilmektedir.
* List koleksiyonunun GetEnumerator() metodu ArrayList’den farklı olarak geriye generic bir numaralayıcı döndürmektedir.

Example

List<int> numbers = new List<int>() { 1, 2, 3 };

IEnumerator<int> enumerator = numbers.GetEnumerator();

enumerator.MoveNext();

Console.WriteLine(enumerator.Current); //1