Loops(Döngüler)

C#’da döngüler diğer programlama dillerinde olduğu gibi kullanılmaktadır. for, while ve do-while gibi standart döngülere ek olarak foreach döngüsü bulunmaktadır.

Example

int[] numbers = { 1, 2, 3 };

foreach (int number in numbers)

{

Console.WriteLine(number);

}

* foreach döngüsü IEnumerable arayüzünü uygulayan dizi ve koleksiyonlar üzerinde gezinmek için kullanılmaktadır.
* Parantez içerisindeki değişkenin tipi ilgili koleksiyon veya dizinin tipine uygun olarak seçilmelidir.
* Elemanın hangi türde bilgi içerdiğini bilmiyorsanız değişkeni aşağıdaki gibi var anahtar kelimesiyle tanımlayabilirsiniz.

Example

foreach (var number in numbers)

{

Console.WriteLine(number);

}

(yield Keyword)

Example

public IEnumerator<int> arr()

{

int[] numbers = { 1, 2, 3, 4 };

for (int i = 0; i < numbers.Length; i++)

{

yield return numbers[i];

}

}

Yukarıdaki örnekte yield anahtar kelimesi dizi elemanlarını bir numaralayıcıya aktarmakta ve daha sonra bu numaralayıcıyı geriye döndürmektedir. Bu anahtar kelime olmasaydı aynı işi yapmak için aşağıdaki kodları kullanmak zorunda kalırdık.

Example

public IEnumerator<int> arr()

{

IEnumerator<int> arr;

int[] numbers = { 1, 2, 3, 4 };

List<int> number\_list = new List<int>();

for (int i = 0; i < numbers.Length; i++)

{

number\_list.Add(numbers[i]);

}

arr = number\_list.GetEnumerator();

return arr;

}

Deyimler

break : Döngüleri sonlandırmak için kullanılır.

continue : Döngü içinde belirli bir koşulun kontrolüyle bir sonraki döngü adımına gidilmesini gerçekleştirir.

return : Metodları sonlandırmak için kullanılır. return; ifadesinden sonra yazılan hiçbir komut çalıştırılmaz.