Arrays

Aynı türde olup bellekte yan yana duran birden fazla değişkenin hepsine birden **dizi değişken** adı verilmektedir.

Example

//bu dizi değişken için bellekte yan yana 10 bölge ayrılır.

int[] numbers = new int[10];

* Diziler tanımlanırken de değer ataması yapılabilmektedir.

Example

//bu durumda eleman sayısını belirtmeye gerek yoktur. (eleman sayısı 3 olur.)

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

* C# programlama dilinde dizi değişken tanımlamak demek bir bakıma Array sınıfının örneğini almak demektir. Zaten diziler üzerinde işlemler yapılırken Array sınıfının metodları kullanılmaktadır. Array abstract bir sınıf olduğu için örneği alınamaz. Array sınıfı IList arayüzünü uygulamış olsa bile Add(), Remove(), Insert() gibi metodlara sahip değildir.
* int türündeki dizilerin başlangıç değeri **0**, bool türündeki dizilerin başlangıç değeri **false**, string türündeki dizilerin başlangıç değeri ise **null(boşluk)** olmaktadır.
* new anahtar kelimesi yeniden kullanılarak dizilerin eleman sayısı değiştirilebilmektedir. Bu durumda dizilerin mevcut elemanlarının içerikleri silinmektedir. Dizi içeriğini silmeden eleman sayısını değiştirmek istiyorsanız Array sınıfının Resize() metodunu kullanabilirsiniz.

Example

string[] cities = new string[3];

cities[0] = "İstanbul";

cities[1] = "Edirne";

cities[2] = "Kocaeli";

Console.WriteLine(cities.Length); //3

//dizinin referansını ve yeni eleman sayısını parametre olarak almaktadır.

Array.Resize(ref cities, 2);

Console.WriteLine(cities.Length); //2

Console.WriteLine(cities[0]); //İstanbul

Console.WriteLine(cities[1]); //Edirne

* object türünde dizi değişken tanımlayıp farklı türde veriler saklayabilirsiniz.
* Dizi değişkenler çok boyutlu olabilmektedir.

Example

int[,] numbers = new int[2, 3];

numbers[0, 0] = 1;

numbers[0, 1] = 2;

numbers[0, 2] = 3;

numbers[1, 0] = 10;

numbers[1, 1] = 20;

numbers[1, 2] = 30;

Console.WriteLine(numbers.Length); //6

Yukarıdaki dizi değişkeni iki satır ve üç sütuna sahip bir tablo gibi düşünebilirsiniz. Yukarıdaki dizi iki boyutludur. Aşağıdaki örnekte ise üç boyutlu bir dizi tanımlanmıştır. (Boyut artabilmektedir.)

Example

int[,,] numbers = new int[1, 2, 3];

numbers[0, 0, 0] = 1;

numbers[0, 0, 1] = 2;

numbers[0, 0, 2] = 3;

numbers[0, 1, 0] = 4;

numbers[0, 1, 1] = 5;

numbers[0, 1, 2] = 6;

Console.WriteLine(numbers.Length); //6

* Length özelliği dizinin eleman sayısını döndürür.
* GetLength() metodu ile çok boyutlu bir dizinin sadece istenilen boyutunun eleman sayısı öğrenilebilir. Parametre olarak boyutun index numarasını almaktadır.
* Dizilerin kaç boyutlu olduğu öğrenilmek istendiğinde Rank özelliği kullanılmaktadır.
* var anahtar kelimesi ile dizi değişken tanımlanabilmektedir. Aşağıdaki diziye tamsayı aktardığımız için int türünde olur.

Example

var years = new[] { 2020, 2021, 2022 };

new anahtar kelimesinden sonra dizinin türü ayarlanabilmektedir.

Example

var numbers = new byte[] { 1, 2, 3 };

Console.WriteLine(numbers.GetType()); //System.Byte[]

* dynamic anahtar kelimesi ile dizi değişken tanımlanabilmektedir.

Example

dynamic array = new dynamic[3];

array[0] = "Eren"; //name

array[1] = "Kocaeli"; //city

array[2] = 28; //age

* Dizi değişkenler içerisinde arama yapılırken IndexOf() ve LastIndexOf() metodları kullanılmaktadır. Bu metodlar geriye aranan elemanların index numaralarını ya da -1 değerini döndürmektedir.

Example

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

//Array.IndexOf(Array array, object value, int startIndex, int count)

//Array.LastIndexOf(Array array, object value, int endIndex, int count)

Console.WriteLine(Array.IndexOf(numbers, 1)); //0

Console.WriteLine(Array.LastIndexOf(numbers, 1)); //5

* string ifadeleri char türünüdeki dizi değişkenlere aktarırken ToCharArray() metodunu kullanabilirsiniz.

Example

string name = "Eren";

char[] arr = name.ToCharArray();

foreach(char item in arr)

{

Console.WriteLine(item);

}

Example

string name = "Eren";

//ToCharArray(int startIndex, int length)

char[] arr = name.ToCharArray(1, 2);

* Dizi elemanlarını seçerken köşeli parantezler içerisinde ^ sembolünü kullanabilirsiniz. Bu yenilik C# 8.0 sürümüyle birlikte gelmiştir.

Example

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

//sondan birinci eleman seçilir.

Console.WriteLine(numbers[^1]); //3

* Bir class ya da struct türünde dizi tanımlanabilmektedir.

Example

class Program

{

static void Main(string[] args)

{

Employee[] employees = new Employee[2];

employees[0] = new Employee();

employees[1] = new Employee();

employees[0].firstName = "Eren";

employees[0].salary = 4000;

employees[1].firstName = "Enis";

employees[1].salary = 8000;

Console.WriteLine(employees[0].firstName); //Eren

Console.WriteLine(employees[1].firstName); //Enis

Console.ReadLine();

}

}

class Employee

{

public string firstName;

public int salary;

}

Example

class Program

{

static void Main(string[] args)

{

Employee[] employees = new Employee[2];

employees[0].firstName = "Eren";

employees[0].salary = 4000;

employees[1].firstName = "Enis";

employees[1].salary = 8000;

Console.WriteLine(employees[0].firstName); //Eren

Console.WriteLine(employees[1].firstName); //Enis

Console.ReadLine();

}

}

struct Employee

{

public string firstName;

public int salary;

}

* Aşağıda gösterildiği gibi bir dizinin elemanları başka bir diziye aktarılabilmektedir. Bu durumda iki dizinin referansları aynı olur ve değer değişikliğinden ikinici dizi de etkilenir.

Example

int[] arr1 = { 1, 2, 3 };

int[] arr2 = arr1;

Console.WriteLine(Object.Equals(arr1, arr2)); //True

Console.WriteLine(Object.ReferenceEquals(arr1, arr2)); //True

arr1[0] = 10;

Console.WriteLine(arr2[0]); //10