

PROGRAMLAMA TEMELLERİ

2

6.HAFTA

C# PROGRAMLAMA RASTGELE SAYI ÜRETME VE DİZİLER

Rastgele sayı üretme

3

- C#'ta program yazarken (özellikle oyun programlarında) rastgele değerlere ihtiyaç duyabiliriz.
- Rastgele sayı üretebilmemiz için öncelikle Random sınıfı türünden bir nesne oluşturmak gereklidir:
- Random **rnd** = new Random();

1) int RastgeleSayi1=**rnd.Next(10,20);**

- 10 ile 20 arasında int türden rastgele bir sayı üretilir, 10 dâhil ancak 20 dâhil değildir.

2) int RastgeleSayi2=**rnd.Next(50);**

- 0 ile 50 arasında int türden rastgele bir sayı üretilir, 0 dâhil ancak 50 dâhil değildir.

3) int RastgeleSayi3=**rnd.Next();**

- int türden pozitif herhangi bir sayı üretilir.

4) double RastgeleSayi4=**rnd.NextDouble();**

- double türden 0.0 ile 1 arasında rastgele bir sayı üretilir.

- Bazı programlarda 200-300 değişkene ihtiyaç duyabiliriz.
- Bunların hepsinin teker teker tanımlanması oldukça zahmetlidir.
- İşte bu yüzden programlama dillerinde dizi diye bir kavram vardır.
- Bir dizi, birbiriyle ilişkili değişkenlerin oluşturduğu bir gruptur.

Dizilerin tanımlanması ve elemanlarının kullanılması

6

□ `int[] dizi=new`

`int[25];` veya

□ `int[] dizi;`

□ `dizi=new int[25];`

MAUZE

- ℳ □ Yukarıdaki iki kodda da `int` türünden 25 elemanlı dizi adında bir dizi tanımlandı ve dizinin her bir elemanına `int` türünün varsayılan değeri atandı.
- Varsayılan değerler, sayısal türler için 0, `string` türü için "", `char` için ' ' (boşluk) ve `bool` için `false` değerleridir.

- Bütün dizilerin birinci elemanı 0. indeksidir.
dizi dizisinin birinci elemanına dizi [0],

- 10. elemanına dizi[9] yazarak erişebilir ve
bu dizi elemanlarını bir değişkenmiş gibi
kullanabiliriz.

Diziler

8

MAUZE

```
using System; class Diziler {  
    static void Main() {  
        int[ ] dizi=new int[20];  
        dizi[5]=30;  
        Console.Write(dizi[5]);  
    }  
}
```

- Bu program ekrana 30 yazacaktır.

Diziler

9

- Dizilerin elemanlarına aşağıdaki gibi de değer atayabiliriz:

```
string[ ]dizi1={"Bir","İki","Üç"};
```

```
int[ ] dizi2={2,-4,6};
```

```
float[ ] dizi3={2f,1.2f,7f};
```

Diziler

10

- ❑ Başka bir dizi tanımlama yöntemi de şöyledir:

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

MAUZE
M

Diziler

11

- `int[] dizi=new int[20];`
- satırında 20'nin illaki sabit olmasınagerek yoktur.
Değişken ya da ifade de olabilir. Örnek:

MAUZE
M

Diziler

12

MAUZE
M

```
using System; class Diziler {  
    static void Main() {  
        int a=Convert.ToInt32(Console.ReadLine());  
        int[ ] dizi=new int[a+5];  
        dizi[5]=30;  
        Console.Write(dizi[5]);  
    }  
}
```

Diziler

13

MAUZE
M

- Eleman sayısı belirlenen bir dizinin eleman sayısı daha sonra değiştirilemez.
- Birden fazla dizi aşağıdaki gibi tanımlanabilir:
- **int[] dizi1=new int[10], dizi2=new int[20];**

Çok boyutlu diziler: Matris diziler

14

Her bir dizi elemanının eşit sayıda dizi içerdiği dizilerdir.

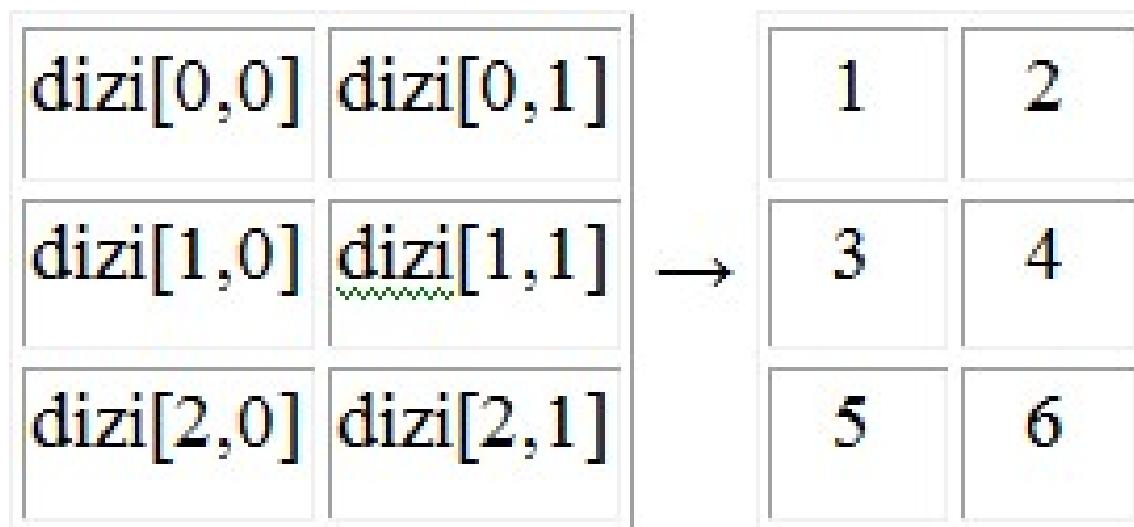
- ***İki boyutlu matris diziler***
- 3×2 boyutunda iki boyutlu bir matris dizi aşağıdaki gibi tanımlanabilir:
- `int[,] dizi=new int[3,2];`
- veya
- `int[,] dizi={{1,2}, {3,4}, {5,6}};`

MAUZE
M

Çok boyutlu diziler: Matris diziler

15

- Bu diziyi matris olarak aşağıdaki gibi gösterebiliriz:



İkiden fazla boyutlu matris diziler

16

Üç boyutlu bir dizi:

□ `int[,,] dizi=new int[3,2,2];`

veya

□ `int[,,] dizi={
 { {1,2} , {3,4} } ,
 { {5,6} , {7,8} } ,
 { {9,10} , {11,12} }
};`

MAUZE

Foreach Döngüsü

17

- foreach yalnızca dizilere uygulanabilen bir döngü yapısıdır.
Kullanımı şu şekildedir:

```
int[ ] dizi={3,2,6,7};
```

```
foreach(int eleman in dizi)
```

```
Console.WriteLine(eleman);
```

MAUZE

M

- Burada dizi dizisinin bütün elemanları teker teker ekrana yazdırılıyor.
- foreach döngüsüyle dizi elemanlarının değerini değiştiremeyiz, sadece ekrana yazdırmak gibi "read-only" işler yapabiliriz.

Foreach Döngüsü

18

```
string[] isimler =  
{ "Ahmet ",“Mehmet”,“Yusuf”, “Mahmut”,“Semra” };  
  
foreach(string eleman in isimler) {  
    Console.WriteLine(eleman);  
}
```

MAUZE