

# Java Programlama ve Diziler

12. Hafta

Dr. Öğr. Üyesi BÜŞRA ÖZDENİZCİ KÖSE

İşletme Bölümü

İşletme Fakültesi

# KarakterlerinAnalizi.java



- Kullanıcıdan alacağınız bir String'in büyük harf, küçük harf ve sayı türündeki karakter sayısını ekrana tablo formatında tab kaçış karakteri kullanarak yazdıran bir Java programı oluşturunuz.

```
run:
Bir String giriniz => bUSRa koSE bUsra123
Analiz          Adet
Büyük Harf      6
Küçük Harf      8
Sayı            3
BUILD SUCCESSFUL (total time: 16 seconds)
```

```
run:
Bir String giriniz => he11o W0r1d
Analiz          Adet
Büyük Harf      2
Küçük Harf      5
Sayı            3
BUILD SUCCESSFUL (total time: 11 seconds)
```

```
Start Page x KarakterlerinAnalizi.java x
Source History
1
2 package karakterlerinanalizi;
3 import java.util.Scanner;
4
5 public class KarakterlerinAnalizi {
6     public static void main(String[] args) {
7         Scanner busra = new Scanner (System.in);
8         System.out.print ("Bir String giriniz => ");
9         String cumle=busra.nextLine();
10
11         char c;
12
13         int i=0;
14
15         int buyukHarf=0, kucukHarf=0, sayiKarakteri=0;
16         while ( i<cumle.length() ) {
17
18             c=cumle.charAt(i);
19
20             if (Character.isUpperCase(c))
21                 buyukHarf++;
22             else if (Character.isLowerCase(c))
23                 kucukHarf++;
24             else if (Character.isDigit(c))
25                 sayiKarakteri++;
26             i++;
27         }
28
29         System.out.println("Analiz " + "\t\t" + "Adet");
30         System.out.println("Büyük Harf " + "\t" + buyukHarf );
31         System.out.println("Küçük Harf " + "\t" + kucukHarf );
32         System.out.println("Sayı " + "\t\t" + sayiKarakteri );
33
34     }
35 }
```



Bir matematiksel işlemler programı geliştireceksiniz. Ekran çıktı örneklerini lütfen dikkatle inceleyiniz. **main()** test programında kullanıcıya yapmak istediği işlemi soracaksınız ve ardından ilgili işlemin metodunu çağıracaksınız. **switch-case** yapısı kullanılacaktır.

```
Output - matematikselislemeler (run)
run:
1- Ortalama hesabı
2- Sayıyı tersine çevirme
3- Palindormik sayı kontrolü
Hangi işlemi yapmak istiyorsunuz? 1, 2 veya 3?
```

- Eğer kullanıcı 1 girerse, önce kullanıcıdan üç tamsayı okunacaktır ve ardından **averageNumbers()** metodu çağrılacaktır. **averageNumbers()** metodunda bu sayıların ortalaması hesaplanarak ekrana yazdırılacaktır.

```
Output - matematikselislemeler (run)
run:
1- Ortalama hesabı
2- Sayıyı tersine çevirme
3- Palindormik sayı kontrolü
Hangi işlemi yapmak istiyorsunuz? 1, 2 veya 3?
1
Üç tamsayı giriniz, ortalamasını hesaplayalım => 30 35 20
Ortalamaşı 28.333333333333332
BUILD SUCCESSFUL (total time: 7 seconds)
```

- Eğer kullanıcı 2 girerse, kullanıcıdan 4 basamaklı bir tamsayı (abcd) okunacaktır ve ardından **reverseNumber()** metodu çağrılacaktır. **reverseNumber()** metodunda, bu sayı tersine çevrilerek yer bir dört basamaklı tamsayı olarak (dcba) geri döndürülecektir; sonuç main() test programında yazdırılacaktır.

```
Output - matematikselislemeler (run)
run:
1- Ortalama hesabı
2- Sayıyı tersine çevirme
3- Palindormik sayı kontrolü
Hangi işlemi yapmak istiyorsunuz? 1, 2 veya 3?
2
Dört basamaklı bir tamsayı giriniz => 2389
Tersi 9832
BUILD SUCCESSFUL (total time: 5 seconds)
```

- Eğer kullanıcı 3 girerse, **palindrome()** metodu direk çağrılacaktır. **palindrome()** metodunda kullanıcıdan 3 basamaklı bir tamsayı okunacaktır; okunan sayının palindromik bir sayı olup olmadığı (true/false) boolean bilgisi geri döndürülecektir; sonuç main() test programında true veya false olarak yazdırılacaktır. Palindromik sayı tersten okunduğunda aynı olan sayıdır; örneğin 252, 787, 101 vb.

```
Output - matematikselislemeler (run)
run:
1- Ortalama hesabı
2- Sayıyı tersine çevirme
3- Palindormik sayı kontrolü
Hangi işlemi yapmak istiyorsunuz? 1, 2 veya 3?
3
Palindromik kontrolü için üç basamaklı bir tamsayı giriniz => 232
Sonuc true
BUILD SUCCESSFUL (total time: 3 seconds)
```

```
Output - matematikselislemeler (run)
run:
1- Ortalama hesabı
2- Sayıyı tersine çevirme
3- Palindormik sayı kontrolü
Hangi işlemi yapmak istiyorsunuz? 1, 2 veya 3?
3
Palindromik kontrolü için üç basamaklı bir tamsayı giriniz => 235
Sonuc false
BUILD SUCCESSFUL (total time: 7 seconds)
```

- Diğer seçimlerde (1, 2, 3 haricindeki seçimlerde) 'Yanlış Seçim...' bildirimi ekrana yazdırılacaktır.

```
Output - matematikselislemeler (run)
run:
1- Ortalama hesabı
2- Sayıyı tersine çevirme
3- Palindormik sayı kontrolü
Hangi işlemi yapmak istiyorsunuz? 1, 2 veya 3?
5
Yanlış seçim...
BUILD SUCCESSFUL (total time: 2 seconds)
```

```
Output - matematikselislemeler (run)
run:
1- Ortalama hesabı
2- Sayıyı tersine çevirme
3- Palindormik sayı kontrolü
Hangi işlemi yapmak istiyorsunuz? 1, 2 veya 3?
6
Yanlış seçim...
BUILD SUCCESSFUL (total time: 1 second)
```



Kodunuzu tamamladıktan sonra çeşitli kullanıcı girdileri ile kodunuzu çalıştırınız ve kontrol ediniz.

*İyi çalışmalar!*

```
1
2 package matematikselislemler;
3 import java.util.Scanner;
4 public class MatematikselIslemler {
5     public static void main(String[] args) {
6         Scanner busra = new Scanner (System.in);
7         System.out.println("1- Ortalama hesabı");
8         System.out.println("2- Sayıyı tersine çevirme");
9         System.out.println("3- Palindormik sayı kontrolü");
10        System.out.println("Hangi işlemi yapmak istiyorsunuz? 1, 2 veya 3?");
11        int secim = busra.nextInt();
12
13        switch (secim) {
14            case 1:
15                System.out.print ("Üç tamsayı giriniz, ortalamasını hesaplayalım => ");
16                int a=busra.nextInt();
17                int b=busra.nextInt();
18                int c=busra.nextInt();
19                averageNumbers(a, b, c);
20                break;
21            case 2:
22                System.out.print ("Dört basamaklı bir tamsayı giriniz => ");
23                int k=busra.nextInt();
24                System.out.println ("Tersi " + reverseNumber(k) );
25                break;
26            case 3:
27                boolean sonuc = palindrome ();
28                System.out.println ("Sonuc " + sonuc);
29                break;
30            default:
31                System.out.println ("Yanlış seçim...");
32                break;
33        }
34    }
```

```

35
36 public static void averageNumbers (double x, double y, double z) {
37     System.out.println("Ortalaması " + (x+y+z)/3 );
38 }
39
40 public static int reverseNumber (int a) {
41     int binler = a/1000;
42     int yuzler = a%1000/100;
43     int onlar = a%100/10;
44     int birler = a%10;
45     int tersten = (birler*1000) + (onlar*100) + (yuzler*10) + (binler*1);
46     return tersten;
47 }
48
49 public static boolean palindrome () {
50     System.out.print ("Palindromik kontrolü için üç basamaklı bir tamsayı giriniz => ");
51     Scanner busra = new Scanner (System.in);
52     int m=busra.nextInt();
53
54     int yuzler = m/100;
55     int birler = m%10;
56
57     if (yuzler ==birler)
58         return true;
59     else
60         return false;
61
62 }
63
64
65 }

```

# Neden Diziler?

- Şu ana kadar, bir(kaç) değer üzerinde işlem yapan programlar yazdık. Bu programların hepsinde az sayıda değişken tanımlaması ve basit işlemler söz konusu idi.
- **Değişken sayısı arttıkça değişkenlerin yönetilmesi oldukça zor ve program kabul edilemez bir uzunluğa ve karmaşıklığa sahip olacaktır.**
- Tanımlama zorluğu yanında, çok sayıda değişken üzerinde işlem yapmak da oldukça zordur. Ayrı ayrı tanımlanmış pek çok değişken üzerinde işlem yapmak için çok sayıda komut yazmak gerekecektir ve bu durum yine işleri daha da zorlaştıracaktır.

# Neden Diziler?

- Çok sayıda verinin işlenmesi ve yönetilmesi amacıyla dizi yapısı kullanılır.
- **Dizi, aynı türdeki çok sayıda veriyi saklamak için tek bir değişken kullanılmasını sağlayan bir yapıdır.**
- Örneğin, 100 adet tamsayıyı **tek bir değişkende** yani bir dizide saklayabilirsiniz ve tek bir değişken ile bu sayılara erişebilirsiniz!



# Nasıl?

- Program içerisinde **sayi0, sayi1, sayi2...** şeklinde 100 tane sayı tanımlamak yerine **sayilar** şeklinde bir dizi değişkeni tanımlayabilirsiniz !
- **sayilar[0], sayilar[1], ... sayilar[99]** şeklinde kullanarak dizide saklanan tüm sayılara erişmeniz mümkün !

# Diziler

- Programlarda dizi kullanmanın pek çok faydaları vardır:
  - **Basitlik:** Pek çok değişken yerine bir değişken kullanımı sade bir program oluşmasını sağlar.
  - **Kolay erişim:** Dizinin istenen herhangi bir elemanına indis kullanarak kolaylıkla erişilebilir.
  - **Kolay işlem:** Tek bir döngüyle tüm elemanlar üzerinde kolaylıkla işlem yapılabilir.
  - **Problem çözümünde kolaylık:** Dizi kullanımı problem çözümünde büyük avantaj sağlar.

# Dizi Tanımlama

- Birinci yöntem, ilk adımda dizi referansının tanımlaması, ikinci adımda da dizinin oluşturulmasıdır.

- Dizi referansının tanımlanması için:

```
veri_turu[] referans_adi;
```

- Örneğin, **double** veri türünde elemanlar içerecek **myList** isimli bir dizi referansı tanımlanmıştır:

```
double[] myList;
```

# Dizi Oluşturma

- **Referans (adres) tanımlaması** tanımlandığında, bu değişken için bellekte göz tahsis edilir. Bu değişken, bir adres bilgisini tutacaktır.
- Referans tanımlamasında dizinin tutacağı veri türü belirlidir, **fakat içinde tutulacak eleman sayısı belirsizdir.**
- Dizi oluşturulurken, dizi içinde tutulacak eleman sayısı –yani dizinin uzunluğu– belirtilir. Bu durumda da dizi için bellekte gerektiği kadar göz tahsis edilir

```
referans_adi = new veri_turu[dizi_uzunlugu];  
myList = new double[10];
```

# Dizi Tanımlama ve Oluşturma

Beraber Yapılabilir:

- Diğer bir yöntem ise dizi referansı tanımlama ve dizi oluşturma işlemlerinin tek satırda gerçekleştirilmesidir.
- Dizi referansının tanımlanması ve dizinin oluşturulması tek bir komut olarak da yazılabilir:

```
veri_turu [] referans_adi = new veri_turu [dizi_uzunlugu];
```

```
double[] myList= new double[10];
```

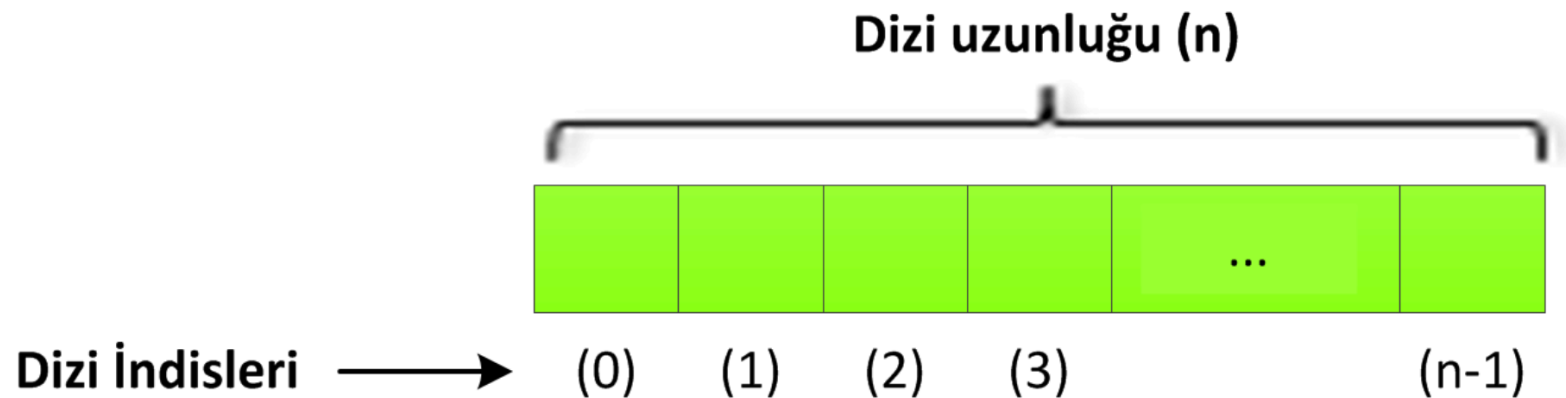
```
Start Page x DiziTanimlama.java x
Source History
1
2 package dizitanimlama;
3
4 public class DiziTanimlama {
5
6     public static void main(String[] args) {
7
8         int[] sayilar;
9         /*
10        int değer içerecek bir dizinin adresi bilgisini tutacak olan
11        sayilar isimli dizi adres değişkeni tanımlandı
12        */
13
14        sayilar = new int[5];
15
16        /*
17        5 tane int değer içerecek bir dizi tanımlandı.
18        Bellekte 5 tane int değer tutacak kadar göz ayrıldı.
19        Tahsis edilen gözlerin başlangıç adres, sayilar değişkenine atandı.
20        */
21
22        char[] alfabe;
23        alfabe = new char[29];
24
25        double[] notlar = new double [5];
26
27    }
28 }
```



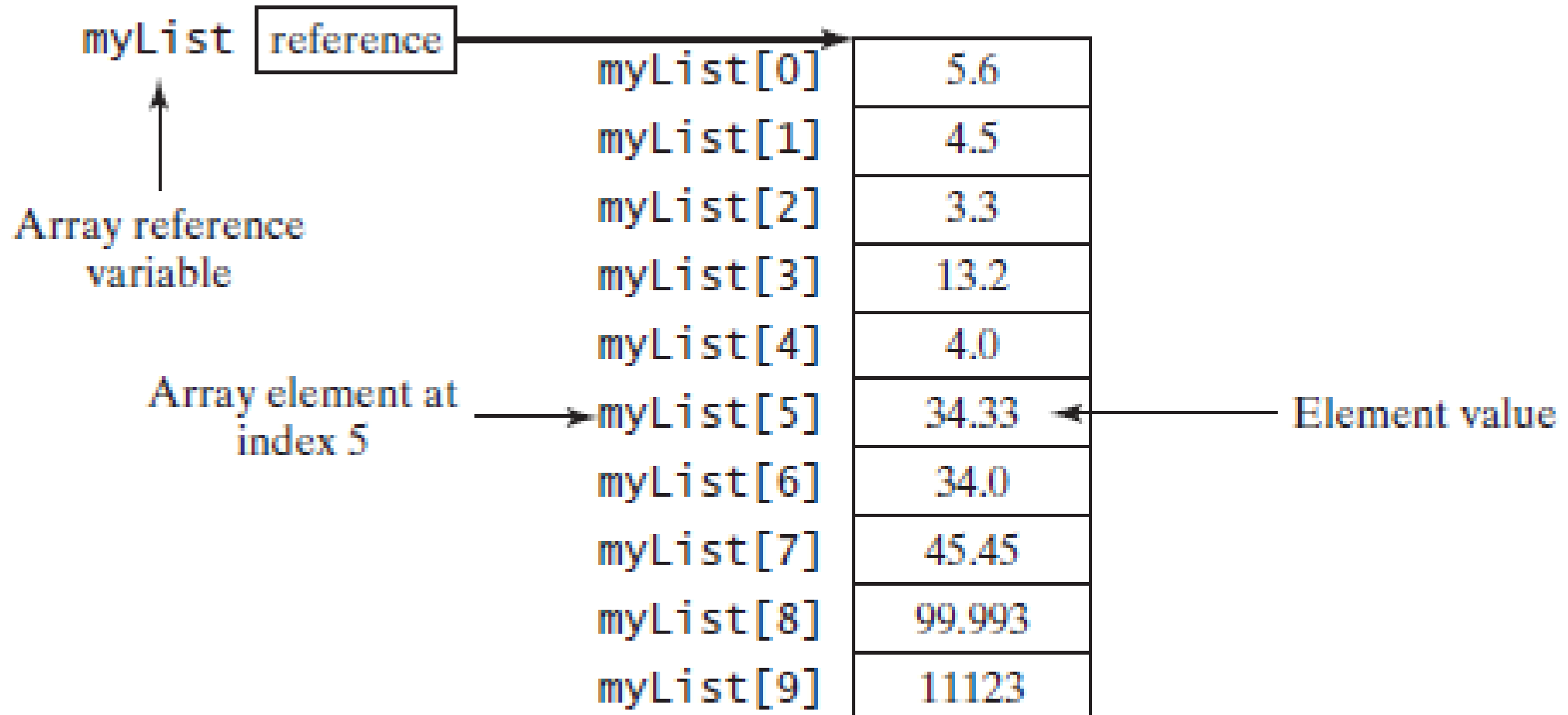
# Dizi Değerleri Tanımlama

- Dizi referansı tanımlandıktan ve dizi oluşturulduktan sonra tek tek dizi indislerine değerler atanabilir:

`referans_adi[indis] = deger;`



```
double[] myList = new double[10];
```





# Dizi Değerleri Tanımlama

- Örneğin tek tek dizi indislerine değer atama yöntemi:

`myList[0] = 5.6;`

`myList[1] = 4.5;`

`myList[2] = 3.3;`

`myList[3] = 13.2;`

`myList[4] = 4.0;`

`myList[5] = 34.33;`

`myList[6] = 34.0;`

`myList[7] = 45.45;`

`myList[8] = 99.993;`

`myList[9] = 11123;`

```
Start Page x DiziOrnegim.java x
Source History
1
2 package diziornegim;
3
4 public class DiziOrnegim {
5
6     public static void main(String[] args) {
7
8         int[] sayilar = new int[5];
9
10        sayilar[0]= 1;
11        sayilar[1]= 3;
12        sayilar[2]= 5;
13        sayilar[3]= 7;
14        sayilar[4]= 9;
15
16        System.out.println("Dizinin ilk elemanı = " + sayilar[0]);
17
18        System.out.println("Dizinin son elemanı = " + sayilar[4]);
19
20    }
21 }
```



Referans	Bellek		Erişim Kodu	Adres
sayilar ->		1	sayilar[0]	F1A3
		3	sayilar[1]	F1A4
		5	sayilar[2]	F1A5
		7	sayilar[3]	F1A6
		9	sayilar[4]	F1A7

```
Start Page x DiziOrnegim.java x DiziTanimlama.java x KarakterAdedi.java x
Source History
1 package diziornegim;
2
3
4 import java.util.Scanner;
5
6 public class DiziOrnegim {
7
8     public static void main(String[] args) {
9
10         Scanner input=new Scanner(System.in);
11
12         char[] alfabe = new char[29];
13
14         alfabe[0] = 'a';
15         alfabe[1] = 'b';
16         alfabe[2] = 'c';
17         alfabe[27] = 'y';
18         alfabe[28] = 'z';
19
20         System.out.println("Dizinin ilk elemanı = " + alfabe[0]);
21
22         System.out.println("Dizinin son elemanı = " + alfabe[28]);
23
24         System.out.println("Dizinin dördüncü elemanını giriniz: ");
25
26         alfabe[3] = input.nextLine().charAt(0);
27
28         System.out.println("Dizinin dördüncü elemanı = " + alfabe[3]);
29
30
31     }
32 }
```



```
run:
Dizinin ilk elemanı = a
Dizinin son elemanı = z
Dizinin dördüncü elemanını giriniz:
d
Dizinin dördüncü elemanı = d
BUILD SUCCESSFUL (total time: 3 seconds)
```

# Diğer Alternatif

- Bu yöntem kullanıldığında **new** operatörünün kullanılmadığına ve dizi uzunluğunun ayrıca belirtilmediğine dikkat ediniz.
- Dizi uzunluğu tanımlanan eleman sayısı kadar otomatik olarak belirlenecektir:

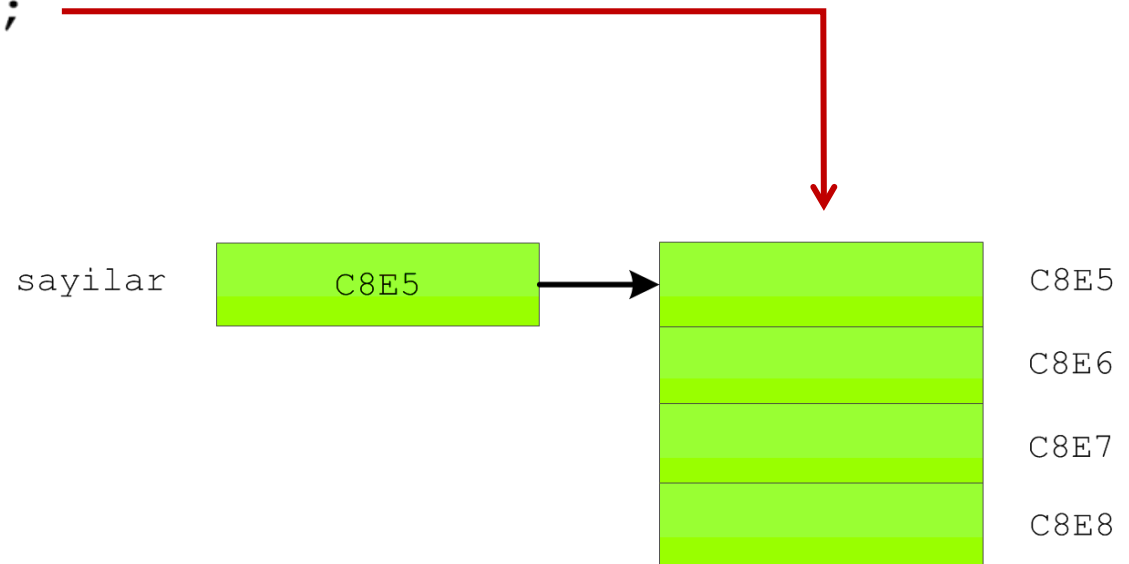
```
veri_turu[] referans_adi = { deger_0, ..., deger_n };
```

DO IT  
NOW!

01	<code>int[] sayilar = { 1, 3, 5, 7, 9 };</code>	Doğru
02	<code>double[] notlar = { 80.3, 90.4 };</code>	
03	<code>char[] harfler = { 'A', 'B', 'C' };</code>	
04	<code>int[] sayilar;</code>	Hatalı
05	<code>sayilar = {147, 323, 89, 933, 1};</code>	

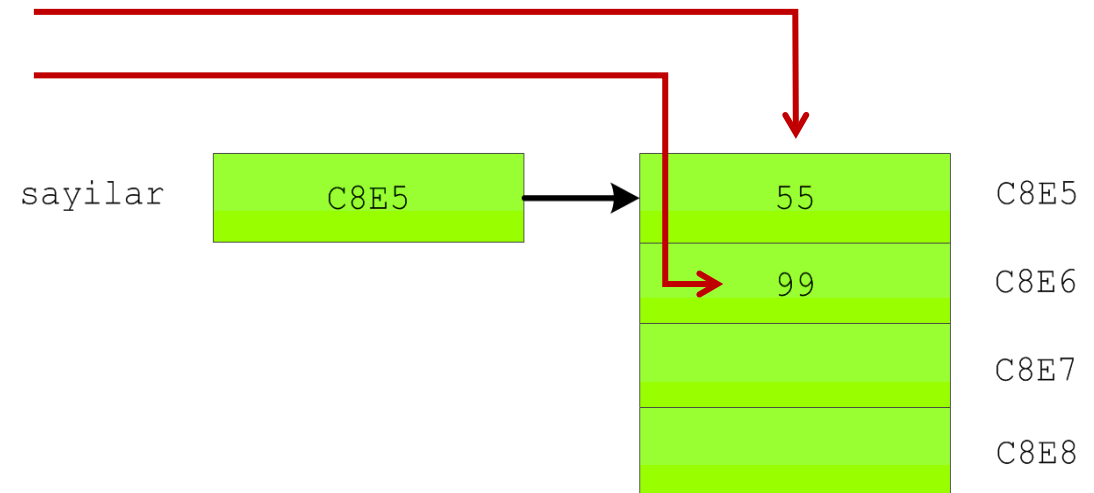
# Dizi ve Dizi Referansı

```
public class DiziOlusturalim
{
    public static void main( String[] args)
    {
        int[] sayilar;
        sayilar = new int[4];
        sayilar[0] = 55;
        sayilar[1] = 99;
    }
}
```



# Dizi ve Dizi Referansı

```
public class DiziOlusturalim
{
    public static void main( String[] args)
    {
        int[] sayilar;
        sayilar = new int[4];
        sayilar[0] = 55;
        sayilar[1] = 99;
    }
}
```



# Dizi ve Dizi Referansı

- Örneğin notlar isimli bir dizi referansı tanımlandığında:  
**int[] notlar;**
- **notlar** dizi ismi değildir, **dizi referansının ismidir!**
- Java'da **dizi ismi** diye bir kavram **yoktur!**



```
int[] sayilar;
```

```
sayilar = new int[2];
```

sayilar

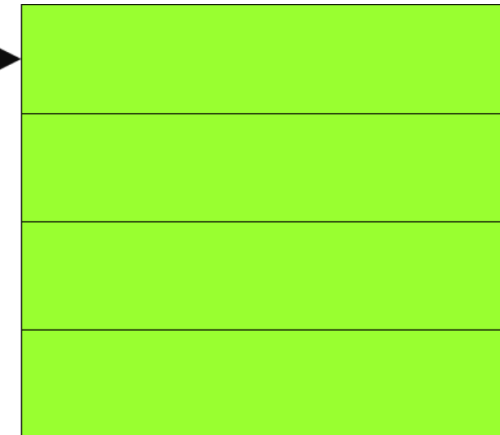


F2E0

F2E1

```
sayilar = new int[4];
```

sayilar



F2E2

F2E3

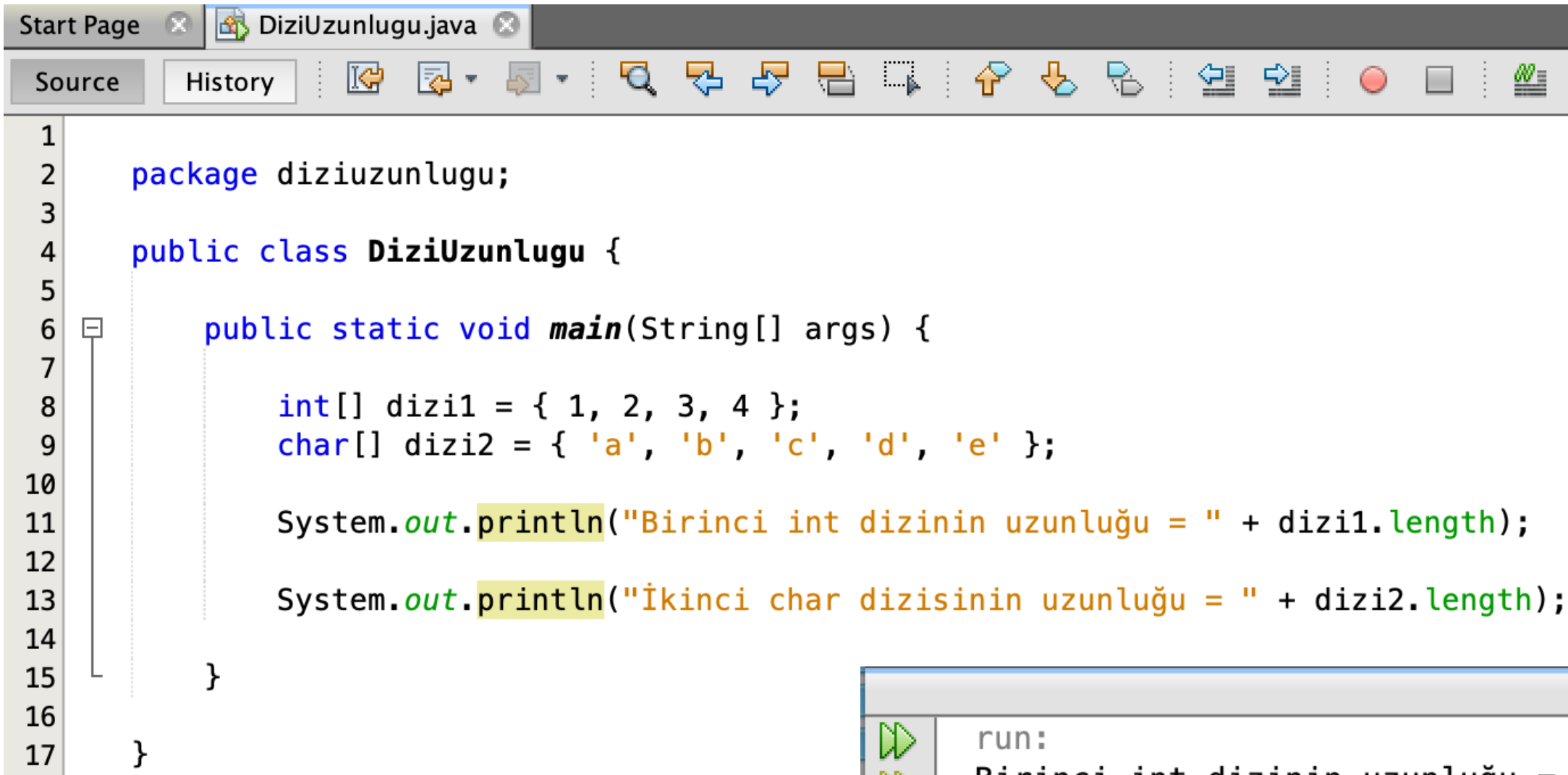
F2E4

F2E5

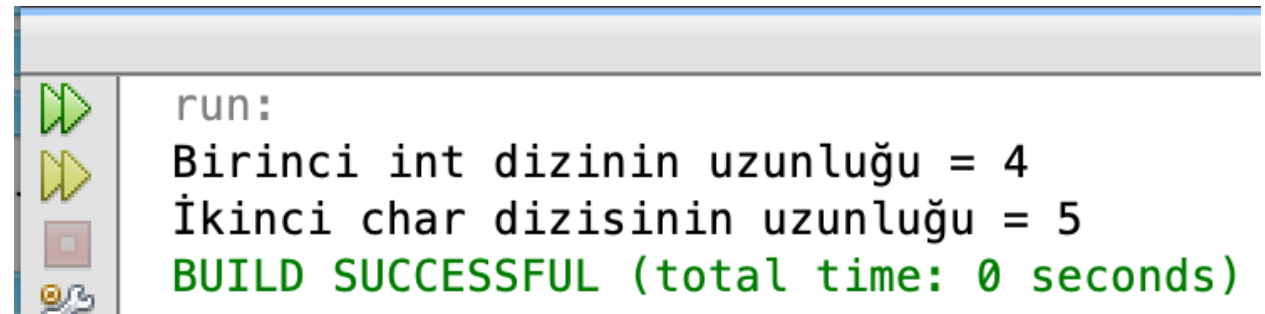
**Dangling Reference  
(Kayıp Bağlantı)**

# Dizi Uzunluğu

- Bir dizi ilk oluşturulduğunda **-eğer değerleri atanmamış ise-** varsayılan tüm indis değerleri:
  - sayısal veri türünde oluşturulmuş diziler için 0 (sıfır),
  - mantıksal veri türünde oluşturulmuş diziler için false,
  - karakter veri türünde oluşturulmuş diziler için '\u0000' olacaktır.
- Bir dizinin uzunluğu dizinin başlangıç adresini içeren dizi referansı kullanılarak **referans\_adi.length** komutuyla öğrenilebilir.



```
1 package diziuzunlugu;
2
3 public class DiziUzunlugu {
4
5     public static void main(String[] args) {
6
7         int[] dizi1 = { 1, 2, 3, 4 };
8         char[] dizi2 = { 'a', 'b', 'c', 'd', 'e' };
9
10        System.out.println("Birinci int dizinin uzunluđu = " + dizi1.length);
11
12        System.out.println("İkinci char dizisinin uzunluđu = " + dizi2.length);
13
14    }
15
16 }
17 }
```



```
run:
Birinci int dizinin uzunluđu = 4
İkinci char dizisinin uzunluđu = 5
BUILD SUCCESSFUL (total time: 0 seconds)
```

# Dizi Elemanlarına Erişim

- n tane eleman içeren dizi elemanlarına **indis (index)** ile erişilir. İndis, dizi elemanlarının sırasını ifade eden bir tamsayıdır. n elemanlı bir dizinin birinci elemanın indisi 0, ikinci elemanın indisi 1, son elemanın indisi ise n-1'dir.
- **myList** dizi referansı 10 adet **double** değer içermektedir ve indisleri 0'dan 9'a kadardır. **myList[0]** ilk elemanı, **myList[9]** son elemanı gösterir:

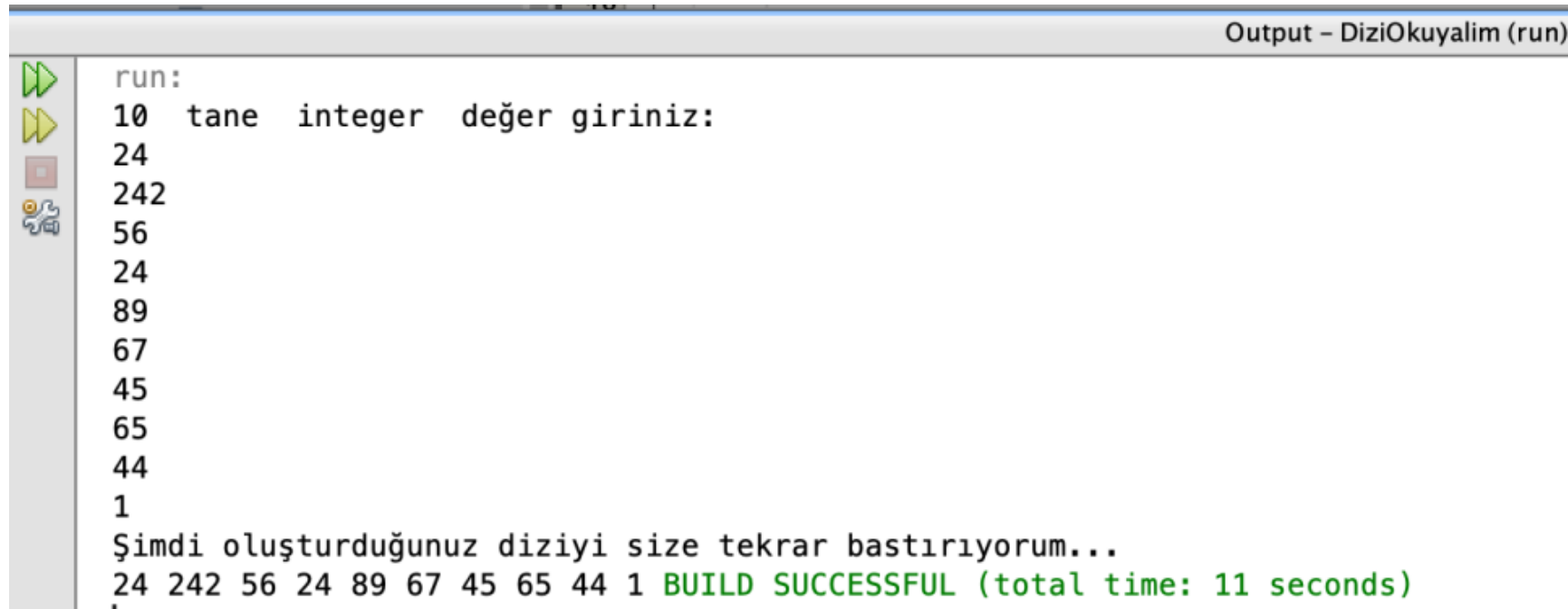
`referans_adi[indis];`

- Örneğin, **myList[0]** ve **myList[1]** değerlerinin toplamı **myList[2]**'ye atanmıştır:

`myList[2] = myList[0] + myList[1];`

# DiziOkuyalim.java

- 10 değer içerecek bir **myList[]** dizi referansı tanımlayarak, kullanıcıdan dizi değerlerini for döngüsü ile okuyalım. Ardından oluşturduğumuz dizinin elemanlarını tekrar ekrana yazdıralım.

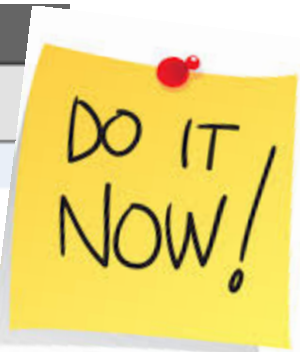


```
run:
10 tane integer değer giriniz:
24
242
56
24
89
67
45
65
44
1
Şimdi oluşturduğunuz diziyi size tekrar bastırıyorum...
24 242 56 24 89 67 45 65 44 1 BUILD SUCCESSFUL (total time: 11 seconds)
```

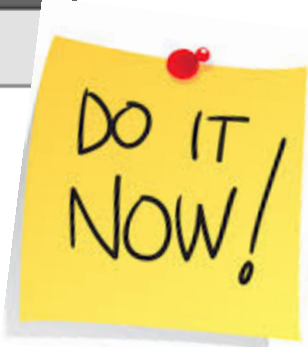
Start Page x DiziOkuyalim.java x

Source History

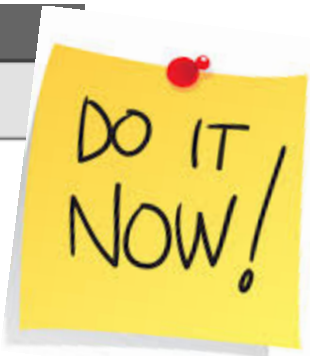
```
1
2 package diziokuyalim;
3
4 import java.util.Scanner;
5
6 public class DiziOkuyalim {
7     public static void main(String[] args) {
8
9         Scanner input = new Scanner(System.in);
10
11         int[] myList= new int[10]; // uzunluk degisebilir
12
13         System.out.println( "10 tane integer deęer giriniz: ");
14
15         for (int i = 0; i < 10; i++)
16
17             myList[i] = input.nextInt();
18
19         System.out.println("Şimdi oluřturduęunuz diziyi size tekrar bastırıyorum... ");
20
21         for (int i = 0; i < 10; i++)
22
23             System.out.print( myList[i] + " ");
24     }
25
26 }
```



```
Start Page x DiziOkuyalim.java x
Source History
1
2 package diziokuyalim;
3
4 import java.util.Scanner;
5
6 public class DiziOkuyalim {
7     public static void main(String[] args) {
8
9         Scanner input = new Scanner(System.in);
10
11         int[] myList= new int[10]; // uzunluk degisebilir
12
13         int uzunluk = myList.length; // dizi uzunluğunu böyle de alabilirim
14
15         System.out.println( uzunluk + " tane integer değer giriniz: ");
16
17         for (int i = 0; i < uzunluk; i++)
18             myList[i] = input.nextInt();
19
20         System.out.println("Şimdi oluşturduğunuz diziyi size tekrar bastırıyorum... ");
21
22         for (int i = 0; i < uzunluk; i++)
23             System.out.print( myList[i] + " ");
24
25     }
26 }
27
28 }
```



```
Start Page x DiziOkuyalım.java x
Source History
1
2 package diziokuyalım;
3
4 import java.util.Scanner;
5
6 public class DiziOkuyalım {
7     public static void main(String[] args) {
8
9         Scanner input = new Scanner(System.in);
10
11         int[] myList= new int[10]; // uzunluk degisebilir
12
13         System.out.println( myList.length + " tane integer değeri giriniz: ");
14
15         for (int i = 0; i < myList.length; i++)
16             myList[i] = input.nextInt();
17
18         System.out.println("Şimdi oluşturduğunuz diziyi size tekrar bastırıyorum... ");
19
20         for (int i = 0; i < myList.length; i++)
21             System.out.print( myList[i] + " ");
22     }
23 }
24
25
26 }
```





# DiziOlusturalim.java

- Kullanıcının belirleyeceği uzunlukta bir **myList[]** dizi referansı tanımlayınız. Kullanıcıdan dizi değerlerini for döngüsü ile okuyalım. Ardından oluşturduğumuz dizinin elemanlarını tekrar ekrana yazdıralım.

```
Output
run:
Oluşturmak istediğiniz integer dizisinin uzunluğunu belirleyiniz:
5
Dizinin için integer değerleri giriniz:
12
34
89
4
6
Şimdi oluşturduğunuz diziyi size tekrar bastırıyorum...
12 34 89 4 6 BUILD SUCCESSFUL (total time: 6 seconds)
```

```
Output
run:
Oluşturmak istediğiniz integer dizisinin uzunluğunu belirleyiniz:
3
Dizinin için integer değerleri giriniz:
12
2
4
Şimdi oluşturduğunuz diziyi size tekrar bastırıyorum...
12 2 4 BUILD SUCCESSFUL (total time: 4 seconds)
```

```
1
2 package diziolusturalim;
3
4 import java.util.Scanner;
5
6 public class DiziOlusturalim {
7
8     public static void main(String[] args) {
9
10         Scanner input = new Scanner(System.in);
11
12         System.out.println("Oluşturmak istediğiniz integer dizisinin uzunluğunu belirleyiniz:");
13
14         int uzunluk = input.nextInt(); // dizi uzunluğunu kullanıcı belirlesin
15
16         int[] myList= new int[uzunluk];
17         // kullanıcının girdiği integer ile array oluşturduk
18
19         System.out.println( "Dizinin için integer değerleri giriniz: ");
20
21         for (int i = 0; i < uzunluk; i++)
22             myList[i] = input.nextInt(); // değerleri okuyorum ve atıyorum
23
24         System.out.println("Şimdi oluşturduğunuz diziyi size tekrar bastırıyorum... ");
25
26         for (int i = 0; i < uzunluk; i++)
27             System.out.print( myList[i] + " "); // değerleri tek tek yazdırıyorum
28
29     }
30
31 }
32 }
```



DO IT  
NOW!

# Dizi Sınırı Hatası

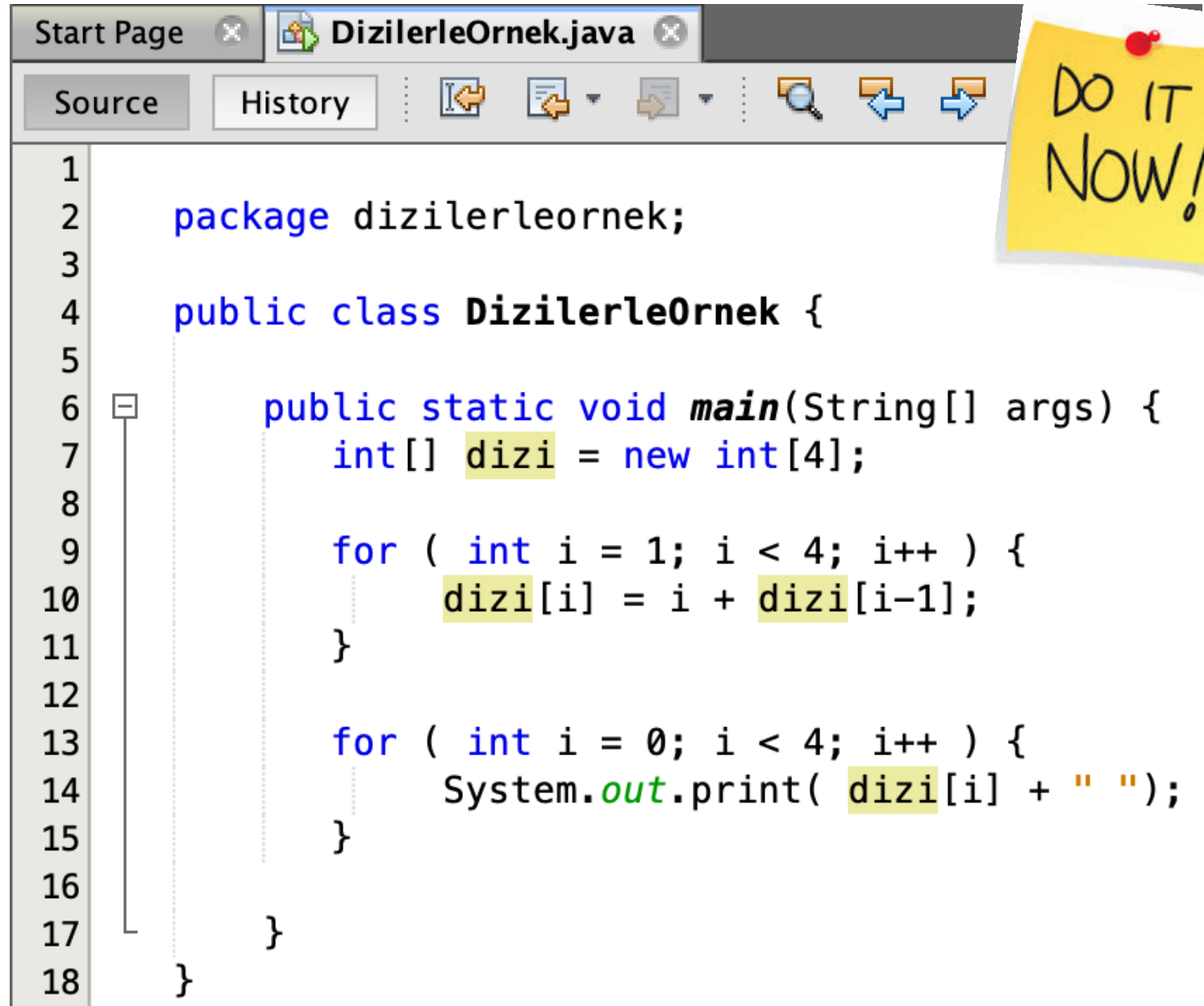
- n elemanı olan bir dizi için indis aralığının 0 ile n-1 arasında olduğunu unutmayalım.
- Sıfırdan küçük ya da n-1'den büyük bir indis kullanılması durumunda JVM, ekrana **ArrayIndexOutOfBoundsException** hata mesajını yazarak programı çalıştırmayı sonlandırır.

```
for (int i = 0; i <= list.length; i++)
```

```
System.out.print(list[i] + " ");
```

- **<= ifadesi, < şeklinde değiştirilmelidir !**

# Ekran Çıktısı Nedir? 😊



```
1 package dizilerleornek;
2
3
4 public class DizilerleOrnek {
5
6     public static void main(String[] args) {
7         int[] dizi = new int[4];
8
9         for ( int i = 1; i < 4; i++ ) {
10             dizi[i] = i + dizi[i-1];
11         }
12
13         for ( int i = 0; i < 4; i++ ) {
14             System.out.print( dizi[i] + " ");
15         }
16
17     }
18 }
```

```
int[] dizi = new int[5];
```

A5E5 (dizi)	0
	0
	0
	0

```
dizi[1] = 1 + dizi[0];
```

A5E5 (dizi)	0
	1
	0
	0

A5E5  
A5E6  
A5E7  
A5E8

```
dizi[2] = 2 + dizi[1];
```

A5E5 (dizi)	0
	1
	3
	0

```
dizi[3] = 3 + dizi[2];
```

A5E5 (dizi)	0
	1
	3
	6

A5E5  
A5E6  
A5E7  
A5E8

# Any Questions?