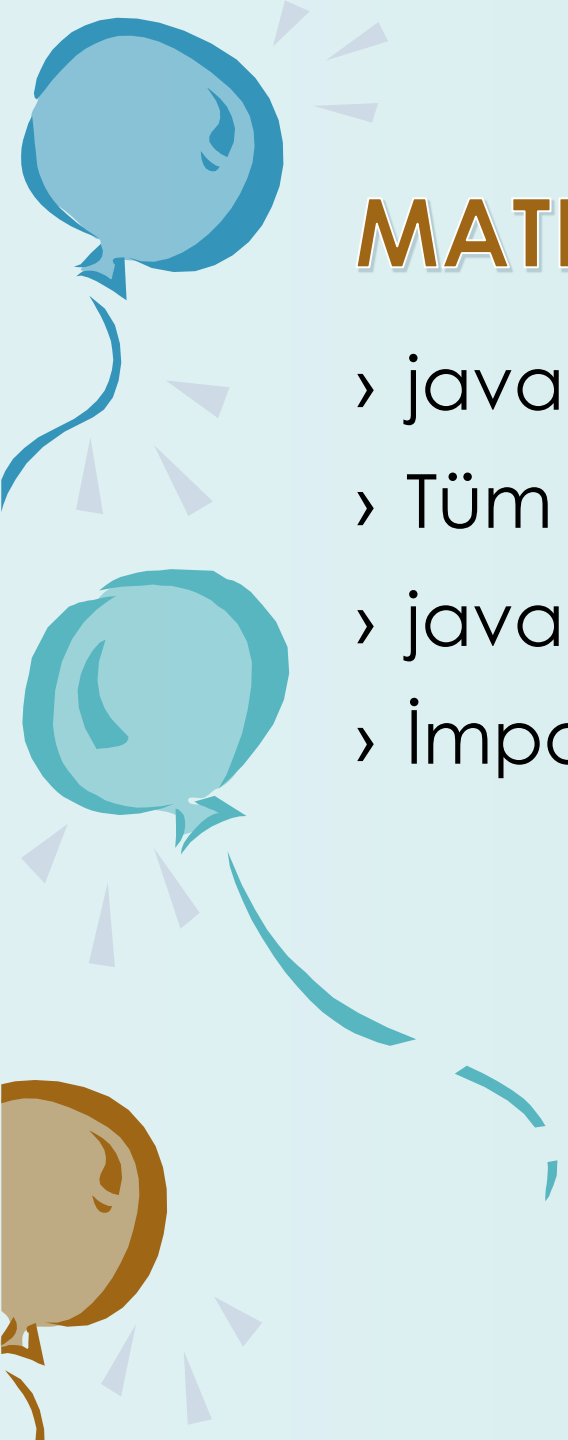
MATH SINIFI



MATH SINIFI

› java.lang paketinin içindedir.

› Tüm paketler import ile çağrılır.

› java.lang açılışta otomatik olarak dahil edilir.

› İmport etmeden kullanılabilir.

MATH SINIFI

- Math.PI
- Abs(x); → Mutlak değer
- Ceil(x); →Ondalıklı sayıyı üst sayıya yuvarlar
- Math.raounds();→Ondalıklı sayıyı yuvarlar.
- Sin(x); →x n sinüs ünü alır →Tan(x);, Acos(x);, Asin(x);,
- pow(x,y);→x in y ninci kuvvetini alır
- sqrt(x);→ karekök ünü alır
- Max(x,y); →en büyük olanını verir.
- Min(x,y); →en küçük olanını verir.
- Random(); → Rastgele sayı üretir 0 ile 1 arasında.

```java
package deneme;
import java.util.*;
public class Deneme {
    public static void main(String[] args) {
    double x=-3.12, y=3.5, z=3.6;
    System.out.println(Math.PI);
    System.out.println(Math.abs(x));
    System.out.println(Math.ceil(x));
    System.out.println(Math.round(x));
    System.out.println(Math.round(y));
    System.out.println(Math.round(z));
    System.out.println(Math.sin(30));
    System.out.println(Math.pow(2,3));
    System.out.println(Math.sqrt(9));
    System.out.println(Math.max(y,z));
    System.out.println(Math.random());
    }
}
```

STRİNGLER

STRİNGLER

› String sınıfı Java'da metinler tanımlamak için kullanılır ve bize metinler üzerinde çeşitli işlemler gerçekleştirmemiz için yardımcı fonksiyonlar sunar.

```
1  public class StringOrnek1
2  {
3      public static void main(String[] args)
4  {
5      // string tanımlanması
6      String myString = "Merhaba dunya";
7  }
8  }
```

STRİNGLER-Lenght

› Bu metinin uzunluğunu (karakter sayısını) merak edersek **length** komutunu kullanırız.

```
5  public static void main(String[] args) {
6      ///STRING MANİPULASYONLARI
7      String s="nmacit@gelisim.edu.tr";
8      System.out.println(s.length() );
9      //string in uzunluğunu bulan fonksiyon
```

Console

<terminated> D1StringManipulasyonlar [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.h

21

STRİNGLER-CharAt

› String in seçilen indisine ulaşılmasını sağlar.

```
11          s="nmacit@gelisim.edu.tr";
12          System.out.println(s.charAt(s.length()-1));
13          //string in istenilen indisine ulaşmanızı sağlar
```

Console

<terminated> D1StringManipulasyonlar [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20

```
r
```

STRİNGLER-Substring

› Örnekte 7 – 14 arasındaki (7 dahil 14 hariç) karakterler konsola yazılacaktır

```
14
15          s="nmacit@gelisim.edu.tr";
16          String kesilenKisim=s.substring(7,14);
17          System.out.println(kesilenKisim);
18          //substring bir string deki belirli bir kısmın alınması nı sağlar
19          //ilk indis başlangıç indisi, ikinci sayı bitiş indisi ama dahil değil
20
21
```

Console

<terminated> D1StringManipulasyonlar [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe (28 Eyl 2025 08:

```
gelisim
```

n	m	a	c	i	t	@	g	e	l	i	s	i	m	.	e	d	u	.	t	r
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

STRİNGLER-Substring

```
21          s="nmacit@gelisim.edu.tr";
22          String kesilenKisim2=s.substring(7);
23          System.out.println(kesilenKisim2);
24          //substring 2. kullanım şekli
25          //belirlenen indis no dan itibaren sonuna kadar bütün değerleri alır
26
```

Console

<terminated> D1StringManipulasyonlar [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe (28 Eyl 2025 (

```
gelisim.edu.tr
```

STRİNGLER-Startswith

- String istenilen karakter ile başlıyor mu kontrolünü sağlar
- Geriye true ya da false bilgisi döner.

```
33    s="nmacit@gelisim.edu.tr";
34    boolean sonuc2 =s.startsWith("m");
35    System.out.println(sonuc2);
36    //istenilen karakter ile başlıyor mu kontrolü yapmamızı sağlar.
37
```

Console

<terminated> D1StringManipulasyonlar [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.ex

```
false
```

STRİNGLER-Contains

› Belirtilen metin string içinde var mı kontrolü yapar.

› Geriye true ya da false bilgisini döndürür.

```
27      s="nmacit@gelisim.edu.tr";
28      boolean sonuc=s.contains("gelisim.edu.tr");
29      if(sonuc==true)System.out.println("doğru format");
30      else System.out.println("yanlış format ");
31      //belirtilen string içerisinde belirtilen veri olup olmadığını kontrol eder
32
```

Console

<terminated> D1StringManipulasyonlar [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe (28 Eyl 2025 C

```
doğru format
```

STRİNGLER-indexOf

› Bir harfin ya da String değerinin kaçıncı dizinden (index) itibaren başladığını merak ediyorsanız, **indexOf** metodunu kullanabilirsiniz. Örneğin "gelisim" kelimesinin kaçıncı dizinde olduğunu öğrenmek için

```
38      s="nmacit@gelisim.edu.tr";
39      int indeksNo=s.indexOf("gelisim");
40      System.out.println(indeksNo);
41      /*Bir harfin ya da String değerinin kaçıncı dizinden
42       * (index) itibaren başladığını merak ediyorsanız,
43       * indexOf metodunu kullanabilirsiniz. */
```

Console

<terminated> D1StringManipulasyonlar [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v2025

```
7
```

n	m	a	c	i	t	@	g	e	l	i	s	i	m	.	e	d	u	.	t	r
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

STRİNGLER-equals-equalsIgnoreCase

```
46      s="nmacit@gelisim.edu.tr";
47      if (s.equalsIgnoreCase("NMACİT@gelisim.edu.tr"))
48          System.out.println("Aynı değer");
49      else System.out.println("Farklı değer");
50      /*Bir metni başka bir metinle karşılaştırmak için
51       * equals ya da equalsIgnoreCase metodlarını kullanabilirsiniz.
52       * equalsIgnoreCase metodu, karşılaştırılan metni öncelikle küçük
53       * harflere çevirir ve ardından karşılaştırma yapar.
54       * Böylelikle büyük-küçük harf durumundan doğacak sorunun
55       *  önüne geçilmiş olur.*/
```

Console

<terminated> D1StringManipulasyonlar [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.e

```
Aynı değer
```

STRİNGLER-toLowerCase-toUpperCase

```
56    s="nmacit@gelisim.edu.tr";
57    System.out.println(s.toUpperCase());
58    /*Bir metni küçük harflere çevirmek için toLowerCase,
59     * büyük harflere çevirmek içinse  toUpperCase metodlarını
60     *  kullanabilirsiniz:*/
```

Console

<terminated> D1StringManipulasyonlar [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-09

```
NMACİT@GELİSİM.EDU.TR
```

STRİNGLER-trim

› Bir metin içerisindeki boşluklardan kurtulmak için **trim** metodu kullanılır. **trim** metodu, metnin sonunda ve başında yer alan boşlukları yok ederken kelime aralarındaki boşluklara dokunmaz. Bu metod özellikle kullanıcıların formlar ile gönderdiği verilerde (isim, e-posta vs.) veritabanına kayıt etmeden önce kullanılmalıdır. Böylece olası kullanıcı hataları biraz olsun azaltılabilir ve ileride doğabilecek karşılaştırma hatalarının önüne geçilmiş olur.

```
62    s="    nmacit@gelisim.edu.tr      ";
63    System.out.println("başladı/"+s.trim()+"/bitti");
```

Console

<terminated> D1StringManipulasyonlar [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_

```
başladı/nmacit@gelisim.edu.tr/bitti
```

STRİNGLER-split

```
74   s="nmacit@gelisim.edu.tr";
75   String [] dizi=s.split("@");
76   System.out.println(dizi[0]);
77   System.out.println(dizi[1]);
78   /*Bir metni parçalara bölmek için
79    * (örneğin boşluklara göre ayırıp kelimeleri ayıklama)
80    * split metodu kullanılır. split metodu regex bir ifade
81    * içerisindeki kurallara göre String değişkenini böler
82    * ve bir String dizisi haline getirir.*/
```

Console

<terminated> D1StringManipulasyonlar [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v2025

```
nmacit
gelisim.edu.tr
```

STRİNGLER-replaceall

//replace() methodu, 2.parametredekini 1.nin yerine koyar.
//Küçük büyük harfe duyarladır

```
84  s="nmacit@gelisim.edu.tr";
85  System.out.println(s.replaceAll("gelisim", "gmail"));
86  //replace() methodu, 2.parametredekini 1.nin yerine koyar.
87  //Küçük büyük harfe duyarladır
```

Console

<terminated> D1StringManipulasyonlar [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v2

```
nmacit@gmail.edu.tr
```

STRİNGLER-replace

```
89          s="nmacit@gelisim.edu.tr";
90          System.out.println(s.replace(".", ""));
91          //string içerisindeki . ları boşluk ile değiştirdi.
92          //Yani sildi
```

Console

<terminated> D1StringManipulasyonlar [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v202505(

```
nmacit@gelisimedutr
```

STRİNGLER-replace

Regex (Regular Expression), metinler üzerinde belirli desenlere uyan kısımları bulmak ya da değiştirmek için kullanılan özel ifadelerdir. Aşağıda sık kullanılan bazı örnekler verilmiştir:

REGEX	ANLAMI	ÖRNEK KOD	ÇIKTI
[0-9]	Tüm rakamlar	`"a1b2".replaceAll("[0-9]", "*")`	a*b*
[a-z]	Küçük harfler	`"Java".replaceAll("[a-z]", "*")`	J**A
[A-Z]	Büyük harfler	`"Java".replaceAll("[A-Z]", "*")`	*ava
[a-zA-Z]	Tüm harfler (küçük + büyük)	`"J2a!".replaceAll("[a-zA-Z]", "*")`	*2*!
[a-zA-Z0-9]	Harfler ve rakamlar	`"J2a!".replaceAll("[a-zA-Z0-9]", "*")`	***!
\\p{Punct}	Noktalama işaretleri	`"Hello!" .replaceAll("\\p{Punct}", "*")`	Hello*
[aeiouAEIOU]	Sesli harfler	`"Java".replaceAll("[aeiouAEIOU]", "*")`	J*v*
[axy]	Belirli harfler (a, x, y)	`"Syntax".replaceAll("[axy]", "*")`	S*nt**

STRİNGLER-replace

Regex (Regular Expression), metinler üzerinde belirli desenlere uyan kısımları bulmak ya da değiştirmek için kullanılan özel ifadelerdir. Aşağıda sık kullanılan bazı örnekler verilmiştir:

REGEX	ANLAMI	ÖRNEK KOD	ÇIKTI
[A-Ga-g]	A–G ve a–g arasındaki harfler	`"Hello".replaceAll("[A-Ga-g]", "*")`	H*llo
[^a-z]	Küçük harfler dışındaki karakterler	`"Java123".replaceAll("[^a-z]", "*")`	*ava***
[^a-zA-Z]	Harf dışındaki karakterler	`"Java123!".replaceAll("[^a-zA-Z]", "*")`	Java***
\\s	Boşluk karakteri	`"a b".replaceAll("\\s", "*")`	a*b
\\S	Boşluk harici karakterler	`"a b".replaceAll("\\S", "*")`	* *
\\d	Rakamlar	`"a1b2".replaceAll("\\d", "*")`	a*b*
\\D	Rakam haricindeki karakterler	`"a1b2".replaceAll("\\D", "*")`	*1*2

STRİNGLER-replace

Regex (Regular Expression), metinler üzerinde belirli desenlere uyan kısımları bulmak ya da değiştirmek için kullanılan özel ifadelerdir. Aşağıda sık kullanılan bazı örnekler verilmiştir:

```
95    s="nmacit@gelisim.edu.tr";
96    System.out.println(s.replaceAll("[^a-zA-z]",""));
97    //Büyük küçük harfler dışında herşeyi sil
98    /*
```

Console

<terminated> D1StringManipulasyonlar [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502

```
nmacitgelisimedutr
```

REGEX	ANLAMI	ÖRNEK KOD	ÇIKTI
[0-9]	Tüm rakamlar	`"a1b2".replaceAll("[0-9]", "*")`	a*b*
[a-z]	Küçük harfler	`"Java".replaceAll("[a-z]", "*")`	J**A
[A-Z]	Büyük harfler	`"Java".replaceAll("[A-Z]", "*")`	*ava
[a-zA-Z]	Tüm harfler (küçük + büyük)	`"J2a!".replaceAll("[a-zA-Z]", "*")`	*2*!
[a-zA-Z0-9]	Harfler ve rakamlar	`"J2a!".replaceAll("[a-zA-Z0-9]", "*")`	***!
\\p{Punct}	Noktalama işaretleri	`"Hello!" .replaceAll("\\p{Punct}", "*")`	Hello*
[aeiouAEIOU]	Sesli harfler	`"Java".replaceAll("[aeiouAEIOU]", "*")`	J*v*
[axy]	Belirli harfler (a, x, y)	`"Syntax".replaceAll("[axy]", "*")`	S*nt**
[A-Ga-g]	A–G ve a–g arasındaki harfler	`"Hello".replaceAll("[A-Ga-g]", "*")`	H*llo
[^a-z]	Küçük harfler dışındaki karakterler	`"Java123".replaceAll("[^a-z]", "*")`	*ava***
[^a-zA-Z]	Harf dışındaki karakterler	`"Java123!".replaceAll("[^a-zA-Z]", "*")`	Java***
\\s	Boşluk karakteri	`"a b".replaceAll("\\s", "*")`	a*b
\\S	Boşluk harici karakterler	`"a b".replaceAll("\\S", "*")`	* *
\\d	Rakamlar	`"a1b2".replaceAll("\\d", "*")`	a*b*
\\D	Rakam haricindeki karakterler	`"a1b2".replaceAll("\\D", "*")`	*1*2

Soru

- **Kurallar:**
- Şifre en az **8 karakter** uzunluğunda olmalıdır.
- Şifre **boşluk karakteri** içermemelidir.
- Şifre en az bir **büyük harf** içermelidir.
- Şifre en az bir **küçük harf** içermelidir.
- Şifre en az bir **rakam** içermelidir.
- Şifre en az bir **sembol** (noktalama işareti veya özel karakter) içermelidir.
- Program, kullanıcıya hangi kural(lar)ı sağlamadığını ekrana yazmalı; eğer tüm kurallar sağlanıyorsa **"Şifre geçerli"** mesajı vermelidir.

```java
Scanner klavye = new Scanner(System.in);
System.out.print("Şifrenizi giriniz: ");       String sifre = klavye.nextLine();


boolean uzunlukKontrol = sifre.length() >= 8; // i) Uzunluk en az 8 karakter olmalı
boolean boslukKontrol = !sifre.contains(" "); // ii) Boşluk olmamalı
boolean buyukHarfKontrol = sifre.replaceAll("[^A-Z]", "").length() > 0; // iii) En az bir büyük harf olmalı
boolean kucukHarfKontrol = sifre.replaceAll("[^a-z]", "").length() > 0; // iv) En az bir küçük harf olmalı
boolean rakamKontrol = sifre.replaceAll("[^0-9]", "").length() > 0; // v) En az bir rakam olmalı
boolean sembolKontrol = sifre.replaceAll("[^\\p{Punct}]", "").length() > 0; // vi) En az bir sembol olmalı

// Kullanıcıya eksik olan kısımları bildir
if (!uzunlukKontrol) {System.out.println("Şifre en az 8 karakter olmalıdır.");}
if (!boslukKontrol) {System.out.println("Şifre boşluk içermemelidir.");}
if (!buyukHarfKontrol) {System.out.println("Şifre en az bir büyük harf içermelidir.");}
if (!kucukHarfKontrol) {System.out.println("Şifre en az bir küçük harf içermelidir.");}
if (!rakamKontrol) {System.out.println("Şifre en az bir rakam içermelidir.");}
if (!sembolKontrol) {System.out.println("Şifre en az bir sembol içermelidir.");}


// Tüm koşullar sağlanıyorsa geçerli şifre
boolean gecerliMi = uzunlukKontrol && boslukKontrol && buyukHarfKontrol && kucukHarfKontrol && rakamKontrol &&
sembolKontrol;


if (gecerliMi) {System.out.println("✅ Şifre geçerli.");}
else { System.out.println("❌ Şifre geçerli değil."); }
```

Soru

Ürünlerin Fiyatlarını Toplama

Bir mağazada satışa sunulan ürünlerin fiyat bilgileri **para birimi işareti ($)** ile birlikte String tipinde tutulmaktadır.

Aşağıdaki ürünlerin toplam fiyatını hesaplayan bir Java programı yazınız:

- Telefon: $299.50
- Tablet: $450.75

Beklenen çıktı:

Toplam fiyat: 750.25

Soru

```
16          String telefon = "$299.50";
17          String tablet = "$450.75";
18
19          // Para işaretini kaldır ve double'a çevir
20          double telefonFiyat = Double.parseDouble(telefon.replace("$", ""));
21          double tabletFiyat = Double.parseDouble(tablet.replace("$", ""));
22
23          double toplam = telefonFiyat + tabletFiyat;
24
```

Console

<terminated> D2Soru1 [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe (28 Eyl 2025 09:17:51 – 09:17:51 ela

```
Toplam fiyat: 750.25
```

STRİNGLER-isempty

//isEmpty() metodu sadece hiçliği kontrol eder (true verir).

- Sadece karakter uzunluğunu kontrol eder.
- Yani length() == 0 demektir.
- Eğer String içinde hiç karakter yoksa true döner.

```
4
5       public static void main(String[] args) {
6           String s1 = "";
7           System.out.println(s1.isEmpty());  // true
8           System.out.println(s1.isBlank());  // true
9
10
```

Console

<terminated> D3StringManipulasyon [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_

```
true
true
```

STRİNGLER-isblank

//isBlank() metodu ise hem hiçliği hem de space'i kontrol eder(true verir)

- Java 11 ile geldi.
- String boş mu ya da sadece **boşluklardan (space, tab, \n vs)** mı oluşuyor onu kontrol eder.
- Yani " " (sadece boşluk) için true döner, ama isEmpty() burada false döner.

```
10          String s2 = "   ";
11          System.out.println(s2.isEmpty());
12          // false (çünkü 3 karakter var)
13          System.out.println(s2.isBlank());
14          // true  (çünkü sadece boşluklardan oluşuyor)
15
```

Console

<terminated> D3StringManipulasyon [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.

```
true
true
```

STRİNGLERDE EŞİTLİK KONTROLÜ

Java'da String ifadelerin eşitliğini kontrol ederken dikkat etmemiz gereken önemli bir nokta vardır.
•== operatörü, iki String'in bellekteki adresini (referansını) karşılaştırır.
•Yani değişkenler aynı nesneyi gösteriyor mu ona bakar.
•equals() metodu ise, String'in içindeki değerleri (içeriğini) karşılaştırır.
•İki String'in yazılışı aynıysa true döner.

```
String str1 = new String("merhaba");
String str2 = new String("merhaba");
String str3 = str1;

System.out.println(str1 == str2);
// false → çünkü farklı adresler
System.out.println(str1.equals(str2));
// true  → çünkü içerikleri aynı
System.out.println(str1 == str3);
// true  → çünkü aynı adresi gösteriyorlar
```

✅ Özet

- `==` → Referans (adres) kontrolü yapar.
- `equals()` → Değer (içerik) kontrolü yapar.