

# NESNE TABANLI PROGRAMLAMA

JAVA

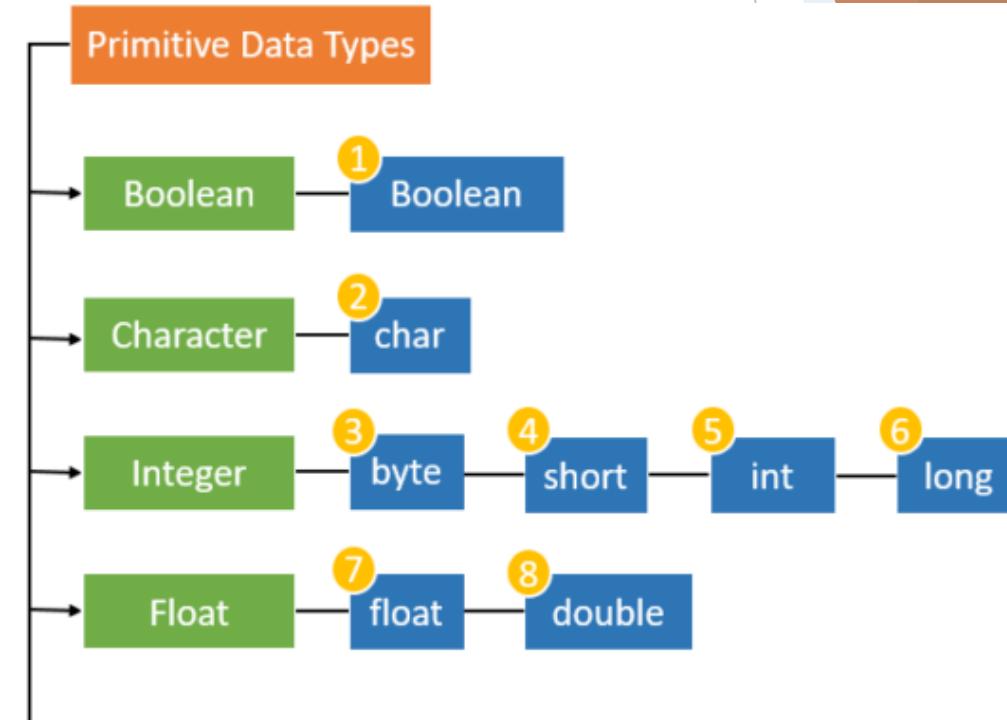
# iÇİNDEKİLER

- ▶ Wrapper Class
- ▶ Tür Dönüşümü
- ▶ Dışarıdan Veri Almak
- ▶ Scope(Faaliyet Alanı)

# WRAPPER CLASS

# Wrapper Class

- ▶ **Wrapper Class lar**
- ▶ **İlkel Veri Tipleri**
- ▶ ilkel değişkenlere bazen metodlar ve çeşitli özellikler eklemek isteriz
- ▶ wrapper class java da primitive değişkenleri object veri tipine dönüştürmek için kullanılır.
- ▶ Java da bulunan koleksiyonlar(arraylist vs) genellikle non primitive değişkenleri kullandıkları için bu işleme ihtiyaç duyuyoruz
- ▶ En önemlisi de girilen veri üzerinde metod kullanımı açar



# Wrapper Class

```
3 public class D1WrapperClass {  
4  
5     public static void main(String[] args) {  
6         int a=5; //primitive  
7         Integer b=4; //wrapper class  
8 //wrapper'lar büyük harfle başlar çünkü class'tırlar  
9 // primitive          : char      - boolean - byte - short - int      - long   - float  - double  
10 //wrapper class       : Character - Boolean - Byte - Short - Integer - Long  - Float  - Double  
11         b.  
12     }  
13 }  
14 }  
15 }
```

Console X  
<terminated> De14Mantiksal  
true

The screenshot shows a Java code editor with the following code:

```
3 public class D1WrapperClass {  
4  
5     public static void main(String[] args) {  
6         int a=5; //primitive  
7         Integer b=4; //wrapper class  
8 //wrapper'lar büyük harfle başlar çünkü class'tırlar  
9 // primitive          : char      - boolean - byte - short - int      - long   - float  - double  
10 //wrapper class       : Character - Boolean - Byte - Short - Integer - Long  - Float  - Double  
11         b.  
12     }  
13 }  
14 }  
15 }
```

A code completion dropdown is open at line 11, position 'b.', listing methods for the Integer type:

- byteValue() : byte - Integer
- compareTo(Integer anotherInteger) : int - Integer
- describeConstable() : Optional<Integer> - Integer
- doubleValue() : double - Integer
- equals(Object obj) : boolean - Integer
- floatValue() : float - Integer
- getClass() : Class<?> - Object
- hashCode() : int - Integer
- intValue() : int - Integer
- longValue() : long - Integer
- notify() : void - Object

A tooltip for the byteValue() method is displayed, providing the following information:

Returns the value of this Integer as a byte after a narrowing primitive conversion.

**Overrides:** [byteValue\(\)](#) in [Number](#)

**Returns:**  
the numeric value represented by this object after conversion to type byte.

**@jls**  
5.1.3 Narrowing Primitive Conversion

Press 'Ctrl+Space' to show Template Proposals

Press 'Tab' from proposal table or click for focus

- ▶ b. Yazdığımızda wrapper olan b ye ait bütün metodlar listelendi.

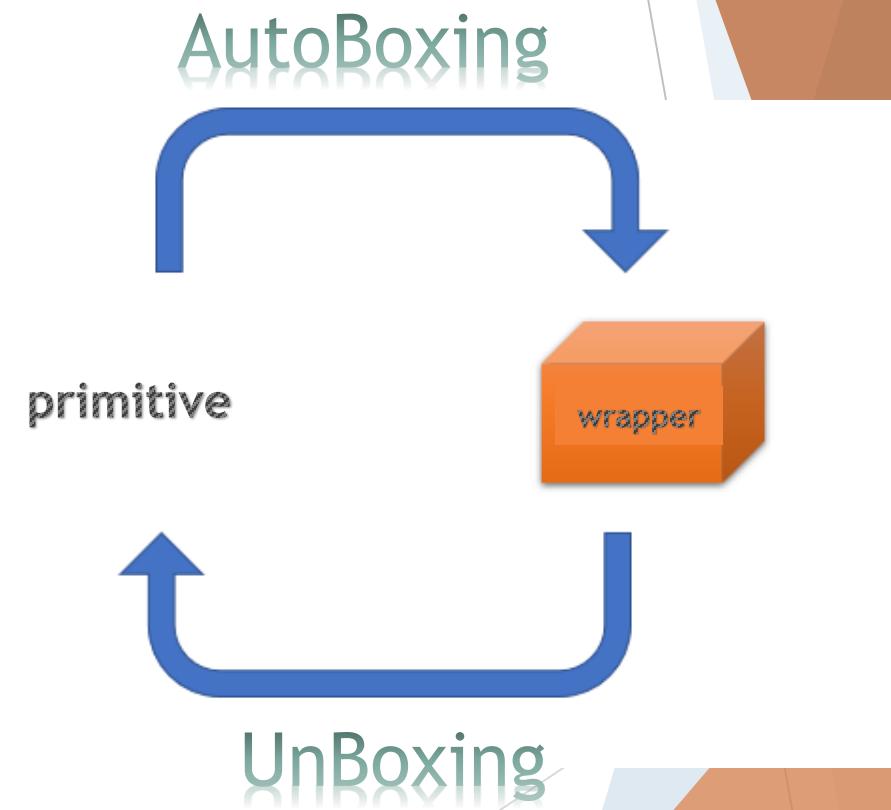
# Wrapper Class

```
3 public class D1WrapperClass {  
4  
5     public static void main(String[] args) {  
6         int a=5; //primitive  
7         Integer b=4; //wrapper class  
8 //wrapper'lar büyük harfle başlar çünkü class'tırlar  
9 // primitive          : char           - boolean - byte - short - int           - long   - float   - double  
10 //wrapper class      : Character      - Boolean - Byte - Short - Integer        - Long    - Float    - Double  
11     a.|  
12 }  
13 }  
14 }  
15 }
```

- ▶ a. Yazdığımızda ilkel veri tipi olduğu için hiçbir metodu yok.

# Wrapper Class -Primitive Tür Dönüşümü

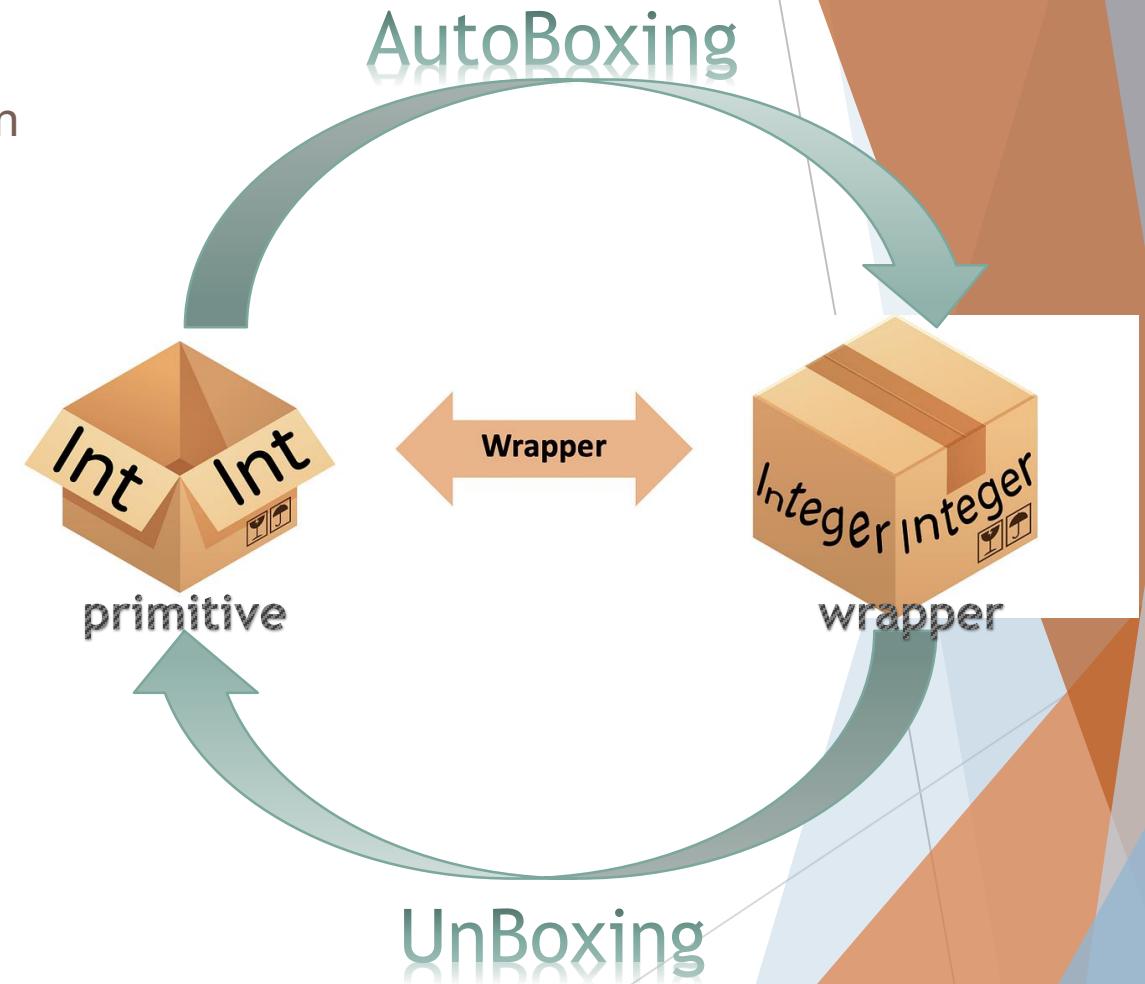
- ▶ Autoboxing - Unboxing
- ▶ Java'da primitive(ilkel) türler ve wrapper sınıfları arasında otomatik dönüşüm yapabilen autoboxing ve unboxing mekanizmaları vardır



# Wrapper Class -Primitive Tür Dönüşümü

- Autoboxing, primitive bir türü otomatik olarak onun wrapper sınıfına çevirir. Örneğin, long'u Long'a dönüştürür

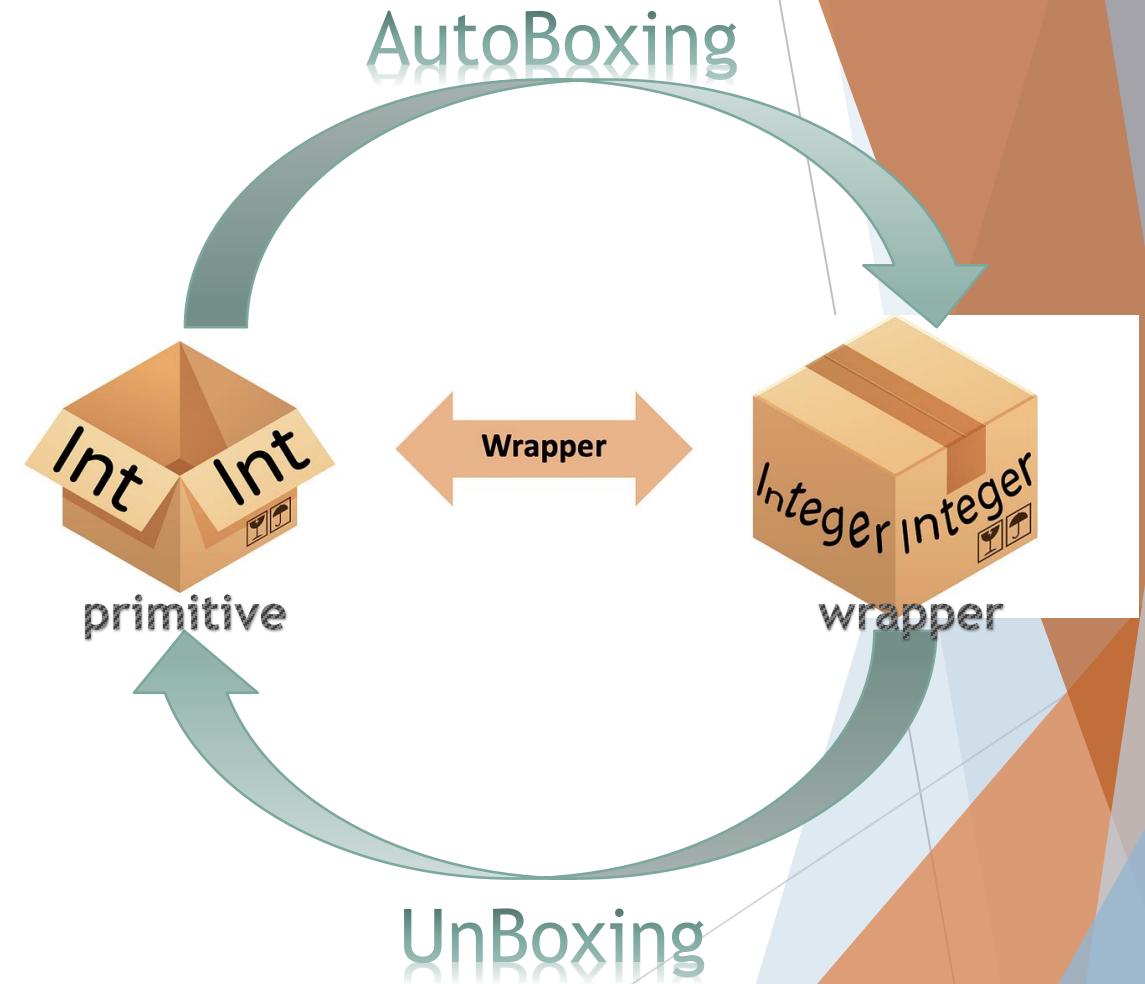
```
//AUTOBOXING  
int x=4;  
Integer y=x; //wrapper<=primitive
```



# Wrapper Class -Primitive Tür Dönüşümü

- Unboxing ise tam tersi bir işlem yapar, Long'u long'a dönüştürür. (Wrapper'i primitive'e dönüştürür) İkisi de otomatik gerçekleşir.

```
//UNBOXING
Integer k=4;
int l=k; //primitive<=wrapper
```



# TÜR DÖNÜŞÜMLERİ

## TYPE CASTING

Primitive data tiplerinin birbirine dönüştürülmesine denir.

# Tür Dönüşümü

Bazı durumlarda elimizdeki veri tipi işimize yaramaz ve onu ihtiyacımız olan veri tipine dönüştürmek zorunda kalırız.

```
public static void main(String[] args) {  
    // TODO code application logic here  
    int a;  
    float b=3.14f;  
    a=b;  
    System.out.println(a);  
}
```

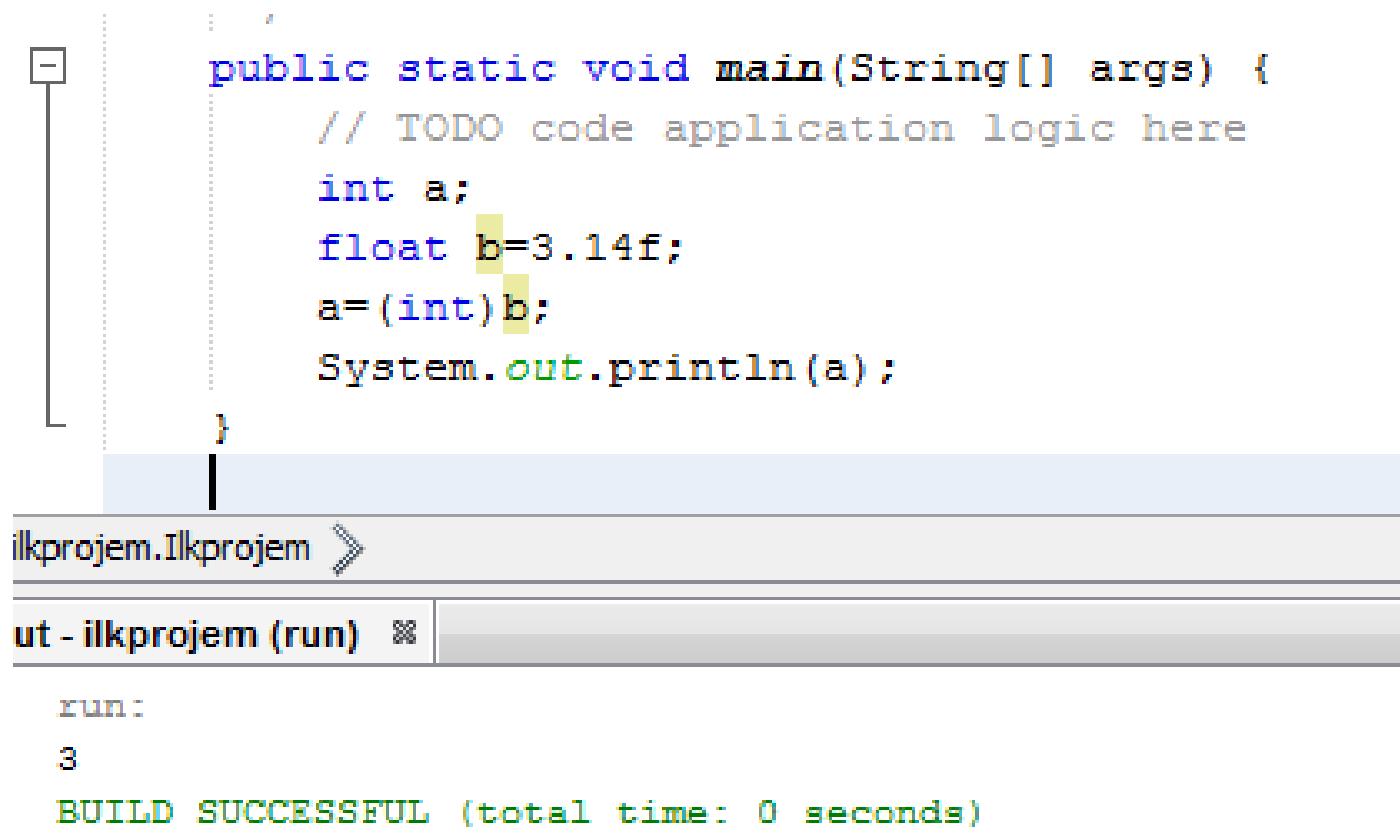
# Tür Dönüşümü Yöntemi

- ▶ Bu işlemi yapmak için tür dönüşümü yapmak zorundayız.
- ▶ **Veri\_tipi degisken=(Veri\_tipi)dönüşecek\_veri;**

```
public static void main(String[] args) {  
    // TODO code application logic here  
    int a;  
    float b=3.14f;  
    a=(int)b;  
    System.out.println(a);  
}
```

# Tür Dönüşümü Yöntemi

- int sayılar virgülden sonraki kısmını tutamayacağı için o kısmını atacaktır.



The screenshot shows a Java code editor with the following code:

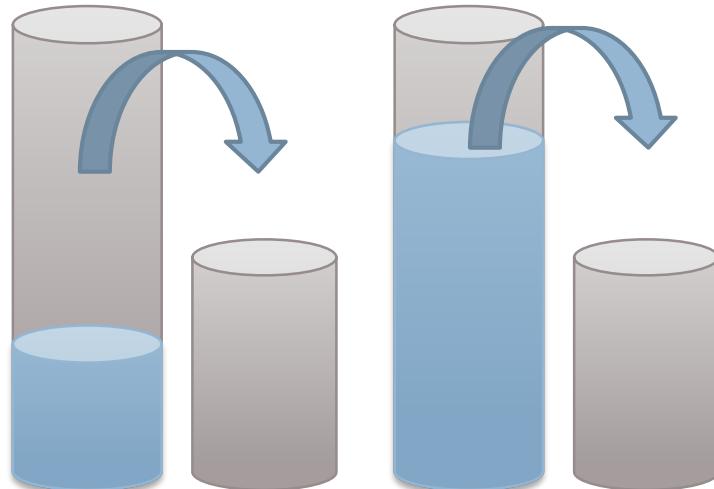
```
public static void main(String[] args) {
    // TODO code application logic here
    int a;
    float b=3.14f;
    a=(int)b;
    System.out.println(a);
}
```

Below the code editor, there are two tabs: "ilkprojem.Ilkprojem" and "ut - ilkprojem (run)". The "ut - ilkprojem (run)" tab is active, showing the output of the run:

```
run:
3
BUILD SUCCESSFUL (total time: 0 seconds)
```

# Tür Dönüşümü

Autowidenig



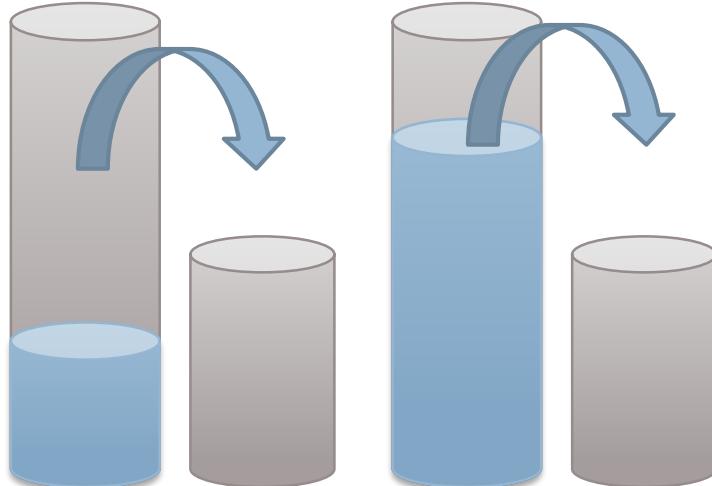
Explicit Narrowing

→ Küçük kaptaki veriyi  
büyük kaba döksek neler  
olabilir?

→ Büyük kaptaki veriyi  
küçük kaba döksek neler  
olabilir?

# Tür Dönüşümü

Autowidenig



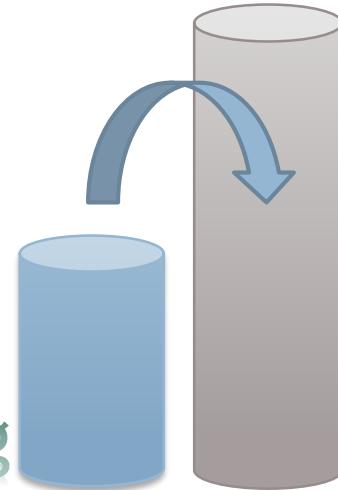
Explicit Narrowing

- ▶ Küçük kaptaki veriyi büyük kaba döksek neler olabilir?
- ▶ Sıkıntı yok

- ▶ Büyük kaptaki veriyi küçük kaba döksek neler olabilir?
- ▶ Veri kaybı olabilir

# Tür Dönüşümü

Autowidening

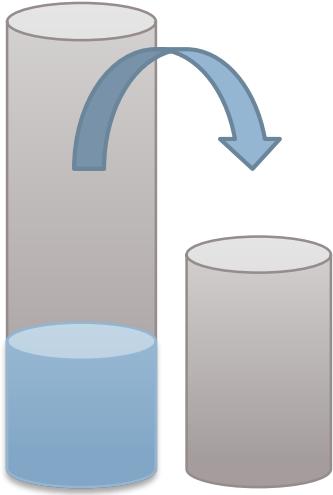


- Küçük data type'larini büyük data type'lara çevirmeyi Java otomatik olarak yapabilir.
- Bu işleme “AutoWidening” (Otomatik genişletme) denir

```
byte a=45;  
int b;  
b=a;      //bilinçsiz tür dönüşümü  
b=(int)a; //bilinçli tür dönüşümü
```

//Küçük kaptaki veriyi büyük kaba döksek neler olabilir?  
//sıkıntı çıkmaz veriler aktarılır.  
//hata mesajı bile almayız

# Tür Dönüşü



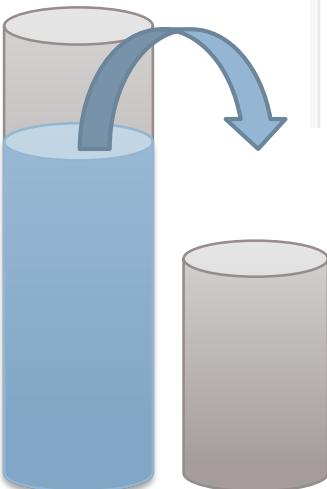
```
byte y;
int x=15;
y=x; //hata alındı
/*
Type mismatch: cannot convert from int to byte
* 3 quick fixes available:
  Add cast to 'byte'
  Change type of 'y' to 'int'
  Change type of 'x' to 'byte'
Press 'F2' for focus
```

```
22
23 byte y;
24 int x=15;
25 //y=x; //hata alındı
26 y=(byte)x;
27 System.out.println(y);
```

Büyük data type' larini küçük data type' larina çevirmek riskli bir iştir.

Veri kaybı oluşabilir.  
Java bu işi otomatik olarak yapmaz. Bu işlemi kod yazanlar yapar.  
Bu işleme "Explicit Narrowing"(Açıkta Daraltma)denir.

# Tür Dönüşümü



```
30 byte n;  
31 int m=3152;  
32 n=(byte)m;  
33 System.out.println(n);  
34 //veri kaybı var  
35
```

```
Console X  
<terminated> D2TurDonusumu [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.ecli  
80
```

Büyük data type' larini küçük data type' larina çevirmek riskli bir iştir.

Veri kaybı oluşabilir.  
Java bu işi otomatik olarak yapmaz. Bu işlemi kod yazanlar yapar.  
Bu işleme "Explicit Narrowing"(Açıkta Daraltma)denir.

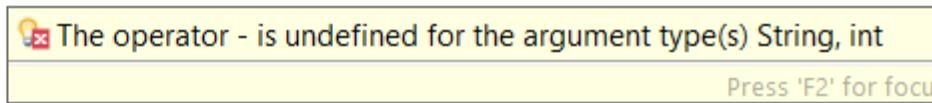
# Java da Tür Dönüşümünü Ne Zaman Kullanırız

- ▶ 1- Daha Büyük Bir Veri Türünden Daha Küçüğe Geçiş:  
Örneğin, double bir sayıyı int'e çevirmek istediğinizde.  
Bu tarz dönüşümlerde çoğu zaman veri kaybı riski vardır.
- ▶ 2- Farklı Türler Arasında Uyum Sağlama:  
Çeşitli tipteki değişkenleri veya nesneleri birlikte kullanmak gerekiğinde dönüşüm gerekebilir.
- ▶ 3- Nesne Yönelimli Programlamada (OOP) Kalıtım İlişkileri:  
Bir alt sınıf (child) nesnesini üst sınıf (parent) tipine dönüştürmek  
ya da bunun tersini yapmak tip dönüşümüne örnektir.
- ▶ 4- API ve Kütüphane Kullanımlarında:  
Bazı hazır metodlar sadece belirli türleri kabul eder.  
Örneğin, int parametre bekleyen bir metoda float değer göndermek için dönüşüm  
yapılmalıdır.
- ▶ 5- Sayısal İşlemlerde:  
Birden fazla farklı veri türünü aynı matematiksel işlemde kullanmak istediğinizde, tip  
dönüşümü devreye girer.

# String Tür Dönüşümü

- ▶ "12" → ?
- ▶ 12 → ?

```
///SAYISAL VERİLERİ STRING E ÇEVİRME
String xString="123";
System.out.println(xString-3);
}
```



# String Tür Dönüşümü

```
55  ///SAYISAL VERİLERİ STRING E ÇEVİRME  
56  String xString="123";  
57  int xInteger=Integer.parseInt(xString);  
58  System.out.println(xInteger-3);  
59
```

Console X

```
<terminated> D2TurDonusumu [Java Application] C:\Users\BIDB\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre
```

120

# String Tür Dönüşümü

```
String xString="123";
int y=xString;
```

Type mismatch: cannot convert from String to int

2 quick fixes available:

- ⚡ [Change type of 'y' to 'String'](#)
- ⚡ [Change type of 'xString' to 'int'](#)

Press 'F2' for focus

# String Tür Dönüşümü

```
60 int yInteger=Integer.valueOf(xString);  
61 System.out.println(y);  
62
```

Console X

```
<terminated> D2TurDonusumu [Java Application] C:\Users\BIDB\.p2\pool\plugins\org.eclipse.justj.openjdk.ho
```

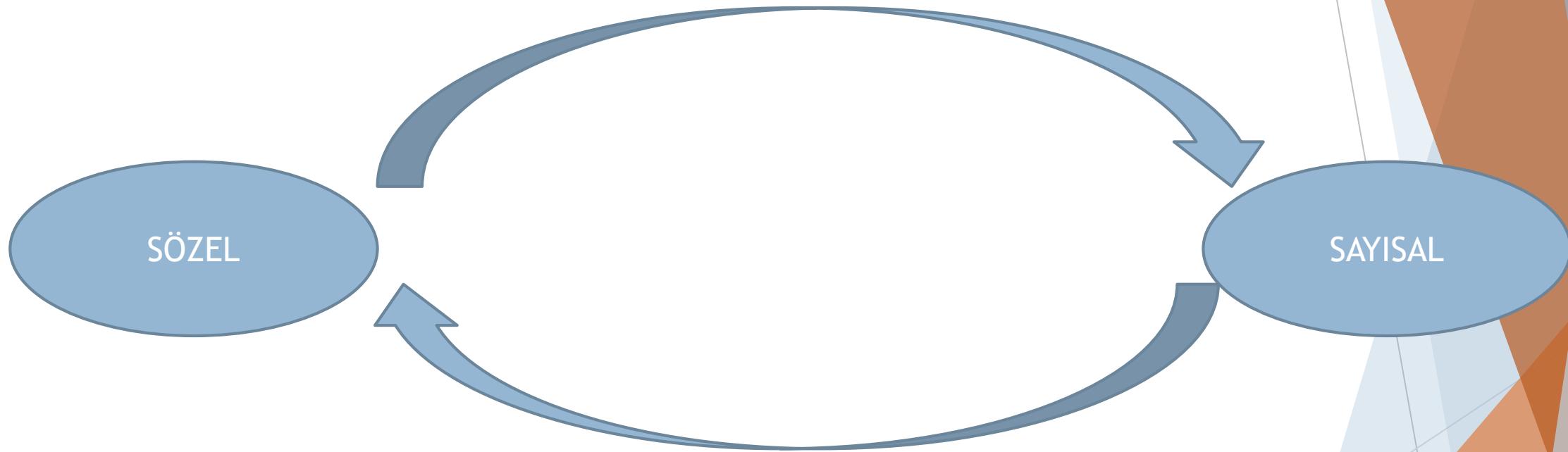
15

# String Tür Dönüşümü

- ▶ **Veri\_tipi** degisken=**veri\_tipi.valueOf(string\_deger);**
- ▶ **Veri\_tipi** degisken=**veri\_tipi.parseveri\_tipi(string\_deger);**

Tür Dönüşüm **Integer.parseInt(sözelveri);**

**Integer.valueOf(sözelveri);**



**String.valueOf(sayısalveri);**

**Integer.toString(sayısalveri);**

# DIŞARIDAN VERİ ALMA

Scanner Sınıfı

# Dışarıdan Veri Almak

```
public class Sube2
{
    public static void main(String[] args)
    {
        System.out.println("adınızı girin");
    }
}
```

ibe2.Sube2 > main >

- sube2 (run) x

run:

adınızı girin

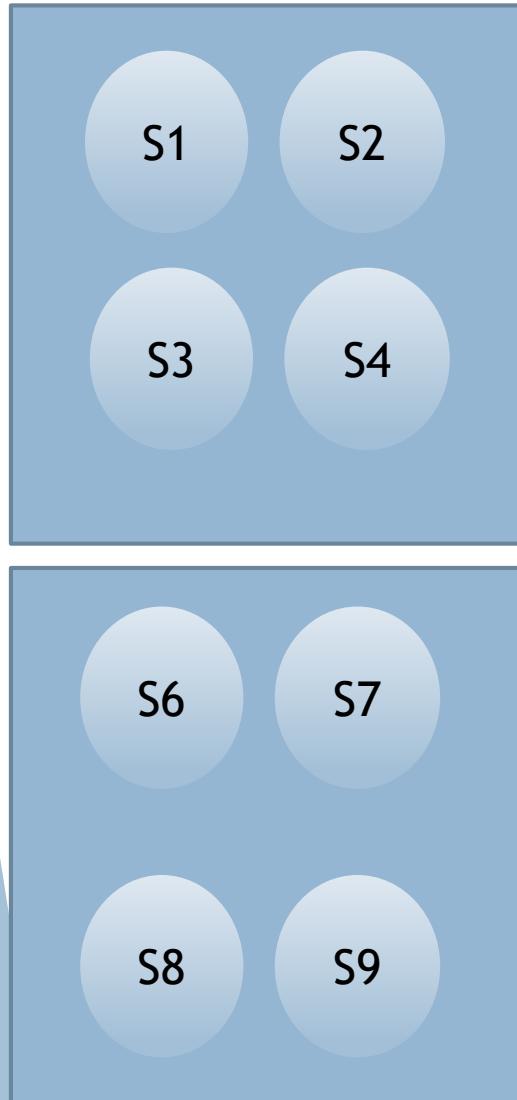
BUILD SUCCESSFUL (total time: 0 seconds)

# Paket Kavramı



- ▶ Sınıflar paketlerin alt birimleridir.
- ▶ Java bir çok sınıfından meydana gelir
- ▶ Bu sınıflar çeşitli paketlerde toplanmıştır.
- ▶ Paketler Java ya da Javax ön ekleri ile başlarlar.
- ▶ Ön eki;  
Java olanlar → Standart API ye bağlıdır.  
Javax olanlar → sonradan ekleme sınıflarıdır. (extension)
- ▶ ***java.paketAdi.sinifAdi***
- ▶ ***java.util.scanner;***

# Paket Kavramı



- ▶ Bir sınıf kendi ile aynı pakette olmayan bir sınıfı kullanacaksa,
- ▶ Kendinden önce o sınıfı import etmeli.
  
- ▶ `import paketAdi.sınıfAdi;`
  - ▶ `import java.util;`
  
- ▶ `import paketAdi.sınıfAdi.*;`  
// paketteki tüm sınıfları çağrıır.
- ▶ `import java.util.*;`

# Dışarıdan Veri Almak

- Giriş çıkış işlemleri için `java.util.scanner` sınıfı kullanılır.
- Bu sınıfın ait yöntemler kullanılabilmesi için sınıfı projenize dahil etmeniz gereklidir.

```
package sube2;
import java.util.Scanner;
public class Sube2
{
    public static void main(String[] args)
    {
        System.out.println("adınızı girin");
        Scanner tara=new Scanner(System.in);
        String s1=tara.nextLine();
    }
}
```

# Dışarıdan Veri Almak

```
package sube2;
import java.util.Scanner;
public class Sube2
{
    public static void main(String[] args)
    {
        System.out.println("adınızı girin");
        Scanner tara=new Scanner(System.in);
        String s1=tara.nextLine();
    }
}
```

```
ibe2.Sube2 > main > s1 >
:- sube2 (run) x
```

```
run:
adınızı girin
naciye macit sez|
```

## Dışarıdan Veri Almak

- ▶ `next();` klavyeden girilen ilk kelimeyi boşluğa kadar okur.
- ▶ `nextLine();` klavyeden girilen tüm ifadeyi boşluklar dahil okur.
- ▶ `findLine(".").charAt(0);` klavyeden girilen ifadeyi karakterlerine ayırarak okur.
- ▶ `nextInt();` klavyeden girilen ifadeyi integer tipine çevirir.
- ▶ `nextFloat();` klavyeden girilen ifadeyi float tipine çevirir.
- ▶ `nextDouble();` klavyeden girilen ifadeyi double tipine çevirir.

## Dışarıdan Veri Almak

- ▶ **System.in** klavyeden veri girileceğinde
- ▶ **System.out** işlem ve mesajların ekrana yazdırılacağındır
- ▶ **System.err** hata mesajlarında kullanılmaktadır mesajları kırmızı yazdırır.

# Sorun!!

- Sayısal bir veri aldıktan sonra sözel veriyi almadan programı bitiriyor ya da diğer işlemlere geçiyor

```
5 public class D3ScannerSoru {      ➤ NEDEN?  
6  
7     public static void main(String[] args) {  
8         Scanner input = new Scanner(System.in);  
9         System.out.println("Yasinizi giriniz");  
10        int age = input.nextInt(); //Sayısal bir girdi aldım  
11  
12        System.out.println("Adinizi soyadinizi giriniz");  
13        String name = input.nextLine();  
14    }  
15  
16
```

The screenshot shows the Eclipse IDE interface. The code editor displays the Java code above. Below it, the 'Console' tab is active, showing the application's output. The output starts with the prompt 'Yasinizi giriniz', followed by the user input '23', and then the prompt 'Adinizi soyadinizi giriniz'. The console window has a standard title bar with tabs and icons.

```
Console X  
<terminated> D3ScannerSoru [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'
```

Yasinizi giriniz

23

Adinizi soyadinizi giriniz

# Sorun!!

- ▶ Sayısal bir veri aldıktan sonra sözel veriyi almadan programı bitiriyor ya da diğer işlemlere geçiyor
- ▶ **NEDEN?**  
☞ **Scanner** ile kullanıcıdan giriş alırken `nextLine()` gibi metodlar kullanıldığında, satır sonu (Enter ile oluşan `\n` veya `\r\n`) girdide kalabiliyor.

Bu durum sonraki okuma işlemlerinde boş değer alınmasına ya da beklenmeyen sonuçlara yol açıyor.  
Bunu çözmek için:

# ÇÖZÜM 1

- ▶ Bu kod, Java'da Scanner sınıfı ile metin tabanlı kullanıcı girişleri alınırken kullanılır.
- ▶ `scanner.skip()` metodu, verilen desene uyan kısmı girdiden atlamayı sağlar.
- ▶ Yani, belirtilen kalıba denk gelen bölüm yok sayar. Burada kullanılan desen `"(\r\n|[\n\r\u2028\u2029\u0085])?"` şeklindedir ve satır sonu karakterlerini temsil eder.
- ▶ Deseni açıklayalım:
  - ▶ - `\r\n` : Windows işletim sisteminde satır sonu ifadesi
  - ▶ - `[\n\r\u2028\u2029\u0085]` : Unix/Linux, macOS ve bazı diğer ortamlarda kullanılan satır sonu karakterlerini belirtir.
  - ▶ - `()?` : Bu karakter grubunun isteğe bağlı olduğunu gösterir, yani satır sonu karakteri bulunabilir ya da hiç olmayabilir.

# ÇÖZÜM 1

```
5 public class D3ScannerSorunu {  
6  
7     public static void main(String[] args) {  
8         Scanner input = new Scanner(System.in);  
9         System.out.println("Yasinizi giriniz");  
10        int age = input.nextInt(); //Sayisal bir girdi aldım  
11        input.skip("\r\n|\n\r\u2028\u2029\u0085)?");  
12        System.out.println("Adinizi soyadinizi giriniz");  
13        String name = input.nextLine();  
14    }  
15}
```

Console X

<terminated> D3ScannerSorunu [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 (27)

Yasinizi giriniz

24

Adinizi soyadinizi giriniz  
naciye

## ÇÖZÜM 2

```
System.out.println("Yasinizi giriniz");
int age12 = input.nextInt(); //Sayisal bir girdi aldım
Scanner input12 = new Scanner(System.in);
// Yeni bir Scanner nesnesi oluşturun
System.out.println("Adinizi soyadinizi giriniz");
String name12 = input12.nextLine();
/*ÇÖZÜM 2
 * /*İki farklı Scanner nesnesi oluşturarak sorunu çözebilirsiniz.
 * İlk Scanner nesnesiyle tamsayı değerini alın ve
 * ikinci Scanner nesnesiyle satırı okuyun.*/

```

## ÇÖZÜM 3

```
System.out.println("Çözüm 2 için kodlar");
Scanner input2 = new Scanner(System.in);
System.out.println("Yasinizi giriniz");
int age2 = input2.nextInt(); //Sayisal bir girdi aldım
input2.nextLine(); // Dummy nextLine() call
System.out.println("Adinizi soyadinizi giriniz");
String name2 = input2.nextLine();
/*ÇÖZÜM 3:
 * Dummy nextLine() Çağrısının nextInt()'ın ardından
 * bir nextLine() çağrı yaparak yeni satır karakterini
 * temizleyebilirsiniz.*/
```

## ÇÖZÜM 4

```
System.out.println("Çözüm 3 için kodlar");
Scanner input3 = new Scanner(System.in);
System.out.println("Yasinizi giriniz");
int age3 = Integer.parseInt(input3.nextLine());
// Tüm satırı okuyun ve tamsayıya dönüştürün
System.out.println("Adinizi soyadinizi giriniz");
String name3 = input3.nextLine();
/*ÇÖZÜM 4 Tüm girdiyi nextLine()
 * ile okuyun ve ardından bu satırı parçalayın.*/
```

# SCOPE FAALİYET ALANLARI

SCOPE

# SCOPE

```
1 package hafta2;  
2  
3 public class D4Scope {  
4  
5     public static void main(String[] args) {  
6         int a=4;  
7         int b=6,c=7;  
8         char k=78;  
9         int a=4;  
10        //tek scope alanı içerisinde aynı değişken tanımlanamaz  
11    }  
12 }  
13  
14 } Exception in thread "main" java.lang.RuntimeException: Uncompilable source code -  
variable a is already defined in method main(java.lang.String[])  
at ilkprojem.Ilkprojem.main(Ilkprojem.java:21)
```

C:\Users\NaciyeMacitSezikli\AppData\Local\NetBeans\Cache\8.1\executor-snippets\run.xml:53: Java returned: 1  
BUILD FAILED (total time: 2 seconds)

# SCOPE

```
public class Sube2
{
    //seviye 1
    public static void main(String[] args)
    {
        //seviye 2
        {
            //seviye 3- 1.blok
        }
        {
            //seviye 3- 2. blok
        }
    }
}
```

# SCOPE

```
1 package hafta2;  
2  
3 public class D5Scope1 {  
4  
5     public static void main(String[] args) {  
6         {  
7             int a=5;  
8         }  
9         {  
10            int a=5;  
11        }  
12        //Farklı scope alanlarındaki değişkenler  
13        //aynı isimle tanımlansalarda  
14        //birbirinden farklıdır.  
15    }  
16  
17 }
```

## SCOPE

```
1 package hafta2;  
2  
3 public class D5Scope1 {  
4  
5     public static void main(String[] args) {  
6         {  
7             int a=5;  
8         }  
9         //scope alanı içindeki veriye dışardan erişilemez  
10        System.out.println(a);  
11    {  
12        int a=5;  
13    }  
14  
15    //Farklı scope alanlarındaki değişkenler  
16    //aynı isimle tanımlansalarda  
17    //birbirinden farklıdır.  
18 }  
--
```



## SCOPE

```
1 package hafta2;  
2  
3 public class D6Scope2 {  
4  
5     public static void main(String[] args) {  
6         int a=5; //scope alanları dışında tanımlı  
7         {  
8             System.out.println(a);  
9             //icerdeki scope dan ulaşılır  
10        }  
11        System.out.println(a);  
12        // aynı scope dan da erişilir  
13        {  
14            System.out.println(a);  
15        }  
16    }  
17}  
18  
19 }
```

```
public class Sube2
{
    //seviye 1
    public static void main(String[] args)
    { //seviye 2
        int a=4;
        {
            System.out.println(a);
            int b=3; //seviye 3- 1. blok
        }
        System.out.println(a);
        System.out.println(b);
        {
            System.out.println(a); //seviye 3- 2. blok
            System.out.println(b);
        }
    }
}
```

# SCOPE

## (Faaliyet Alanları)

{

{

}

{

}

}