

# ALGORİTMA VE PROGRAMLAMA I

## HAFTA 6 GİRİŞ/ÇIKIŞ KOMUTLARI VE OPERATÖRLER

# Ders İçeriği: Hafta 6

- Veri Çıkış Komutları
- Veri Giriş Komutları
- Operatörler
- Örnekler

# Veri Çıkış Komutları

- Programların çıktısını ekrana yazdırmak için kullanılır.
- Java'da giriş/çıkış işlemleri için System sınıfı fonksiyonları kullanılır.
- System sınıfı, java.lang paketi içindedir ve tüm Java programları bu paketi otomatik olarak yükler; bu nedenle import java.lang; satırına gerek yoktur.
- Bu yüzden her Java programı System sınıfını otomatik olarak içerir.

# Veri Çıkış Komutları

- System.out değişkeni standart çıkışı sağlar.
- Bilgisayarların standart çıkış birimi ekrandır.
- System sınıfı içerisinde tanımlı print(), println() ve printf() metotları ile ekrana çıktı göndermek için kullanılır.
  - System.out.print(); //imleç verinin bitiminde bekler
  - System.out.println(); //imleç bir alt satır başında bekler
  - System.out.printf(); //çıktıyı biçimli yazdırmada kullanılır

# Veri Çıkış Komutları

## ➤ Özel Karakterler

- Bazı özel karakterler (escape characters) sözel ifadeler arasında kullanılarak formatlı çıktılar elde edilebilir.

| Özel Karakter       | Anlamı                            |
|---------------------|-----------------------------------|
| <code>\n</code>     | Yeni satır                        |
| <code>\t</code>     | Tab                               |
| <code>\b</code>     | Geri tuşu                         |
| <code>\r</code>     | Satırbaşı                         |
| <code>\\</code>     | Ters bölü çizgisi                 |
| <code>\'</code>     | Tekli kesme işareti               |
| <code>\"</code>     | Çifte kesme (tırnak)              |
| <code>\ddd</code>   | Octal (Sekiz tabanlı)             |
| <code>\xdd</code>   | Hexadecimal (onaltı tabanlı)      |
| <code>\udddd</code> | Unicode karakter Kodu (16 bitlik) |

# Veri Çıkış Komutları

## ➤ Özel Karakterler

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Birinci satır\nİkinci satır");  
    }  
}
```

Birinci satır  
İkinci satır

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Ad\tSoyad\tYaş");  
        System.out.println("Ali\tVeli\t25");  
    }  
}
```

| Ad  | Soyad | Yaş |
|-----|-------|-----|
| Ali | Veli  | 25  |

# Veri Çıkış Komutları

## ➤ Sözel İfadelerin Yazdırılması

- Sözel ifadeler yani metinler yazdırılırken mutlaka çift tırnak ("" ) içerisine yazılmalıdır.

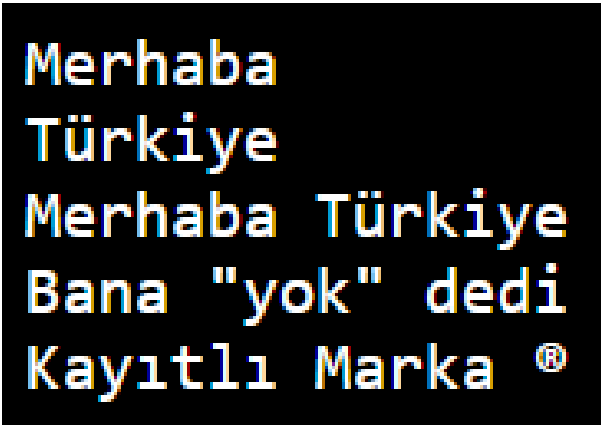
```
public class Main {  
    public static void main(String[] args) {  
  
        System.out.println("Bu cümle ekrana yazdırılır!");  
        System.out.println(Bu cümle hataya neden olur.);  
  
    }  
}
```

# Veri Çıkış Komutları

## ➤ Sözel İfadelerin Yazdırılması

- Özel karakterler ile formatlı çıktılar elde edebilir veya farklı karakterleri yazdırabiliriz.

```
System.out.println("Merhaba\nTürkiye");  
System.out.println("Merhaba\tTürkiye");  
System.out.println("Bana \"yok\" dedi");  
System.out.println("Bana \"yok\" dedi");    \\Hata!  
System.out.println("Kayıtlı Marka \u00AE");
```



```
Merhaba  
Türkiye  
Merhaba Türkiye  
Bana "yok" dedi  
Kayıtlı Marka ®
```

# Veri Çıkış Komutları

## ➤ Sayısal İfadelerin Yazdırılması

- Sayısal değerler ekrana yazdırılırken doğrudan `print()` metodunun içine yazılır.
- Hatta metot içerisinde matematiksel işlemler yapılabilir ve ekrana işlem sonucu yazdırılır.

```
System.out.println(358);  
System.out.println(3 + 3);  
System.out.println(2 * 5);
```

```
358  
6  
10
```

# Veri Çıkış Komutları

## ➤ Değişkenlerin Yazdırılması

- Değişkenler yazdırılırken değişken ismi print() ifadesinin içine yazılır.

```
int a=650;  
String b="Java Metni";  
System.out.println(a);  
System.out.println(b);
```

```
650  
Java Metni
```

- Farklı ifadeleri yan yana yazdırmak için + işareti kullanılır.

```
System.out.println(a + b);  
System.out.println(a+" "+b);  
System.out.println(a+" nin karesi = "+a*a);
```

```
650Java Metni  
650 Java Metni  
650 nin karesi = 422500
```

# Veri Çıkış Komutları

## ➤ Java'da Çıktının Biçimlendirilmesi

- printf ve format metotları kullanılır.
- Biçimlendirme için özel karakterler ile birlikte dönüşüm (conversion) ve biçim belirleyici (format specifiers) karakterler kullanılır.
- Kullanım şekli:

➤ % [Değişken\_damgası\$][Bayrak][Genişlik][.Hassasiyet]Değişim Karakteri

### Örnek

```
public class Main {  
    public static void main(String[] args) {  
        System.out.printf("Merhaba %s! Bugün %d derece.\n", "Dünya", 25);  
    }  
}
```

# Veri Çıkış Komutları

## ➤ Java'da Çıktının Biçimlendirilmesi

- Anlamları:
- %           Dönüşümü başlatan simge
- []           Bu parantezlerin içleri isteğe bağlıdır. Gerekli ise yazılır.
- Değişken\_damgası\$   Değişken değeri için yer tutucu; değişkenin çıktıda nereye konuşlanacağını belirtir.
- Bayraklar: - , ^ , 0 , #
- Değişim Karakteri: d,o,x,f,g,a,c,s,b,h,n harfleri ile özel biçimlendirme sağlanır.

# Veri Çıkış Komutları

## ➤ Java'da Çıktının Biçimlendirilmesi

- Özel biçimlendirme karakterleri

| Karakter | Anlamı           | Karakter | Anlamı               |
|----------|------------------|----------|----------------------|
| b        | Boolean          | s        | String               |
| d        | Desimal          | c        | Karakter             |
| h        | Hex. Küçük harf  | tr ve tT | 12 ve 24 zaman       |
| H        | Hex. Büyük harf  | tc       | Tarih ve zaman       |
| f        | Ondalık sayı     | tl ve tM | Saat ve dakika       |
| e        | Ondalık bilimsel | tb ve tm | Ay adı ve ay sıra no |
| g        | Ondalık kısa     | tD       | Kısa Tarih           |

# Veri Çıkış Komutları

## ➤ Java'da Çıktının Biçimlendirilmesi

- Özel biçimlendirme karakterleri

```
public class Main {  
    public static void main(String[] args) {  
        int sayi = 124;  
        System.out.printf("Sayı: %06d", sayi);  
    }  
}
```

Sayı: 000124

Bu örnekte %06d, 6 haneli bir sayı ve önünde sıfırlarla doldurulmuş biçim kullanır.

# Veri Çıkış Komutları

## ➤ Java'da Çıktının Biçimlendirilmesi

```
int x=10; double d=7.8;
System.out.printf("x=%d \n", x);
System.out.printf("d=%f \n", d);
System.out.printf("x=%d \t d=%f \n", x, d);
System.out.printf("x=%10d d=%30f \n", x, d);
System.out.printf("d=%40.3f ", d);
```

x=10

$d=7,800000$

x=10      d=7,8000000

x= 10 d= 7,800000

d= 7,800

# Veri Giriş Komutları

- Java'da veri girişi yani klavyeden veri girişi için genellikle Scanner sınıfı kullanılır.
- java.util paketi içinde tanımlı bu sınıfı kullanmak için önce paketin programa eklenmesi gerekir (import).
- Sonrasında Scanner sınıfına ait bir nesne üretilir ve bu nesne kullanılarak veri girişi sağlanır.
- Değişken tipine göre veri okumak ve ilgili değişkene aktarmak için Scanner sınıfının farklı metotlarından faydalanılır.

# Veri Giriş Komutları

| Metot         | Anlamı                                                 |
|---------------|--------------------------------------------------------|
| nextBoolean() | boolean tipinde veri girişleri için                    |
| nextByte()    | byte tipinde veri girişleri için                       |
| nextDouble()  | double tipinde veri girişleri için                     |
| nextFloat()   | float tipinde veri girişleri için                      |
| nextInt()     | int tipinde veri girişleri için                        |
| nextLong()    | long tipinde veri girişleri için                       |
| nextShort()   | short tipinde veri girişleri için                      |
| next()        | string tipinde veri girişleri için (ilk boşluğa kadar) |
| nextLine()    | string tipinde veri girişleri için (tüm satırı alır)   |

# Veri Giriş Komutları

```
import java.util.Scanner;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Scanner oku = new Scanner(System.in);
```

```
        System.out.print("Yaşınız :");
```

```
        byte yas=oku.nextByte();
```

```
        System.out.println(yas);
```

```
        System.out.print("Adınız :");
```

```
        String isim=oku.next();
```

```
        System.out.println(isim);
```

```
        System.out.print("Ondalık sayı giriniz :");
```

```
        float sayi=oku.nextFloat();
```

```
        System.out.println(sayi);
```

```
        System.out.print("Sayı giriniz :");
```

```
        int ikili =oku.nextInt( radix: 2); // 2 lik tabanda al
```

```
        System.out.println(ikili);
```

```
    }
```

```
}
```

Yaşınız :19

19

Adınız :Sıla

Sıla

Ondalık sayı giriniz :45

45.0

Sayı giriniz :1001110

78

# Veri Giriş Komutları

```
import java.util.Scanner;
```

```
class Main {  
    public static void main(String[] args) {  
  
        Scanner oku = new Scanner(System.in);  
  
        System.out.println("Cinsiyetinizi Giriniz:");  
        String metin=oku.next();  
        char karakter=metin.charAt(0);  
        System.out.println(karakter);  
  
    }  
}
```

```
Cinsiyetinizi Giriniz:  
Kadın  
K
```

# Operatörler

## ➤ Aritmetik Operatörler

| Operatör | İşlevi       | Örnek    |
|----------|--------------|----------|
| +        | Toplama      | $x + y$  |
| -        | Çıkarma      | $x - y$  |
| *        | Çarpma       | $x * y$  |
| /        | Bölme        | $x / y$  |
| %        | Mod (Kalan)  | $x \% y$ |
| ++       | Bir arttırma | $++x$    |
| --       | Bir azaltma  | $--x$    |

# Operatörler

## ➤ Aritmetik Operatörler

| Operatör | Açıklama         | Örnek    |
|----------|------------------|----------|
| +        | Toplama          | $a + b$  |
| -        | Çıkarma          | $a - b$  |
| *        | Çarpma           | $a * b$  |
| /        | Bölme            | $a / b$  |
| %        | Mod alma (kalan) | $a \% b$ |

# Operatörler

## ➤ Atama Operatörleri

| Operatör | Açıklama         | Örnek  |
|----------|------------------|--------|
| =        | Atama            | a = 5  |
| +=       | Toplayarak atama | a += 2 |
| -=       | Çıkararak atama  | a -= 3 |
| *=       | Çarparak atama   | a *= 4 |
| /=       | Bölerek atama    | a /= 2 |
| %=       | Mod alarak atama | a %= 3 |

# Operatörler

- Karşılaştırma Operatörleri
- Değerler birbiri ile karşılaştırır ve sonuç true veya false değeri alır.

| Operatör           | Açıklama            | Örnek                  |
|--------------------|---------------------|------------------------|
| <code>==</code>    | Eşit mi?            | <code>a == b</code>    |
| <code>!=</code>    | Eşit değil mi?      | <code>a != b</code>    |
| <code>&gt;</code>  | Büyük mü?           | <code>a &gt; b</code>  |
| <code>&lt;</code>  | Küçük mü?           | <code>a &lt; b</code>  |
| <code>&gt;=</code> | Büyük veya eşit mi? | <code>a &gt;= b</code> |
| <code>&lt;=</code> | Küçük veya eşit mi? | <code>a &lt;= b</code> |

# Operatörler

## ➤ Mantıksal Operatörler

- Karşılaştırma operatörlerini birleştirmek için kullanılabilir.
- Değişkenler arası mantıksal duruma göre sonuç true veya false değeri alır.

| Operatör | Anlamı                                      | Örnek                      |
|----------|---------------------------------------------|----------------------------|
| &&       | Her iki ifade de doğru ise sonuç true       | $x < 5 \ \&\& \ x < 10$    |
|          | İfadelerden bir tanesi doğru ise sonuç true | $x < 5 \    \ x < 4$       |
| !        | Mantıksal ifadenin değilini alır            | $!(x < 5 \ \&\& \ x < 10)$ |

# Operatörler

## ➤ Bit Düzeyinde Operatörler

- Bit seviyesinde işlem yapmak için kullanılan operatörlerdir.

| Operatör | Anlamı          | Örnek        |
|----------|-----------------|--------------|
|          | veya            | $a   b$      |
| &        | ve              | $a \& b$     |
| ^        | Özel veya (xor) | $a \wedge b$ |
| ~        | değil           | $\sim a$     |
| <<       | Sola kaydırma   | $a << 2$     |
| >>       | Sağa kaydırma   | $a >> 4$     |

# Operatörler

## ➤ Bit Düzeyinde Operatörler

| Operatör | Açıklama                      | Örnek    | Sonuç (İkili) | Sonuç (Ondalık) |
|----------|-------------------------------|----------|---------------|-----------------|
| &        | Bit düzeyinde VE (AND)        | 5 & 3    | 0101 & 0011   | 0001            |
|          | Bit düzeyinde VEYA (OR)       | 5   3    | 0101   0011   | 0111            |
| ^        | Bit düzeyinde Özel VEYA (XOR) | 5 ^ 3    | 0101 ^ 0011   | 0110            |
| ~        | Bit düzeyinde DEĞİL (NOT)     | ~5       | ~0101         | 1010            |
| <<       | Sol kaydırma                  | 5 << 1   | 0101 << 1     | 1010            |
| >>       | Sağ kaydırma                  | 5 >> 1   | 0101 >> 1     | 0010            |
| >>>      | Sıfır doldurarak sağ kaydırma | -5 >>> 1 | ...1011 >>> 1 | ...111101       |

```
int a = 5; // 0101 (ikili)
int b = 3; // 0011 (ikili)
int result = a & b; // 0001 (ikili) = 1 (ondalık)
System.out.println(result);
```

```
int a = 5; // 0101 (ikili)
int result = a << 1; // 1010 (ikili) = 10 (ondalık)
System.out.println(result);
```

```
int a = 5; // 0101 (ikili)
int result = a >> 1; // 0010 (ikili) = 2 (ondalık)
System.out.println(result);
```

# Operatörler

## ➤ Atama Operatörleri

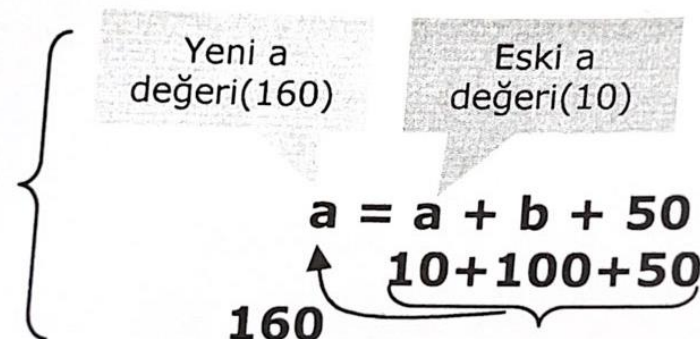
- Atamalarda eşitliğin soluna tek bir değişken gelir ve başka bir değişken veya sayı gelemmez.
- Eşitliğin sağında ise başka değişken, sabit veya operatör olabilir.

$a + b = 120;$  //Hatalı

$a = a + b - 120;$  //Geçerli

- Atamalarda önce eşitliğin sağ tarafı hesaplanır ve sonuç sol tarafa atanır.

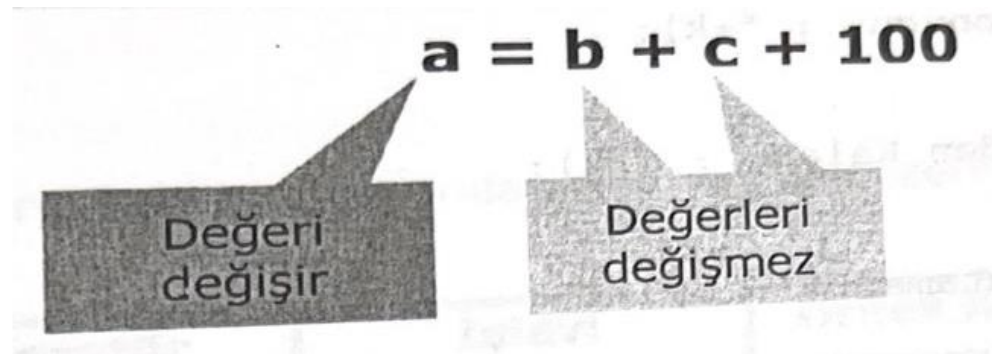
Programlamada  
değer atama  
yapısı gösterimi



# Operatörler

## ➤ Atama Operatörleri

- Atamalarda eşitliğin sağındaki değişkenlerin değeri ++ ve -- operatörleri dışında hiçbir zaman değişmez.



# Operatörler

## ➤ Atama Operatörleri

- ++ ve – operatörleri atama ifadelerinde kullanılırken sağda ve solda olmaları arasında fark vardır.

- Örnek:

```
int a,b=100;
```

```
a=b++;      //a=100 ve b=101
```

```
a=++b;      //a=101 ve b=101
```

```
a=b--;      //a=100 ve b=99
```

```
a=--b;      //a=99 ve b=99
```

# Operatörler

## ➤ Bit Operatörleri

| a | b | a   b | a & b | a ^ b | ~a |
|---|---|-------|-------|-------|----|
| 0 | 0 | 0     | 0     | 0     | 1  |
| 1 | 0 | 1     | 0     | 1     | 0  |
| 0 | 1 | 1     | 0     | 1     | 1  |
| 1 | 1 | 1     | 1     | 0     | 0  |

| a        | b        | a   b    | a & b    | a ^ b    | ~a       |
|----------|----------|----------|----------|----------|----------|
| 00101010 | 00001111 | 00101111 | 00001010 | 00100101 | 11010101 |

42

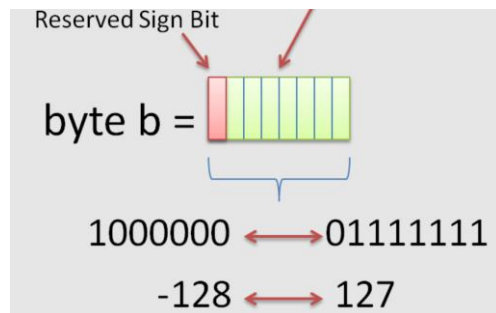
15

47

10

37

-43



Nasıl -43 çıkıyor? İlk bit işaret biti eğer 1 olursa negatif anlamına geliyor.

Örnekte geri kalan değerler toplamı **88**, 128'den çıkarılması gerekiyor.  $128 - 85 = 43$ . İşaret bitiyle birlikte -43

# Operatörler

## ➤ Bit Operatörleri

```
class Main {  
    public static void main(String[] args) {  
  
        byte a = 42;  
        byte b = 15;                                47  
        System.out.println(a | b);                  10  
        System.out.println(a & b);                  37  
        System.out.println(a ^ b);                  -43  
        System.out.println(~ a);                    168  
        System.out.println(42 << 2);  
    }  
}
```

168 hesabında işaret biti bilgisi verilmediğinden işaret biti göz ardı edilerek hesaplanmıştır.

# Operatörler

## ➤ Operatörlerde İşlem Önceliği

| İşlem Önceliği | Operatör           | İşlem Önceliği | Operatör           |
|----------------|--------------------|----------------|--------------------|
| 1              | x++, x--, ., []    | 8              | &                  |
| 2              | ++x, --x, ~, -x, ! | 9              | ^                  |
| 3              | *, /, %            | 10             |                    |
| 4              | +, -               | 11             | &&                 |
| 5              | <<, >>             | 12             |                    |
| 6              | <, >, <=, >=       | 13             | ?:                 |
| 7              | ==, !=             | 14             | Atama operatörleri |

# Örnekler

## ➤ Örnek 1:

```
class Main {  
    public static void main(String[] args) {  
  
        int x = 10;        int y = 3;  
        System.out.println("x + y = " + (x + y) );  
        System.out.println("x - y = " + (x - y));  
        System.out.println("x * y = " + x * y);  
        System.out.println("x / y = " + x / y);  
        System.out.println("x % y = " + x % y);  
        System.out.println("x++    = " + x++ );  
        System.out.println("y--    = " + y-- );  
  
    }  
}
```

x + y = 13

x - y = 7

x \* y = 30

x / y = 3

x % y = 1

x++ = 10

y-- = 3

# Örnekler

## ➤ Örnek 2:

```
public static void main(String[] args) {  
    int x = 10;           float y = 3.0f;           double z = 3.0;  
    System.out.println("x + y = " + (x + y) );  
    System.out.println("x - y = " + (x - y));  
    System.out.println("x * y = " + x * y);  
    System.out.println("x / y = " + x / y);  
    System.out.println();  
    System.out.println("x + z = " + (x + z) );  
    System.out.println("x - z = " + (x - z));  
    System.out.println("x * z = " + x * z);  
    System.out.println("x / z = " + x / z);  
}
```

x + y = 13.0  
x - y = 7.0  
x \* y = 30.0  
x / y = 3.3333333

x + z = 13.0  
x - z = 7.0  
x \* z = 30.0  
x / z = 3.3333333333333335

# Örnekler

## ➤ Örnek 3:

```
public static void main(String[] args) {  
  
    int x = 5; int y = 3; int z = 8; int t = 11;  
    System.out.println("--x * --x = " + --x * --x) ;  
    System.out.println(++y * ++y = " + ++y * ++y) ;  
  
    System.out.println("z-- * z-- = " + z-- * z--) ;  
    System.out.println("t++ * t++ = " + t++ * t++) ;  
  
}
```

|     |   |     |   |     |    |       |   |     |
|-----|---|-----|---|-----|----|-------|---|-----|
| --x | * | --x | = | 12  | \\ | 4*3   | = | 12  |
| ++y | * | ++y | = | 20  | \\ | 4*5   | = | 20  |
| z-- | * | z-- | = | 56  | \\ | 8*7   | = | 56  |
| t++ | * | t++ | = | 132 | \\ | 11*12 | = | 132 |

# Örnekler

## ➤ Örnek 4:

```
public static void main(String[] args) {  
    int x=12;                                15  
    int y=3;                                7  
    int z=4;                                7  
    int toplu = (x+y);                      44  
    y=--x;                                  2  
    int carp = (y*z);                       0  
    int cikar = (x-z);  
    x-=3;  
    int bol = (x/z);  
    y=z++;  
    int kalan = (x%y);  
    System.out.println(toplu);  
    System.out.println(cikar);  
    System.out.println(carp);  
    System.out.println(bol);  
    System.out.println(kalan);  
}
```