

# **BASİT VERİ TÜRLERİ, STRINGLER, VE KONSOL GİRİŞ/ÇIKIŞ**

**BMÜ-101 ALGORİTMA VE PROGRAMLAMA**

**İLHAN AYDIN**

**2013-2014 GÜZ DÖNEMİ**

**FIRAT ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

# AMAÇLAR

- İlkel veri türleri ile tanışmak (sayılar, karakterler, vb.)
- Atama deyimleri ve ifadeler
- Stringler
- Ekranı bilgi yazdırma ve basit klavyeden giriş komutları
- İlkel türler ve ifadeler
- String Sınıfı
- Klavye ve Ekran Giriş/Çıkışı

# İLKEL TÜRLER VE İFADELER

- Değişkenler
- Java tanımlayıcıları
- İlkel türler
- Atama deyimleri
- Özelleştirilmiş atama operatörleri
- Basit ekran çıkışı
- Basit ekran girişi
- Sabitler
- Atama deyimleri
- Aritmetik işlemler
- Parantezler ve Öncelik kuralları
- Arttırma ve azaltma operatörü

# DEĞİŞKENLER VE DEĞERLERİ

- *Değişkenler harf, yazı, sayı gibi verileri tutar.*
  - Değişkenleri veriyi depolamak için yerler olarak düşünün.
  - Hafıza yerleri olarak uygulanır.
- Bir değişken ile depolanan veri onun değeri olarak çağrılır.
  - Değer hafızada saklanır.
- Bir değişkenin değeri değiştirilebilir.

# ORNEK BİR PROGRAMIN ÇALIŞTIRILMASI

```
public class ComputeArea {  
    /** Main method */  
    public static void main(String[] args) {  
        double yaricap;  
        double alan;  
  
        // yarıçapa değer ata  
        yaricap = 20;  
  
        // Alanı hesapla  
        alan = yaricap * yaricap * 3.14159;  
  
        // Sonuçları göster  
        System.out.println("Yarıçapı " + yaricap  
        + " olan çemberin alanı " + alan);  
    }  
}
```

Yaricap için  
hafıza alanı

yaricap

Değer yok

## ORNEK BİR PROGRAMIN ÇALIŞTIRILMASI

```
public class ComputeArea {  
    /** Main method */  
    public static void main(String[] args) {  
        double yaricap;  
        double alan;  
  
        // yarıçapa değer ata  
        yaricap = 20;  
  
        // Alanı hesapla  
        alan = yaricap * yaricap * 3.14159;  
  
        // Sonuçları göster  
        System.out.println(«Yarıçapı " + yaricap + "  
        olan çemberin alanı " + alan);  
    }  
}
```

Hafıza

yaricap

Değer yok

alan

Değer yok

Alan için hafıza  
alanı

## ORNEK BİR PROGRAMIN ÇALIŞTIRILMASI

```
public class ComputeArea {  
    /** Main method */  
    public static void main(String[] args) {  
        double yaricap;  
        double alan;  
  
        // yarıçapa değer ata  
        yaricap = 20;  
  
        // Alanı hesapla  
        alan = yaricap * yaricap * 3.14159;  
  
        // Sonuçları göster  
        System.out.println(«Yarıçapı " + yaricap + "  
        olan çemberin alanı " + alan);  
    }  
}
```

yaricap

alan

Yaricapa 20 ata

20

Değer yok

## ORNEK BİR PROGRAMIN ÇALIŞTIRILMASI

```
public class ComputeArea {  
    /** Main method */  
    public static void main(String[] args) {  
        double yaricap;  
        double alan;  
  
        // yarıçapa değer ata  
        yaricap = 20;  
  
        // Alanı hesapla  
        alan = yaricap * yaricap * 3.14159;  
  
        // Sonuçları göster  
        System.out.println(«Yarıçapı " + yaricap + "  
        olan çemberin alanı " + alan);  
    }  
}
```

Hafıza

yaricap

20

alan

1256.636

Alanı hesapla ve alan  
değişkenine ata



# ORNEK BİR PROGRAMIN ÇALIŞTIRILMASI

```
public class ComputeArea {  
    /** Main method */  
    public static void main(String[] args) {  
        double yaricap;  
        double alan;  
  
        // yarıçapa değer ata  
        yaricap = 20;  
  
        // Alanı hesapla  
        alan = yaricap * yaricap * 3.14159;  
  
        // Sonuçları göster  
        System.out.println(«Yarıçapı " + yaricap + "  
        olan çemberin alanı " + alan);  
    }  
}
```

Hafıza

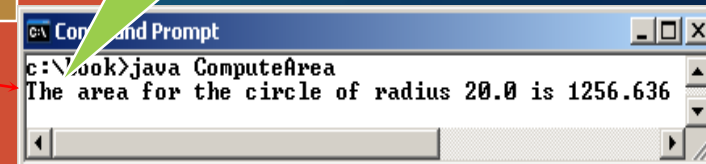
yaricap

20

alan

1256.636

Sonucu ekranda göster



```
CA Command Prompt  
c:\book>java ComputeArea  
The area for the circle of radius 20.0 is 1256.636
```

# DEĞİŞKENLER VE DEĞERLERİ

- Değişkenler

yaricap

alan

- Atanan değerler

yaricap = 20;

alan = yaricap \* yaricap \* 3.14159;

## DEĞİŞKENLERİN TANIMLANMASI VE İSİMLENDİRİLMESİ

► Değişkenlere anlamlı isim verin h ve s yerine hiz ve sayi gibi isimlendirmeler kullanın.

► Bir değişken tanımladığınızda ismini ve tipini verin.

```
int sayi, eggsPerBasket;
```

```
double yaricap;
```

► Bir değişkenin tipi onun tutacağı değer türünü belirler (int→tamsayı, double→noktalı sayı, char→karakter, vb.).

► Bir değişken kullanılmadan önce tanımlanmalıdır.

# SÖZ DİZİMİ VE ÖRNEKLER

- Söz dizimi

```
tip degsikenadi_1, degiskenadi_2, ...;
```

- Örnekler

```
int sayi, toplam;
```

```
double mesafe, ortalama;
```

```
char cinsiyet;
```

- Bir sınıf tipi nesnelerin bir sınıfı için kullanılır ve hem ver hem de metot içerir..
  - “Bugün hava güzel” String türünden sınıfın bir değeridir.
- İlkel bir tür sayı ve karakter türünden değer tutar.
  - `int`, `double`, **ve** `char` ilkel türlerdir.

## İSİMLENDİRME KURALLARI

- Sınıf türleri büyük harf ile başlar (örn. `String`).
- İlkel türler küçük harf ile başlar (örn. `int`, `char`, `double`, `byte`).
- **Değişkenler nerede tanımlanmalıdır?**
  - Değişkeni kullanmadan önce veya
  - «{» ile başlayıp «}» ile biten program bloğunuzun başında.

```
public static void main(String[] args)
{ /* değişkenleri burada tanımlayınız */
```

## DEĞİŞKEN İSİMLERİ

- Değişken isimleri sadece aşağıdakilerden oluşmalıdır.
  - Harfler (Türkçe karakter kullanmayın, İ, ö, ç, ü, ğ, ş)
  - rakamlar (0'dan 9'a kadar)
  - Altçizgi (\_)

Fakat ilk karakter rakam olmamalıdır.

- Değişken isimleri boşluk, nokta (.), yıldız (\*), veya diğer özel karakterleri içermez:

```
7-11  netscape.com  util.* (izin verilmez)
```

- Değişken isimleri gereksiz şekilde uzun olmamalıdır.
- Java büyük-küçük harf duyarlı olduğundan sayi, **Sayi**, ve **SAYI** değişkenleri farklı olarak algılanır.
- Java dilinde kullanılan anahtar kelimeler değişken adı olarak kullanılamazlar.
- Örneğin bir değişkene print, int, public class gibi isimler verilemez.

# İLKEL TÜRLER

- Dört tam sayı türü (`byte`, `short`, `int`, ve `long`)
  - `int` en çok kullanılandır.
- İki noktalı sayı (`float` ve `double`)
  - `double` en kullanılandır.
- Bir karakter türü (`char`)
- Bir mantıksal veri türü (`boolean`)

İsim	Aralık	Depolanma boyutu
<code>byte</code>	$-2^7$ (-128) to $2^7-1$ (127)	8-bit işaretli
<code>short</code>	$-2^{15}$ (-32768) to $2^{15}-1$ (32767)	16-bit işaretli
<code>int</code>	$-2^{31}$ (-2147483648) to $2^{31}-1$ (2147483647)	32-bit işaretli
<code>long</code>	$-2^{63}$ to $2^{63}-1$ (i.e., -9223372036854775808 to 9223372036854775807)	64-bit işaretli
<code>float</code>	Negatif aralık: -3.4028235E+38 to -1.4E-45 Pozitif aralık: 1.4E-45 to 3.4028235E+38	32-bit IEEE 754
<code>double</code>	Negatif aralık: -1.7976931348623157E+308 to -4.9E-324 Pozitif aralık: 4.9E-324 to 1.7976931348623157E+308	64-bit IEEE 754

# İLKEL VERİ TÜRÜ ÖRNEKLERİ

## ► integer türü

0 -1 365 12000

## ► Noktalı sayı türü

0.99 -22.8 3.14159 5.0

## ► Karakter türü

'a' 'A' '#' ' '

## ► boolean türü

true false



# NÜMERİK OPERATÖRLER

İsim	Anlamı	Örnek	Sonuç
+	Toplama	$34 + 1$	35
-	Çıkarma	$34.0 - 0.1$	33.9
*	Çarpma	$300 * 30$	9000
/	Bölme	$1.0 / 2.0$	0.5
%	Kalanı bulma	$20 \% 3$	2

# TAM SAYI BÖLME

$+$ ,  $-$ ,  $*$ ,  $/$ , ve  $\%$

$5 / 2$  ifadesinin sonucu 2 dir.

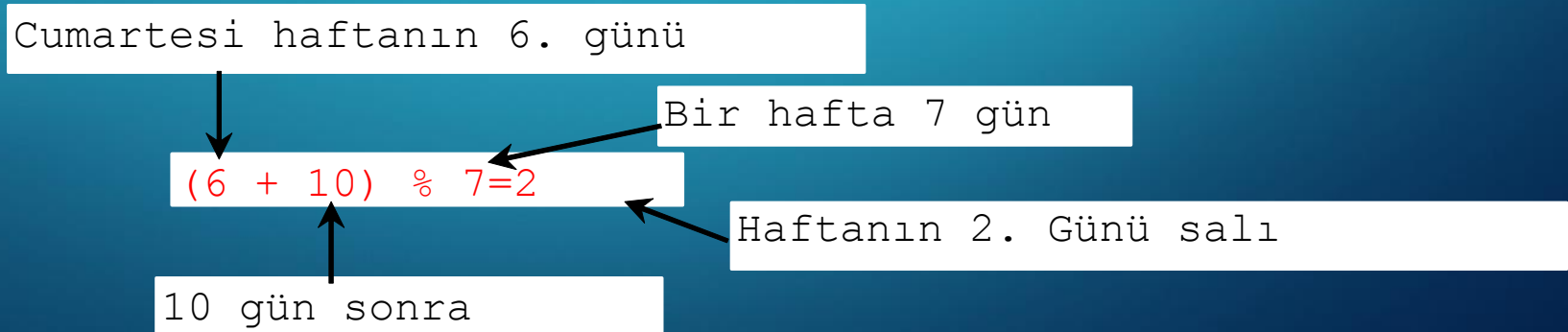
$5.0 / 2$  ifadesinin sonucu 2.5 dir.

$5 \% 2$  sonucu 1'dir (bölme sonucu kalan değer)

# KALANLI BÖLME OPERATÖRÜ

Kalanlı bölme programlamada oldukça kullanışlıdır. Örneğin, bir çift sayının ikiye bölümünden kalan daima 0 iken, tek sayının bölümünden kalan her zaman 1'dir. Bu bilgiyi sayının tek mi çift mi olduğunu bulmak için kullanabilirsiniz.

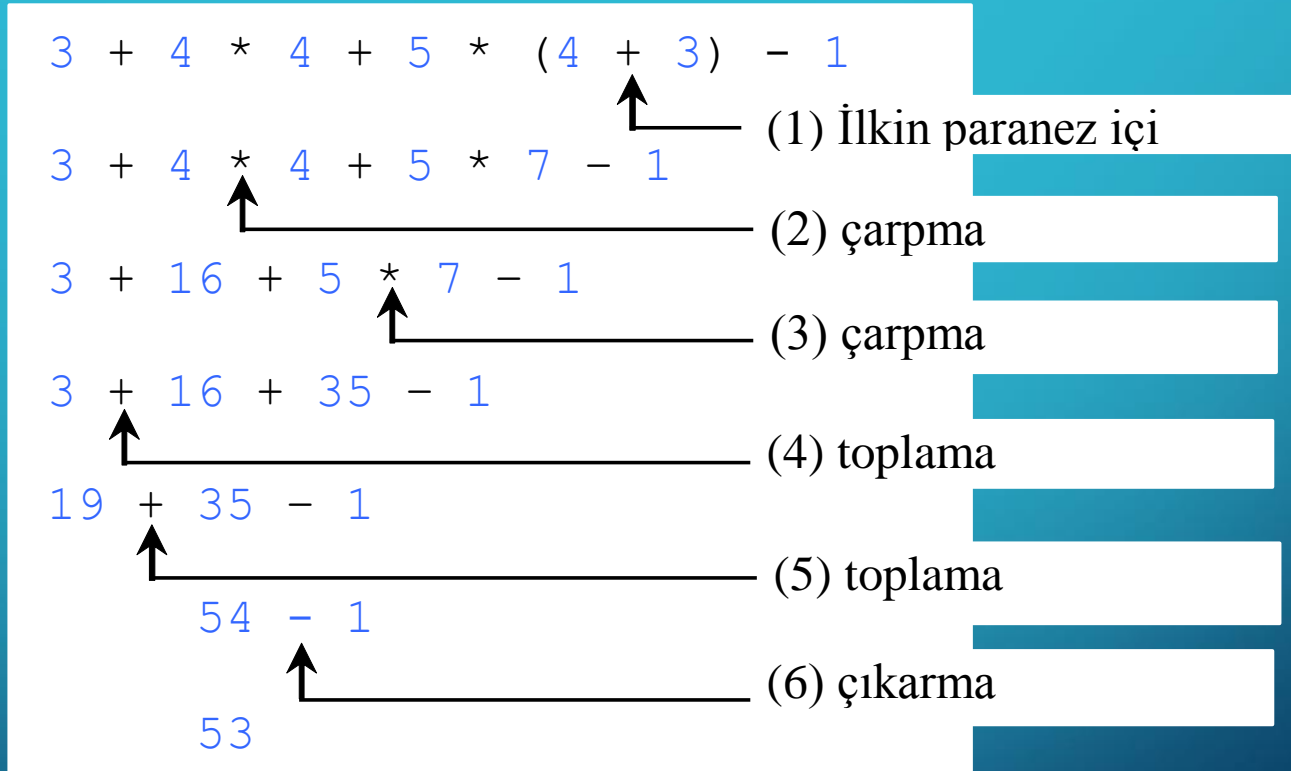
Bugün Cumartesi olduğunu düşünelim ve arkadaşınız 10 gün sonra size gelecek. 10 gün sonra hangi gündür? Aşağıdaki ifade ile bugünün Salı olduğunu bulabilirsiniz.



# ARİTMETİK İFADELER

Matematiksel ifade	Javada gösterim	Javada parantezli gösterim
$rate^2 + delta$	<code>rate*rate + delta</code>	<code>(rate*rate) + delta</code>
$2(salary + bonus)$	<code>2*(salary + bonus)</code>	<code>2*(salary + bonus)</code>
$\frac{1}{time + 3\ mass}$	<code>1/(time + 3*mass)</code>	<code>1/(time + (3*mass))</code>
$\frac{a - 7}{t + 9v}$	<code>(a - 7)/(t + 9*v)</code>	<code>(a - 7)/(t + (9*v))</code>

# BİR İFADENİN DEĞERLENDİRİLMESİ



# KISAYOLDAN ATAMA OPERATÖRÜ

<i>Operator</i>	<i>Örnek</i>	<i>Eşdeğeri</i>
<code>+=</code>	<code>i += 8</code>	<code>i = i + 8</code>
<code>-=</code>	<code>f -= 8.0</code>	<code>f = f - 8.0</code>
<code>*=</code>	<code>i *= 8</code>	<code>i = i * 8</code>
<code>/=</code>	<code>i /= 8</code>	<code>i = i / 8</code>
<code>%=</code>	<code>i %= 8</code>	<code>i = i % 8</code>

# ARTTIRMA VE AZALTMA OPERATÖRLERİ

++say

önartım

(++say) ifadesi say değişkenini 1 arttırır ve artımdan sonra say'daki yeni değeri değerlendirir.

--say

önazaltım

(--say) ifadesi say değişkenini 1 azaltır ve azaltımdan sonra say'daki yeni değeri değerlendirir.

```
int i = 10;  
int newNum = 10 * i++;
```

Karşılığı

```
int newNum = 10 * i;  
i = i + 1;
```

```
int i = 10;  
int newNum = 10 * (++i);
```

Karşılığı

```
i = i + 1;  
int newNum = 10 * i;
```

# NÜMERİK VERİ DÖNÜŞÜMÜ

Aşağıdaki atamaları göz önüne alalım

```
byte i = 100;
```

```
long k = i * 3 + 4;
```

```
double d = i * 3.1 + k / 2;
```

İki farklı tür içeren bir denklemde, java aşağıdaki kuarallara göre dönüşümü otomatik yapar:

1. Değişkenlerden biri double ise, diğeri double'a dönüştürülür.
2. Değilse, eğer biri float ise diğeri float'a dönüştürülür.
3. Değilse, değişkenlerden biri long ise diğeri long'a dönüştürülür.
4. Değilse, her iki değişken int'e dönüştürülür.



# TÜR DÖNÜŞÜMÜ

Anlaşılır dönüştürme

```
double d = 3; (tip genişlemesi)
```

Belirtilmiş dönüştürme

```
int i = (int)3.0; (tip daralması)
```

```
int i = (int)3.9; (Kesirli kısım atılır)
```

Yanlış olan nedir?      `int x = 5 / 2.0;`

Aralık artar

byte, short, int, long, float, double

# APPENDIX B: ASCII CHARACTER SET

ASCII Character Set is a subset of the Unicode from \u0000 to \u007f

TABLE B.1 ASCII Character Set in the Decimal Index

	0	1	2	3	4	5	6	7	8	9
0	nul	soh	stx	etx	eot	enq	ack	bel	bs	ht
1	nl	vt	ff	cr	so	si	dle	dcl	dc2	dc3
2	dc4	nak	syn	etb	can	em	sub	esc	fs	gs
3	rs	us	sp	!	"	#	\$	%	&	'
4	(	)	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[	\	]	^	_	`	a	b	c
10	d	e	f	g	h	i	j	k	l	m
11	n	o	p	q	r	s	t	u	v	w
12	x	y	z	{		}	~	del		

# ASCII CHARACTER SET, CONT.

ASCII Character Set is a subset of the Unicode from \u0000 to \u007f

**TABLE B.2** ASCII Character Set in the Hexadecimal Index

	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>
0	nul	soh	stx	etx	eot	enq	ack	bel	bs	ht	nl	vt	ff	cr	so	si
1	dle	dcl	dc2	dc3	dc4	nak	syn	etb	can	em	sub	esc	fs	gs	rs	us
2	sp	!	“	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	del

# NÜMERİK VE KARAKTER TÜRLERİ ARASINDA DÖNÜŞÜM

```
int i = 'a'; // int i = (int)'a' ile aynı;
```

```
char c = 97; // c = (char)97 ile aynı;
```

# ATAMA İFADELERİ

- Atama ifadesi bir değişkene değer atamak için kullanılır.

```
cevap = 42;
```

- «Eşitlik işareti =» atama operatörü olarak bilinir.

- Yukarıdaki atama ile cevap isimli değişkene 42 atandı veya cevap'a 42 atandığını söyleriz.

- Söz dizimi

```
degisken = ifade
```

*burada ifade bir değişken olabilir, sabit bir değer, veya matematiksel bir işlem olabilir.*

```
miktar = 3.99;
```

```
ilkharf = 'W';
```

```
toplam = burslu + burssuz;
```

```
yumurtasayisi = yumurtasayisi - 2;
```

## BASİT EKRAN GİRİŞ/ÇIKIŞ FONKSİYONU

- Java'da ekrana yazı yazmak için `System.out.print()` ve `System.out.println()` metotları kullanılır.
- Ekrana yazılacak veri veya değişkenler parantezin içine yazılır.
- `System.out.print()` metodunda yazı yazıldıktan sonra aynı satırda kalınır.
- `System.out.println()` ise bilgi yazıldıktan sonra alt satıra geçer.

`System.out.print(«JAVA»);`  
Örnek: `System.out.print(«Programlama»)`

Çıktı: JAVA Programlama

`System.out.println(«JAVA»);`  
`System.out.print(«Programlama»)`

Çıktı:  
JAVA  
Programlama

# BASİT EKРАН GİRİŞ/ÇIKIŞ FONKSİYONU

```
int a=5; double  
b=2.5; boolean  
c=true;  
  
char d='?';  
  
System.out.println(a);  
System.out.println(b);  
System.out.println(c);  
System.out.println(d);
```

Çıktı:  
5  
2.5  
True  
?

```
int a=5, b=17, c=4;  
System.out.println(a);  
System.out.println(b);  
System.out.print(«a=»+a);  
System.out.println(«, a+b= »+a+b);  
System.out.println(«a+b=»+(a+b));  
System.out.print(a+ « »+ b+ « »);  
System.out.print(a+ « »+ b+ « »);  
System.out.println(a+b+c);  
a=b-c;  
System.out.println(«a=»+a+ «, b=»+b+ «, c= »+c);
```

Çıktı:  
5  
17  
a=5, a+b=517  
a+b=22  
517 5 17 26  
a=13, b=17, c=4

# ÖZEL KARAKTERLER

`\»` Çift tırnak  
`\'` tek tırnak  
`\\` ters eğik çizgi  
`\n` Yeni satır. Sonraki satırın başına gider.  
`\r` Satır başı. Geçerli satırın başına gider  
`\t` Tab tuşu. Sonraki tab sekmesine kadar imleci götürür.

- Her bir özel karakteri yazdırmak için başına ters eğik çizgi konulur.



# KLAVYEDEN VERİ OKUNMASI: SCANNER (TARAYICI) SINIFI

- Scanner sınıfı java.util kütüphanesinin içinde yer alır ve kullanılması için aşağıdaki iki satırdan birinin programın başına yazılması gerekir.
  - `import java.util.*;`
  - `import java.util.Scanner;`
- İlk satır java.util kütüphanesini eklerken ikinci satır yalnızca Scanner uygulamasının paketin eklemek için kullanılır.
- Klavyeden veri girişini sağlayan Scanner sınıfının bir nesnesi aşağıdaki gibi oluşturulur.
  - `Scanner klavye= new Scanner(System.in);`
  - `int a=klavye.nextInt();`

# BAZI SCANNER SINIFI ÖRNEKLERİ

- `.nextByte()`→ Klavyeden byte türünde bir sayı okur.
- `.nextShort()`→ Klavyeden Short türünde bir sayı okur.
- `.nextInt()`→ Klavyeden int türünde bir sayı okur.
- `.nextLong()`→ Klavyeden long türünde bir sayı okur.
- `.nextDouble()`→ Klavyeden double türünde bir sayı okur.
- `.nextFloat()`→ Klavyeden Float türünde bir sayı okur.
- `.next()`→ Klavyeden String türünden bir sözcük okur. Okunan sözcük imleç ile bir sonraki boşluk arasında kalan parçadır.
- `.nextLine()`→ Klavyeden String türünde bir satır okur. Satır sonu karakteri okunan satıra dahil değildir.

```
int sayi = keyboard.nextInt();  
double mesafe = keyboard.nextDouble();  
String kelime = keyboard.next();  
String butunsatir = keyboard.nextLine();
```

- Giriş için kullanıcıya bilgi vermeyi unutmayın

```
System.out.print("Bir sayı gir: ");
```

# ÖRNEK: KLAVYEDEN BİR KİŞİNİN ADINI, SOYADINI, VE YAŞ VE BOYUNU OKUYUP EKRANA BASAN BİR JAVA PROGRAMI

YAZALIM.

```
import java.util.*;
```

Scanner sınıfı için kütüphane eklenmesi

```
public class klavyegirisorneği{
```

```
    public static void main()
```

```
{
```

```
    int yas;
```

```
    double boy;
```

```
    String adsoyad;
```

İsim, yaş ve boy bilgilerini tutmak için  
Değişkenlerin tanımlanması

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

Klavyeden okuma için giriş tanımlanması

```
    System.out.println(«Adınızı ve soyadınız aralarında boşluk ile girin»);
```

İsim bilgisinin  
okunması

```
    adsoyad=klavye.nextLine();
```

```
    System.out.println(«yaşınızı giriniz:»);
```

Yaş bilgisinin  
okunması

```
    yas=klavye.nextInt();
```

```
    System.out.println(«Boy bilginizi girin»);
```

Boy bilgisinin  
okunması

```
    boy=klavye.nextDouble();
```

```
    System.out.println(«Adınız:»+adsoyad+« Yaş:»+yas+« Boy:»+boy);
```

Bilgilerin ekrana  
yazılması

```
}
```

## ÖRNEK-2: İKİ SAYI OKUYUP TOPLAMINI BULAN JAVA PROGRAMINI YAZALIM.

```
import java.util.*
public class Toplama {
    public static void main( String [] args)
    {
        System.out.print(«ilk sayı:\t»);
        Scanner klavye=new Scanner(System.in);
        int ilk=klavye.nextInt();
        System.out.println();
        System.out.print(«ikinci sayı:\t»);
        int ikinci=klavye.nextInt();
        int toplam=ilk+ikinci;
        System.out.println(ilk+«+»+ikinci+ «=»+toplam);
    }
}
```

### Çıktı:

```
İlk Sayı: 5
İkinci sayı:8
5+8=13
```

# STRING SINIFI

- String bir karakter dizisidir.
- « » karakterleri arasına yazılan her türlü cümle bir dizedir.
- String java'nın temel türlerinden biri değildir ve sınıf olarak ayrıca tasarlanmıştır.
- Sınıf olarak tasarlanması kendine özgü metot ve özellikleri olduğu anlamına gelir.
- Java'da aşağıdaki örneklerdeki gibi çift tırnak içinde gösterilirler.
  - «Merhaba Java»
  - «Merhaba»
  - « »
  - «Fırat Bilgisayar Bölümü»
- String ileti=«Fırat Üniversitesi Bilgisayar Mühendisliği»;
- System.out.println(ileti);
- Dizi değişkenleri doğrudan system.out.println komutuyla da ekrana yazdırılabilir.

```
System.out.println(«Fırat Üniversitesi Bilgisayar Mühendisliği»);
```

# DİZGİLERİN BİRBİRİNE EKLENMESİ

- İki dizgi birbirine + işleci kullanılarak eklenir. İki kısa dizgi daha uzun bir dizgi oluşturmak için birleştirilebilir.
- String ilk=«Fırat»;
- String ikinci=«Bilgisayar Mühendisliği»
- String firat=ilk+ «Üniversitesi»+ikinci;
  - Fırat dizgisi «Fırat Üniversitesi Bilgisayar Mühendisliği» şeklinde olur.
- İki dizgiyi aralarında boşluk bırakarak birleştirme
  - String ilk=«FIRAT»;
  - String iki=«Üniversitesi»
  - String üniversitem=ilk+ « »+iki;
- Java'da string ile basit veri türleri de birleştirilebilir.
- Bu durumda yeni tür String olur.
  - String tümce=«FIRAT UNIVERSITESİ»+ 1975+ «yılında kuruldu»;

# DİZGİ TÜRÜNDE VERİLERİN UZUNLUĞUNU BULMA

- Dizgilerin uzunluğunu bulmak için length() adlı metot kullanılır.

```
int uzunluk= «Fırat Bilgisayar ».length();
```

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

Örnek program:

```
public class Uygulama1 {  
    public static void main(String[] args)  
    {  
        int uzunluk;  
        String iletı="Bilgisayar Mühendisliđi";  
        uzunluk=iletı.length();  
        System.out.println(iletı+": "+uzunluk);  
        iletı=iletı+ " Lisans programı";  
        uzunluk=iletı.length();  
        System.out.println(iletı+ " = " + uzunluk);  
    }  
}
```

Çıktı:

Bilgisayar Mühendisliđi:23

Bilgisayar Mühendisliđi Lisans programı= 39

# Bir String'teki karakterlerin pozisyonları

- Pozisyon indis olarak alınır.
  - “Java is fun.” stringinde ‘f’ 9. indiste yer alır.

«Java is fun.» stringinde 0 dan 11'e kadar indisler vardır.

0	1	2	3	4	5	6	7	8	9	10	11
J	a	v	a		i	s		f	u	n	.

Boşluklar stringte karakter olarak alınır



# STRING SINIFININ BAZI METOTLARI

➡ .toLowerCase() → Metodu çağıran dizginin tamamen küçük harfe çevrilmiş hali olan dizgiyi geri verir.

➡ String ileti=«MERHABA»;

➡ String kucukHarfi=iletı.toLowerCase();

➡ //kucukHarfli değişkeninin değeri «merhaba» olur

➡ .toUpperCase() → Metodu çağıran dizginin tamamen büyük harfe çevrilmiş hali olan dizgiyi geri verir.

➡ String ileti=«Merhaba»;

➡ String buyukHarfi=iletı.toUpperCase();

➡ //buyukHarfli değişkeninin değeri «MERHABA» olur

➡ .substring(Başlangıç) → Dizgi nesnesinin başlangıç numaralı karakterinden dizgi sonuna kadar olan bölümünü dizgi olarak geri verir.

➡ String ileti=«Merhaba»;

➡ String parca=iletı.substring(4);

➡ //parca değişkeninin içeriği «aba» olur.

M	E	R	H	A	B	A
0	1	2	3	4	5	6

# STRING SINIFININ BAZI METOTLARI

- .substring(Başlangıç,son) → Dizgi nesnesinin başlangıç numaralı karakterinden son numaralı karakterine kadar olan bölümü dizgi olarak geri verir.

- String ileti=«Merhaba»;

- String parca=ileti.substring(1,4);

- //parca değişkeninin içeriği «erh» olur.

M	E	R	H	A	B	A
0	1	2	3	4	5	6

- .indexOf(Bir Dizgi) → String dizgide Bir\_Dizgi ile tanımlı başka bir dizginin ilk görüldüğü yeri bulur. Eğer Bir\_Dizgi dizgisi kaynak dizgide -1 değeri geri döner.

- String ileti=«Bilgisayar Mühendisliği»;

- int yer=ileti.indexOf(«Müh»);

- //yer değişkeninin değeri 11 olur.

- .indexOf(Bir Dizgi,Başlangıç) → Dizgi üzerinde Başlangıç numaralı karakterden sonra Bir\_Dizgi ile tanımlı başka bir dizginin ilk görüldüğü yeri bulur. Eğer Bir\_Dizgi dizgisi kaynak dizgide -1 değeri geri döner.

- String ileti=«Bilgisayar Mühendisliği»;

- int yer=ileti.indexOf(«is»,8);

- //yer değişkeninin değeri 17 olur.

# STRING SINIFININ BAZI METOTLARI

➡ .lastIndexOf(Bir Dizgi) → Dizgi üzerinde Bir\_Dizgi ile tanımlı başka bir dizginin son görüldüğü yeri bulur. Eğer Bir\_Dizgi dizgisi kaynak dizgide -1 değeri geri döner.

➡ String iletı=«Bilgisayar Mühendisliđi»;

➡ int yer=iletı.indexOf(«Müh»);

➡ //yer değışkeninin değeri 11 olur.

➡ .trim() → Dizginin başındaki ve sonundaki boşlukların silindiđi bir dizgi verir.

➡ String iletı= « Merhaba »;

➡ String boslukSil=iletı.trim();

➡ //boslukSil değışkeninin değeri «merhaba»

➡ charAt(konum) → Dizginin üzerinde yer alan konum numaralı karakteri geri verir.

➡ String iletı=«merhaba»;

➡ Char konumdakiKarakter=iletı.charAt(3);

➡ // konumdakiKarakter değışkeninin değeri 'h' olur.

# STRING SINIFININ BAZI METOTLARI

- ▶ .compareTo(Bir dizgi) → Bir\_dizgi nesnesini kaynak dizgi ile alfabetik olarak karşılaştırır. Eğer metodu çağıran nesne daha küçük ise sıfırdan küçük bir değer, iki dizgi eşit ise 0 verir. Eğer metodu çağıran nesne metoda giren parametreden daha büyük ise sıfırdan büyük sonuç verir.
  - ▶ String ileti=«merhaba»;
  - ▶ int esitlik=iletisi.compareTo(«merhaba»);
  - ▶ // esitlik değişkeninin değeri 0 olur.
- ▶ .equals ve .equalsIgnoreCase → equals komutu iki dizginin eşit olup olmadığını sınar. True veya false değeri dönderir.
- ▶ equalsIgnoreCase ise karşılaştırmada büyük küçük hrf ayrımı yapmaz.

```
public class Uygulama1{  
    public static void main(String[] args) {  
        String birinci="FIRAT";  
        String ikinci="ÜNİVERSİTESİ";  
        String ucuncu="fırat";  
        System.out.println(birinci.equals(birinci));  
        System.out.println(birinci.equals(ikinci));  
        System.out.println(birinci.equalsIgnoreCase(ucuncu));  
        System.out.println(birinci.equals(ucuncu.toUpperCase()));  
    }  
}
```

Çıktı

true  
false  
true  
true

## GÖZDEN GEÇİRME SORULARI

1. Aşağıdaki değişkenlerden hangisi geçerlidir? Hangileri java anahtar kelimeleridir?

applet, Applet, a++, --a, 4#R, \$4, #44, apps, class, public, int, x, y, radius

2. Aşağıdaki algoritmayı koda dönüştürün :

- Adım 1: ismi mil olan double türünden bir değişkeni başlangıç değeri 100 olacak şekilde tanımlayın
- Adım 2: donuşum adında double türünden sabiti değeri 1.609 olacak şekilde tanımlayın
- Adım 3: Kilometre adında bir değişken tanımlayın ve değerine mil ile donuşumun çarpımını atayın.
- Adım 4: kilometre sonucunu ekranda gösterin.

Dördüncü adım sonunda kilometrenin değeri nedir?

```
public class Uygulama1{  
    public static void main(String[] args) {  
        double mil=100; // Adım 1  
        final double donusum=1.609;// Adım 2  
        double kilometre=mil*donusum;//Adım 3  
        System.out.println("kilometre = " + kilometre);//Adım 4  
    }  
}
```

# GÖZDEN GEÇİRME SORULARI

3. `int a = 1` ve `double d = 1.0`, alındığını varsayarak her bir ifadeyi bağımsız olarak değerlendirin?

- Aşağıdaki ifadelerin sonuçları nedir?

a) `a = 46 / 9;`

b) `a = 46 % 9 + 4 * 4 - 2;`

c) `a = 45 + 43 % 5 * (23 * 3 % 2);`

d) `a %= 3 / a + 3;`

e) `d = 4 + d * d + 4;`

f) `d += 1.5 * 3 + (++a);`

g) `d -= 1.5 * 3 + a++;`

4. Eğer bugün Salı ise 100 gün sonra hangi olacaktır?

5. `25/4` ifadesinin sonucu nedir? Noktalı sonuç elde etmek için nasıl yazmak gerekir?

6. Aşağıdaki ifadeler doğru mudur? Eğer doğru ise çıktıları yazınız?

```
System.out.println("25 / 4 is " + 25 / 4);
```

```
System.out.println("25 / 4.0 is " + 25 / 4.0);
```

```
System.out.println("3 * 2 / 4 is " + 3 * 2 / 4);
```

```
System.out.println("3.0 * 2 / 4 is " + 3.0 * 2 / 4);
```

7. Aşağıdaki aritmetik ifadeyi Java'da nasıl yazarsınız?

$$\frac{4}{3(r + 34)} - 9(a + bc) + \frac{3 + d(2 + a)}{a + bd}$$

# GÖZDEN GEÇİRME SORULARI

8. m ve r tamsayı değişkenler olsun.  $mr^2$  yi hesaplamak için bir Java programı yazınız?

9. Aşağıdaki programdaki hataları düzeltiniz?

```
1 public class Test {  
2     public void main(string[] args) {  
3         int i;  
4         int k = 100.0;  
5         int j = i + 1;  
6         System.out.println("j:" + j + " ve k: " + k);  
7     }  
8 }
```

10. Aşağıdaki kod parçasının çıktısını veriniz?

```
float f = 12.5F;
```

```
int i = (int)f;
```

```
System.out.println("f: " + f);
```

```
System.out.println("i : " + i);
```

11. Aşağıdaki ifadeleri değerlendirin:

```
int i = '1';
```

```
int j = '1' + '2';
```

```
int k = 'a';
```

# GÖZDEN GEÇİRME SORULARI

**12.**Aşağıdaki programın çıktısını veriniz :

```
public class Test {  
    public static void main(String[] args) {  
        char x = 'a';  
        char y = 'c';  
        System.out.println(++x);  
        System.out.println(y++);  
        System.out.println(x - y);  
    }  
}
```

**13.** System.out.println ile System.out.print arasında ne fark vardır?

**14.** Aşağıdaki program parçasının çıktısı nedir?

```
String cümle=«Merhaba JaVa»;  
Cumle=cümle.toUpperCase();  
System.out.println(cumle);
```

**15.** Aşağıdaki programın çıktısını veriniz?

```
String harfler=«klmnoprs»;  
System.out.println(harfler.substring(3));
```