

10. Ders: JAVA Programlama Dili (Mat. fonk., Unicode, ASCII, char, printf(), String)

Fırat Üniversitesi Teknoloji Fakültesi Yazılım Mühendisliği Bölümü

YMH111 Algoritma ve Programlama-I

Dr. Öğr. Üyesi Yaman Akbulut

JAVA Programlama Dili

- <http://www.kriptarium.com/algoritma.html> (Yardımcı kaynak)
- JAVA Programlama Dili (Ders: 47-66) video. (**Mutlaka İzleyiniz!**)
 - Ders 47: Yorum/Açıklama Satırları (izle)
 - Ders 48: Java String (izle)
 - Ders 49: Java String Sınıf Neden Değiştirilemezdir? (izle)
 - Ders 50: Java String Karşılaştırma İşlemleri (izle)
 - Ders 51: Java String Karşılaştırma İşlemleri Örnek (izle)
 - Ders 52: Java String Ekleme İşlemleri (izle)
 - Ders 53: Java Substring (izle)
 - Ders 54: Java String Sınıfının Bazı Metotları (izle)
 - Ders 55: Java StringBuffer Sınıfı (izle)
 - Ders 56: Java StringBuilder Sınıfı (izle)
 - Ders 57: String ve StringBuffer Arasındaki Farklar (izle)
 - Ders 58: StringBuffer ve StringBuilder Arasındaki Farklar (izle)
 - Ders 59: Java toString() method (izle)
 - Ders 60: StringTokenizer (izle)
 - Ders 61: Tip Dönüşümleri I (izle)
 - Ders 62 - Tip Dönüşümleri II (izle)
 - Ders 63 - Tip Dönüşümleri III (izle)
 - Ders 64 - Tip Dönüşümleri IV (izle)
 - Ders 65 - Tip Dönüşümleri V (izle)
 - Ders 66 - Tip Dönüşümleri VI (izle)

JAVA'da program yazma ve çalıştırma ortamı

1. JDK + `notepad++`

2. IDE `NetBeans`

Özel Karakterler

{ }	braces
()	parentheses
[]	brackets
//	double slashes
" "	quotation marks
;	semicolon

Java Keywords

abstract	double	int	super
assert	else	interface	switch
boolean	enum	long	synchronized
break	extends	native	this
byte	final	new	throw
case	finally	package	throws
catch	float	private	transient
char	for	protected	try
class	goto	public	void
const	if	return	volatile
continue	implements	short	while
default	import	static	
do	instanceof	strictfp	

Matematiksel fonksiyonlar-1

Math sınıfında trigonometrik fonksiyon işlemleriyle ilgili aşağıdaki metotlar bulunmaktadır.

Trigonometrik Metotlar

metot

sin(radyan)

cos(radyan)

tan(radyan)

toRadians(derece)

toDegree(radyan)

asin(a)

acos(a)

atan(a)

açıklama

bir radyan açının trigonometrik sinüsünü döndürür.

bir radyan açının trigonometrik kosinüsünü döndürür.

bir radyan açının trigonometrik tanjantını döndürür.

derece açının radyan değerini döndürür.

radyan açının derece değerini döndürür.

arcsin radyan

arccos radyan

arctan radyan

Trigonometrik örnekler

Math.toDegrees(Math.PI / 2) sonucu 90.0

Math.toRadians(30) sonucu 0.5236 ($\pi/6$ ile aynı)

Math.sin(0) sonucu 0.0

Math.sin(Math.toRadians(270)) sonucu -1.0

Math.sin(Math.PI / 6) sonucu 0.5

Math.sin(Math.PI / 2) sonucu 1.0

Math.cos(0) sonucu 1.0

Math.cos(Math.PI / 6) sonucu 0.866

Math.cos(Math.PI / 2) sonucu 0

Math.asin(0.5) sonucu 0.523598333 ($\pi/6$ ile aynı)

Math.acos(0.5) sonucu 1.0472 ($\pi/3$ ile aynı)

Math.atan(1.0) sonucu 0.785398 ($\pi/4$ ile aynı)

DENEYİN!

ölçün!

kontrol edin!

Matematiksel fonksiyonlar-2

Math sınıfında 5 üstel metot bulunmaktadır.

Üstel Metotlar

metot

açıklama

exp(x)

e^x değerini döndürür.

log(x)

doğal logaritma $\ln(x)$ değerini döndürür. ($\log_e x$)

log10(x)

10 tabanında logaritma x değerini döndürür. ($\log_{10} x$)

pow(a, b)

a^b değerini döndürür.

sqrt(x)

karekök x değerini döndürür.

Üstel örnekler

DENEYİN!

ölçün!

kontrol edin!

Math.exp(1) sonucu 2.71828

Math.log(Math.E) sonucu 1.0

Math.log10(10) sonucu 1.0

Math.log10(100) sonucu 2.0

Math.pow(2, 3) sonucu 8.0

Math.pow(3, 2) sonucu 9.0

Math.pow(4.5, 2.5) sonucu 22.91765

Math.sqrt(4) sonucu 2.0

Math.sqrt(16) sonucu 4.0

Math.sqrt(10.5) sonucu 4.24

Matematiksel fonksiyonlar-3

Math sınıfında 4 yuvarlama metodu bulunmaktadır.

Yuvarlama Metotları

metot	açıklama
ceil (x)	x üstteki en yakın tamsayıya yuvarlanır, dönen değer double olur.
floor (x)	x alttaki en yakın tamsayıya yuvarlanır, dönen değer double olur.
rint (x)	x en yakın tamsayıya yuvarlanır, dönen değer double olur. x alt ve üst taraftaki tamsayıya eşit uzaklıkta ise çift sayı döner.
round (x)	x, float ise (int)Math.floor(x + 0.5) sonucunu döndürür. x, double ise (long)Math.floor(x + 0.5) sonucunu döndürür.

Yuvarlama örnekleri

Math.ceil(2.1) sonucu 4.0

Math.ceil(2.0) sonucu 2.0

Math.ceil(-2.0) sonucu -2.0

Math.ceil(-2.1) sonucu -2.0

Math.floor(2.1) sonucu 2.0

Math.floor(2.0) sonucu 2.0

Math.floor(-2.0) sonucu -2.0

Math.floor(-2.1) sonucu -4.0

Math rint(2.1) sonucu 2.0

Math rint(-2.1) sonucu -2.0

Math rint(4.5) sonucu 4.0

Math rint(-2.0) sonucu -2.0

Math rint(2.5) sonucu 2.0

Math rint(-2.5) sonucu -2.0

Math.round(2.6f) sonucu 3

// int döner

Math.round(2.0) sonucu 2

// long döner

Math.round(-2.0f) sonucu -2

// int döner

Math.round(-2.6) sonucu -3

// long döner

Math.round(-2.4) sonucu -2

// long döner

DENEYİN!

ölçün!

kontrol edin!

Matematiksel fonksiyonlar-4

min, max ve abs metotları

min, max, abs Metotları

metot	açıklama
<code>min(x, y)</code>	iki sayıdan küçük olanı döndürür. (sayılar: int, long, float, double)
<code>max(x, y)</code>	iki sayıdan büyük olanı döndürür. (sayılar: int, long, float, double)
<code>abs(x)</code>	bir sayının mutlak değerini döndürür. (sayılar: int, long, float, double)

min, max, abs örnekleri

Math.max(2, 3) sonucu 3

Math.max(2.5, 3) sonucu 3.0

Math.min(2.5, 4.6) sonucu 2.5

Math.abs(-2) sonucu 2

Math.abs(-2.1) sonucu 2.1

DENEYİN!

ölçün!

kontrol edin!

Matematiksel fonksiyonlar-5

random metodu

metot	açıklama
Math.random()	<p>Bu yöntem 0.0'dan büyük veya eşit ve 1.0'den küçük rastgele bir double değer üretir.</p> $0 \leq \text{Math.random()} < 1.0$

random örnekleri

`(int)(Math.random() * 10)`

0 ile 9 arasında rastgele bir tamsayı üretir.

`50 + (int)(Math.random() * 50)`

50 ile 99 arasında rastgele bir tamsayı üretir.

Genelleştirilmiş şekliyle:

`a + Math.random() * b`

a ile a + b arasında rastgele bir tamsayı üretir. (a+b hariç)

DENEYİN!

ölçün!

kontrol edin!

char Veri Tipi

char sadece bir karakteri temsil eder.

```
char harf = 'A';  
char sayiChar = '4';
```

char için 'tek tırnak'
String için "çift tırnak"
kullanılır.

Unicode ve ASCII karakterler **char** veri tipi ile ifade edilebilir.

Unicode

Original Unicode, Unicode Konsorsiyumu tarafından kurulmuştur.

16-bit karakter kodlaması olarak tasarlanmıştır.

char veri tipiyle bu karakterler temsil edilebilir.

Java **Unicode**'u destekler.

16-bit orijinal Unicode 65.536 karakteri barındırır ancak bu tüm dünyadaki karakterleri tanımlamak için yeterli değildir.

Bu nedenle Unicode 1.112.064 karakter sayısına genişletilmiştir. Genişletilmiş Unicode veya Ek Unicode şeklinde söylenebilir.

Unicode

Original Unicode karakterler 16-bit'tir ve bu karakterler **char** veri tipinde saklanabilir.

16-bit, 2 Byte eder. Bu da **\u0000** hexadecimal (onaltılık) sayı sisteminde 4-bitlik 4 hane ile gösterilebilir.

\u hexadecimal (onaltılık) gösterim için kullanılır. Karakter aralığı **\u0000**'dan **\uFFFF**'e kadardır.

Yunan harfleri α , β ve γ 'nın Unicode'ları **\u03b1**, **\u03b2**, **\u03b3** şeklindedir. Decimal (onluk) karşılıkları ise sırasıyla **945**, **946** ve **947**. Klavyede ALT + **945** yaparsanız alfa karakterini ortaya çıkar. (945 numlock tarafında yazılmalı)

ASCII (American Standard Code for Information Interchange)

ASCII kodlama sistemi 8-bit'tir.

Birçok bilgisayar büyük harf, küçük harf, rakamlar, noktalama işaretleri, kontrol işaretleri için ASCII kullanır.

Unicode, ASCII nin 7-bitlik `\u0000` ile `\u007F` lik bölümünü (128 adet ASCII) içermektedir.

Alfabemizde bulunan A, B ve C karakterlerinin ASCII kodları `\u0041`, `\u0042`, `\u0043` şeklindedir. Decimal (onluk) karşılıkları ise sırasıyla 65, 66 ve 67'dir. Klavyede ALT + 65 yaparsanız büyük A karakterini yazabilirsiniz. (65 numlock tarafında yazılmalı)

ASCII (American Standard Code for Information Interchange)

Sık kullanılan ASCII kodları

Karakter	Decimal değeri	Unicode değeri
'0' – '9'	48 – 57	\u0030 – \u0039
'A' – 'Z'	65 – 90	\u0041 – \u005A
'a' – 'z'	97 – 122	\u0061 – \u007A

```
char harf = 'A';  
char harf = '\u0041';
```

```
char ch = 'a';  
System.out.println(++ch);
```

ASCII karakter seti (decimal)

	<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>
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 karakter seti (hexadecimal)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
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

Hexadecimal Binary Decimal

<i>Hexadecimal</i>	<i>Binary</i>	<i>Decimal</i>
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Escape karakteri (Kaçış karakteri \)

```
System.out.println("Burası "Yazılım Mühendisliği");  
// hata verir
```

```
System.out.println("Burası \"Yazılım Mühendisliği\"");
```

Çıktı sonucu: Burası "Yazılım Mühendisliği"

Escape karakteri (Kaçış karakteri \)

escape sequence

kaçış kodları	adı	Unicode kodu	decimal değeri
\b	backspace	\u0008	8
\t	tab	\u0009	9
\n	linefeed	\u000A	10
\f	formfeed	\u000C	12
\r	carriage return	\u000D	13
\\	backslash	\u005C	92
\"	double quote	\u0022	34

Escape karakteri (Kaçış karakteri \)

```
System.out.println("\\t tab karakteridir.");
```

Çıktı sonucu:

```
"\t tab karakteridir."
```

```
System.out.println("Path su sekilde C:\\Program Files\\Java\\jdk-15.0.1\\bin");
```

Çıktı sonucu:

```
"Path su sekilde C:\Program Files\Java\jdk-15.0.1\bin"
```

char ve nümerik tipler arasında çevrim-1

char herhangi bir nümerik tipe çevrilebilir ve bunun tersi de geçerlidir. Bir tamsayı char'a çevrilirken en düşük 16-bit kullanılır kalanı göz ardı edilir.

Örnek olarak:

```
char ch = (char)0xAB0041;    // En düşük 16-bit hexadecimal 0041
                               // ch değişkenine atanıyor
System.out.println(ch);      // ch, A karakteridir.
```

char ve nümerik tipler arasında çevrim-2

Bir ondalık sayı bir karaktere (**char**) dönüştürüldüğünde, ondalık sayı önce bir `int`'e dönüştürülür ve daha sonra bir karaktere (**char**) dönüştürülür.

Örnek olarak:

```
char ch = (char)65.25;           // Onluk 65 değeri, ch değişkenine atanıyor.
```

```
System.out.println(ch);          // ch, A karakteridir.
```

char ve nümerik tipler arasında çevrim-3

Bir karakter (char) nümerik bir tipe dönüştürüldüğünde, karakterin Unicode'u belirtilen nümerik tipe dönüştürülür.

Örnek olarak:

```
int i = (int)'A';           // A karakterinin Unicode değeri, i değişkenine atanıyor.
```

```
System.out.println(i); // i, 65 olur.
```

char ve nümerik tipler arasında çevrim-4

Dönüşüm yapılan değer hedef değişkene uyuyorsa aşağıdaki gibi kapalı çevrimle kullanılır.

```
byte b = 'a'; // 'a' nın Unicode değeri 97  
int i = 'a';
```

Dönüşüm yapılan değer hedef değişkene uymuyorsa açık çevrimle yapılmalıdır.

```
byte b = '\uFFF4'; // bu hata verir.
```

Açık çevrim şu şekilde kullanılır:

```
byte b = (byte)\uFFF4;
```

char ve nümerik tipler arasında çevrim-5

Onaltılık olarak 0 ile FFFF arasındaki herhangi bir pozitif tam sayı, kapalı çevrimle bir karaktere dönüştürülebilir. Bu aralıkta olmayan herhangi bir sayı açık çevrimle bir karaktere dönüştürülmelidir.

```
int i = '2' + '3';           // (int)'2', 50 ve (int)'3', 51 eder.  
System.out.println("i: " + i); // i: 101
```

```
int j = 2 + 'a';             // (int)'a', 97 eder.  
System.out.println("j: " + j); // j: 99
```

```
System.out.println(j + ", " + (char)j // 99, c nin Unicode karakteridir.  
+ " nin Unicode karakteridir.");
```

```
System.out.println("Chapter " + '2'); // Chapter 2
```

char karşılaştırma-1

İki karakter sayılar gibi Unicode değerleri üzerinden karşılaştırılabilir.

'a' < 'b' **true**, 'a' nın (97) Unicode değeri, 'b' nin Unicode (98) değerinden küçüktür.

'a' < 'A' **false**, 'a' nın (97) Unicode değeri, 'A' nın Unicode (65) değerinden büyüktür.

'1' < '8' **true**, '1' in (49) Unicode değeri, '8' in Unicode (56) değerinden küçüktür.

char karşılaştırma-2

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
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

Program yazarken bir karakterin harf mi sayı mı, büyük harf mi yoksa küçük harf mi diye test etmemiz gerekebilir. Bunun için karakterlerin Unicode'larının bulunduğu ASCII karakter tablosundan faydalanabiliriz. Tablodan 'a' dan 'z' ye, 'A' dan 'Z' ye ve '0' dan '9' sıralı bir şekilde ardışık olarak kodlar devam etmektedir.

```
if (ch >= 'A' && ch <= 'Z')
```

```
    System.out.println(ch + " karakteri büyük harftir");
```

```
else if (ch >= 'a' && ch <= 'z')
```

```
    System.out.println(ch + " karakteri küçük harftir ");
```

```
else if (ch >= '0' && ch <= '9')
```

```
    System.out.println(ch + " sayısal karakterdir.");
```

char karşılaştırma-3

Character sınıfında aşağıdaki metotlar vardır.

metot

isDigit(ch)

isLetter(ch)

isLetterOfDigit(ch)

isLowerCase(ch)

isUpperCase(ch)

toLowerCase(ch)

toUpperCase(ch)

açıklama

Belirtilen karakter sayı ise **true** döndürür.

Belirtilen karakter harf ise **true** döndürür.

Belirtilen karakter sayı veya harf ise **true** döndürür.

Belirtilen karakter küçük harf ise **true** döndürür.

Belirtilen karakter BÜYÜK harf ise **true** döndürür.

Belirtilen karakteri küçük harfe çevirir.

Belirtilen karakteri BÜYÜK harfe çevirir.

char karşılaştırma örnekleri

```
System.out.println("isDigit('a') " + Character.isDigit('a')); // isDigit('a') false
```

```
System.out.println("isLetter('a') " + Character.isLetter('a')); // isLetter('a') true
```

```
System.out.println("isLowerCase('a') " + Character.isLowerCase('a')); // isLowerCase('a') true
```

```
System.out.println("isUpperCase('a') " + Character.isUpperCase('a')); // isUpperCase('a') false
```

```
System.out.println("toLowerCase('T') " + Character.toLowerCase('T')); // toLowerCase('T') t
```

```
System.out.println("toUpperCase('q') " + Character.toUpperCase('q')); //toUpperCase('q') Q
```

String Tipi

char

sadece bir karakteri temsil eder.

```
char ch = 'A';
```

char için 'tek tırnak'
String için "çift tırnak"
kullanılır.

String

birden fazla karakter dizisini ifade eder.

```
String mesaj = "Merhaba Dünya";
```

String tipi için temel metotlar

metot	açıklama	<div>whitespace karakterler: ' ', \t, \f, \r, \n</div>
length()	dizideki karakter sayısını döndürür.	
charAt(index)	dizideki belirtilen indeksteki karakteri döndürür.	
concat(s1)	bu dizi ile s1 dizisini birleştirerek yeni bir dizi döndürür.	
toUpperCase()	dizideki bütün karakterleri büyük harf yapıp yeni bir dizi döndürür.	
toLowerCase()	dizideki bütün karakterleri küçük harf yapıp yeni bir dizi döndürür.	
trim()	dizinin her iki yanındaki whitespace karakterleri (' ', \t, \f, \r, \n) yok ederek yeni bir dizi döndürür.	

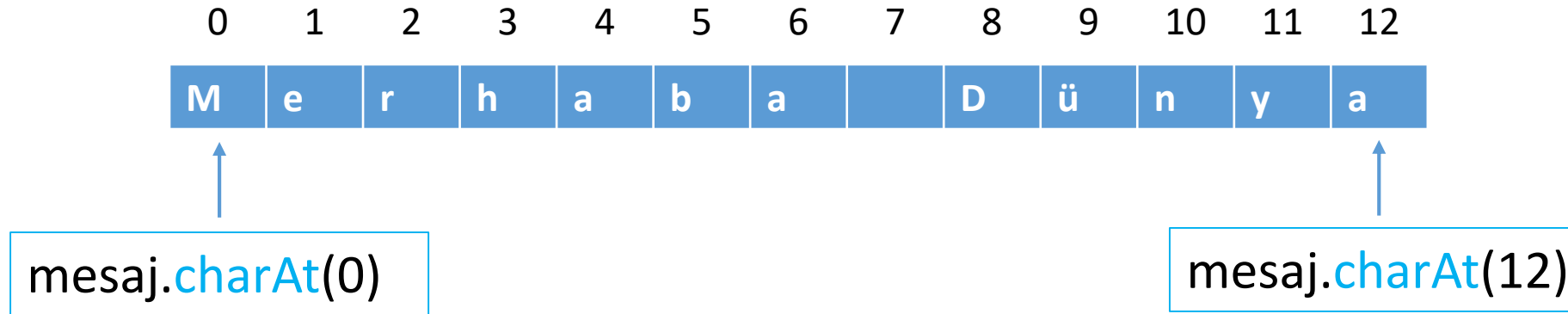
length() dizinin uzunlugu

```
String mesaj = "Merhaba Dünya";  
System.out.println(mesaj + " mesajinin uzunlugu " + mesaj.length() + " dir. ");  
// Merhaba Dünya mesajinin uzunlugu 13 dir.
```

```
String mesaj2 = "";  
System.out.println(mesaj2 + " mesajinin uzunlugu " + mesaj2.length() + " dir. ");  
// ?? deneyiniz
```

DE
NE
Yİ
Nİ
Z!

charAt(index) indekse göre karakter getirme



concat(s1) concatenating, dizileri birleştirme

`String s3 = s1.concat(s2);` `=` `String s3 = s1 + s2;`

`String benimMesaj = mesaj + " ve " + " arkadaşlar";`


```
String mesaj = "Merhaba " + "Dünya " + "ve " + "arkadaşlar" ;
```

```
String s1 = "Ek-" + 'B' ;  
// s1, Ek-B olur
```

```
String s = "Bölüm" + 2 ;  
// s, Bölüm2 olur
```

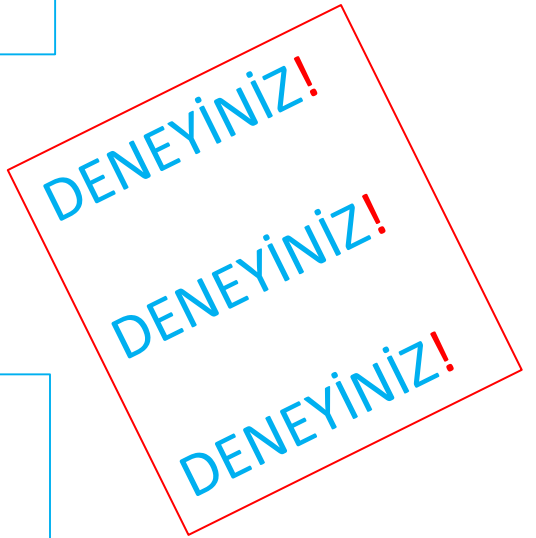
Daha önceki mesaj içeriği "Merhaba Dünya" idi.
Şimdi buna += operatörü ile " ve arkadaşlar hoşgeldiniz" dizisini ekleyebiliyoruz.

```
mesaj += " ve arkadaşlar hoşgeldiniz";
```

```
//yeni mesaj, "Merhaba Dünya ve arkadaşlar hoşgeldiniz" olur.
```

```
i = 2, j = 3 ise  
System.out.println(" i + j " + i + j);  
// sonucu nedir?
```

```
i = 2, j = 3 ise  
System.out.println(" i + j " + (i + j));  
// sonucu nedir?
```



toLowerCase(), toUpperCase(), trim()

```
"Merhaba".toLowerCase()  
// merhaba haline gelir
```

```
"Merhaba".toUpperCase()  
// MERHABA haline gelir
```

trim() bir dizinin her iki yanındaki whitespace karakterlerini yok eder.

whitespace karakterler: ' ', \t, \f, \r, \n

```
"\t      İyi Akşamlar      \n".trim()  
// İyi Akşamlar haline gelir
```

next() Konsoldan string (dizi) girişı

```
Scanner giris = new Scanner(System.in);  
String s = giris.next();
```

Örnek 1:

```
1 //String Tipi
2 import java.util.Scanner;
3
4 public class StringTipi {
5     public static void main(String [] args) {
6         Scanner giris = new Scanner(System.in);
7         System.out.print("Boslukla ayrilmis 3 kelime yaziniz: ");
8         String s1 = giris.next();
9         String s2 = giris.next();
10        String s3 = giris.next();
11        System.out.println("s1: " + s1);
12        System.out.println("s2: " + s2);
13        System.out.println("s3: " + s3);
14    }
15 }
```

```
C:\Program Files\Java\jdk-15.0.1\bin\yeni>javac StringTipi.java
```

```
C:\Program Files\Java\jdk-15.0.1\bin\yeni>java StringTipi
Boslukla ayrilmis 3 kelime yaziniz: Yazilim Muhendisligi Bolumu
s1: Yazilim
s2: Muhendisligi
s3: Bolumu
```

nextLine() Konsoldan satır girişi

```
Scanner giris = new Scanner(System.in);  
String s = giris.nextLine();
```

Örnek 2: nextLine() tüm satır

```
1 //String Tüm Satır
2 import java.util.Scanner;
3
4 public class StringTumSatir {
5     public static void main(String [] args) {
6         Scanner giris = new Scanner(System.in);
7         System.out.print("Bir satir yaziniz: ");
8         String s = giris.nextLine();
9         System.out.println("Girilen satir: " + s);
10    }
11 }
```

```
C:\Program Files\Java\jdk-15.0.1\bin\yeni>javac StringTumSatir.java
```

```
C:\Program Files\Java\jdk-15.0.1\bin\yeni>java StringTumSatir
```

```
Bir satir yaziniz: Yazilim Muhendisligi Bolumu Algoritma ve Programlama Dersi
```

```
Girilen satir: Yazilim Muhendisligi Bolumu Algoritma ve Programlama Dersi
```

nextLine() Konsoldan karakter(char) okuma

```
Scanner giris = new Scanner(System.in);  
String s = giris.nextLine();  
char ch = s.charAt(0);
```


Örnek 3: nextLine() karakter

```
1 //Karakter Okuma
2 import java.util.Scanner;
3
4 public class KarakterOkuma {
5     public static void main(String [] args) {
6         Scanner giris = new Scanner(System.in);
7         System.out.print("Bir karakter yaziniz: ");
8         String s = giris.nextLine();
9         char ch = s.charAt(0);
10        System.out.println("Girilen karakter: " + ch);
11    }
12 }
```

```
C:\Program Files\Java\jdk-15.0.1\bin\yeni>java KarakterOkuma
Bir karakter yaziniz: Yazılım
Girilen karakter: Y
```

```
C:\Program Files\Java\jdk-15.0.1\bin\yeni>java KarakterOkuma
Bir karakter yaziniz: Y
Girilen karakter: Y
```

Karakter Dizilerinin (String) Karşılaştırılması

metot (yöntem)	açıklama
<code>equals(s1)</code>	dizi, s1 dizisine eşit ise true döndürür.
<code>equalsIgnoreCase(s1)</code>	dizi, s1 dizisine eşit ise true döndürür, büyük küçük harf dikkate alınmaz.
<code>compareTo(s1)</code>	dizi, s1 dizisinden büyükse sıfırdan büyük bir sayı, eşitse sıfır ve küçükse sıfırdan küçük bir sayı döndürür.
<code>compareToIgnoreCase(s1)</code>	dizi, s1 dizisinden büyükse sıfırdan büyük bir sayı, eşitse sıfır ve küçükse sıfırdan küçük bir sayı döndürür, büyük küçük harf dikkate alınmaz
<code>startsWith(prefix)</code>	dizi, belirlenen bir önek ile başlamışsa true döndürür.
<code>endsWith(suffix)</code>	dizi, belirlenen bir son ek ile bitmişse true döndürür.
<code>contains(s1)</code>	dizinin içinde s1 alt stringi mevcut ise true döndürür.

```
String s1 = "Merhaba Dunya";  
String s2 = "Merhaba Dunya";  
String s3 = "Merhaba Mars";  
System.out.println(s1.equals(s2)); // true  
System.out.println(s1.equals(s3)); // false
```

```
s1.compareTo(s2)  
// bunun sonucunda 0 döner.
```

String karşılaştırma için
>, >=, <, <=
kullanmayınız.

compareTo sonucunda
sıfır,
pozitif tamsayı ya da
negatif tam sayı döner.

```
String s1 = "Merhaba Dunya";  
String s2 = "Merhaba Dunya";  
String s3 = "Merhaba Mars";
```

```
s1.startsWith("Me"); // true  
s1.startsWith("me"); // false
```

```
s1.endsWith("ya"); // true  
s1.endsWith("y"); // false
```

```
s1.contains("Du"); // true  
s1.contains("du"); // false
```

Örnek 4:

```
1 //İki Sehir Siralama
2 import java.util.Scanner;
3
4 public class IkiSehirSiralama {
5     public static void main(String [] args) {
6         Scanner giris = new Scanner(System.in);
7
8         System.out.print("İlk sehri yaziniz: ");
9         String sehir1 = giris.nextLine();
10        System.out.print("İkinci sehri yaziniz: ");
11        String sehir2 = giris.nextLine();
12
13        if (sehir1.compareTo(sehir2) < 0)
14            System.out.println("Şehirlerin alfabetik sirasi: " +
15                               sehir1 + ", " + sehir2);
16        else
17            System.out.println("Şehirlerin alfabetik sirasi: " +
18                               sehir2 + ", " + sehir1);
19    }
20 }
```

```
C:\Program Files\Java\jdk-15.0.1\bin\yeni>java IkiSehirSiralama
İlk sehri yaziniz: Istanbul
İkinci sehri yaziniz: Ankara
Şehirlerin alfabetik sirasi: Ankara, Istanbul
```

```
C:\Program Files\Java\jdk-15.0.1\bin\yeni>java IkiSehirSiralama
İlk sehri yaziniz: Ankara
İkinci sehri yaziniz: Istanbul
Şehirlerin alfabetik sirasi: Ankara, Istanbul
```

substring(index1, [index2]) Alt string elde etme

metot (yöntem)
açıklama

substring(baslamaIndeksi)

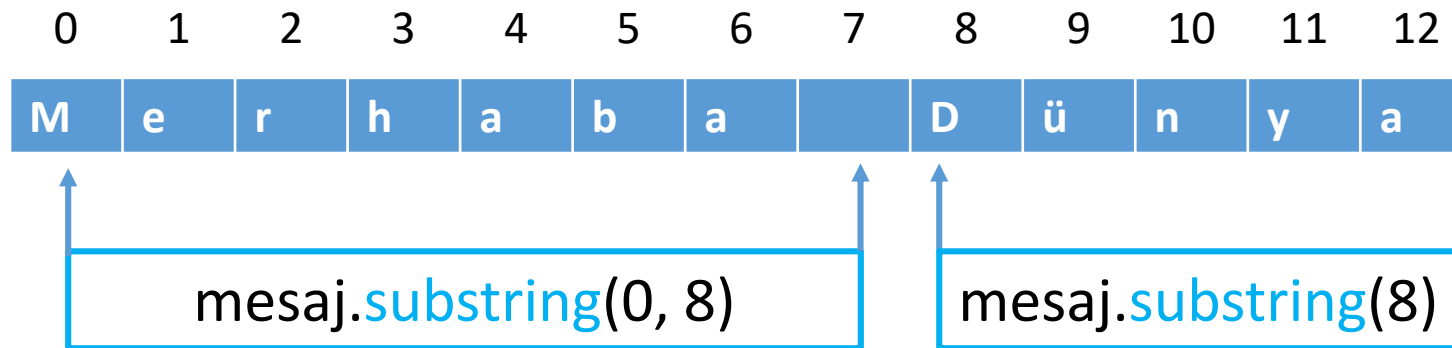
dizideki karakterleri **baslamaIndeksi**nden dizinin sonuna kadar elde eder.

substring(baslamaIndeksi, bitisIndeksi)

dizideki karakterleri **baslamaIndeksi**nden **bitisIndeksi-1**'e kadar elde eder.

substring(index) alt stringleri elde etme

mesaj



başlamaİndeksi

bitisİndeksi'ne eşit ise
boş döner.

başlamaİndeksi

bitisİndeksi'nden büyük
ise hata verir.

```
String altMesaj1 = mesaj.substring(0, 8); // "Merhaba "  
String altMesaj2 = mesaj.substring(8);   // "Dünya"
```

Karakter ya da alt string Bulma

metot (yöntem)

açıklama

`indexOf(ch)`

dizi içindeki **ilk** ch'nin indeksini döndürür, eşleşme olmazsa -1 döndürür.

`indexOf(ch, indeksSonrasi)`

dizi içindeki `indeksSonrasi`'ndan itibaren **ilk** ch'nin indeksini döndürür, eşleşme olmazsa -1 döndürür.

`indexOf(s)`

dizi içindeki **ilk** s'nin indeksini döndürür, eşleşme olmazsa -1 döndürür.

`indexOf(s, indeksSonrasi)`

dizi içindeki `indeksSonrasi`'ndan itibaren **ilk** s'nin indeksini döndürür, eşleşme olmazsa -1 döndürür.

Karakter ya da alt string Bulma

metot (yöntem)

açıklama

`lastIndexOf(ch)`

dizi içindeki **son** ch'nin indeksini döndürür, eşleşme olmazsa -1 döndürür.

`lastIndexOf(ch, indeksSonrasi)`

dizi içindeki `indeksSonrasi`'ndan itibaren **son** ch'nin indeksini döndürür, eşleşme olmazsa -1 döndürür.

`lastIndexOf(s)`

dizi içindeki **son** s'nin indeksini döndürür, eşleşme olmazsa -1 döndürür.

`lastIndexOf(s, indeksSonrasi)`

dizi içindeki `indeksSonrasi`'ndan itibaren **son** s'nin indeksini döndürür, eşleşme olmazsa -1 döndürür.

indexOf()

ilk görünen indeksi bulma

```
"Merhaba Dünya".indexOf('M');           // 0 döner.  
"Merhaba Dünya".indexOf('a');           // 4 döner.  
"Merhaba Dünya".indexOf('a', 5);        // 6 döner.  
  
"Merhaba Dünya".indexOf("haba");        // 3 döner.  
"Merhaba Dünya".indexOf("Dünya", 5);    // 8 döner.  
"Merhaba Dünya".indexOf("dünya", 5);    // -1 döner.
```

lastIndexOf() son görünen indeksi bulma

```
"Merhaba Dünya".lastIndexOf('M');      // 0 döner.  
"Merhaba Dünya".lastIndexOf('a');      // 12 döner.  
"Merhaba Dünya".lastIndexOf('a', 5);      // 12 döner.  
  
"Merhaba Dünya".lastIndexOf("haba");      // 3 döner.  
"Merhaba Dünya".lastIndexOf("Dunya", 5);      // 8 döner.  
"Merhaba Dünya".lastIndexOf("dunya", 5);      // -1 döner.
```

printf()

konsol çıktısı biçimlendirme-3

```
double miktar = 12618.98;  
double ucretOrani = 0.0013;  
double ucret = miktar * ucretOrani;  
System.out.println("Ucret: " + ucret + " TL"); // 16.404674 TL
```

```
double miktar = 12618.98;  
double ucretOrani = 0.0013;  
double ucret = miktar * ucretOrani;  
System.out.println("Ucret: " + (int)(ucret * 100) / 100.0 + " TL"); // 16.4 TL
```

```
double miktar = 12618.98;  
double ucretOrani = 0.0013;  
double ucret = miktar * ucretOrani;  
System.out.printf("Ucret: %4.2f TL", ucret); // 16.40 TL
```

printf()

konsol çıktısı biçimlendirme-1

`System.out.printf()` ile konsol (siyah ekran) üzerinde program çıktılarının biçimlerini düzenleyebilirsiniz.

```
System.out.printf(" biçimli metin ", nesne1, nesne2, ..., nesneN);
```

```
// nesne: sayı, karakter, true/false değer, string olabilir.
```

```
// " biçimli metin ": nesneleri ve diğer istenen metinleri (yazıları) barındıran bölüm.
```

printf() Sık kullanılan biçim tanımlayıcıları

biçim tanımlayıcısı	çıktı	örnek
%b	boolean değer	true veya false
%c	karakter	'a'
%d	onluk tam sayı	200
%f	ondalık sayı	45.460000
%e	sayının bilimsel gösterimi	4.556000e+01
%s	karakter dizisi, string	"Java eğlencelidir"

printf()

biçim tanımlayıcısı

```
System.out.printf("Ucret: %4.2f TL", ucret); // 16.40 TL
```

biçim tanımlayıcısı

% 4 . 2 f


alan genişliği

hassasiyet

dönüşüm kodu

printf() biçim tanımlayıcıları

```
int adet = 5;  
double miktar = 45.56;  
System.out.printf("Adet %d ve miktar %f", adet, miktar);  
  
// Adet 5 ve miktar 45.560000
```



printf() içinde %'nin kendisini kullanmak için %% yazınız.

printf() Alan genişliği ve hassasiyet örnekleri

örnek	çıktı
%5c	karakteri çıktı verir, karakterden önce 4 boşluk ekler.
%6b	boolean değeri çıktı verir, true ise öncesine 2 karakter, false ise öncesine 1 karakter ekler.
%5d	tam sayıyı çıktı verir, 5'ten küçük ise öncesine boşluk ekler, 5'ten büyük ise alan genişliği otomatik olarak artar.
%10.2f	noktadan sonra 2 hane olacak, toplam genişlik 10 olduğundan tam kısma 7 hane kalır. 7'den küçük ise boşluk ile doldurulur, 7'den büyük ise otomatik olarak genişlik artar.
%10.2e	sayının bilimsel gösterimi (DENEYİNİZ.)
%12s	String en az 12 karakter uzunluğunda olmalı, 12'den az ise boşluk ile doldurulur. 12'den fazla ise otomatik olarak alan genişliği artar.

Örnek 5: printf() kullanımı

```
1 //Bicim Demosu
2 public class BicimDemosu {
3     public static void main(String[] args) {
4         // tablonun basligini gosterme
5         System.out.printf("%-10s%-10s%-10s%-10s%-10s\n", "Derece",
6             "Radyan", "Sinus", "Cosinus", "Tanjant");
7
8         // 30 derece icin deger gosterimi
9         int derece = 30;
10        double radyan = Math.toRadians(derece);
11        System.out.printf("%-10d%-10.4f%-10.4f%-10.4f%-10.4f\n",
12            derece, radyan, Math.sin(radyan), Math.cos(radyan),
13            Math.tan(radyan));
14
15        // 60 derece icin deger gosterimi
16        derece = 60;
17        radyan = Math.toRadians(derece);
18        System.out.printf("%-10d%-10.4f%-10.4f%-10.4f%-10.4f\n",
19            derece, radyan, Math.sin(radyan), Math.cos(radyan),
20            Math.tan(radyan));
21    }
22 }
```

```
C:\Program Files\Java\jdk-15.0.1\bin\yeni>java BicimDemosu
Derece      Radyan      Sinus      Cosinus     Tanjant
30           0,5236     0,5000     0,8660     0,5774
60           1,0472     0,8660     0,5000     1,7321
```

Ödev 1:

Aşağıdaki programın çıktısını gösteriniz.

```
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);  
    }  
}
```

Ödev 2:

`mesaj` dizisindeki harfleri (karakterleri) kullanarak `charAt()` yöntemi ile `"baMya"` kelimesini oluşturunuz. (bir indeks en fazla bir kere kullanılabilir)

0	1	2	3	4	5	6	7	8	9	10	11	12
M	e	r	h	a	b	a		D	ü	n	y	a

Ödev 3:

`adSoyad` dizisi içindeki bir kişiye ait adı ve soyadı ayırıp; `ad` ve `soyad` dizilerine kaydeden ve konsolda gösteren programı yazınız.

0	1	2	3	4	5	6	7	8	9	10	11	12
Y	a	m	a	n		A	k	b	u	l	u	t

Ödev 4:

Java ve Bilişim terimleri olarak

ceil, floor, abs, radian, asin, max, min, string, String,

nedir?

İngilizce ve Türkçe karşılıklarını yazınız.