

Java Programlama, Metotlar ve Karakterler

11. Hafta

Dr. Öğr. Üyesi BÜŞRA ÖZDENİZCİ KÖSE

İşletme Bölümü

İşletme Fakültesi

Karakterler (char)

- Karakterler tek tırnak içerisinde tanımlanır:

`char not = 'A';`

`char harf = 'm';`

`char sayisalKarakterim = '4';`

- İlk ifadede A karakteri char veri türündeki `not` değişkenine atanmıştır.
- Üçüncü ifadede 4 karakteri char veri türündeki `sayisalKarakterim` değişkenine atanmıştır.

Karakterler ve Stringler Arasındaki Fark !

- String ise çift tırnak (" ") içerisinde tanımlanır.
- "A" bir String'dir, fakat 'A' bir karakterdir.

ASCII Karakter Seti

- Konuşmanın ve yazının en küçük birimi olarak karakterlerin bilgisayarlarda işlenebilmesi için, iki sütundan oluşan tablolar kullanılır. Bu tablolarda ikinci sütun bir karakteri, ilk sütun da bu karakterin tablo içindeki sıra numarasını gösterir. Böylece her bir karakter ile bu karakterin sıra numarası birbiri ile eşleşmiş olur.
- 1963 yılında bir standart olarak tanımlanmış olan ASCII karakter tablosu, bilgisayar programlarında kullanılan ilk karakter tablosudur.
- ASCII tablosunda sadece İngilizce harflerin mevcut olduğunu, örneğin Türkçemizdeki { ç, Ç, ğ, Ğ, ş, Ş, ı, İ, ö, Ö } harflerinin mevcut olmadığına dikkat ediniz

ASCII Karakter Seti (128 Karakter)

Escape Characters between 0-32

*Numbers {0...9}
between 48-57*

*Uppercase Letters (26 letters)
between 65-90*

*Lowercase Letters (26 letters)
between 97-122*

No	Ü	Karakter	No	Karakter	No	Karakter	No	Karakter
0		[Boş]	32	¸	64	@	96	`
1			33	!	65	A	97	a
2			34	"	66	B	98	b
3		(Bizim	35	#	67	C	99	c
4		İçin	36	\$	68	D	100	d
5		Önemsiz)	37	%	69	E	101	e
6			38	&	70	F	102	f
7		[Zil]	* 39	'	* 71	G	103	g
8		[Geri Al]	* 40	(72	H	104	h
9		[Yatay Sekme]	* 41)	73	I	105	i
10		[Satır Besleme]	* 42	*	74	J	106	j
11		[Dikey Sekme]	* 43	+	75	K	107	k
12		[Form Besleme]	* 44	,	76	L	108	l
13		[Satır Başı]	* 45	-	77	M	109	m
14			46	.	78	N	110	n
15			47	/	79	O	111	o
16			48	0	80	P	112	p
17			49	1	81	Q	113	q
18			50	2	82	R	114	r
19		(Bizim	51	3	83	S	115	s
20		İçin	52	4	84	T	116	t
21		Önemsiz)	53	5	85	U	117	u
22			54	6	86	V	118	v
23			55	7	87	W	119	w
24			56	8	88	X	120	x
25			57	9	89	Y	121	y
26			58	:	90	Z	122	z
27		[Çıkış]	* 59	;	91	[123	{
28		(Bizim	60	<	92	\	* 124	
29		İçin	61	=	93]	125	}
30		Önemsiz)	62	>	94	^	126	~
31			63	?	95	_	127	

Extended ASCII Karakter Seti (256 Karakter)

- Bilgisayarların biraz daha gelişmesi sonrasında, **256 karakter** içeren Genişletilmiş ASCII (Extended ASCII) tablosu kullanılmaya başlanmıştır.

No	Karakter	No	Karakter	No	Karakter	No	Karakter
128	€	160		192	À	224	à
129		161	ı	193	Á	225	á
130	,	162	ç	194	Â	226	â
131	f	163	£	195	Ã	227	ã
132	„	164	¤	196	Ä	228	ä
133	...	165	¥	197	Å	229	å
134	†	166		198	Æ	230	æ
135	‡	167	§	199	Ç	231	ç
136	^	168	¨	200	È	232	è
137	‰	169	©	201	É	233	é
138	Š	170	ª	202	Ê	234	ê
139	‹	171	«	203	Ë	235	ë
140	Œ	172	¬	204	Ì	236	ì
141		173		205	Í	237	í
142	Ž	174	®	206	Î	238	î
143		175	¯	207	Ï	239	ï
144		176	°	208	Ð	240	ð
145	‘	177	±	209	Ñ	241	ñ
146	’	178	²	210	Ò	242	ò
147	“	179	³	211	Ó	243	ó
148	”	180	´	212	Ô	244	ô
149	•	181	µ	213	Õ	245	õ
150	—	182	¶	214	Ö	246	ö
151	—	183	·	215	×	247	÷
152	~	184	,	216	Ø	248	ø
153	™	185	¹	217	Ù	249	ù
154	š	186	º	218	Ú	250	ú
155	›	187	»	219	Û	251	û
156	œ	188	¼	220	Ü	252	ü
157		189	½	221	Ý	253	ý
158	ž	190	¾	222	Þ	254	þ
159	ÿ	191	¿	223	ß	255	ÿ

Unicode Karakter Seti

- Bilgisayarların daha da gelişmesi sonrasında ise Arapça, Çince gibi pek çok dilde kullanılan karakterler ile, konuşma dili dışında başka amaçlar için de kullanılan karakterleri de içerecek şekilde Unicode (Universal Character Set) karakter tablosu oluşturulmuştur.
- Günümüz itibari ile Java programlarında Genişletilmiş ASCII ile Unicode karakter tabloları kullanılabilir.
- **Genişletilmiş ASCII tablosunda 256, Unicode tablosunda ise 120.000'den fazla karakter mevcut olduğundan, her bir ASCII karakter Unicode karakter olarak işlenebilmektedir.**

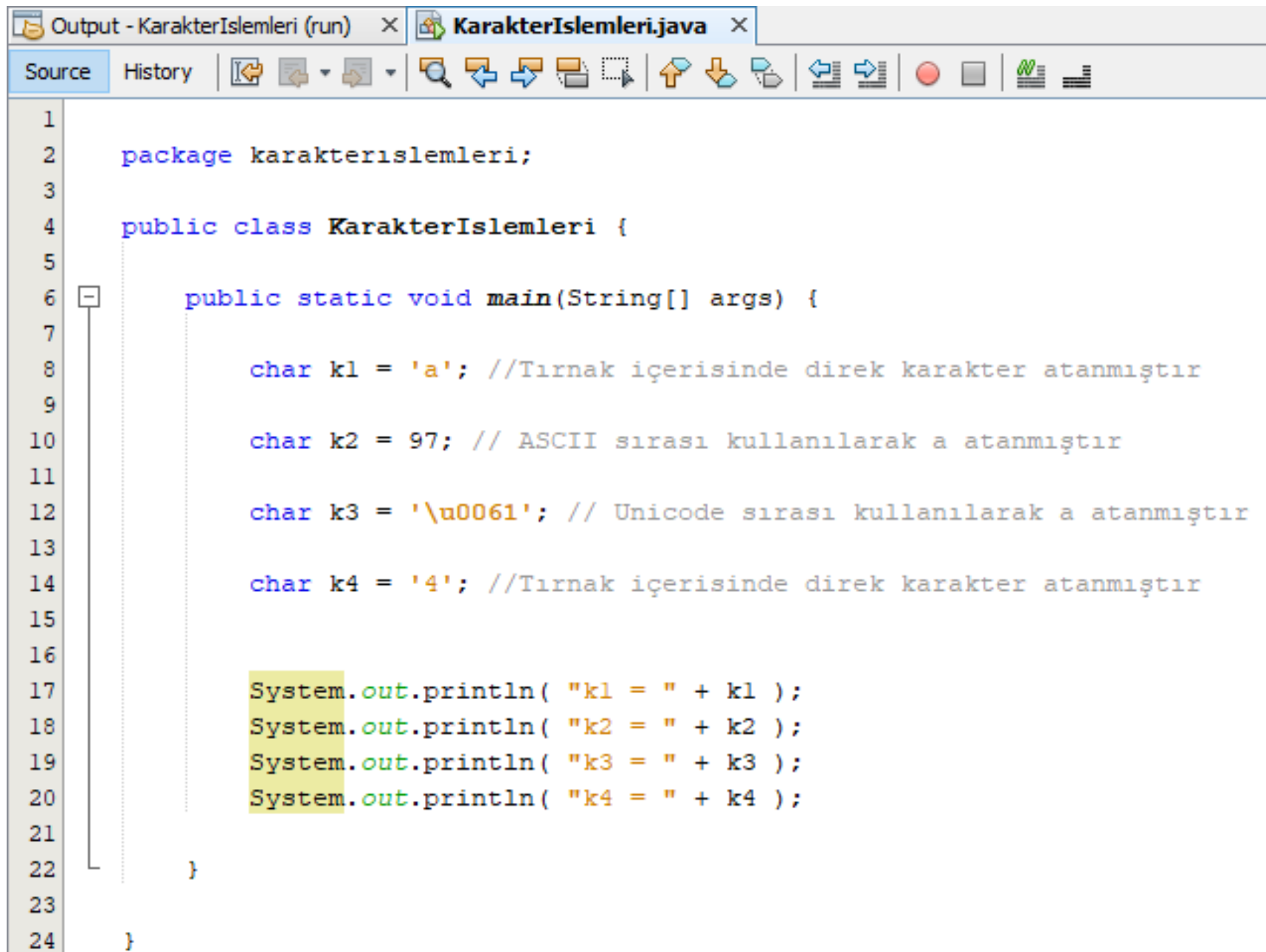
Unicode Karakter Seti

- Unicode karakter tablosu, başlangıçta **65.536 karakter** içerecek şekilde oluşturulmuştu ve bu durumda her bir karakteri adreslemek için **16 bit (2 byte)** yeterli oluyordu.
- Buna karşın zaman içinde dünyadaki tüm olası alfabeler ile diğer karakterler için 32 bit uzunluktaki adreslere ihtiyaç duyan Unicode UTF-32 versiyonu tanımlanmıştır.
- Karakter tanımlaması için **32 bit (4 basamaklı Onaltılı sayı)** için kullanılmaya başlanmıştır → **1,112,064 karakter !**

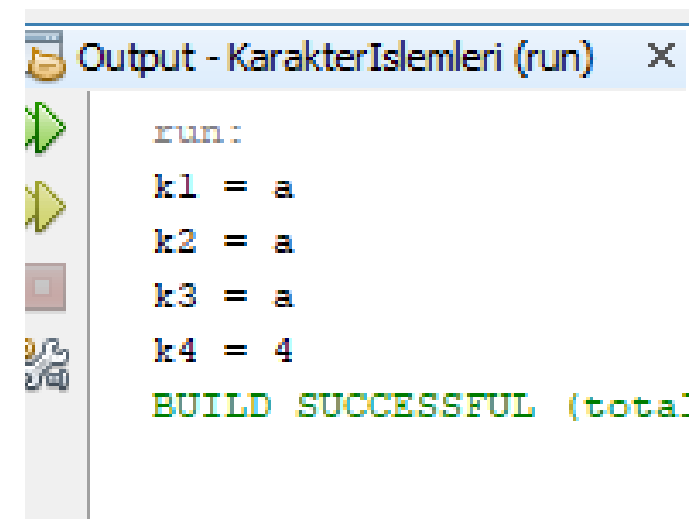
Unicode Karakter Seti

- Unicode \u ifadesi ile başlar ve 4 basamaklı onaltılı sayı şeklinde ifade edilir: \u0000 ile \uFFFF arasındadır.
- Java'da isterseniz ASCII karakter veya Unicode karakter kullanabilirsiniz
- Örneğin:
char letter = 'A';
char letter = '\u0041'; // Karakter A'nın Unicode değeri

Character	Extended ASCII		Unicode	
	Sıra (Onlu)	Kod	Sıra (Onlu)	Kod
a	97	‘a’	97	‘\u0061’
X	88	‘X’	88	‘\u0058’
4	52	‘4’	52	‘\u0034’
*	42	‘*’	42	‘\u002A’
^	94	‘^’	94	‘\u005E’
Ç	199	‘Ç’	130	‘\u00C7’
€	128	‘€’	8364	‘\u20AC’
¶	182	‘¶’	166	‘\u00B6’
Σ	(mevcut değil)	(mevcut değil)	8721	‘\u2211’



```
1 package karakterislemleri;
2
3
4 public class KarakterIslemleri {
5
6     public static void main(String[] args) {
7
8         char k1 = 'a'; //Tırnak içerisinde direk karakter atanmıştır
9
10        char k2 = 97; // ASCII sırası kullanılarak a atanmıştır
11
12        char k3 = '\u0061'; // Unicode sırası kullanılarak a atanmıştır
13
14        char k4 = '4'; //Tırnak içerisinde direk karakter atanmıştır
15
16
17        System.out.println( "k1 = " + k1 );
18        System.out.println( "k2 = " + k2 );
19        System.out.println( "k3 = " + k3 );
20        System.out.println( "k4 = " + k4 );
21
22    }
23
24 }
```



```
Output - KarakterIslemleri (run) X
run:
k1 = a
k2 = a
k3 = a
k4 = 4
BUILD SUCCESSFUL (total: 0s)
```

Karakter İşlemleri

- Increment ve decrement operatörleri char değişkenleri üzerinde uygulanabilir, bir sonraki veya önceki Unicode karakterin okunmasını sağlayabilir
- Örneğin, aşağıdaki komut b karakterinin okunmasını sağlar:
char ch = 'a';
System.out.println (++ch);

Output - Karakterİslemleri (run) X Karakterİslemleri.java X

Source History

A set of standard IDE navigation icons including arrows for navigation, a magnifying glass for search, and a refresh icon.

```
1
2 package karakterislemeleri;
3
4 public class Karakterİslemleri {
5
6     public static void main(String[] args) {
7
8         char k1 = 'a'; //Tırnak içerisinde direk karakter atanmıştır
9
10        char k2 = 97; // ASCII sırası kullanılarak a atanmıştır
11
12        char k3 = '\u0061'; // Unicode sırası kullanılarak a atanmıştır
13
14        char k4 = '4'; //Tırnak içerisinde direk karakter atanmıştır
15
16
17        System.out.println( "k1 = " + (++k1) );
18
19        System.out.println( "k2 = " + (++k2) );
20
21        System.out.println( "k3 = " + (--k3) );
22
23        System.out.println( "k4 = " + (--k4) );
24
25    }
26
27 }
28
```



Output - Karakterİslemleri (run) X

A green button with a white right-pointing arrow and a small white square, used to execute the code.

run:

k1 = b

k2 = b

k3 = `

k4 = 3

BUILD SUCCESSFUL (total:

Komut	Sonuç	Açıklama
<pre>char c; int i;</pre>		
<pre>c = 'a' + 1;</pre>	<pre>c = 'b'</pre>	<p>toplama işlemi yapılacağı için tamsayı oluşturmak üzere 'a' karakterinin değeri, ASCII sırası olan 97'ye çevrilir.</p> <p>97'ye 1 eklenir.</p> <p>char değeri beklendiğinden ASCII sırası 98 olan 'b' karakteri döndürülür.</p>
<pre>i = 'a' + 1;</pre>	<pre>i = 98</pre>	<p>toplama işlemi yapılacağı için tamsayı oluşturmak üzere 'a' karakteri, ASCII sırası olan 97'ye çevrilir</p> <p>97'ye 1 eklenir.</p> <p>tamsayı değeri beklendiğinden (1+97=) 98 döndürülür.</p>
<pre>c = 'a' ;</pre>	<pre>c = 'a'</pre>	<p>c'nin değeri 'a' olur.</p>
<pre>c = c + 1;</pre>	<pre>c = 'b'</pre>	<p>toplama işlemi yapılacağı için tamsayı oluşturmak üzere 'a' karakteri, ASCII sırası olan 97'ye çevrilir.</p> <p>97'ye 1 eklenir.</p> <p>char değeri beklendiğinden ASCII sırası 98 olan 'b' karakteri döndürülür.</p>



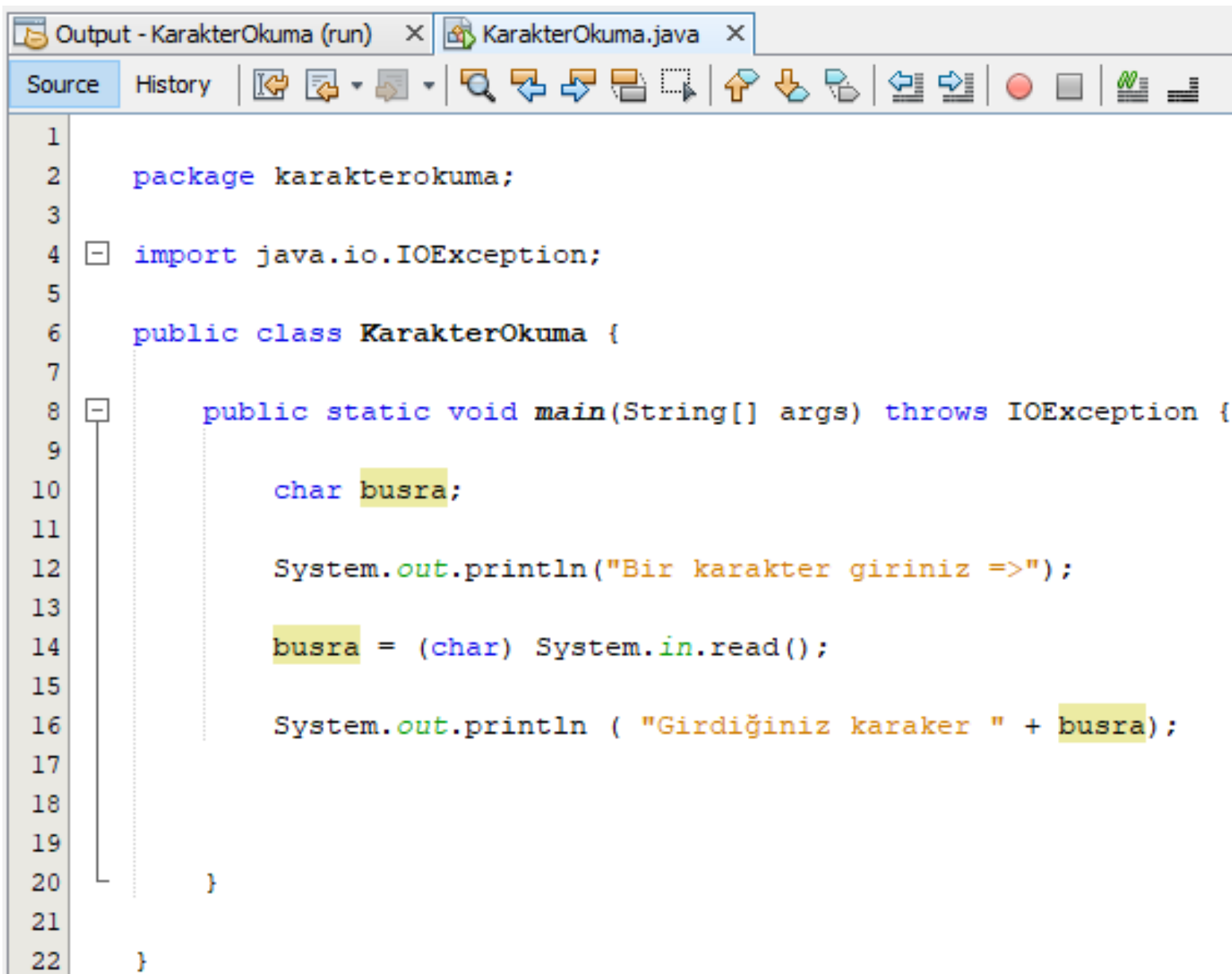


Komut	Sonuç	Açıklama
<code>char k;</code>		
<code>k = '\u0061'</code>	<code>'a'</code>	<i>\u, sonrasındaki Onaltılı sayı kullanılarak Unicode karakterin alınacağını işaret eder. Unicode karakter tablosunda 0061 sıralı karakter alınır.</i>
<code>k = '\u0061\u0027'</code>	(error)	<i>k, tek bir karaktere atanabilir. Tek tırnaklar arasında iki tane değer vardır</i>
<code>k = ''</code>	(error)	<i>tırnaklar arasında karakter görünmemektedir</i>
<code>k = ' '</code>	<code>' '</code>	<i>k, boşluk karakterine atanmıştır</i>
<code>k = ' '</code>	(error)	<i>tırnaklar arasında iki boşluk karakteri görünmektedir. k, tek bir karaktere atanabilir</i>

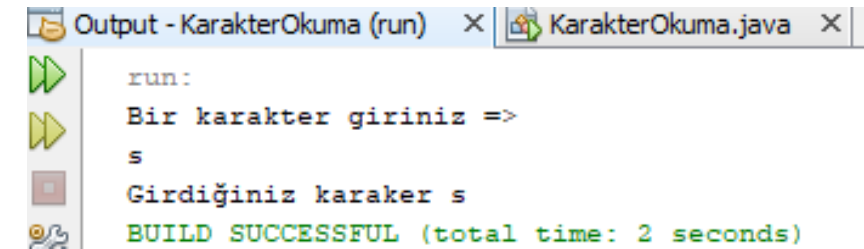
Kullanıcıdan Karakter Okuma:

1. Alternatif

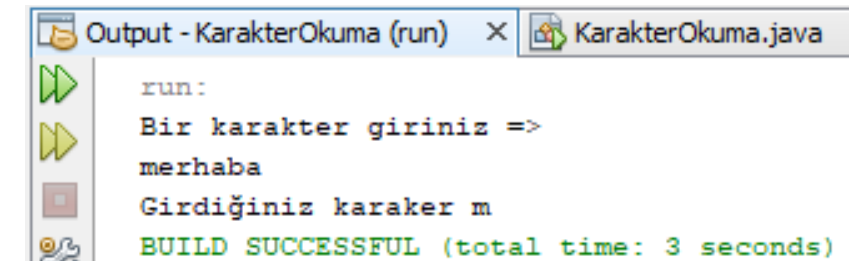
Metot	Tanım	Örnek
<code>(char) System.in.read()</code>	Klavyeden girilen ilk Karakteri değer olarak alır.	<i>Örneğin "M" ya da "Merhaba" yazılırsa "M" karakteri okunur.</i>



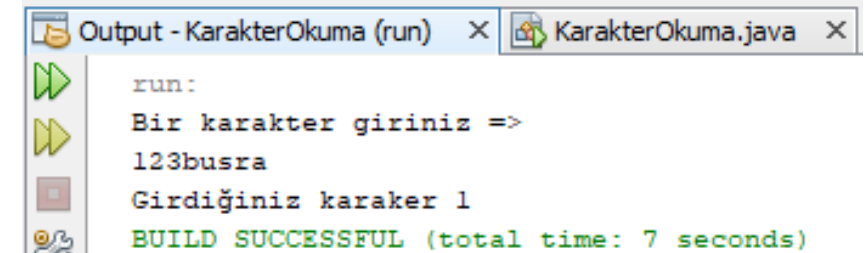
```
1
2 package karakterokuma;
3
4 import java.io.IOException;
5
6 public class KarakterOkuma {
7
8     public static void main(String[] args) throws IOException {
9
10         char busra;
11
12         System.out.println("Bir karakter giriniz =>");
13
14         busra = (char) System.in.read();
15
16         System.out.println ( "Girdiğiniz karakter " + busra);
17
18     }
19
20 }
21
22 }
```



```
Output - KarakterOkuma (run) x KarakterOkuma.java x
run:
Bir karakter giriniz =>
s
Girdiğiniz karakter s
BUILD SUCCESSFUL (total time: 2 seconds)
```



```
Output - KarakterOkuma (run) x KarakterOkuma.java
run:
Bir karakter giriniz =>
merhaba
Girdiğiniz karakter m
BUILD SUCCESSFUL (total time: 3 seconds)
```



```
Output - KarakterOkuma (run) x KarakterOkuma.java x
run:
Bir karakter giriniz =>
123busra
Girdiğiniz karakter 1
BUILD SUCCESSFUL (total time: 7 seconds)
```

Kullanıcıdan String ve Karakter Okuma:

2. Alternatif

Metot	Tanım	Örnek
<code>next()</code>	Klavyeden girilenleri ilk boşluğa kadar alır. String okuyan Scanner metodu	<i>"Merhaba Dünya" yazılırsa sadece "Merhaba" kelimesi okunur.</i>
<code>nextLine()</code>	Klavyeden girilenleri ilk Enter tuşuna kadar alır. String okuyan Scanner metodu	<i>"Merhaba Dünya" yazılırsa tüm "Merhaba Dünya" satırını okur.</i>

next() veya nextLine() Scanner metotları kullandığımız zaman, charAt() metodu da kullanarak ilgili String içerisindeki karakterleri okuyabiliriz!

Önce String Tanımlama...

- String'ler bir karakter dizisidir.

String busra;

String cumle = "Merhaba Dünya !!! ";



```
Output - KarakterOkuma (run) x KarakterOkuma.java x
Source History
1
2 package karakterokuma;
3 import java.util.Scanner;
4 public class KarakterOkuma {
5
6     public static void main(String[] args) {
7
8         char busra;
9
10        Scanner input = new Scanner (System.in);
11
12        System.out.println("Bir karakter giriniz =>");
13
14        busra = input.next().charAt(0);
15
16        System.out.println ( "Girdiğiniz karakter " + busra);
17
18    }
19
20 }
```

```
Output - KarakterOkuma (run) x KarakterOkuma.java x
run:
Bir karakter giriniz =>
busra
Girdiğiniz karakter b
BUILD SUCCESSFUL (total time: 2 seconds)
```

```
Output - KarakterOkuma (run) x KarakterOkuma.java x
run:
Bir karakter giriniz =>
12badgdh
Girdiğiniz karakter 1
BUILD SUCCESSFUL (total time: 6 seconds)
```

```
Output - KarakterOkuma (run) x KarakterOkuma.java x
Source History
1
2 package karakterokuma;
3 import java.util.Scanner;
4 public class KarakterOkuma {
5
6     public static void main(String[] args) {
7
8         String busra; //String nesnesi tanımladım
9
10        Scanner input = new Scanner (System.in);
11
12        System.out.println("Bir karakter giriniz =>");
13
14        busra = input.next(); // String okudum
15
16        char karakterim = busra.charAt(0);
17        // String içindeki 0. konumdaki (indexli) ilk karakteri charAt() metoduyla okudum
18
19        System.out.println ( "Girdiğiniz String " + busra);
20
21        System.out.println ( "Girdiğiniz ilk karakter " + karakterim);
22
23        System.out.println ( "Girdiğiniz üçüncü karakter " + busra.charAt(2) );
24        //charAt() metodunu bu şekilde yerinde de kullanabilirsiniz
25
26
27    }
28
29 }
```



```
Output - KarakterOkuma (run) x KarakterOkuma.java x
run:
Bir karakter giriniz =>
hey hey hey
Girdiğiniz String hey
Girdiğiniz ilk karakter h
Girdiğiniz üçüncü karakter y
BUILD SUCCESSFUL (total time: 3 seconds)
```

```
Output - KarakterOkuma (run) x KarakterOkuma.java x
Source History
1
2 package karakterokuma;
3 import java.util.Scanner;
4 public class KarakterOkuma {
5
6     public static void main(String[] args) {
7
8         String busra; //String nesnesi tanımladım
9
10        Scanner input = new Scanner (System.in);
11
12        System.out.println("Bir karakter giriniz =>");
13
14        busra = input.nextLine(); // String okudum
15
16        char karakterim = busra.charAt(0);
17        // String içindeki 0. konumdaki (indexli) ilk karakteri charAt() metoduyla okudum
18
19        System.out.println ( "Girdiğiniz String " + busra);
20
21        System.out.println ( "Girdiğiniz ilk karakter " + karakterim);
22
23        System.out.println ( "Girdiğiniz üçüncü karakter " + busra.charAt(2) );
24        //charAt() metodunu bu şekilde yerinde de kullanabilirsiniz
25
26
27    }
28
29 }
```



```
Output - KarakterOkuma (run) x KarakterOkuma.java x
run:
Bir karakter giriniz =>
hey hey hey
Girdiğiniz String hey hey hey
Girdiğiniz ilk karakter h
Girdiğiniz üçüncü karakter y
BUILD SUCCESSFUL (total time: 7 seconds)
```

Bir String'deki Karakter Sayısını Bulma:

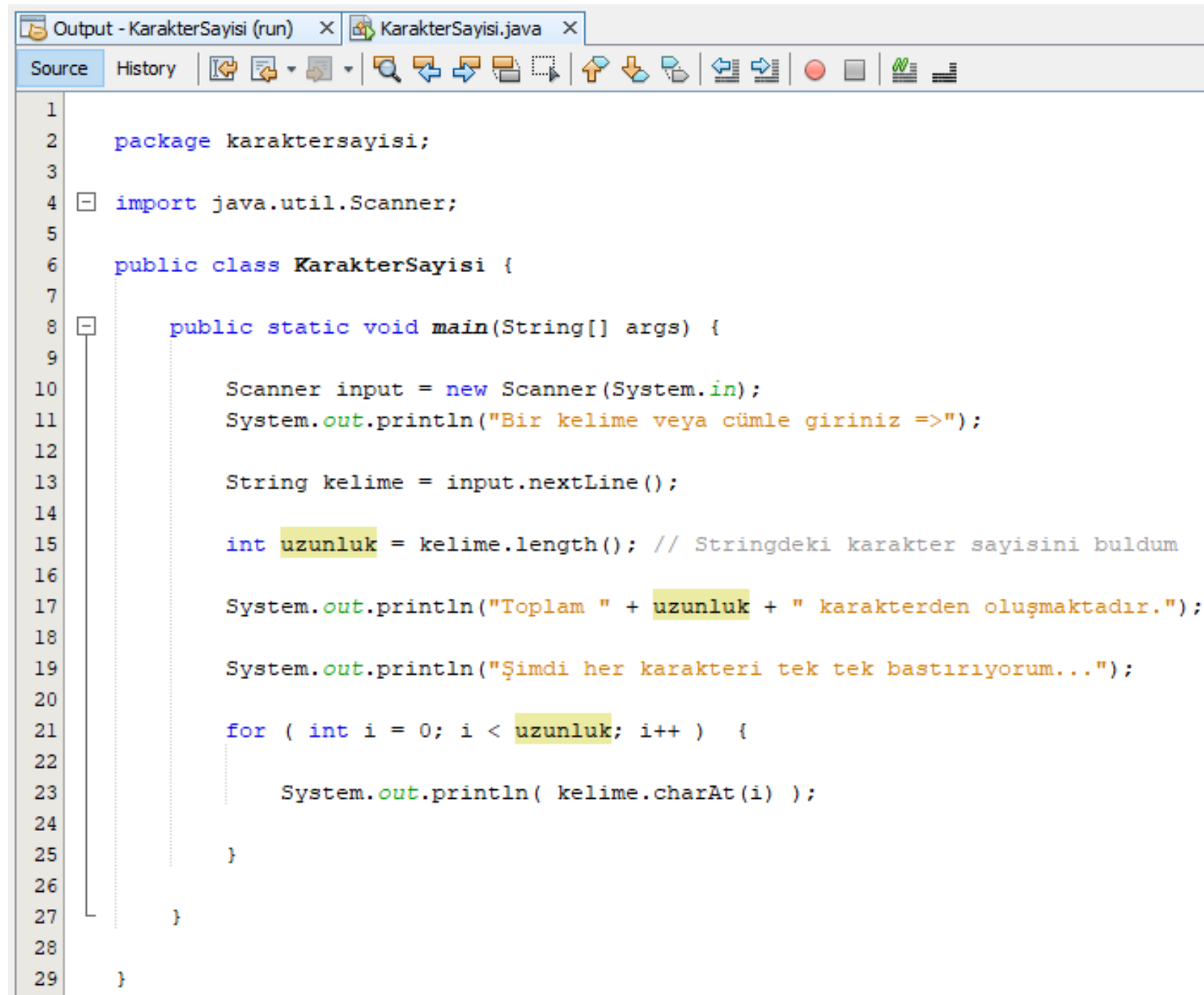
length() String sınıfı metodu



```
Output - KarakterSayisi (run) X KarakterSayisi.java X
run:
Bir kelime veya cümle giriniz =>
OZDENIZCI KOSE
Toplam 14 karakterden oluşmaktadır.
Şimdi her karakteri tek tek bastırıyorum...
O
Z
D
E
N
I
Z
C
I

K
O
S
E

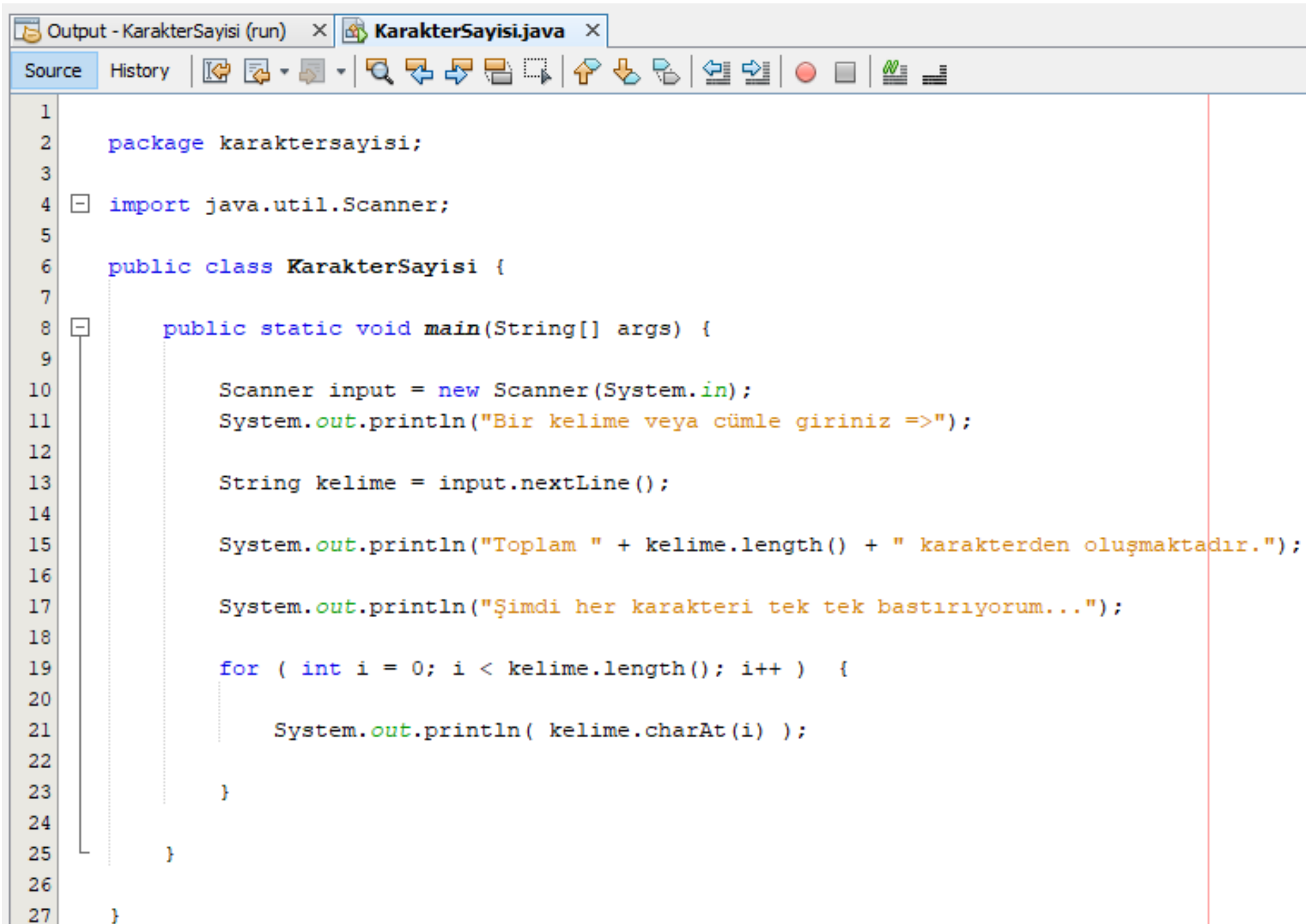
BUILD SUCCESSFUL (total time: 3 seconds)
```



The screenshot shows an IDE window titled "Output - KarakterSayisi (run)" and "KarakterSayisi.java". The code is as follows:

```
1
2 package karactersayisi;
3
4 import java.util.Scanner;
5
6 public class KarakterSayisi {
7
8     public static void main(String[] args) {
9
10         Scanner input = new Scanner(System.in);
11         System.out.println("Bir kelime veya cümle giriniz =>");
12
13         String kelime = input.nextLine();
14
15         int uzunluk = kelime.length(); // Stringdeki karakter sayisini buldum
16
17         System.out.println("Toplam " + uzunluk + " karakterden oluşmaktadır.");
18
19         System.out.println("Şimdi her karakteri tek tek bastırıyorum...");
20
21         for ( int i = 0; i < uzunluk; i++ ) {
22
23             System.out.println( kelime.charAt(i) );
24
25         }
26
27     }
28
29 }
```

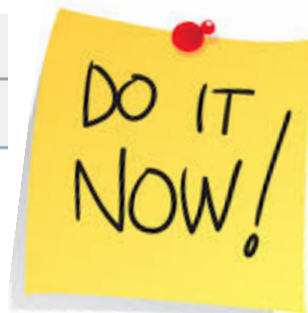




The screenshot shows an IDE window titled "KarakterSayisi.java". The code is as follows:

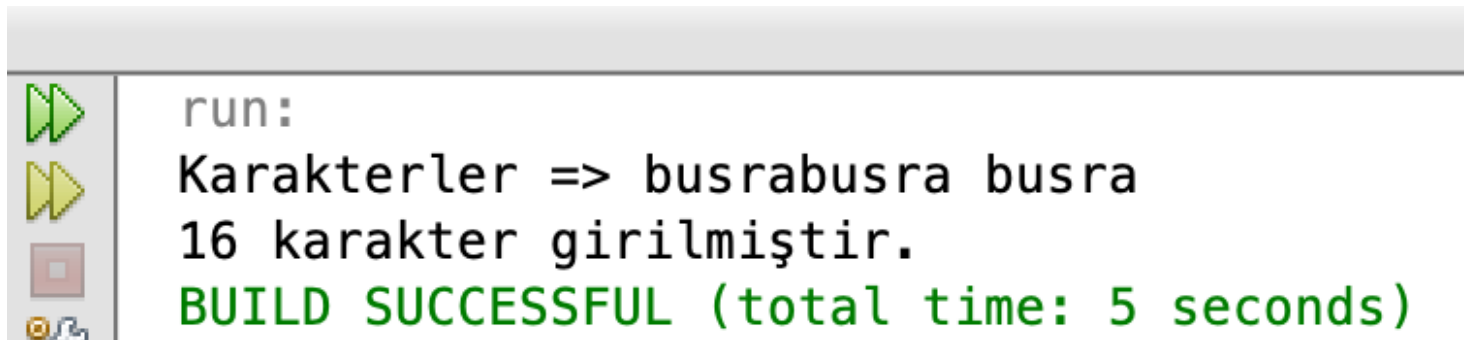
```
1
2 package karactersayisi;
3
4 import java.util.Scanner;
5
6 public class KarakterSayisi {
7
8     public static void main(String[] args) {
9
10         Scanner input = new Scanner(System.in);
11         System.out.println("Bir kelime veya cümle giriniz =>");
12
13         String kelime = input.nextLine();
14
15         System.out.println("Toplam " + kelime.length() + " karakterden oluşmaktadır.");
16
17         System.out.println("Şimdi her karakteri tek tek bastırıyorum...");
18
19         for ( int i = 0; i < kelime.length(); i++ ) {
20
21             System.out.println( kelime.charAt(i) );
22
23         }
24
25     }
26
27 }
```

The code is a Java program that takes a string input from the user and prints the total number of characters in the string, followed by printing each character on a new line. The package is named "karactersayisi".

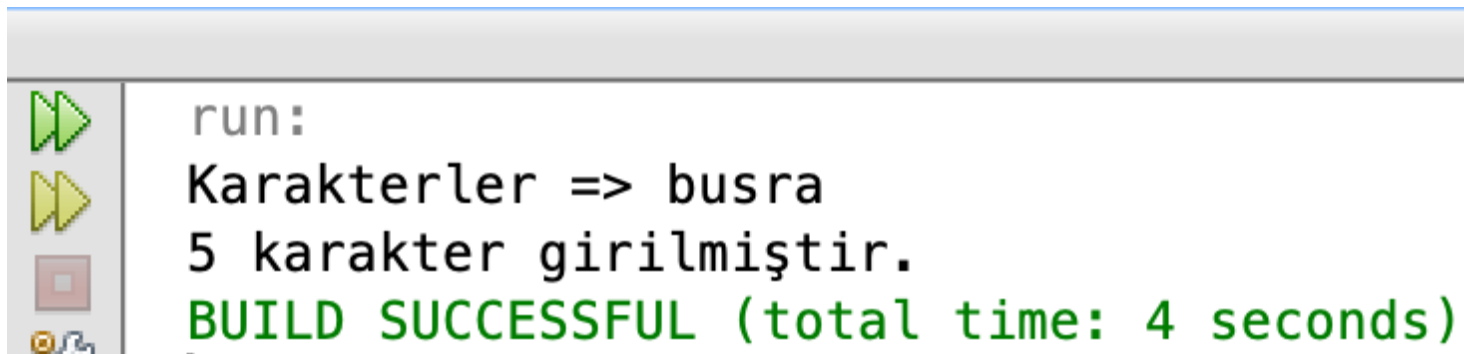


KarakterAdedi.java

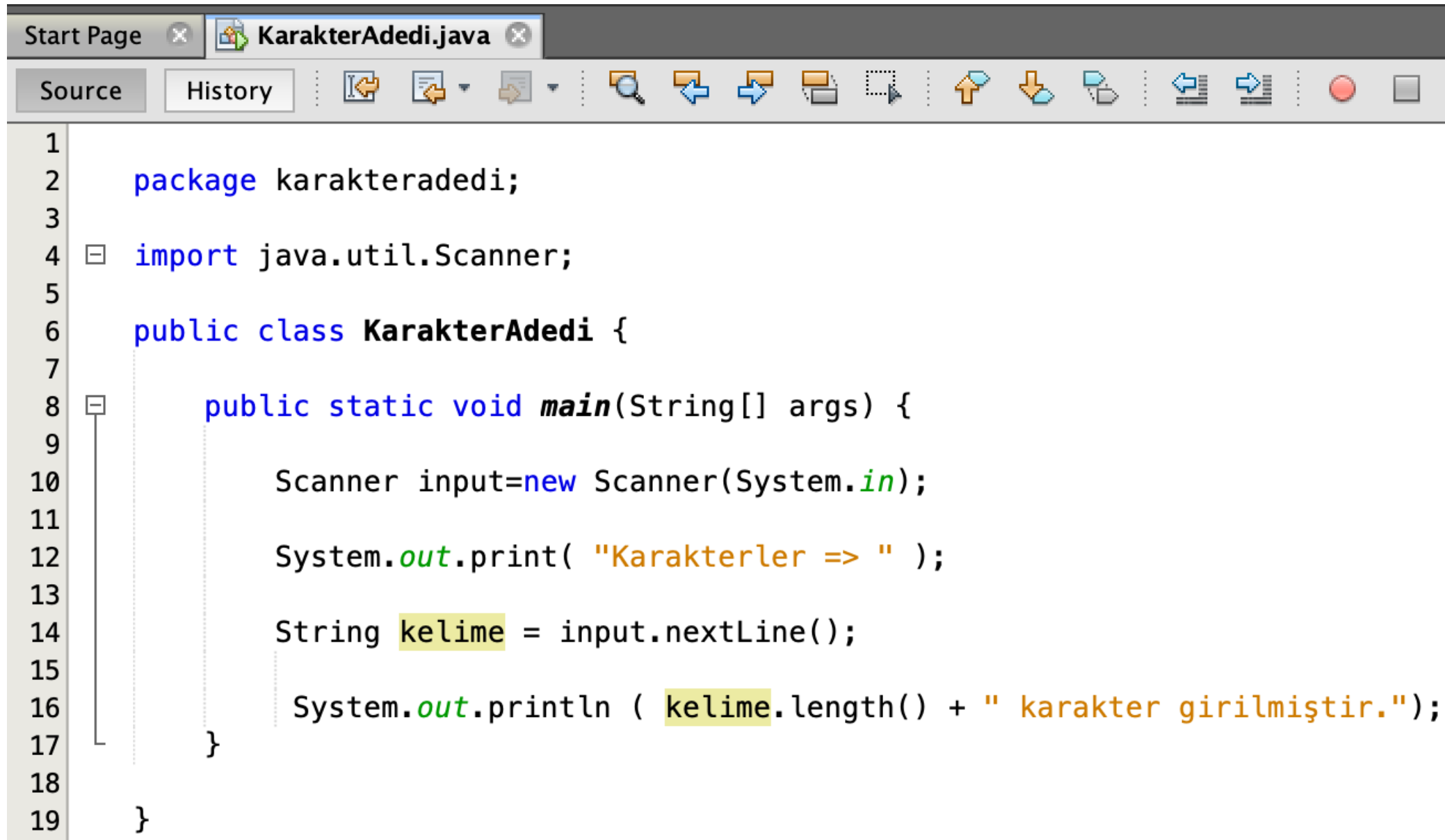
- Kullanıcı Enter tuşuna basana kadar girdiği karakter adedini hesaplayan programı yazınız.

A screenshot of an IDE's Run Console window. It shows the execution of a Java program. The output text is: 'run: Karakterler => busrabusra busra', '16 karakter girilmiştir.', and 'BUILD SUCCESSFUL (total time: 5 seconds)'. On the left side of the console, there are four icons: a green play button, a yellow play button, a red square stop button, and a small bug icon.

```
run:
Karakterler => busrabusra busra
16 karakter girilmiştir.
BUILD SUCCESSFUL (total time: 5 seconds)
```

A screenshot of an IDE's Run Console window, similar to the one above. It shows the execution of the same Java program with different input. The output text is: 'run: Karakterler => busra', '5 karakter girilmiştir.', and 'BUILD SUCCESSFUL (total time: 4 seconds)'. The same four icons (green play, yellow play, red square stop, and bug) are visible on the left.

```
run:
Karakterler => busra
5 karakter girilmiştir.
BUILD SUCCESSFUL (total time: 4 seconds)
```



The screenshot shows an IDE window titled 'KarakterAdedi.java'. The code is as follows:

```
1
2 package karakteradedi;
3
4 import java.util.Scanner;
5
6 public class KarakterAdedi {
7
8     public static void main(String[] args) {
9
10         Scanner input=new Scanner(System.in);
11
12         System.out.print( "Karakterler => " );
13
14         String kelime = input.nextLine();
15
16         System.out.println ( kelime.length() + " karakter girilmiştir.");
17     }
18
19 }
```

The code is a Java program that prompts the user to enter a string and then prints the length of the string. The variable `kelime` is highlighted in yellow in the original image. The IDE interface includes a 'Source' tab, a 'History' tab, and a toolbar with various icons for file operations and navigation.



Kaçış Karakterleri

Escape Characters

- Aşağıdaki komut doğru bir ifade midir?

`System.out.println(" He said "Java is fun " ");`

- Hayır, **Compiler hatası** verecektir.
- Çünkü ikinci çift tırnak, String sonunu ifade edeceğinden compiler geri kalan karakterleri anlamlandıramayacak ve bir hata üretecektir.

Kaçış Karakterleri

Escape Characters

- Bu sorunu çözmek için özel karakterler kullanılır.
- Kaçış karakterleri taksim (\) karakteri ile başlar.
- Örneğin, \t bir kaçış karakteridir, Tab karakterini sağlar veya bunu \u0020 şeklinde Unicode olarak da yazmak mümkündür.

Kaçış Karakterleri

Escape Characters

- Komutu bu şekilde yazarsak:

```
System.out.println("He said \"Java is fun\" ");
```

- Çıktı:

```
He said "Java is fun"
```

Karakter	Tanım	ASCII Sayısı	Neden?
\a	Zil	07 ₁₀	<p><i>Bu karakterleri görsel olarak göremeyiz ve klavye kullanarak yazamayız</i></p> <p><i>Örneğin yeni satıra geçme karakteri, klavye kullanarak oluşturulamaz</i></p> <p><i>Oysa klavye kullanarak '\n' yazabiliriz ve bu karakter bilgisayar tarafından yeni satıra geçme karakteri olarak algılanır</i></p>
\b	Geri Al	08 ₁₀	
\t	Yatay Sekme	09 ₁₀	
\n	Yeni Satır	10 ₁₀	
\v	Dikey Sekme	11 ₁₀	
\f	Form Besleme	12 ₁₀	
\r	Satır Başı	13 ₁₀	
\e	Çıkış	27 ₁₀	
\"	Çift Tırnak	34 ₁₀	<p><i>String değerlerin solunda ve sağında kullanılan sonlandırma tırnağı ile karışmaması için kullanılır.</i></p>
\'	Tek Tırnak	39 ₁₀	<p><i>Karakter değerlerin solunda ve sağında kullanılan sonlandırma tırnağı ile karışmaması için kullanılır.</i></p>
\\	Taksim	92 ₁₀	<p><i>\ kullanıldığında hemen sonrasında bir kaçış karakteri bekleneceğinden karışıklık olmaması için kullanılır.</i></p>

```
Start Page x KacisKarakterOrnek.java x
Source History
1
2 package kaciskarakteriornek;
3
4 import java.util.Scanner;
5
6 public class KacisKarakterOrnek {
7
8     public static void main(String[] args) {
9
10         System.out.println("Sıra No\tİsim:\tSoyisim:");
11
12         Scanner input = new Scanner(System.in);
13
14         String isim, soyisim;
15
16         for ( int i = 1 ; i <= 3; i++ )
17         {
18             isim = input.nextLine();
19             soyisim = input.nextLine();
20             System.out.print(i+ "\t" +isim+ "\t" +soyisim );
21             System.out.print("\n");
22         }
23
24     }
25
26 }
```



```
run:
Sıra No İsim:   Soyisim:
a
b
1          a      b
c
d
2          c      d
d
e
3          d      e
BUILD SUCCESSFUL (total time: 8 seconds)
```


Character Sınıfı Metotları

TABLE 4.6 Methods in the Character Class

<i>Method</i>	<i>Description</i>
<code>isDigit(ch)</code>	Returns true if the specified character is a digit.
<code>isLetter(ch)</code>	Returns true if the specified character is a letter.
<code>isLetterOfDigit(ch)</code>	Returns true if the specified character is a letter or digit.
<code>isLowerCase(ch)</code>	Returns true if the specified character is a lowercase letter.
<code>isUpperCase(ch)</code>	Returns true if the specified character is an uppercase letter.
<code>toLowerCase(ch)</code>	Returns the lowercase of the specified character.
<code>toUpperCase(ch)</code>	Returns the uppercase of the specified character.

```
1
2 package karactersinifiornekler;
3
4 public class KarakterSinifiOrnekler {
5
6     public static void main(String[] args) {
7
8         System.out.println("a karakteri bir sayı mıdır? " + Character.isDigit('a'));
9         System.out.println("a karakteri bir harf midir? " + Character.isLetter('a'));
10        System.out.println("a karakteri küçük bir harf midir? " + Character.isLowerCase('a'));
11        System.out.println("a karakteri büyük bir harf midir? " + Character.isUpperCase('a'));
12        System.out.println("T karakterini küçük karf yapalım: " + Character.toLowerCase('T'));
13        System.out.println("q karakterini büyük karf yapalım: " + Character.toUpperCase('q'));
14
15        boolean sonuc = Character.isDigit('4');
16
17        char sonuc2 = Character.toUpperCase('p');
18
19        char sonuc3 = Character.toLowerCase('B');
20
21        System.out.println("4 karakteri bir sayı mıdır? " + sonuc );
22
23        System.out.println("p karakterini büyük harf yapalım: " + sonuc2 );
24
25        System.out.println("B karakterini küçük harf yapalım: " + sonuc3 );
26
27    }
28
29 }
```













```
run:
a karakteri bir sayı mıdır? false
a karakteri bir harf midir? true
a karakteri küçük bir harf midir? true
a karakteri büyük bir harf midir? false
T karakterini küçük karf yapalım: t
q karakterini büyük karf yapalım:  Q
4 karakteri bir sayı mıdır? true
p karakterini büyük harf yapalım: P
B karakterini küçük harf yapalım: b
BUILD SUCCESSFUL (total time: 0 seconds)
```

SayimiDegilmi.java





- Kullanıcının gireceği bir karakterin, sayı türünde bir karakter olup olmadığını bulan programı yazınız.







```
run:
Bir karakter giriniz => 3
Sayı !
BUILD SUCCESSFUL (total time: 1 second)
```



```
run:
Bir karakter giriniz => !
Sayı değil !
BUILD SUCCESSFUL (total time: 4 seconds)
```



```
run:
Bir karakter giriniz => x
Sayı değil !
BUILD SUCCESSFUL (total time: 2 seconds)
```



```
run:
Bir karakter giriniz => 7
Sayı !
BUILD SUCCESSFUL (total time: 1 second)
```

```
Start Page x SayimiDegilmi.java x
Source History
1
2 package sayimidegilmi;
3
4 import java.io.IOException;
5
6 public class SayimiDegilmi {
7
8     public static void main(String[] args) throws IOException {
9
10         char c;
11         System.out.print( "Bir karakter giriniz => " );
12         c = (char) System.in.read();
13
14         boolean b = Character.isDigit(c);
15
16         if (b)
17             System.out.print( "Sayı ! " );
18
19         else
20             System.out.print( "Sayı değil ! " );
21     }
22 }
```



KarakterlerinAnalizi.java



- Kullanıcıdan alacağınız bir String'in büyük harf, küçük harf ve sayı türündeki karakter sayısını ekrana tablo formatında tab kaçış karakteri kullanarak yazdıran bir Java programı oluşturunuz.

```
run:
Bir String giriniz => bUSRa koSE bUsra123
Analiz          Adet
Büyük Harf      6
Küçük Harf      8
Sayı            3
BUILD SUCCESSFUL (total time: 16 seconds)
```

```
run:
Bir String giriniz => he11o W0r1d
Analiz          Adet
Büyük Harf      2
Küçük Harf      5
Sayı            3
BUILD SUCCESSFUL (total time: 11 seconds)
```

Java'da Özel Anlamalı Kelimeler

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

Soru 1: Bir dik üçgenin hipotenüsünü hesaplayan ve sonucu döndüren hipotenusHesapla() metodunu oluşturunuz. Karekök almak için Math.sqrt() metodunu kullanabilirsiniz. Kullanıcıdan bir dik üçgenin iki kısa kenar uzunluğunu okuyan ve hipotenusHesapla() metodunu çağıran main() test programını da geliştiriniz.

$$\text{hipotenüs}^2 = \text{kenar1}^2 + \text{kenar2}^2$$

İki kısa kenarı giriniz => 3 4

Hipotenüs 5

```
MetotSorular.java
package metotsorular;

import java.util.Scanner;

public class MetotSorular {

    public static void main(String[] args) {

        Scanner busra = new Scanner (System.in);

        System.out.println("Birinci kenar giriniz: ");
        double kenar1 = busra.nextDouble();

        System.out.println("İkinci kenar giriniz: ");
        double kenar2 = busra.nextDouble();

        double sonuc = hipotenusHesapla (kenar1, kenar2);

        System.out.println ("Hipotenüs " + sonuc);

    }

    public static double hipotenusHesapla (double a, double b) {

        double hipoKare = a*a + b*b;

        double hipo = Math.sqrt (hipoKare);

        return hipo;

    }

}
```

```
Output - MetotSorular (run)
run:
Birinci kenar giriniz:
3
İkinci kenar giriniz:
4
Hipotenüs 5.0
BUILD SUCCESSFUL (total time: 2 seconds)
```


Soru 2: Bir sayının faktöriyel değerini hesaplayan ve sonucu ekrana yazdıran faktoriyelHesapla() metodunu oluşturunuz. Kullanıcıdan bir sayı okuyan ve faktoriyelHesapla() metodunu çağıran main() test programını da geliştiriniz. $F(n) = n * (n-1) * (n-2) * \dots * 1$

Bir sayı giriniz => 5

5! = 120

Bir sayı giriniz => 3

3! = 6

```
MetotSorular.java
package metotsorular;

import java.util.Scanner;

public class MetotSorular {

    public static void main(String[] args) {

        Scanner busra = new Scanner (System.in);

        System.out.println("Bir sayı giriniz: ");
        int x = busra.nextInt();

        faktoriyelHesapla (x);

    }

    public static void faktoriyelHesapla (int a) {

        int sonuc=1;

        for (int i=a; i>0; i--)
            sonuc=sonuc*i;

        System.out.println(sonuc);

    }

}
```

```
Output - MetotSorular (run)
run:
Bir sayı giriniz:
5
120
BUILD SUCCESSFUL (total time: 1 second)
```

Soru 3: Celcius cinsinden hava sıcaklığı değerini, Fahrenheit cinsine çeviren ve sonucu döndüren `fahrenheitCevir()` metodu yazınız. Kullanıcıdan Celcius cinsinden hava sıcaklığı değerini gerçel sayı şeklinde okuyan ve `fahrenheitCevir()` metodunu çağıran `main()` test programını da geliştiriniz.

NOT: $F = C \times (9 / 5) + 32$

Celcius cinsinden hava sıcaklığı => 10

Fahrenheit cinsinden hava sıcaklığı 50

```
MetotSorular.java x
package metotsorular;

import java.util.Scanner;

public class MetotSorular {

    public static void main(String[] args) {

        Scanner busra = new Scanner (System.in);

        System.out.println("Sıcaklık celsius degerini giriniz: ");
        double a = busra.nextDouble();

        double sonuc= fahrenheitCevir (a);

        System.out.println("Fahrenheit cinsinden hava sıcaklığı "+sonuc);

    }

    public static double fahrenheitCevir (double c) {

        return ( ( c * 9/5) +32 ) ;

    }

}
```

```
Output - MetotSorular (run) x
run:
Sıcaklık celsius degerini giriniz:
10
Fahrenheit cinsinden hava sıcaklığı 50.0
BUILD SUCCESSFUL (total time: 2 seconds)
```

Soru 4: Aşağıda verilen fonksiyonu hesaplayan ve sonucu ekrana yazdıran fonksiyonHesapla() metodunu geliştiriniz. Kullanıcıdan bir tamsayı olarak ilgili fonksiyonu çağıran main() test programını da geliştiriniz.

$$f(n) = 1^2 + 2^2 + 3^2 + \dots n^2$$

Bir sayı giriniz => 4

Sonuc 30

```
MetotSorular.java x
package metotsorular;

import java.util.Scanner;

public class MetotSorular {

    public static void main(String[] args) {

        Scanner busra = new Scanner (System.in);

        System.out.println("Bir sayı giriniz: ");
        int n = busra.nextInt();

        fonksiyonHesapla (n);

    }

    public static void fonksiyonHesapla (int k) {

        int toplam=0;

        for (int i=1; i<=k; i++)
            toplam = toplam + (i*i);

        System.out.println ("Sonuc " +toplam);

    }

}
```

```
Output - MetotSorular (run) x
run:
Bir sayı giriniz:
10
Sonuc 385
BUILD SUCCESSFUL (total time: 3 seconds)
```

Soru 5: Üç tamsayıdan büyük olanı hesaplayan ve sonucu döndüren `buyukSayi()` metodunu; küçük olanı hesaplayan ve sonucu döndüren `kucukSayi()` metodunu; üç tamsayının ortalamasını hesaplayan ve sonucu ekrana yazdıran `ortalamaHesapla()` metodunu geliştiriniz. Kullanıcıdan üç tamsayı olarak ilgili fonksiyonları çağıran `main()` test programını da geliştiriniz.

Üç tamsayı giriniz => 40 24 26

En küçük sayı 24

En büyük sayı 54

Ortalama 30.0

```
MetotSorular.java x
package metotsorular;

import java.util.Scanner;

public class MetotSorular {

    public static void main(String[] args) {
        Scanner busra = new Scanner (System.in);
        System.out.println("Üç tamsayı giriniz: ");
        int a = busra.nextInt();
        int b = busra.nextInt();
        int c = busra.nextInt();

        int buyuk = buyukSayi (a, b, c);
        int kucuk = kucukSayi (a, b, c);
        System.out.println ("En büyük sayı "+buyuk);
        System.out.println ("En küçük sayı "+kucuk);

        ortalamaHesapla (a, b, c);

    }

    public static int buyukSayi (int k, int l, int m) {
        if ( k> l && k >m)
            return k;
        else if (l>k && l>m)
            return l;
        else
            return m;
    }
}
```

```
public static int kucukSayi (int x, int y, int z) {
    if (x<y && x<z)
        return x;
    else if (y<x && y<z)
        return y;
    else
        return z;
}
```

```
public static void ortalamaHesapla (int x, int y, int z) {
    System.out.println("Ortalama " + (x+y+z)/3 );
}
```

Any Questions?