



clarusway



clarusway



clarusway



clarusway

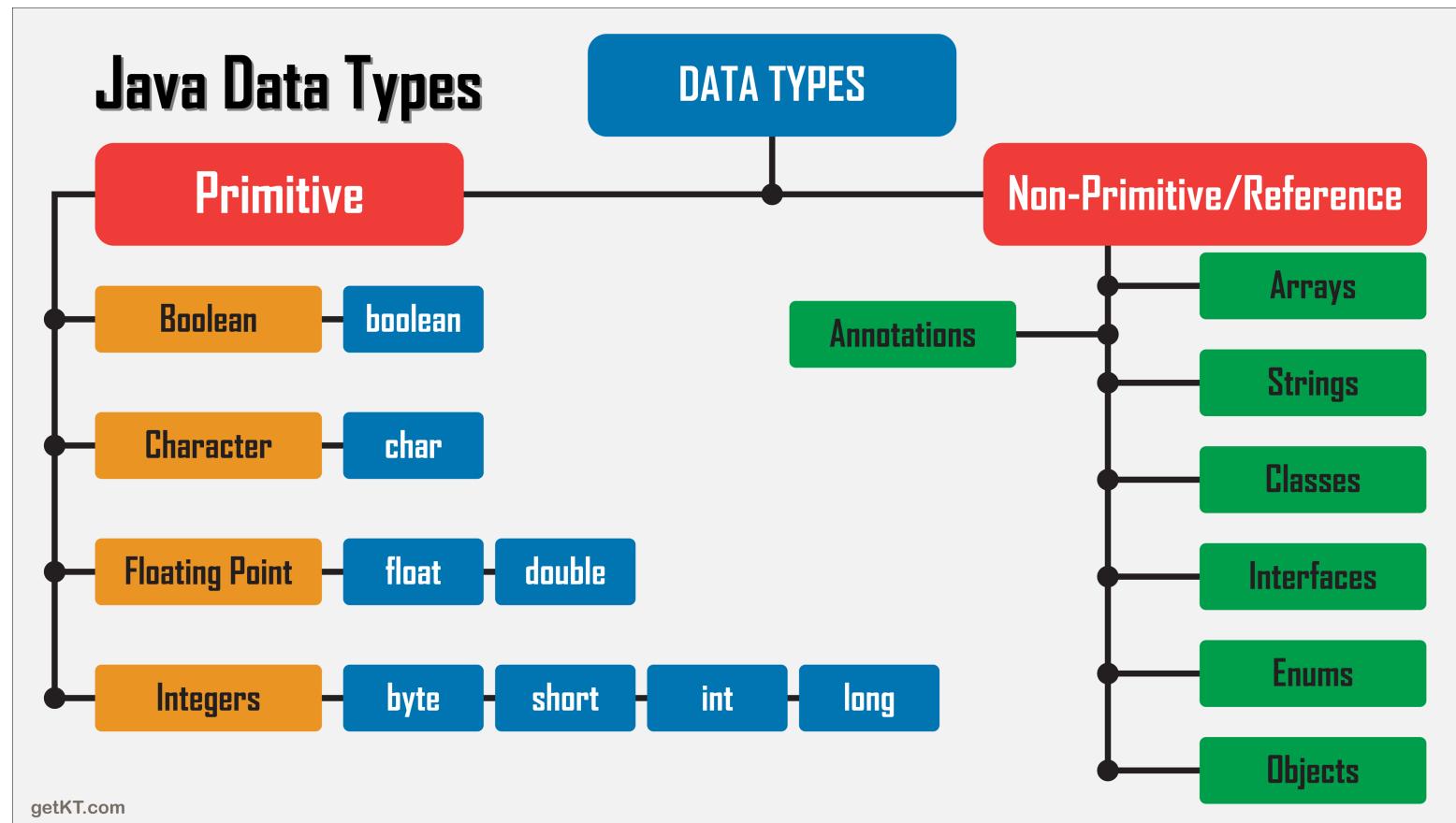


clarusway

Data Types Wrapper Classes

Lesson:
JAVA Chapter 02

Data Types (Veri Türleri)



Data Types (Veri Türleri)

JAVA'da iki data tipi kullanılmaktadır.,

- **Primitive Data Types:** boolean, char, byte, short, int, long, float ve double
- **Non-Primitive Data Types:** String, Array(Diziler), List Class object

Tamsayı

Data Turu	Hafiza Boyut	minimum değer	maximum değer
3- byte	8 bit	$-2^7 = -128$	$2^7 - 1 = 127$
4- short	16 bit	$-2^{15} = -32.768$	$2^{15} - 1 = 32.767$
5- int	32 bit	$-2^{31} = -2.147.483.648$	$2^{31} - 1 = 2.147.483.647$
6- long	64 bit	$-2^{63} = -9.223.372.036.854.755.808$	$2^{63} - 1 = 9.223.372.036.854.755.807$

Ondalıklı Sayı

Data Turu	Hafiza Boyut	min-max değer	Ondalık basamak sayısı
7- float	32 bit	$\pm 3.40282347E+38F$	6-7 basamak
8- double	64 bit	$\pm 1.79769313486231570E+308$	15-16 basamak



Primitive Data Types (İllkel Veri Türleri)

1) **boolean** Data Type: True veya false değerlerini alır. Hafızada **1 bit** kullanır.

Sadece doğru veya yanlış sonuç verilecek variable'larda kullanılır.

2) **char** Data Type: Tek karakter değerini alır. Hafızada **16 bit** kullanır.

Harf, sayı veya sembol bakılmaksızın sadece 1 karakter kullanacak variable'larda kullanılır.

```
boolean manager = true;  
boolean tester = false;
```

```
char letter = 'a';  
char digit = '3';  
char symbol = '#';
```

* **TRICK :)** char değerleri iki '' (single quote) arasına yazılır.



Primitive Data Types (İlkel Veri Türleri)

3) byte Data Type: -128'den 127'e (dahil) tamsayılar için kullanılabilir. Hafızada **8 bit** kullanır.

`byte age = 33;`

4) short Data Type: -32.768'den 32.767'e (dahil) tamsayılar için kullanılabilir. Hafızada **16 bit** kullanır.

`short aracKm = 27,324;`

5) int Data Type: -2.147.483.648'den 2.147.483.647'e (dahil) tamsayılar için kullanılabilir. Hafızada **32 bit** kullanır.

`int motorOmurKm = 67,324.564;`

6) long Data Type: -9.223.372.036.854.755.808'den ,223.372.036.854.755.807'e (dahil) tamsayılar için kullanılabilir.

Hafızada **64 bit** kullanır.



Primitive Data Types (İlkel Veri Türleri)

7) float Data Type: Küçük ondalık sayılar için kullanılabilir. Hafızada **64 bit** kullanır.

```
float floatNum = -2.123456f;
```

8) double Data Type: Büyük ondalık sayılar için kullanılabilir. Hafızada **64 bit** kullanır.

```
double doubleNum = -2.1234567907800000000123
```

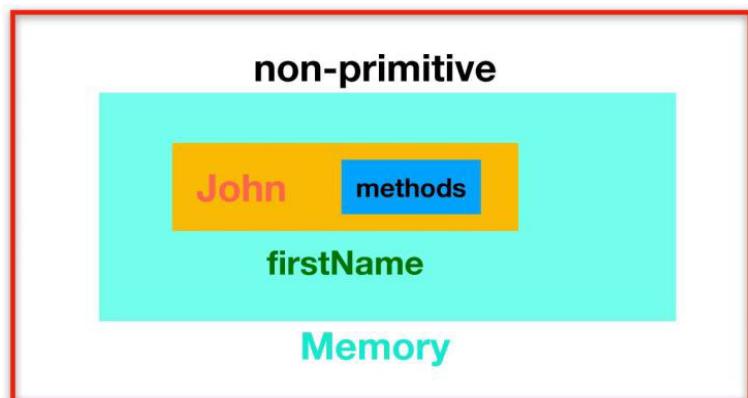
* **TRICK :** float sayıların sonunda “ **f** ” konulmalıdır, yoksa JAVA sayıyı double olarak derler.



Non-Primitive Data Types (ilkel Olmayan Veri Türleri)

String Data Type:

String bir character dizisidir. Ancak Java'da dize, bir karakter dizisini temsil eden bir nesnedir. Kelimeler, cümleler, matematiksel işlem yapılmayacak sayısal değerler de **String** object olarak tanımlanabilir.



```
String bottcamp = "The IT Career of Your Dreams Starts Here!";  
String phoneNumber = "+1 (571) 360 66 77";  
String firstCharacter = "C";
```

* TRICK :) String ifadeler çift tırnak “ ” (double quotes) arasına yazılır yoksa derleme hatası alınır.

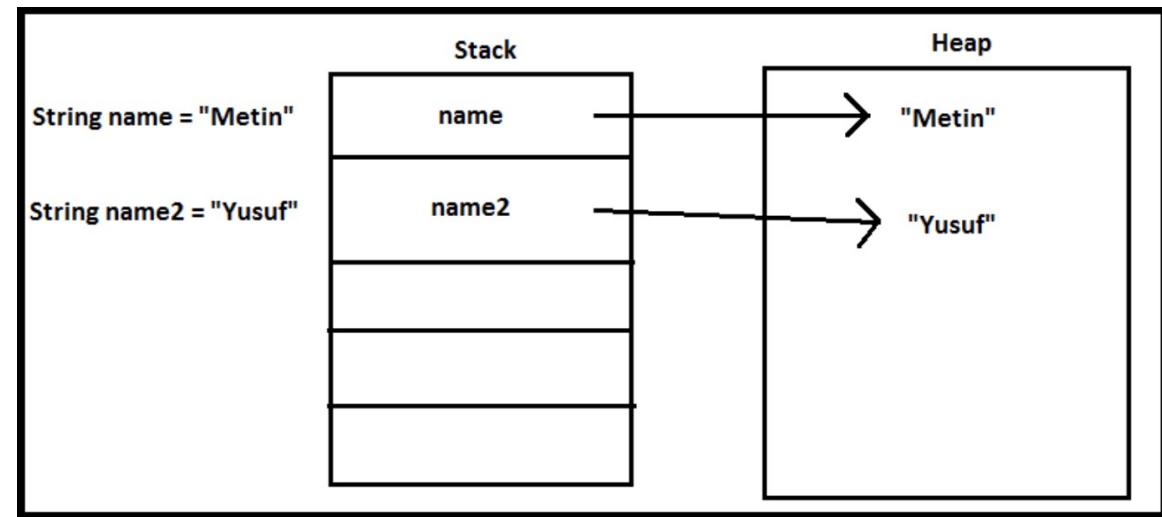
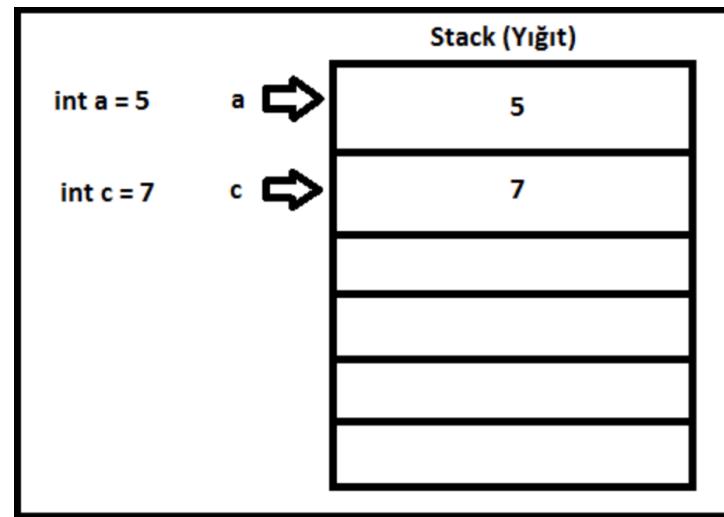


Primitive & Non-Primitive Data Types (İlkel & İlkel olmayan Veri Türleri)

Primitives	Non-Primitives
Tumu Java tarafından olusturulmustur.	Java tarafından olusturulanlar oldugu gibi biz de olusturabiliriz
Sadece deger icerirler, variable ile kullanilacak hazir method'lari yoktur.	Icerdikleri degerin yaninda olusturulduklari class'dan gelen hazir method'lar da barindirirlar.
Bir deger atamadan olusturulabilir ama kullanabilmek icin mutlaka deger atanmalidir.	Deger atanmadan null olarak isaretlenebilirler.
Data turu isimleri kucuk harfle baslar (int, char vb)	Data turu isimleri buyuk harfle baslar . (String vb..)
Primitive data turundeki variable'larin hafizada kapladiklari alan sabittir. Degeri kucuk de olsa, buyuk de olsa hafizada belirlenen miktarda alan ayrılır.	Hafizada kapladiklari alan sabit degildir. Data turu ve icerdigi datanin buyuklugune gore hafizada yer kaplarlar. (bir kelime veya binlerce kelime iceren String'lerin boyutlari farkli olacaktir)



Non-Primitive Data Types (ilkel Olmayan Veri Türleri)

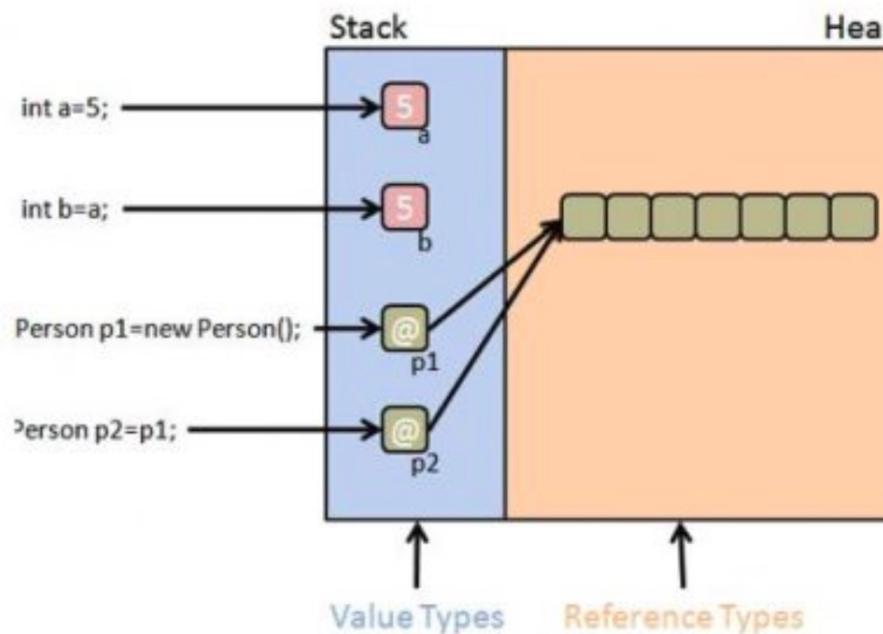


- 1) Primitive data type'ları sadece value içerir, non-primitive data type'lari value ve methodlar içerir.
- 2) Primitive data type'ları küçük harf ile, non-primitive data type'lari büyük harf ile başlar.
- 3) Primitive data type'ları JAVA tanımlamıştır. Biz primitive data type create edemez.
- 4) Primitive data type'ları büyülükleri data type'ing göre sabittir. Non-primitive data type'lari için sabit büyülük söz konusu değildir.



Memory (Hafıza) Kullanımı

JAVA'da iki farklı memory kullanılır.



Stack => small

Heap => huge

1) Stack Memory: Value type (değer tipli) nesneler Java'da Primitive tipler byte, char, int, long, double, boolean... gibi tiplere değerleri ve Non-primitive datalara (Object) ait referansları (adres) içerir.

2) Heap Memory: Non-primitive data'ları depolamak (**store**) için kullanılır.



Memory (Hafıza) Kullanımı

Kısaca Stack

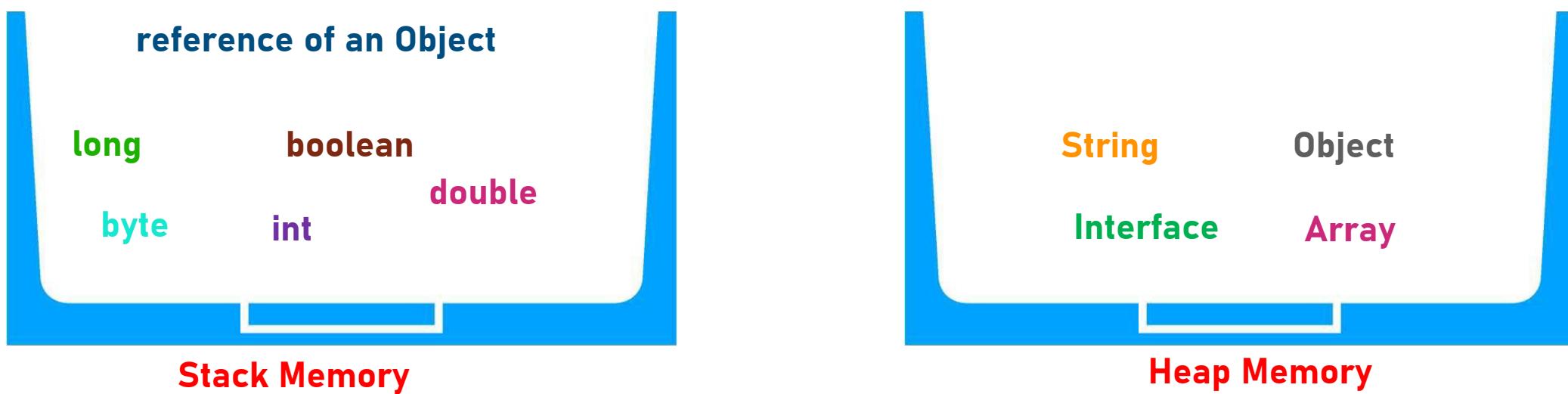
- **LIFO** (Last in First out) son giren ilk çıkar mantığında çalışır.
- Veri depolama alanı çok geniş olmadığından kullanımı **kolay ve hızlıdır**.
- Veriler **Big and Little Endian** (artan ya da azalan) adres mantığında tutulur.
- Derleme zamanında oluşturulur.
- Life time (yaşam süresi) kısa olan değişkenler tutulur. Ör; local variables (yerel değişkenler).
- **Static allocation** (Kullanılacak depolama alanının boyutu biliniyorsa stack işe yarayacaktır.)
- Bir Java uygulamasında sadece tek bir stack yoktur. Her bir thread'in kendi stack'i vardır.
- Bir stack üzerindeki veriye kendi thread'inden başka bir thread erişemez.
- Doğru kullanılmadığında **java.lang.StackOverflowError** hatası alınır.

Kısaca Heap

- Heap stack'e göre **daha büyük boyuta sahiptir**.
- Stack'e göre daha fazla alana sahip olduğundan stack'e göre daha yavaştır.
- Heap'teki veriler karışık şekilde sıralanır.
- Çalışma zamanında oluşturulur.
- **Dynamic allocation** (Kullanılacak depolama alanının boyutu bilinmiyorsa ya da sürekli değişken olacak ise heap kullanmak doğru olacaktır.)
- Bir Java uygulamasında tüm thread'ler için sadece bir tane heap bulunmaktadır.
- Doğru kullanılmadığında **java.lang.OutOfMemoryError** hatası alınır.



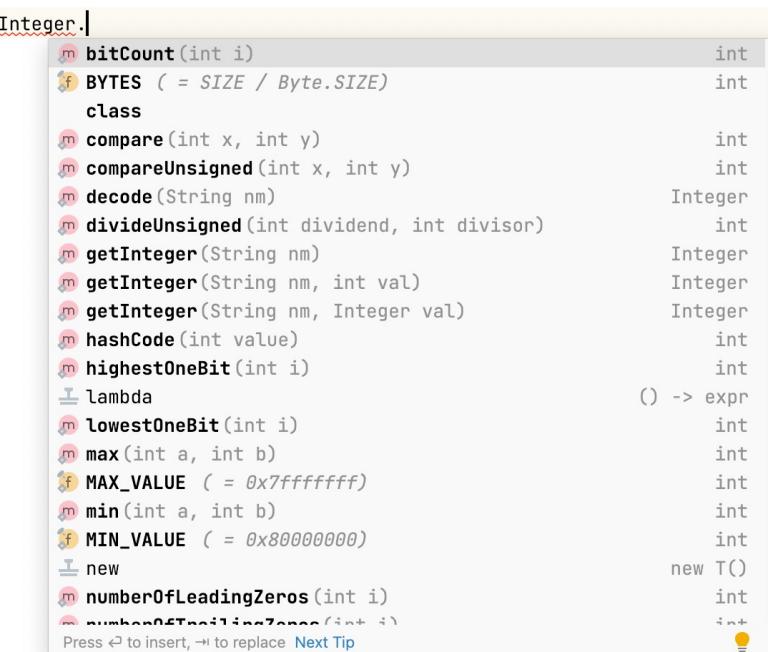
Variables (Değişken) Oluşturma



Wrapper Class

- ▶ Primitive data type sadece data saklayıp Class olmadıkları için method barındıramazlar.
- ▶ **Wrapper Class'lar primitive(ilkel) veri türleri olan **int, float, short, long** vs. gibi türleri birer nesne (object) olarak kullanarak method uygulama imkanı sağlar.**
- ▶ **Wrapper Class'lardan** create edilen objelere primitive data Type değer atanabilir.

```
int age = 48;  
  
Integer yas = 33;  
  
yas = age;
```



Wrapper Class

- ▶ **Wrapper Class'lar primitive(ilkel) veri türleri olan int, float, short, long vs. gibi türleri birer nesne (object) olarak kullanarak method uygulama imkanı sağlar.**

Primitif	Sarmalayıcı (Wrapper)
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
boolean	Boolean
char	Character

```
int sayi=10;
Integer sayiW= 20;
System.out.println(Integer.MAX_VALUE); // 2147483647
System.out.println(Integer.max( a: 34, b: 465)); // 465

boolean kontrol=true;
Boolean kont=false;
String knt="false";
boolean sonuc = Boolean.valueOf(knt);

char chr='*';
Character ch='p';
char chr2=101;
System.out.println(Character.valueOf(chr2)); // e
System.out.println(Character.isDigit( ch: '5')); // true
System.out.println(Character.isAlphabetic( codePoint: '9')); // false
System.out.println(Character.isAlphabetic( codePoint: 'a')); //true
```

ASCII Table

ASCII control characters		ASCII printable characters								Extended ASCII characters							
00	NULL (Null character)	32	space	64	@	96	`	128	Ç	160	á	192	Ł	224	Ó		
01	SOH (Start of Header)	33	!	65	A	97	a	129	Ü	161	í	193	ł	225	ß		
02	STX (Start of Text)	34	"	66	B	98	b	130	é	162	ó	194	ł	226	ö		
03	ETX (End of Text)	35	#	67	C	99	c	131	â	163	ú	195	ł	227	ô		
04	EOT (End of Trans.)	36	\$	68	D	100	d	132	ä	164	ñ	196	—	228	ö		
05	ENQ (Enquiry)	37	%	69	E	101	e	133	à	165	N	197	+	229	Ö		
06	ACK (Acknowledgement)	38	&	70	F	102	f	134	à	166	»	198	å	230	¶		
07	BEL (Bell)	39	.	71	G	103	g	135	ç	167	°	199	Ã	231	þ		
08	BS (Backspace)	40	(72	H	104	h	136	é	168	¿	200	Ł	232	ı		
09	HT (Horizontal Tab)	41)	73	I	105	i	137	ë	169	®	201	Ł	233	ü		
10	LF (Line feed)	42	*	74	J	106	j	138	è	170	¬	202	Ł	234	ø		
11	VT (Vertical Tab)	43	+	75	K	107	k	139	í	171	½	203	Ł	235	ú		
12	FF (Form feed)	44	,	76	L	108	l	140	î	172	¼	204	Ł	236	ý		
13	CR (Carriage return)	45	-	77	M	109	m	141	í	173	í	205	=	237	ÿ		
14	SO (Shift Out)	46	.	78	N	110	n	142	À	174	«	206	‡	238			
15	SI (Shift In)	47	/	79	O	111	o	143	Á	175	»	207	¤	239	‘		
16	DLE (Data link escape)	48	0	80	P	112	p	144	É	176	„	208	ð	240	≡		
17	DC1 (Device control 1)	49	1	81	Q	113	q	145	æ	177	„	209	ð	241	±		
18	DC2 (Device control 2)	50	2	82	R	114	r	146	Æ	178	„	210	É	242	„		
19	DC3 (Device control 3)	51	3	83	S	115	s	147	ò	179	—	211	È	243	¾		
20	DC4 (Device control 4)	52	4	84	T	116	t	148	ö	180	—	212	È	244	¶		
21	NAK (Negative acknowl.)	53	5	85	U	117	u	149	ò	181	À	213	í	245	§		
22	SYN (Synchronous idle)	54	6	86	V	118	v	150	ó	182	Á	214	í	246	÷		
23	ETB (End of trans. block)	55	7	87	W	119	w	151	ú	183	À	215	í	247	:		
24	CAN (Cancel)	56	8	88	X	120	x	152	ÿ	184	©	216	í	248	:		
25	EM (End of medium)	57	9	89	Y	121	y	153	Ó	185	—	217	í	249	:		
26	SUB (Substitute)	58	:	90	Z	122	z	154	Ü	186	—	218	í	250	:		
27	ESC (Escape)	59	:	91	[123	{	155	ø	187	—	219	í	251	:		
28	FS (File separator)	60	<	92	\	124		156	£	188	—	220	í	252	:		
29	GS (Group separator)	61	=	93]	125	}	157	Ø	189	¢	221	í	253	:		
30	RS (Record separator)	62	>	94	^	126	~	158	×	190	¥	222	í	254	:		
31	US (Unit separator)	63	?	95	_			159	ƒ	191	—	223	í	255	nnbsp		
127	DEL (Delete)																

```
char harf= 'a';
int sayi= 100;

System.out.println( harf + sayi); // 197
System.out.println( harf+1); // 98

char yeniharf=(char)(harf+1);
System.out.println(yeniharf); // b
```



Interview Question

Interview Question

Interview

1- Verilen sayı1 ve sayı2 variable'larının değerlerini değiştiren (SWAP) bir program create ediniz.

ÖRN: sayı1 = 20 ve sayı2 = 34;

Kod çalıştırıktan sonra

sayı1 = 34 ve sayı2 = 20

2- Verilen sayı1 ve sayı2 variable'larının değerlerini 3. bir variable olmadan değiştiren (SWAP) bir program yazınız.



THANKS!

Any questions?

HalUk Bilgin | JAVA Backend Developer

BAŞARI GAYRETE AŞIKTIR ☺

