CLARUSWAY

WAY TO REINVENT YOURSELF

Data Types Wrapper Classes

Lesson: JAVA Chapter 02


clarusway

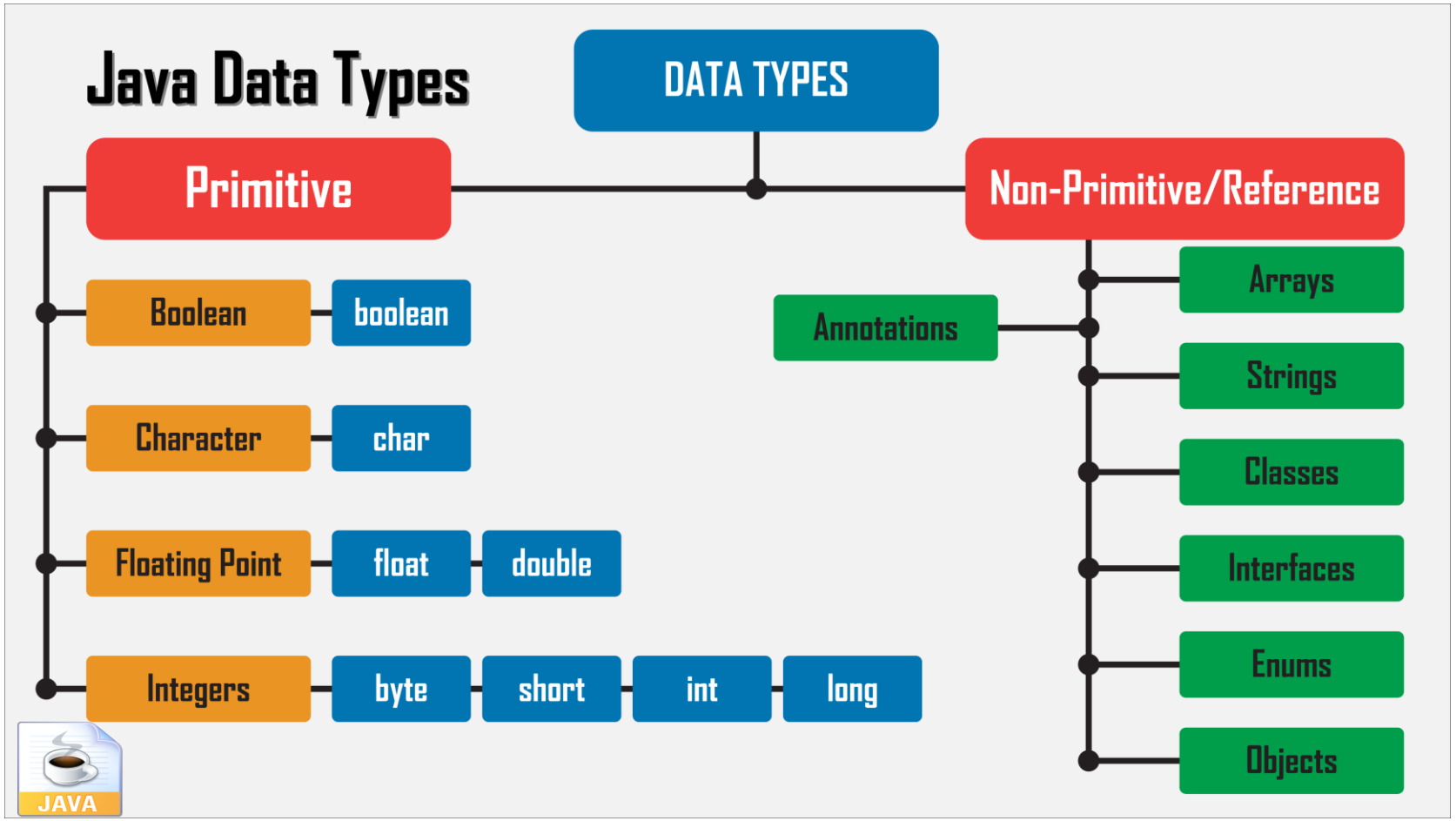

clarusway


clarusway


clarusway


clarusway

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

Data Type	Size	Description
`byte`	1 byte	Stores whole numbers from -128 to 127
`short`	2 bytes	Stores whole numbers from -32,768 to 32,767
`int`	4 bytes	Stores whole numbers from -2,147,483,648 to 2,147,483,647
`long`	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
`float`	4 bytes	Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits
`double`	8 bytes	Stores fractional numbers. Sufficient for storing 15 decimal digits
`boolean`	1 bit	Stores true or false values
`char`	2 bytes	Stores a single character/letter or ASCII values

Primitive Data Types (İlkel Veri Türleri)

1) boolean Data Type: True veya false değerlerini alır. Hafızada 1 bit kullanır.
Sadece doğru veya yanlış sonuç verilebilecek 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 cymbol = '#';
```

* 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;

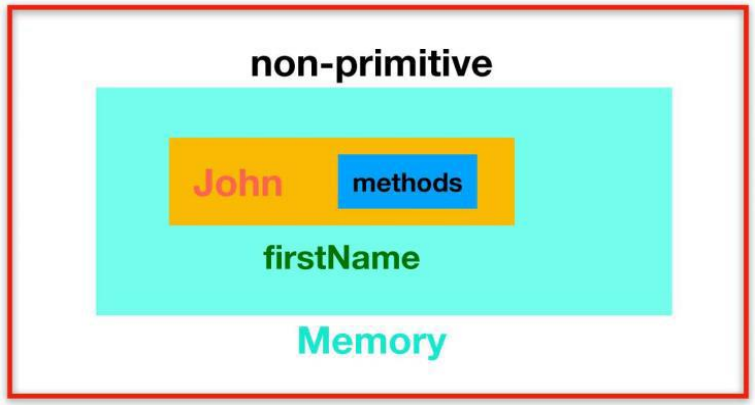
8) double Data Type: Büyük ondalık sayılar için kullanılabilir. Hafızada 64 bit kullanır.
double doubleNum = -2.12345679078000000000123

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

Non-Primitive Data Types (İlkel Olmayan Veri Türleri)

String Data Type:

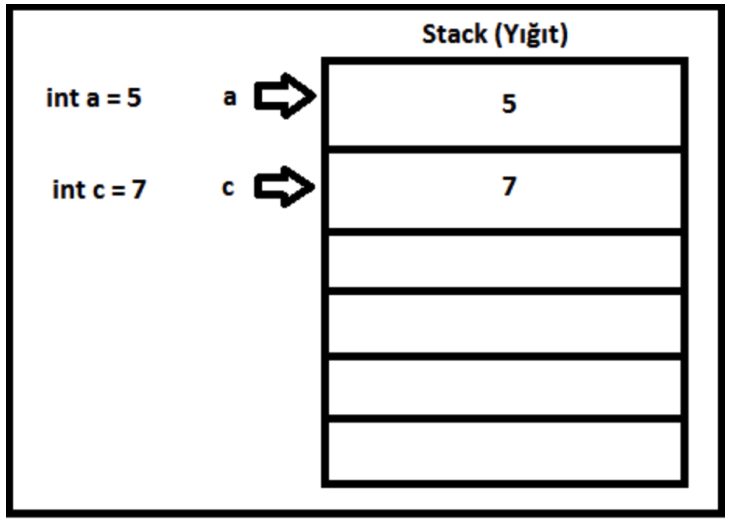
String bir character dizisidir. Ancak Java'da dize, bir karakter dizisini temsil eden bir nesnedir.
Kelimeler, cümleler, matematiksel işlem yapılmayacak sayısal değerler de String object olarak tanımlanabilir.

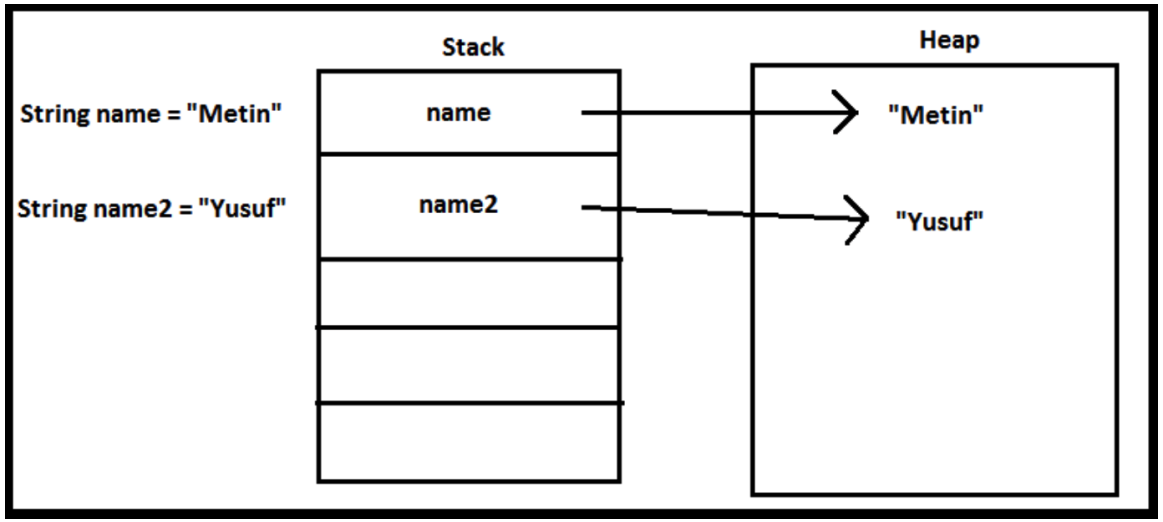



```
String bottcamp = "The IT Career of Your Dreams Starts Here! ";
String phoneNumber = "+1 (571) 360 66 77";
String firstCharacter = "C";
```

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

Non-Primitive Data Types (İlkel Olmayan Veri Türleri)

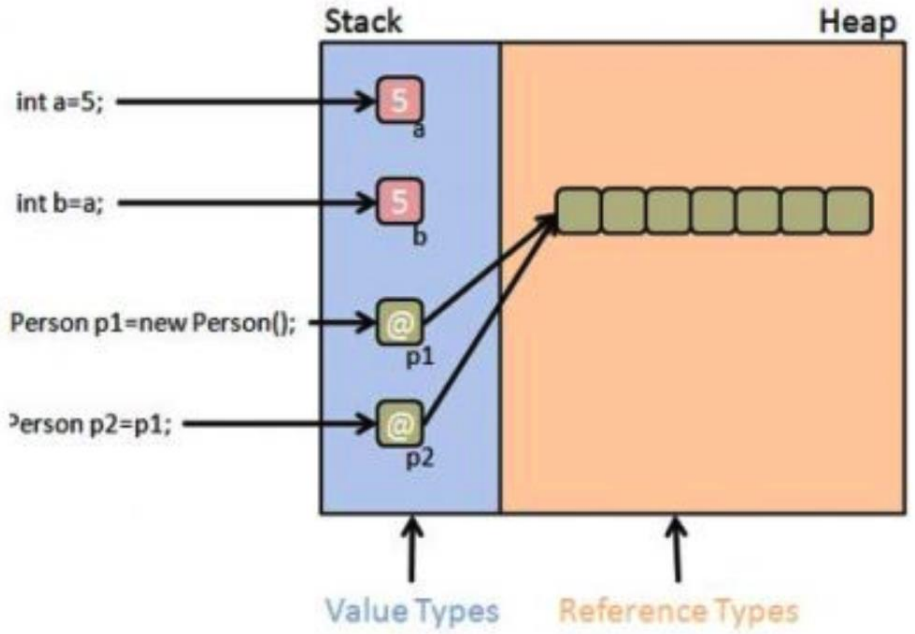




1) Primitive data type'ları sadece value içerir, non-primitive data type'lari value ve methodlar içerir.
2) Primitive data type'ları küçük harf ile, non-primitive data type'lari büyük harf ile başlar.
3) Primitive data type'ları JAVA tanımlamıştır. Biz primitive data type create edemez.
Non-primitive data'ları biz de JAVA da create edebilir. ÖRN: String'i JAVA'ya ait bir data type'tır.
4) Primitive data type'ları büyüklükleri data type'ing göre sabittir. Non-primitive data type'ları için sabit büyüklük 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 tiplerbyte, 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

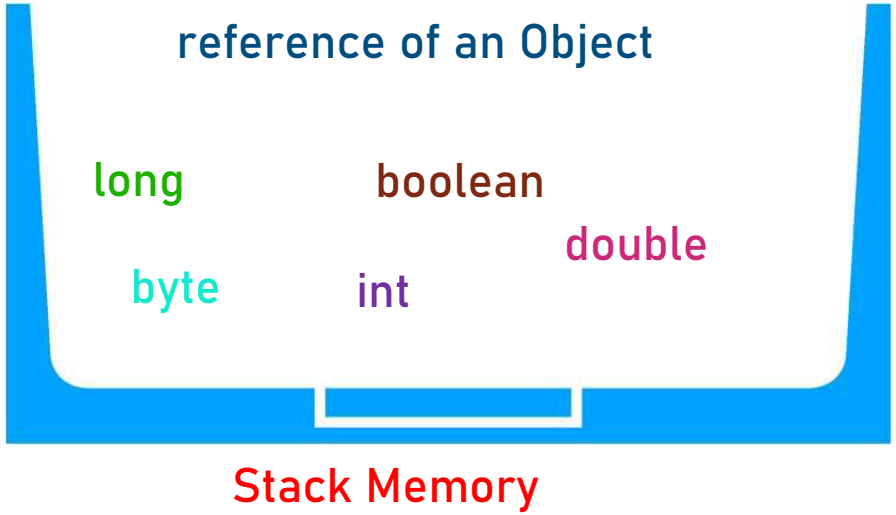
Kısaca Stack

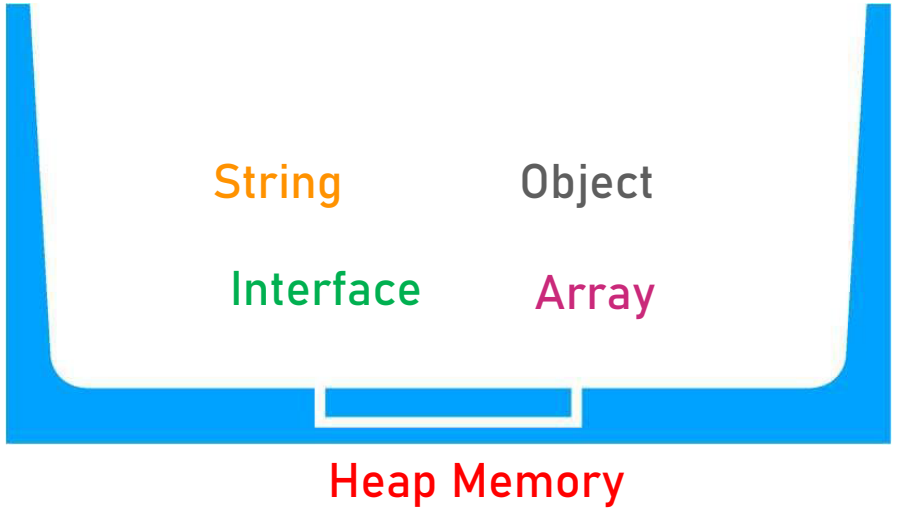
- LIFO (Last in First out) son giren ilk çıkar mantığında çalışır.
- Veri depolama alanı çok geniş olmadığından kullanımı kolay ve hızlıdır.
- Veriler Big and Little Endian (artan ya da azalan) adres mantığında tutulur.
- Derleme zamanında oluşturulur.
- Life time (yaşam süresi) kısa olan değişkenler 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

Wrapper Class'lar primitive (ilkel) veri türleri olan int, float, short, long vs. gibi türleri birer nesne (object) olarak kullanarak method uygulamaları için bize bir yol sunmaktadır.

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

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

    int num1 = Integer.MIN_VALUE;
    System.out.println(num1);

    int num2 = Integer.MAX_VALUE;
    System.out.println(num2);

    int num3 = Byte.MIN_VALUE;
    System.out.println(num3);

    int num4 = Byte.MAX_VALUE;
    System.out.println(num4);

}
}
```

```
-2147483648

2147483647

-128

127
```

ASCII control characters

00	NULL	(Null character)
01	SOH	(Start of Header)
02	STX	(Start of Text)
03	ETX	(End of Text)
04	EOT	(End of Trans.)
05	ENQ	(Enquiry)
06	ACK	(Acknowledgement)
07	BEL	(Bell)
08	BS	(Backspace)
09	HT	(Horizontal Tab)
10	LF	(Line feed)
11	VT	(Vertical Tab)
12	FF	(Form feed)
13	CR	(Carriage return)
14	SO	(Shift Out)
15	SI	(Shift In)
16	DLE	(Data link escape)
17	DC1	(Device control 1)
18	DC2	(Device control 2)
19	DC3	(Device control 3)
20	DC4	(Device control 4)
21	NAK	(Negative acknowl.)
22	SYN	(Synchronous idle)
23	ETB	(End of trans. block)
24	CAN	(Cancel)
25	EM	(End of medium)
26	SUB	(Substitute)
27	ESC	(Escape)
28	FS	(File separator)
29	GS	(Group separator)
30	RS	(Record separator)
31	US	(Unit separator)
127	DEL	(Delete)

ASCII printable characters

32	space	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

Extended ASCII characters

128	Ç	160	á	192	└	224	Ó
129	ü	161	í	193	┴	225	ß
130	é	162	ó	194	┬	226	Ô
131	â	163	ú	195	├	227	Ò
132	ä	164	ñ	196	─	228	õ
133	à	165	Ñ	197	┼	229	Õ
134	å	166	ª	198	ã	230	µ
135	ç	167	º	199	Ã	231	þ
136	ê	168	¿	200	╚	232	Þ
137	ë	169	®	201	╔	233	Ú
138	è	170	¬	202	╩	234	Û
139	ï	171	½	203	╦	235	Ù
140	î	172	¼	204	╠	236	ý
141	ì	173	¡	205	═	237	Ý
142	Ä	174	«	206	╬	238	¯
143	Å	175	»	207	¤	239	´
144	É	176	░	208	ð	240	≡
145	æ	177	▒	209	Ð	241	±
146	Æ	178	▓	210	Ê	242	‗
147	ô	179	│	211	Ë	243	¾
148	ö	180	┤	212	È	244	¶
149	ò	181	Á	213	ı	245	§
150	û	182	Â	214	Í	246	÷
151	ù	183	À	215	Î	247	¸
152	ÿ	184	©	216	Ï	248	°
153	Ö	185	╣	217	┘	249	¨
154	Ü	186	║	218	┌	250	·
155	ø	187	╗	219	█	251	¹
156	£	188	╝	220	▄	252	³
157	Ø	189	¢	221	¦	253	²
158	×	190	¥	222	Ì	254	■
159	ƒ	191	┐	223	▀	255	nbsp

Interview Question

Interview Question

1- Verilen sayi1 ve sayi2 variable'larının değerlerini değiştiren (SWAP) bir program create ediniz.

ÖRN: sayi1 = 20 ve sayi2 = 34;
Kod çalıştıktan sonra
sayi1 = 34 ve sayi2 = 20

2- Verilen sayi1 ve sayi2 variable'larının değerlerini 3. bir variable olmadan değiştiren (SWAP) bir program yazınız.

THANKS!

Any questions?

Garry T. - Full Stack Automation Engineer
Haluk B. - JAVA Backend Developer