



Variables (Degisken) Olusturma

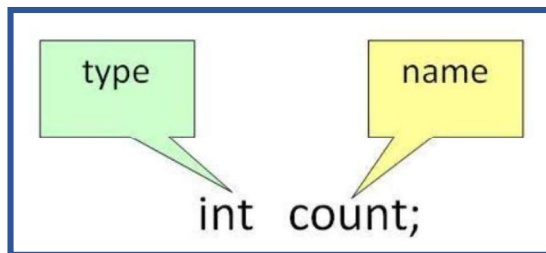
Declaration

Variable bellekte (**memory**) ayrılmis olan alanin (**reserved area**) adidir.

Variable icinde deger saklayan bir konteynirdir (**container**).

Bir değışkende saklanan değ r, program y r t l rken değıştirilebilir.

Java'da, t m değışkenler kullanılmadan  nce deklare edilmelidir (variable declaration)



Variable declaration icin iki seyi belirtmemiz gerekiyor

- 1- Data type (data turu)
- 2- Variable Name (degisken ismi)



Variables Deger Atama (Assignment)

Varolan bir variable'a deger atamaya assignment (atama) denir.

1- Deger atamasi yapilirken data turune uygun deger atanmalidir. Diger turlu Java hata verir.

```
5 public class Example {  
6  
7     String isim ="Mehmet";  
8     boolean ogrenciMi =false;  
9     int not=85;  
10    double ortalama= 78.3;  
11  
12    String ad =75;  
13    boolean emekliMi ="true";  
14    int maas=true;  
15    double yas= "kuru";  
16 }
```



Variables Deger Atama (Assignment)

2- İlk once declaration, daha sonra atama yapılabilir.

```
String isim ;  
boolean ogrenciMi;  
int not;  
double ortalama;  
  
isim ="Mehmet";  
ogrenciMi =false;  
not=85;  
ortalama= 78.3;
```

3- Bir defa declaration yapıldıktan sonra, birden fazla atama yapılabilir. Java son degeri tutar, oncekini siler.

```
5 public class Example {  
6     public static void main(String[] args) {  
7  
8  
9         int level=1;  
10  
11  
12  
13         level=2;  
14  
15  
16  
17         level=3;  
18  
19  
20     }  
21 }
```



Variables Deger Atama (Assignment)

4- Ayni data turunde birden fazla variable tek komutla deklare edilebilir.

```
9      int level,yas,maas;  
10  
11     level=5;  
12     yas=20;  
13     maas=10000;
```

5- Ayni data turunde birden fazla variable tek komutla deklare edilip deger atanabilir.

```
9      int level=5, yas=20, maas=10000;
```



Data Types

Java'da iki data tipi kullanılmaktadır

- **Primitive Data Types** : boolean, char, byte, short, int, long, float ve double
- **Non- Primitive Data Types** : String,

ilerleyen derslerde göreceğimiz primitive olmayan Array, List, Object gibi her data non-primitive'dir.



Primitive Data Types

1) **boolean** Data Type: true veya false barindirir. Hafizada **1 bit** kullanir

Sadece dogru veya yanlis seklinde cevap verilebilecek variable'larda kullanilir

```
boolean isExpensive = true;
```

```
boolean isCold = false;
```

2) **char** Data Type : Tek karakter barindirir. Hafizada **16 bit** kullanir

Harf, sayi veya sembol bakilmaksizin sadece 1 karakter kullanacak variable'larda kullanilir

```
char letter = 'a';
```

```
char digit = '3';
```

```
char symbol = '#';
```

Note: char degerlerini single quote arasina yazilir.



Primitive Data Types

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

byte age = 73;

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

short koyNufusu = 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 turkiyeNufusu = 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

7) **float** Data Type: Kucuk ondalik sayilar icin kullanilabilir. Hafizada **64 bit** kullanir

```
float floatVar2 = -2.123456f;
```

Not: float sayilarin sonunda “ **f** ” yazilmalidir, yazilmazsa java sayiyi double kabul eder

8) **double** Data Type: Buyuk ondalik sayilar icin kullanilabilir. Hafizada **64 bit** kullanir

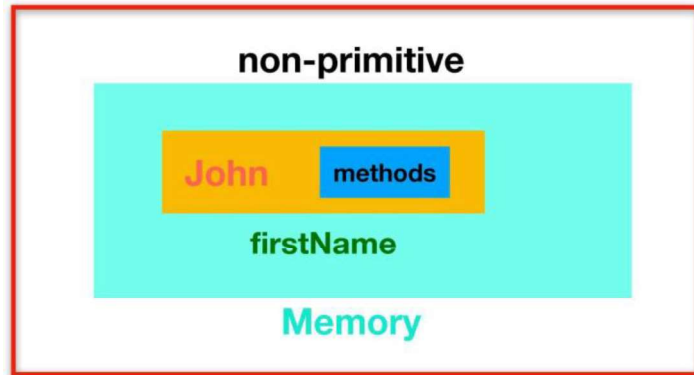
```
double doubleVar2 = -2.12345679078000000000123
```




Non-Primitive Data Type

String Data Type:

String pes pese dizilmiş char'lardan oluşur. Kelimeler, cümleler, matematiksel işlem yapılmayacak sayısal değerler de String olarak tanımlanabilir



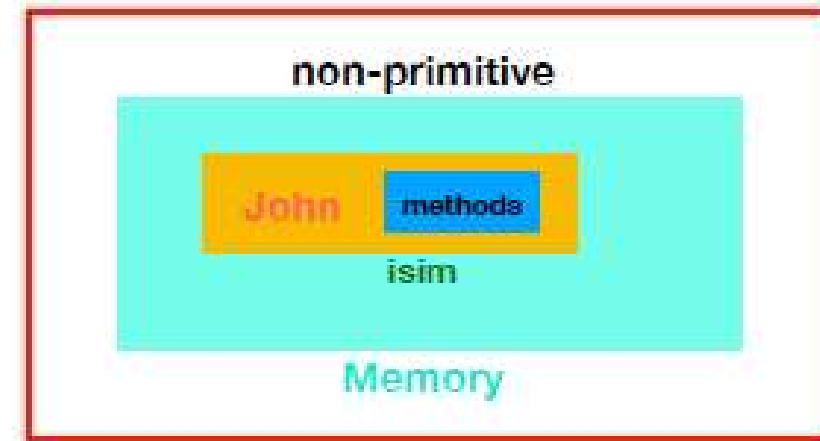
```
String okulAdi = "Yildiz Koleji, Cankaya Ankara #";  
String telNo   = "5321234567";  
String ilkHarf = "A";
```

Note: String'ler çift tırnak (double quotes) arasına yazılır.

Note: Baska non-primitive data type'lar da var, daha sonra öğreneceğiz.



Primitive VS Non-Primitive Data Types



- 1) Primitive'ler sadece value içerir, non-primitive'ler value ve methodlar içerir.
- 2) Primitive'ler küçük harf ile, non-primitive'ler büyük harf ile baslar.
- 3) Primitive'leri Java olusturur biz primitive data turu olusturamayiz.
Non-primitive'leri biz de olusturabiliriz, Java da olusturabilir. Or: String'i Java olusturmudur.
- 4) Primitive'lerin buyuklukleri data type'ing gore sabittir. non-primitive'ler icin sabit buyukluk soz konusu degildir.



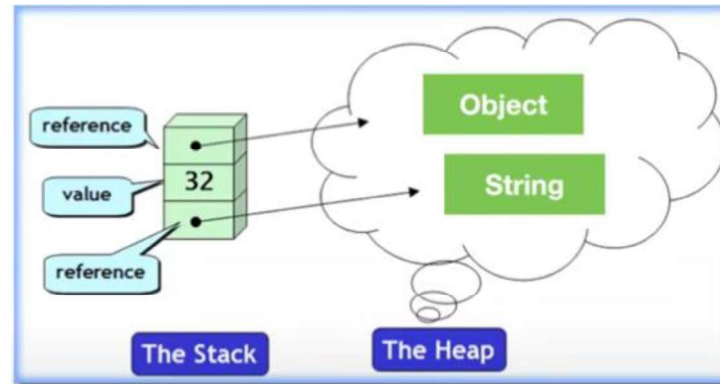
Variable ve Method'lar Nasıl Adlandırılır

1. Java variable isimleri **case sensitive** (Buyuk kucuk harfe duyarlidir)dir.
“money”, “Money” veya “MONEY” birbirinden farklıdır
2. Java variable isimleri “harf”,“\$” veya “_” ile başlamalıdır.
Fakat “\$” ve “_” ile başlamak tavsiye edilmez.
3. Java variable isimlerinde, ilk harften sonra sayı, “\$” ve “_” kullanılabilir.
4. Variable isimleri için Java'ya özel terimler (key word) kullanılamaz. (int, for, if, import vb).
5. Variable isimleri küçük harflerle başlar, camel case kullanılır
6. Variable isimleri 1'den fazla kelime içeriyorsa, ilk kelimedenden sonraki her kelimenin ilk harfi büyük harf ile başlamalıdır. firstName, bigApple, ageJohnWalker gibi. Buna camelCase denir.



Memory (Hafıza) Kullanımı

Javada kullanılan iki hafıza vardır



Stack => small

Heap => huge

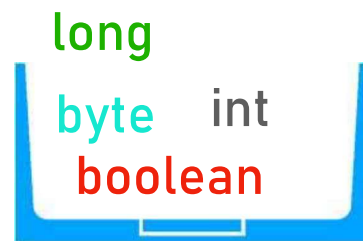
1- **Stack Memory** : primitive data tiplerine ait degerleri ve Non-primitive datalara (Object) ait referanslari(adres) barindirir

2- **Heap Memory** : Non-primitive data'lari depolamak(store) icin kullanilir

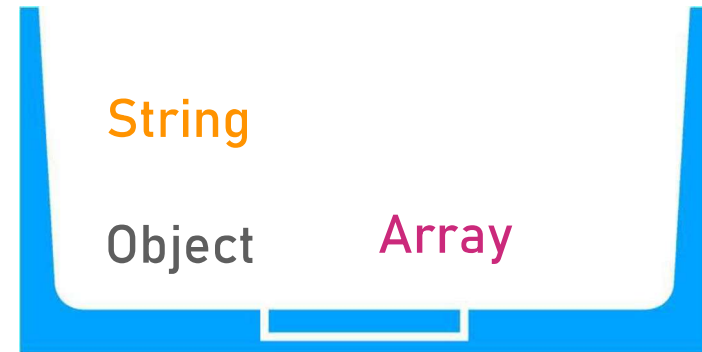


Memory (Hafiza) Kullanimi

reference of an Object



Stack



Heap



Variables Class Work

- 1- Farkli 3 data turunde variable olusturun ve bunlari yazdirin
- 2- isim ve soyisim icin iki variable olusturun ve bunlari
isminiz : Mehmet
soyisminiz : Bulutluoz
sekinde yazdirin
- 3- Iki farkli tamsayi data turunde 2 variable olusturun bunlari toplamini yazdirin
- 4- Bir tamsayi ve bir ondalikli variable olusturun ve bunlari toplamini yazdirin
- 5 – char data turunde bir variable olusturun ve yazdirin
- 6- Bir tamsayi, bir de char degisken olusturun ve bunlari toplamini yazdirin.



ASCII Table

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

Extended ASCII characters			
128	Ç	160	á
129	ü	161	í
130	é	162	ó
131	â	163	ú
132	ä	164	ñ
133	à	165	Ñ
134	á	166	ª
135	ç	167	º
136	ê	168	¿
137	ë	169	@
138	è	170	¬
139	ï	171	½
140	î	172	¼
141	í	173	¡
142	Ä	174	«
143	Å	175	»
144	É	176	•
145	æ	177	•
146	Æ	178	•
147	ö	179	•
148	ø	180	•
149	ò	181	À
150	û	182	Á
151	ù	183	Â
152	ÿ	184	©
153	Ö	185	•
154	Ü	186	•
155	ø	187	•
156	£	188	•
157	Ø	189	•
158	×	190	¥
159	f	191	•
192	Ł	224	Ó
193	ł	225	ß
194	Ł	226	Ô
195	ł	227	Ò
196	Ł	228	ō
197	ł	229	Õ
198	Ł	230	μ
199	ł	231	þ
200	Ł	232	þ
201	Ł	233	Ú
202	Ł	234	Û
203	Ł	235	Ü
204	Ł	236	ý
205	Ł	237	Ý
206	Ł	238	—
207	Ł	239	•
208	Ł	240	≡
209	Ł	241	±
210	Ł	242	≡
211	Ł	243	¾
212	Ł	244	¶
213	Ł	245	§
214	Ł	246	÷
215	Ł	247	•
216	Ł	248	•
217	Ł	249	•
218	Ł	250	•
219	Ł	251	•
220	Ł	252	•
221	Ł	253	•
222	Ł	254	■
223	Ł	255	nbsp



Variables Class Work

Interview Question

1- Verilen sayi1 ve sayi2 variable'larinin degerlerini degistiren (SWAP) bir program yaziniz

Orn : sayi1=10 ve sayi2=20;
kod calistiktan sonra
sayi1=20 ve sayi2=10

2- Verilen sayi1 ve sayi2 variable'larinin degerlerini 3.bir variable olmadan degistiren (SWAP) bir program yapiniz



Kullanıcıdan Deger Alma

1) `Scanner scan = new Scanner(System.in);`

scan : olusturdugumuz scanner'in ismidir ve istedigimiz ismi vermemiz mumkundur. Ancak genelde scan ismi kullanilir.

Bu tur isimlendirmelerde genel kurallara uymamiz kodumuzun anlasilabilir olmasi acisinden faydali olacaktır.

2) `System.out.println("Lutfen 100'den kucuk pozitif iki tamsayi giriniz");`

Kullaniciya girmesini istedigimiz degerler icin aciklayici bilgi vermeliyiz.

Burada aciklama olarak ne yazdirsak kodumuz calisir, hatta birsey yazdirmasak da calisir ancak kullanıcı kendisinden ne istedigimizi bilmezse deger girmesi gerektigini veya ne tur bilgi girmesi gerektigini bilemez



Kullanıcıdan Deger Alma

3) `scan.nextInt()` ile girilen degerleri alabiliriz. Istedigimiz data tipine gore next'ten sonra yazilacak kisim degisir.

```
int num1 = scan.nextInt()  
int num2 = scan.nextInt()
```

<code>nextBoolean()</code>	→	Reads a boolean value from the user
<code>nextByte()</code>	→	Reads a byte value from the user
<code>nextDouble()</code>	→	Reads a double value from the user
<code>nextFloat()</code>	→	Reads a float value from the user
<code>nextInt()</code>	→	Reads a int value from the user
<code>nextLine()</code>	→	Reads a String value from the user
<code>nextLong()</code>	→	Reads a long value from the user
<code>nextShort()</code>	→	Reads a short value from the user



Kullanıcıdan Deger Alma

Sorular

- Soru 1)** Kullanıcıdan iki tamsayı alıp bu sayıların toplam,fark ve carpimlerini yazdırın
- Soru 2)** Kullanıcıdan karenin bir kenar uzunlugunu alın ve karenin cevresini ve alanini hesaplayıp yazdırın
- Soru 3)** Kullanıcıdan yarıçap isteyip cemberin cevresini ve dairenin alanini hesaplayıp yazdırın
- Soru 4)** Kullanıcıdan dikdörtgenler prizmasının uzun, kısa kenarlarını ve yüksekliğini isteyip prizmanın hacmini hesaplayıp yazdırın
- Soru 5)** Kullanıcıdan ismini ve soyismini isteyip aşağıdaki şekilde yazdırın
- İsminiz : Mehmet
Soyisminiz : Bulut
Kursumuza katılımınız alınmıştır,tesekkür ederiz
- Soru 6)** Kullanıcıdan ismini ve soyismini alıp aralarında bir boşluk oluşturarak aşağıdaki şekilde yazdırın
- İsim – soyisim : Mehmet Bulutluoz
- Soru 7)** Kullanıcıdan ismini alıp isminin baş harfini yazdırın.



Data Casting / Veri Sinifi Degistirme

- Java'da kod yazarken bir veri tipinden diğer bir veri tipine aktarım yapmamız gerekebilir.
- Veri tiplerinde bir variable'a , olusturulduđu data tipinden farklı bir data turunden deger atanmasina Data Casting denir.
- Data casting yaparken aklimizdan cikarmamamız gereken konu data tiplerinin sinirlaridir. Data tipinin sinirlarini asan data casting islemlerinde hata almamamız için dikkat etmemiz gereken bazı durumlar olacaktır.
- Hatırlayacağımız şekilde Java'da sayılarla ilgili data tiplerinin siralamasi su şekildeydi

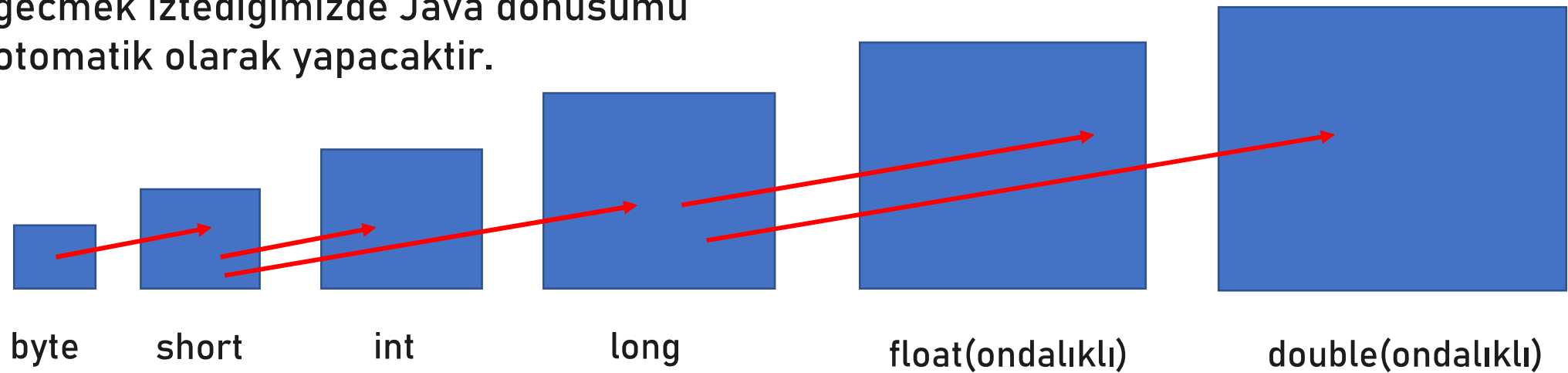
byte < short < int < long < float(ondalıklı) < double(ondalıklı)



Data Casting / Veri Sinifi Degistirme

1) Auto Widening (Otomatik Genisletme)

Dar veri tipinden daha genis bir veri tipine gecmek istedigimizde Java donusumu otomatik olarak yapacaktır.



Orn :

```
byte num1 = 12;  
short num2 = num1;      // yazdirirsak 12 olarak yazdirir  
int num3 = num2;      // yazdirirsak 12 olarak yazdirir  
float num4=num3;      // yazdirirsak 12.0 olarak yazdirir  
double num5=num4;      // yazdirirsak 12.0 olarak yazdirir
```



Data Casting

2) Explicit Narrowing (Manuel Daraltma)

```
public class Main {  
    public static void main(String[] args) {  
        double myDouble = 9.78;  
        int myInt = (int) myDouble; // Manual casting: double to int  
  
        System.out.println(myDouble); // Outputs 9.78  
        System.out.println(myInt);    // Outputs 9  
    }  
}
```

- Genis veri tipinden daha dar bir veri tipine gecmek istedigimizde Java donusumu otomatik olarak YAPMAYACAKTIR.
- Bu durumda Java Casting'in bir problem olusturabilecegini varsayarak sizden MANUEL ONAY isteyecektir.
- Narrowing Casting bazi datalari kaybetmemize yol acabilir, bazen de sayiyi kendi sinirlari icinde kalan baska bir sayiya donusturebilir



Data Casting

- Soru 1)** byte veri tipinde bir degisken olusturun, short,int,float ve double data tiplerinde birer degisken olusturup adim adim widening yapin ve yazdirin
- Soru 2)** int veri turunde bir degisken olusturun ve adim adim narrowing yapin ve yazdirin
- Soru 3)** Float data turunde bir variable olusturun ve yazdirin
- Soru 4)** double 255.36 sayisini int'a ve sonra da olusturdugunuz int sayiyi byte'a cevirip yazdirin
- Soru 5)** int 2 sayiyi birbirine boldurun ve sonucu yazdirin
- Soru 6)** int bir sayiyi double bir sayiya bolun ve sonucu yazdirin
- Soru 7)** Farkli data tipleri ile islem yapip, sonuclarini yazdiralim