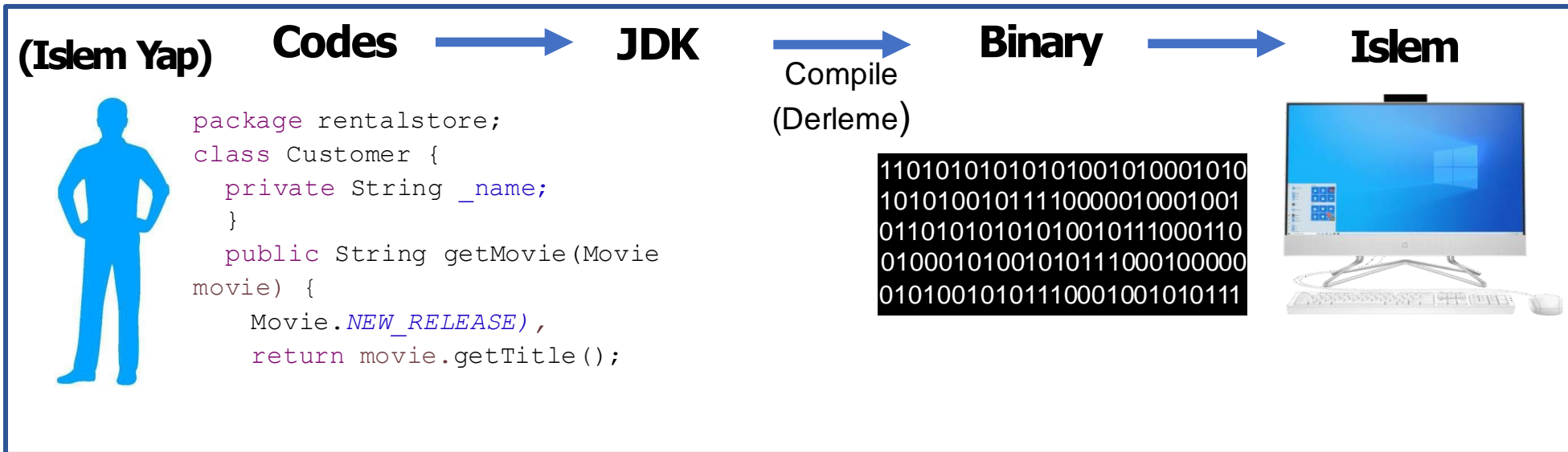
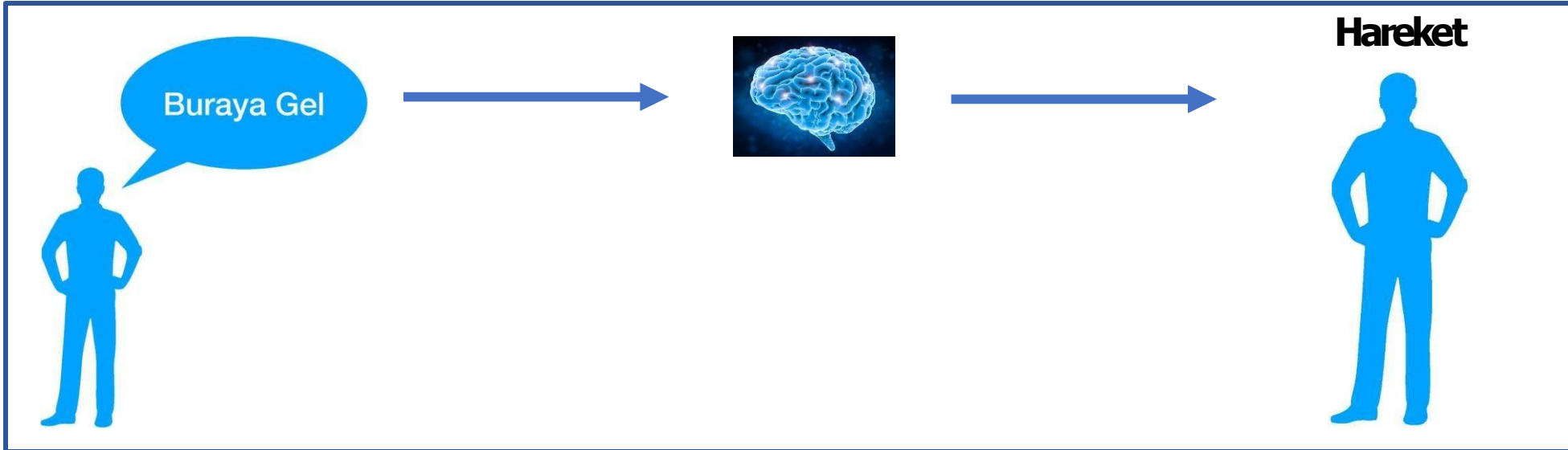




Programlama Dili Nedir ?





Nicin Java ?

1 Öğrenmesi kolay

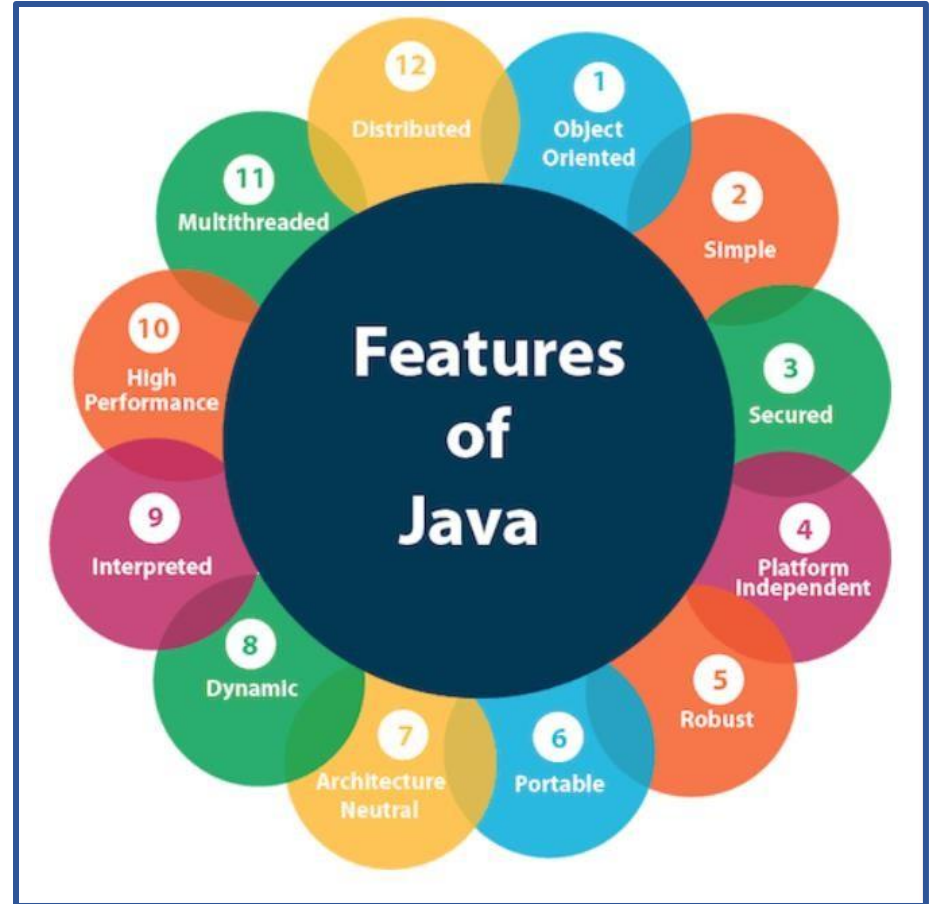
2 Dünyada en çok kullanılan programlama dili

Sun'a göre 3 milyar cihaz Java kullanıyor. Şu anda Java'nın kullanıldığı birçok cihaz var.

Bunlardan bazıları şu şekildedir:

- Acrobat reader, medya oynatıcı, antivirüs vb.
- Masaüstü Uygulamaları
- Bankacılık uygulamaları gibi Kurumsal Uygulamalar
- Cep Telefonu
- Akıllı kart uygulamaları
- Robotik uygulamaları
- Oyunlar

3 Java "Object Oriented Programming (OOP)" Language' dir.





Object Oriented Programming Nedir?



Objects (Nesne)

Application (Urun)

- 1 Feature (Fields veya Variables) Pasif ozellik (renk, sekil, isim)
- 2 Functionality (Method) Aktif ozellik (tasima, degistirme)



Bir Object Nasıl Olusturulur?



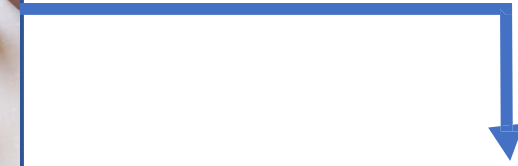
Class(Object Kalibi)

Field
(**Variables**)

Method
(**Functions**)



Object



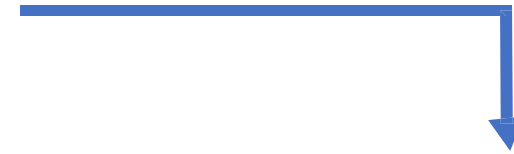
Birden fazla Obje birlestirilir



Application



Bir Object Nasıl Olusturulur?



Class(Object Kalibi)

Object

Application





Object Nasıl Olusturulur?



Class

Variables

(Fields)



Size
Color

Methods

(Functions)



Buy
Travel



Object *(Instance)*



Application



Object Nasıl Kullanilir ?



Ogretmen



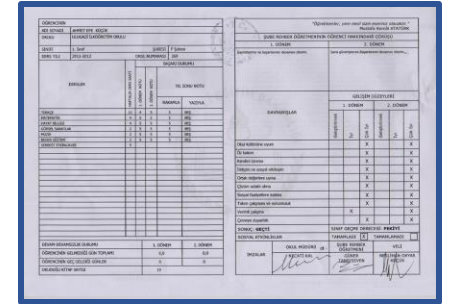
Ogrenci

Dersler

09:00	TÜRKÇE-1
09:30	MATEMATİK-1
10:00	TÜRKÇE-2
10:30	MATEMATİK-2
11:00	TÜRKÇE-3
11:30	MATEMATİK-3
12:00	TÜRKÇE-4
12:30	MATEMATİK-4
13:00	İYEP TÜRKÇE



Personel



Notlar



2

3

```
} // Class sonu
```

- 1– Class Declaration**
- 2 – Curly braces : Suslu parantez**
- 3 – Class Body : Suslu parantezler arasında kalan ve kodlarımızı yazdığımız bölüm**



Bir Class'ın İçinde Neler Bulunur?

```
public class C2_MethodCreation2 {
```

```
    private double ortalama;  
    public int sonuc;
```

1

```
    public static void main(String[] args) {  
        ortalama(85.2 ,90.3); // method call  
    }
```

2

```
    public static void ortalama(double sayi1, double sayi2) {  
        System.out.println("girdiginiz iki sayinin ortalamasi : " + (sayi1+sayi2)/2);  
    }
```

3

```
} // Class sonu
```

1– Field / Variables

2– Main Method

3– Method



Class Olustururken (Declaration) Kullanilan Keyword'ler

```
public class MyFirstClass {}  
1      2      3      4
```

- 1 public :** Access Modifier (Erisim duzenleyici) : class'a kimlerin erisebilecegini belirler. Public olursa her yerden erisilebilir
default : Sadece bulunduгу Package'den kullanilabilir
- 2 class :** Yazdigimiz kodun class oldugunu belirtir
- 3 MyFirstClass :** Olusturdugumuz class'in ismidir. Class'a istedigimiz ismi verebiliriz ancak isim verilirken genelde class'da yapilan isleme uygun bir isim secilmesine dikkat edilir.
Isim mutlaka buyuk harfle baslar, birden fazla kelimedenden olursa sonraki kelimelerin ilk harfleri de buyuk harf yazilir (Camel Case)
- 4 Body (Class Body) :** {} arasinda kalan kodlarimizi yazdigimiz bolumdur



Method Olusutururken Kullanilan Keyword'ler

```
public int myFirstMethod () {}  
1    2      3          4 5
```

- 1 public** : Access Modifier (Erisim duzenleyici):methoda'a kimlerin erisebilecegini belirler
private: Sadece bulunduğu class'da kullanılabilir
protected : Sadece içinde bulunduğu class ve child class'lardan kullanılır
- 2 Int** : Return Type, methodun ne urettigini ve bize dondurdugunu belirtir
- 3 myFirstMethod** :Olusturdugumuz method'un ismidir. Isim mutlaka kucuk harfle baslar, birden fazla kelimeden olusursa sonraki kelimelerin ilk harfleri buyuk harf yazilir (Camel Case)
- 4 () parantez**: Methodlarda isimden sonra parantez kullanilir ve gerektiğinde parantez içinde parametre yazilir.
- 5 Body (Method Body)** : {}arasinda kalan kodlarimizi yazdigimiz bolumdur



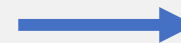
Main Method

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



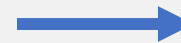
- main method, java'nin çalışmaya başladığı giristir. (Entry Point)
- main method oluşturulurken yazılması gereken syntax (kod dizimi) değiştirilemez
- Parantez içinde yazılan (String[] args) java'nin çalışması için gerekli olan parametreleri bildirir ve olması şarttır.

Araba



Motor

Java Project



Main Method



Yorum Cümlesi (Comment) Nasıl Eklenir ?

```
public class Example {  
    // Bir satiri comment haline getirmek icin // kullanilir  
    String isim ="Mehmet";  
  
    /*  
    Eger birden fazla  
    satiri yorum haline  
    getirmek istiyorsak  
    kullanilir  
  
    int sayi=10;  
    double not=75.70;  
    */  
    boolean ogrenciMi =false;  
}
```

➤ **Comments** : Java tarafından çalıştırılmayan, amacı kodların açıklanması veya bir konuda bilgi vermek olan cümlelerdir

➤ Genelde iki kullanım vardır

1) Tek satirlik comment

2) Çok satirlik comment



Data Nedir?

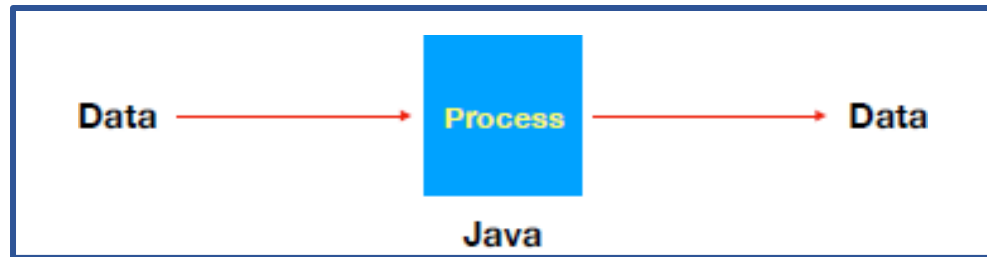
Data bilgisayar tarafından işlenen (**processed**) veya depolanan (**stored**) bilgidir.

Joe, Smith, 1234 Daire, SLC, UT, 8404,8015553211

0143 0157 0155 0160 0165 0164 0145 0162 0040 0150 0157 0160 0145

011000110110111101101101110000011101010111010001100101011100100010000001101000000
101

Java'nin kullandığı (**use**) veya ürettiği (**produce**) her şey data'dır.

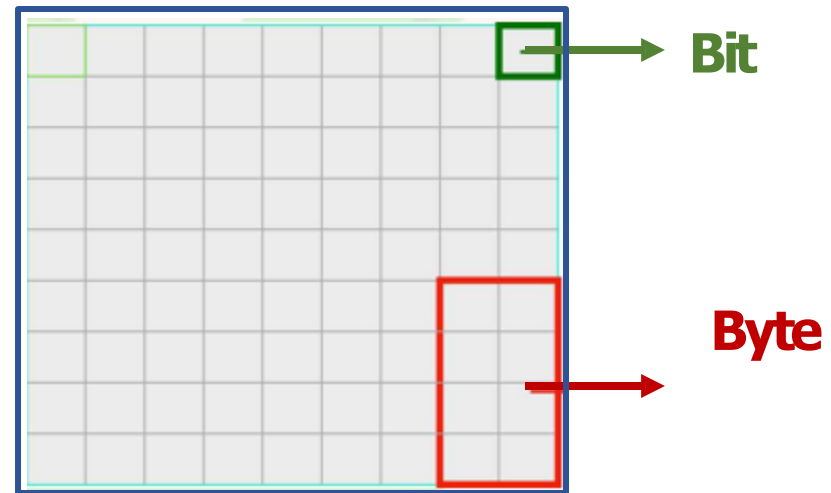
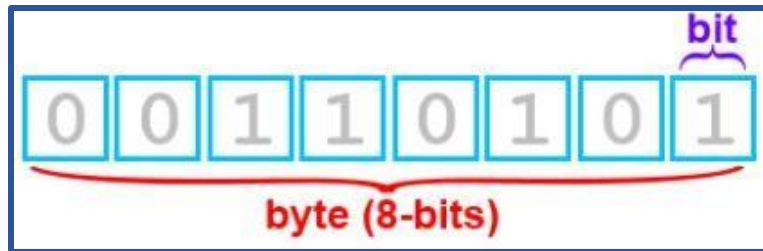




Bit

bit hafızadaki en küçük data parçasıdır. Her "bit" bir binary value içerir, 0 veya 1

Note: 8 bit =1byte



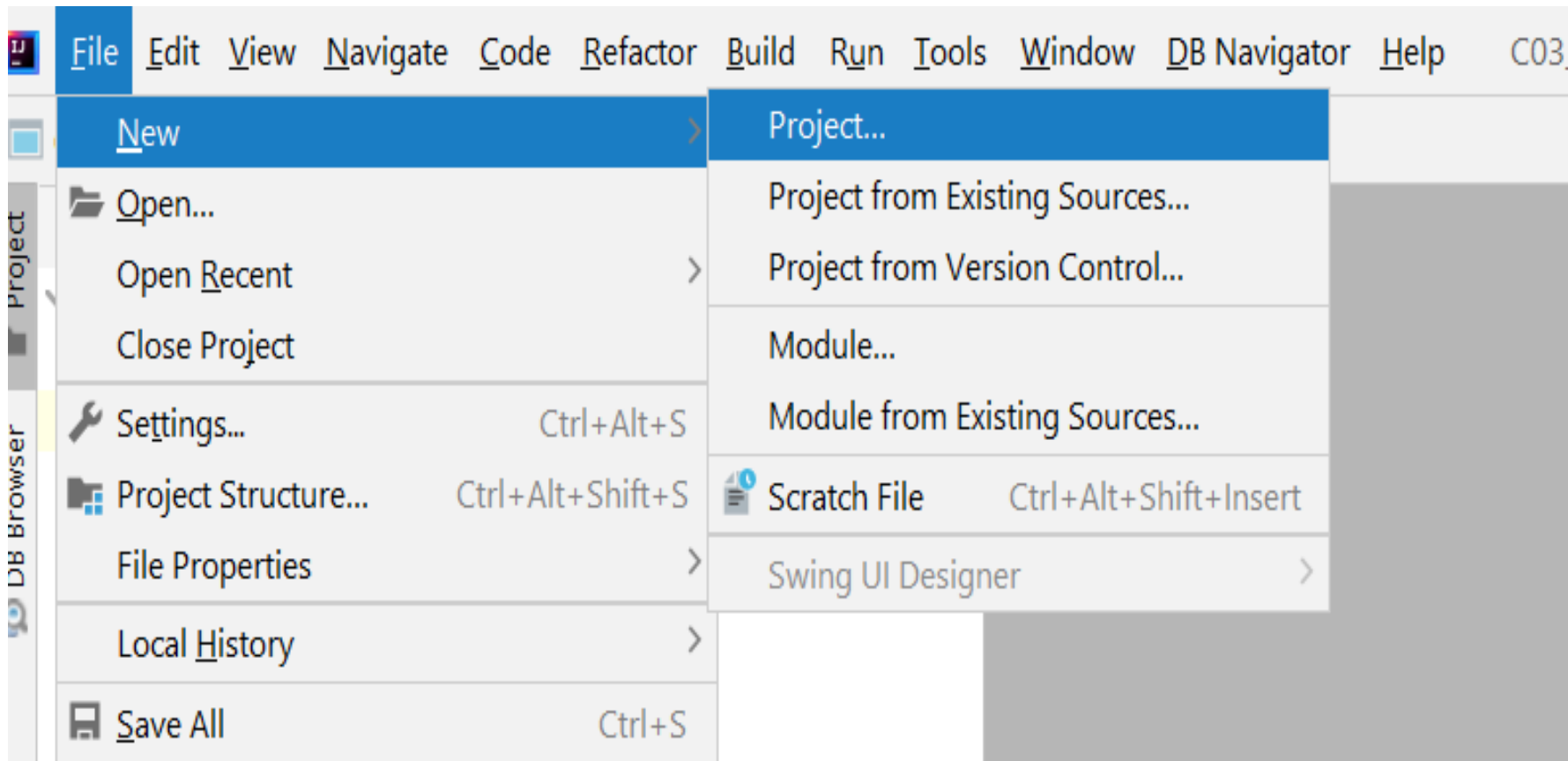
**Memory
(Hafıza)**



IntelliJ Kullanım

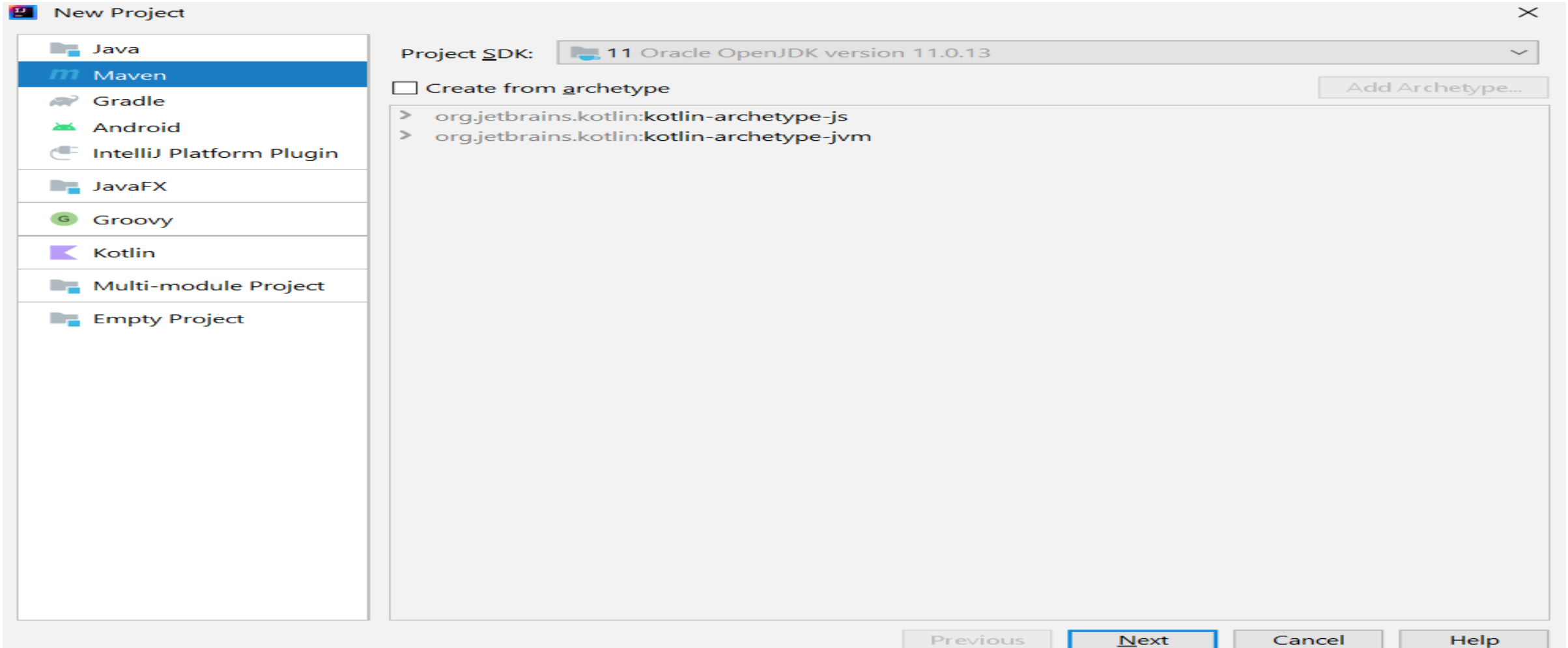
1 Proje oluşturma

File → New → Project → (Java Project) Next → java2022WinterTr → finish






Maven → Next






Name=java2022tr→Finish

 New Project ✕

Name:

Location: 

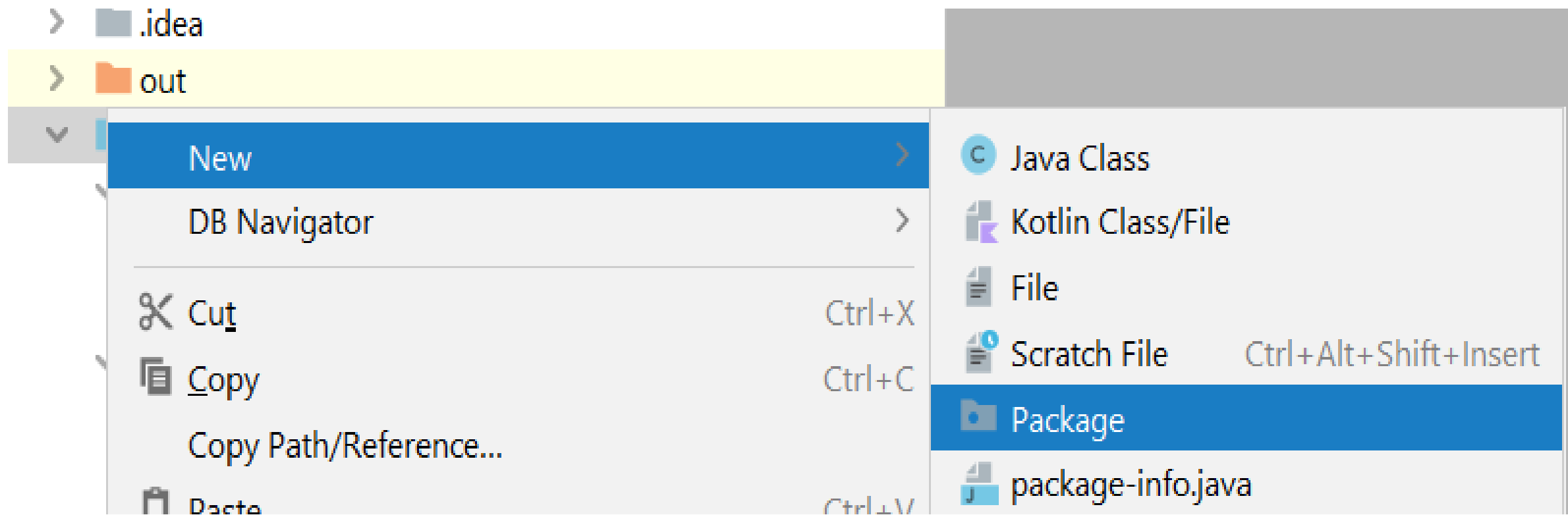
▶ Artifact Coordinates



Package Olusturma

2 Package (paket) olusturma

src dosyasina sag click -- New -- Package -- day01variables-- finish





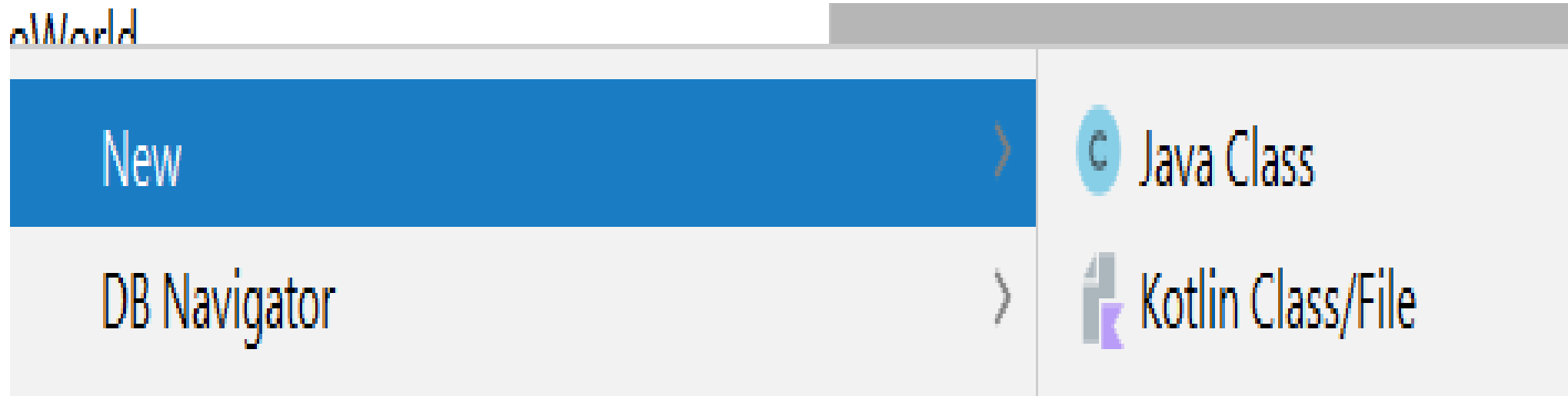
Class Olusturma

3 Class olusturma

Package ismine sag click → New → Class → C01_Variables01 → finish

4- Main method olusturma

public static void main(String[] args) yazarak main methodu olusturalim





Variables (Degisken) Olusturma

Dedaration

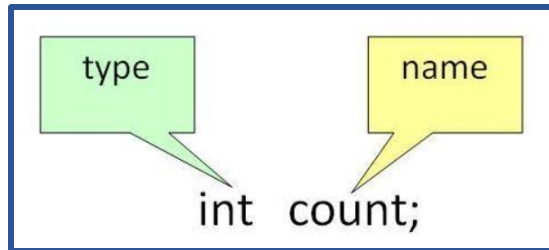
Variable bellekte (**memory**) ayrılmis olan alanin (**reserved area**) adidir.

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

Bir değışkende saklanan değır, program yürütölürken değıştirilebilir.



Java'da, tüm değışkenler kullanılmadan önce deklare edilmelidir (variable dedaration)



Variable dedaration için iki şeyi 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 dedaration, daha sonra atama yapılabilir.

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

3- Bir defa dedaration 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;
```

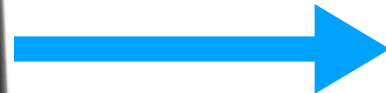



Data Types

Java'da iki data tipi kullanılmaktadır



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



- **Non- Primitive Data Types** : String, ArrayList, Object



Primitive Data Types

1) boolean Data Type: true veya false barindirir. Hafızada **1bit** kullanır

Sadece dogru veya yanlis seklinde cevap verilebilecek variable'larda kullanilir

```
boolean isExpensive =true;
```

```
boolean isCold =false;
```

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

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

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

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

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

Note: char degerlerini single quote arasina yazilir.



JAVA DATATYPES

Primitive Data Types

BOOLEAN

boolean

char

NUMERIC

Character

Tamsayi

byte

short

int

long

Integral

Non Primitive Data Types

Strings

Arrays

Objects

Ondalikli Sayi

float

double





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 **16bit** 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
```



ASCII= AMERIKAN STANDART KODLAMA SISTEMI

ASCII TABLE

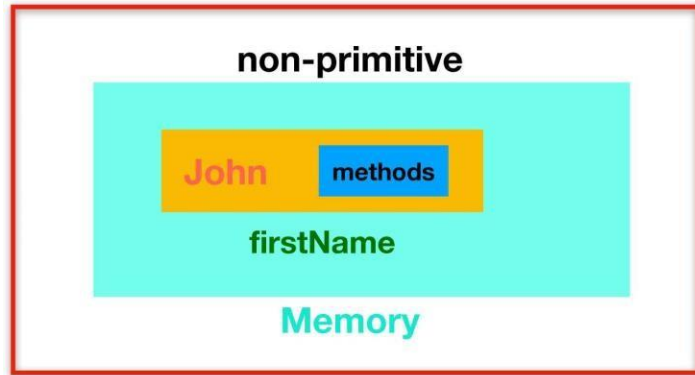
Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]	48	30	110000	60	0	96	60	11000000	140	`
1	1	1	1	[START OF HEADING]	49	31	110001	61	1	97	61	11000001	141	a
2	2	10	2	[START OF TEXT]	50	32	110010	62	2	98	62	11000010	142	b
3	3	11	3	[END OF TEXT]	51	33	110011	63	3	99	63	11000011	143	c
4	4	100	4	[END OF TRANSMISSION]	52	34	110100	64	4	100	64	11001000	144	d
5	5	101	5	[ENQUIRY]	53	35	110101	65	5	101	65	11001001	145	e
6	6	110	6	[ACKNOWLEDGE]	54	36	110110	66	6	102	66	11001010	146	f
7	7	111	7	[BELL]	55	37	110111	67	7	103	67	11001011	147	g
8	8	1000	10	[BACKSPACE]	56	38	111000	70	8	104	68	11010000	150	h
9	9	1001	11	[HORIZONTAL TAB]	57	39	111001	71	9	105	69	11010001	151	i
10	A	1010	12	[LINE FEED]	58	3A	111010	72	:	106	6A	11010100	152	j
11	B	1011	13	[VERTICAL TAB]	59	3B	111011	73	;	107	6B	11010101	153	k
12	C	1100	14	[FORM FEED]	60	3C	111100	74	<	108	6C	11011000	154	l
13	D	1101	15	[CARRIAGE RETURN]	61	3D	111101	75	=	109	6D	11011001	155	m
14	E	1110	16	[SHIFT OUT]	62	3E	111110	76	>	110	6E	11011010	156	n
15	F	1111	17	[SHIFT IN]	63	3F	111111	77	?	111	6F	11011011	157	o
16	10	10000	20	[DATA LINK ESCAPE]	64	40	1000000	100	@	112	70	11100000	160	p
17	11	10001	21	[DEVICE CONTROL 1]	65	41	1000001	101	A	113	71	11100001	161	q
18	12	10010	22	[DEVICE CONTROL 2]	66	42	1000010	102	B	114	72	11100010	162	r
19	13	10011	23	[DEVICE CONTROL 3]	67	43	1000011	103	C	115	73	11100011	163	s
20	14	10100	24	[DEVICE CONTROL 4]	68	44	1000100	104	D	116	74	11101000	164	t
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]	69	45	1000101	105	E	117	75	11101001	165	u
22	16	10110	26	[SYNCHRONOUS IDLE]	70	46	1000110	106	F	118	76	11101010	166	v
23	17	10111	27	[ENG OF TRANS. BLOCK]	71	47	1000111	107	G	119	77	11101011	167	w
24	18	11000	30	[CANCEL]	72	48	1001000	110	H	120	78	11110000	170	x
25	19	11001	31	[END OF MEDIUM]	73	49	1001001	111	I	121	79	11110001	171	y
26	1A	11010	32	[SUBSTITUTE]	74	4A	1001010	112	J	122	7A	11110010	172	z
27	1B	11011	33	[ESCAPE]	75	4B	1001011	113	K	123	7B	11110011	173	{
28	1C	11100	34	[FILE SEPARATOR]	76	4C	1001100	114	L	124	7C	11111000	174	
29	1D	11101	35	[GROUP SEPARATOR]	77	4D	1001101	115	M	125	7D	11111001	175	}
30	1E	11110	36	[RECORD SEPARATOR]	78	4E	1001110	116	N	126	7E	11111010	176	~
31	1F	11111	37	[UNIT SEPARATOR]	79	4F	1001111	117	O	127	7F	11111011	177	[DEL]
32	20	100000	40	[SPACE]	80	50	1010000	120	P					
33	21	100001	41	!	81	51	1010001	121	Q					
34	22	100010	42	"	82	52	1010010	122	R					
35	23	100011	43	#	83	53	1010011	123	S					
36	24	100100	44	\$	84	54	1010100	124	T					
37	25	100101	45	%	85	55	1010101	125	U					
38	26	100110	46	&	86	56	1010110	126	V					
39	27	100111	47	'	87	57	1010111	127	W					
40	28	101000	50	(88	58	1011000	130	X					
41	29	101001	51)	89	59	1011001	131	Y					
42	2A	101010	52	*	90	5A	1011010	132	Z					
43	2B	101011	53	+	91	5B	1011011	133	[
44	2C	101100	54	,	92	5C	1011100	134	\					
45	2D	101101	55	-	93	5D	1011101	135]					
46	2E	101110	56	.	94	5E	1011110	136	^					
47	2F	101111	57	/	95	5F	1011111	137	_					



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

Note: String'ler çift tırnak (double quotes) arasına yazılır, Büyük harfle başlar.

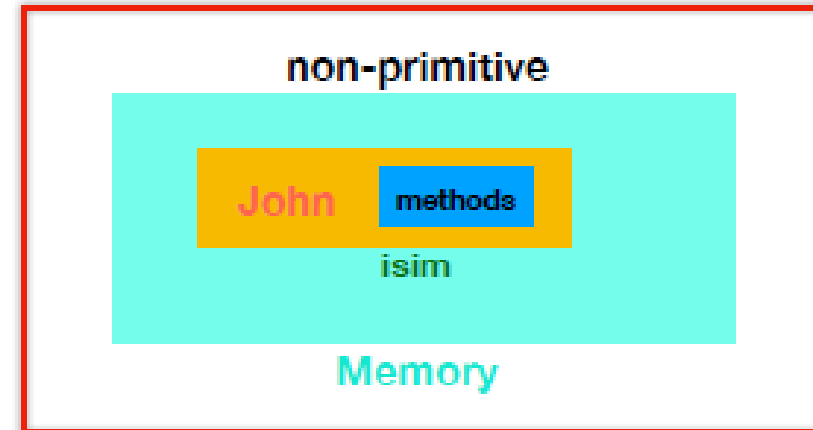
Note: Java tarafından üretilbildiği gibi bizler tarafından da üretilbilen değer yanında methodlarda barındıran Data Types

Note: Non-primitives aynı büyüklüktedir.





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 oluşturur biz primitive data turu olusturamayiz.
Non-primitiveleri 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.



Önceki Dersten Aklımızda Kalanlar

1. **OOP concept : nesne tabanlı programlama demektir, biz oluşturduğumuz class'lar sayesinde objeler uretebiliriz ve bu objeler birleştirerek kompleks uygulamalar geliştirebiliriz(lego gibi)**
2. **Clas hangi bolumlerden oluşur ?**
 - **class dedaration : keyword ler sayesinde class'ı kimlerin kullanabileceğini görebiliriz, class ismi büyük harfle başlar ve CamelCase şeklinde yazılır**
 - **{ } curly braces / suslu parantez : Classın nerede başlayıp bittiğini gösterir**
 - **Class Body : curly braces arasında kalan ve kodlarımızı yazdığımız bölümdür**
- 3 **Class içerisinde neler olur ?**
 - **Main method :arabanın motoru gibidir, Java kodlarımız çalıştırmaya main method'dan başlar**
 - **normal method'lar : method'lar bizim adımıza belirlediğimiz işlemleri yapıp, işlem sonunda da istediğimiz sonucu bize döndürürler**
 - **variable : bizim için değerleri saklayan konteynir'lerdir**



Önceki Dersten Aklımızda Kalanlar

- 4 **variable** : bizim istedigimiz degeri koyabilmemiz icin hafiza da ayrılan bolumun adidir. Örneğin bir oyunda level bilgisi için bir variable tanımlarsak, oyunun hangi asamasında olursa olsun level variable'ına baktığımızda içinde level değerimizi görebiliriz. Biz programımız içerisinde ne zaman variable ismini yazsak, java o variable yerine en son atanan degeri kullanır.
- 5 **Data türleri**
 - primitive data türleri : boolean, char, byte, short, int, long, float, double
 - non-primitive data türleri : String (ilerde pek çok cesidini göreceğiz, biz de istersek non-primitive data türü oluşturabiliriz)
- 6 **iki data türü arasındaki farklar**
 - p'ler sadece depolama yapar, np'ler ise hem depolama yapar hem de kendilerine özel methodlar sayesinde istedigimiz degisimleri yaparlar
 - p data türlerinin isimleri küçük harfle baslar, np data türlerinin isimleri büyük harfle baslar
 - p'ler 8 tanedir ve biz yeni p data türü üretemeyiz, ancak np'leri java da üretebilir biz de üretebiliriz, dolayısıyla np data türü sayısı sınırlanmaz



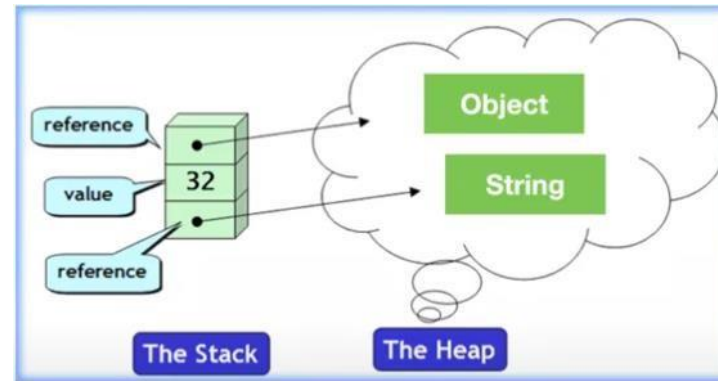
Variable ve Method'lar Nasıl Adlandırılır

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 hafi büyük harf ile başlamalıdır. firstName, bigApple, ageJohnWalker gibi. Buna camelCase denir.



Memory (Hafıza) Kullanımı

Javada kullanılan iki hafıza vardır



Stack ⇒ **s**mall

Heap ⇒ **h**uge

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

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



Memory (Hafiza) Kullanimi

reference of an Object

long

int

byte

boolean

Object

Array

String



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 : Aydin

soyisminiz : Yilmaz

sekinde yazdirin

3İki 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			ASCII printable characters			Extended ASCII characters		
00	NULL	(Null character)	32	space	64	@	96	`
01	SOH	(Start of Header)	33	!	65	A	97	a
02	STX	(Start of Text)	34	"	66	B	98	b
03	ETX	(End of Text)	35	#	67	C	99	c
04	EOT	(End of Trans.)	36	\$	68	D	100	d
05	ENQ	(Enquiry)	37	%	69	E	101	e
06	ACK	(Acknowledgement)	38	&	70	F	102	f
07	BEL	(Bell)	39	'	71	G	103	g
08	BS	(Backspace)	40	(72	H	104	h
09	HT	(Horizontal Tab)	41)	73	I	105	i
10	LF	(Line feed)	42	*	74	J	106	j
11	VT	(Vertical Tab)	43	+	75	K	107	k
12	FF	(Form feed)	44	,	76	L	108	l
13	CR	(Carriage return)	45	-	77	M	109	m
14	SO	(Shift Out)	46	.	78	N	110	n
15	SI	(Shift In)	47	/	79	O	111	o
16	DLE	(Data link escape)	48	0	80	P	112	p
17	DC1	(Device control 1)	49	1	81	Q	113	q
18	DC2	(Device control 2)	50	2	82	R	114	r
19	DC3	(Device control 3)	51	3	83	S	115	s
20	DC4	(Device control 4)	52	4	84	T	116	t
21	NAK	(Negative acknowl.)	53	5	85	U	117	u
22	SYN	(Synchronous idle)	54	6	86	V	118	v
23	ETB	(End of trans. block)	55	7	87	W	119	w
24	CAN	(Cancel)	56	8	88	X	120	x
25	EM	(End of medium)	57	9	89	Y	121	y
26	SUB	(Substitute)	58	:	90	Z	122	z
27	ESC	(Escape)	59	;	91	[123	{
28	FS	(File separator)	60	<	92	\	124	
29	GS	(Group separator)	61	=	93]	125	}
30	RS	(Record separator)	62	>	94	^	126	~
31	US	(Unit separator)	63	?	95	_		
127	DEL	(Delete)						
128	Ç		160	á	192	Ł	224	Ó
129	ù		161	í	193	ł	225	ô
130	é		162	ó	194	Ł	226	ö
131	â		163	ú	195	ł	227	÷
132	ä		164	ñ	196	—	228	ø
133	à		165	Ñ	197	†	229	Õ
134	á		166	°	198	ä	230	µ
135	ç		167	°	199	Å	231	þ
136	ê		168	¿	200	Ł	232	þ
137	ë		169	®	201	Œ	233	Ú
138	è		170	™	202	Œ	234	Û
139	ï		171	½	203	Œ	235	Ü
140	î		172	¼	204	Œ	236	Ý
141	ì		173	ı	205	=	237	Ÿ
142	Ä		174	«	206	†	238	—
143	Å		175	»	207	‡	239	·
144	É		176	•	208	ø	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	•	213	İ	245	§
150	û		182	•	214	ı	246	÷
151	ù		183	•	215	İ	247	•
152	ÿ		184	•	216	ı	248	•
153	Ö		185	•	217	ı	249	•
154	Ü		186	•	218	ı	250	•
155	ø		187	•	219	ı	251	•
156	£		188	•	220	ı	252	•
157	Ø		189	•	221	ı	253	•
158	×		190	•	222	ı	254	•
159	f		191	•	223	ı	255	nbsp



Variables Class Work

Interview Question

1- Verilen sayı1 ve sayı2 variable'larının degerlerini degistiren (SWAP) bir program yazınız

**Örneğin : sayı1=10 ve sayı2=20;
kod çalıştıktan sonra
sayı1=20 ve sayı2=10**

2- Verilen sayı1 ve sayı2 variable'larının degerlerini 3. bir variable olmadan degistiren (SWAP) bir program yapınız



Kullanıcıdan Deger Alma

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

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

Bu tur isimlendirmelerde genel kurallara uymamiz kodumuzun anlasilabilir olmasi acisindan 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 Değer Alma

3) **scan.nextInt()** ile girilen değerleri alabiliriz. İstedigimiz data tipine göre next'ten sonra yazılacak kısım değişir.

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

nextBoolean()	→	Reads a boolean value from the user
nextByte()	→	Reads a byte value from the user
nextDouble()	→	Reads a double value from the user
nextFloat()	→	Reads a float value from the user
nextInt()	→	Reads a int value from the user
nextLine()	→	Reads a String value from the user
nextLong()	→	Reads a long value from the user
nextShort()	→	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 uzunluğunu alın ve karenin çevresini ve alanını hesaplayıp yazdırın
- Soru 3)** Kullanıcıdan yarıçap isteyip çemberin çevresini ve dairenin alanını 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 : Ali
- Soyisminiz : Can
- Kursumuza katılımınız alınmıştır, teşekkür ederiz
- Soru 6)** Kullanıcıdan ismini ve soyismini alıp aralarında bir boşluk oluşturarak aşağıdaki şekilde yazdırın
- İsim – soyisim : Ali Can
- Soru 7)** Kullanıcıdan ismini alıp isminin baş harfini yazdırın.



Önceki Dersten Aklımızda Kalanlar

1. **Scanner class'ı kullanıcıdan bilgi almak için kullanılır.**
2. **Uc adımda kullanıcıdan değeri alırız**
 - **Scanner oluşturma, parameter olarak System.in yazmalıyız**
 - **Kullanıcıya ne gireceğini söyleyen bir mesaj yazdırma**
 - **Olusturdugumuz scan objesi ve girilecek dataya uygun next method'u ile kullanıcının girdigi değeri alıp, gelen dataya uygun data turunde olusturdugumuz variable'a atama yaparız**
- 3 **Kullanıcıdan aldığımız değer metin ise next() veya nextLine() method'u kullanılır. Next method'u sadece ilk boşluğa kadar olan metni alırken, nextLine tüm satiri alır**
- 4 **Java da kullanılan 2 tür hafıza vardır.**
 - stack- primitive dataturundeki variable'ların aldığı değerler ve non-primitive'lerin referansları bulunur**
 - Heap : Non primitive data turundeki datalar**



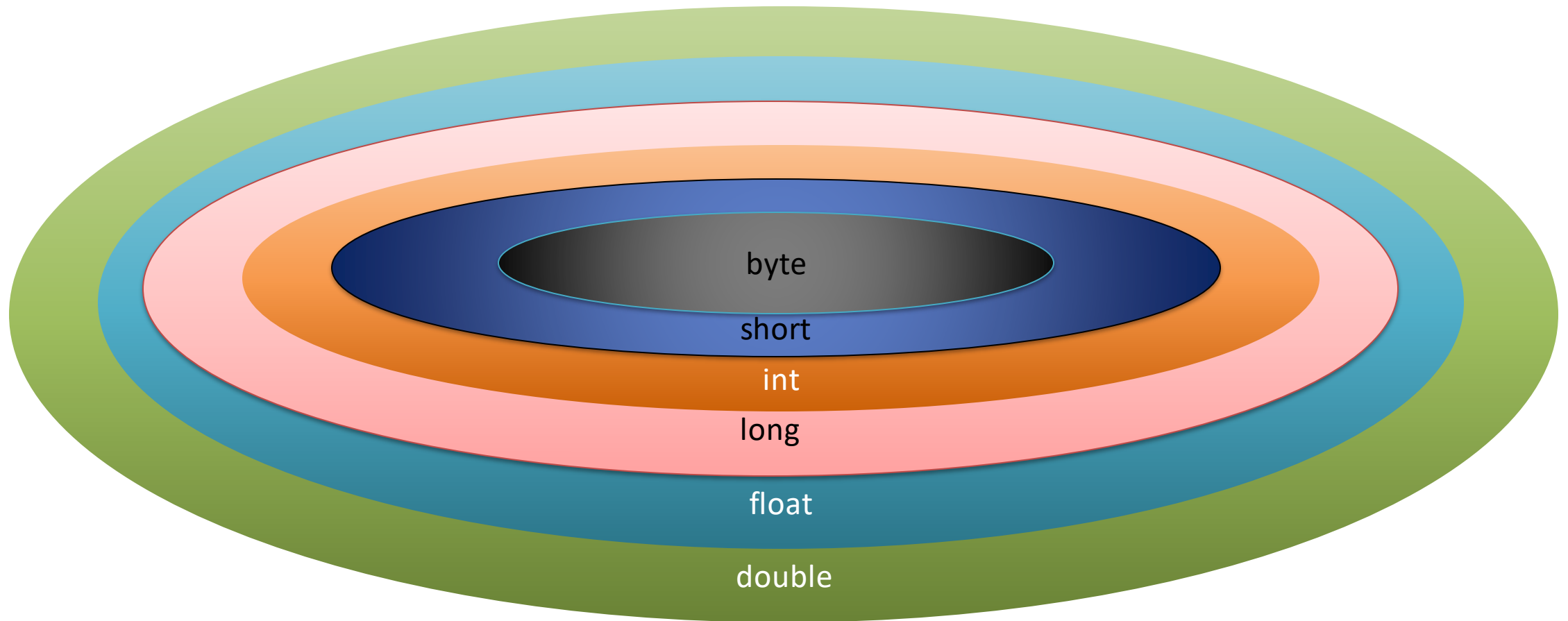
Data Casting / Veri Sinifi Degistirme

- **Java'da kod yazarken bir veri tipinden diğer bir veri tipine aktarım yapmamız gerekebilir.**
- **Veri tiplerinde bir variable'a , olusturuldugu data tipinden farkli bir data turunden deger atanmasina Data Casting denir.**
- **Data casting yaparken aklimizdan cikarmamamiz gereken konu data tiplerinin sinirlaridir. Data tipinin sinirlarini asan data casting islemlerinde hata almamamiz icin dikkat etmemiz gereken bazi durumlar olacaktır.**
- **Hatırlayacagimiz sekilde Java'da sayilarla ilgili data tiplerinin siralamasi su sekildeydi**

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



DATA TYPE CASTING

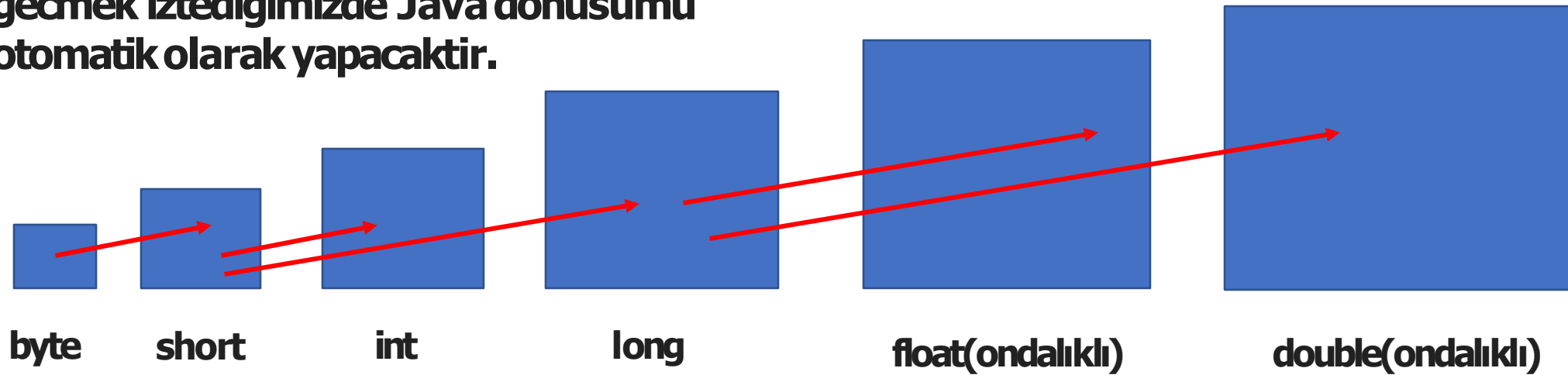




Data Casting / Veri Sinifi Degistirme

1) Auto Widening (Otomatik Genisletme)

Dar veri tipinden daha geniş bir veri tipine geçmek istediğimizde Java bunu otomatik olarak yapacaktır.



Örn :

```
byte num1 = 12;  
short num2 = num1; // yazdırırsak 12 olarak yazdırır  
int num3 = num2; // yazdırırsak 12 olarak yazdırır  
float num4 = num3; // yazdırırsak 120 olarak yazdırır  
double num5 = num4; // yazdırırsak 1200 olarak yazdırır
```



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



Increment / Bir Variable'in Degerini Artirma Yontemleri

```
int numA = 2 ;  
numA = numA + 3 ;
```

veya

```
numA += 3
```

?

```
int numB = 10 ;  
numB = numB * 7 ;
```

veya

```
numB *= 7
```

?

```
int numC = 7 ;  
numC++;
```

?

```
int numD = 11 ;  
numD++ ;
```

?



Decrement / Bir Variable'in Degerini Azaltma Yontemleri

```
int numA = 2 ;  
numA = numA - 3 ;
```

veya

```
numA -= 3
```

?

```
int numB = 20 ;  
numB = numB / 5 ;
```

veya

```
numB /= 5
```

?

```
int numD = 7 ;  
numD -- ;
```

?

```
int numE = 11 ;  
numE -- ;
```

?



Önceki Dersten Aklımızda Kalanlar

1. **Data Casting (Veri turunu degistirme):** Java her data turunu birbirine cevirmmez, ornegin Boolean bir degiskene String bir deger atayamayiz.

Ancak sayisal veri turlerini birbirlerine cevirebiliriz

- Eger variable turu(esitligin solu) daha kapsamli ise, Java bu casting islemini otomatik olarak yapar (Auto Widening)
- Eger deger'in turu (esitligin sagi) daha kapsamli ise Java bunu otomatik olarak yapmaz. Cunku daha kapsamli bir data turunden daha dar kapsamli bir data turune gecis sirasinda data kayiplari veya farkli deger alma ihtimali olusur.

Bu durumda Java sorumlulugu manuel olarak almamizi ister. Sorumlulugu alabilmek Icin esitligin sagina parantez icerisinde variable'in data turunu yazariz

```
double sayi1=10.28;
```

```
int sayi2 =(int)sayi1;
```

- 2 **Increment ve Decrement :** artirma veya azaltma demektir

- `sayi1=sayi1+3;` bu cok tercih edilmez
- `sayi1+=3;` genelde bu tercih edilir
- `sayi1++;` sadece 1artirip azaltacaksak bunu kullaniriz



Pre-Increment & Post Increment

- **Pre-Increment ve Post Increment** operatorlerinin her ikisi de artırma işlemi için kullanılır
- **Pre-Increment** işleminde variable statement'da kullanılmadan önce artırılır veya azaltılır

```
public static void main(String[] args) {  
    int a=15;  
    int b=++a;  
    System.out.println(b);  
}
```

Output: 16

- **Post Increment** işleminde variable statement'da kullanılır, sonra artırılır veya azaltılır

```
public static void main(String[] args) {  
    int a=15;  
    int b=a++;  
    System.out.println(b);  
}
```

Output: 15



Javada Matematiksel Operatorler

- 1 Ustel islemler
- 2 Parantez ici
- 3 Carpma-Bolme
- 4 Toplama-cikarma

Ornek 1:

$$38 / 2 * (4 + 3) * 2 =$$

Ornek 2 :

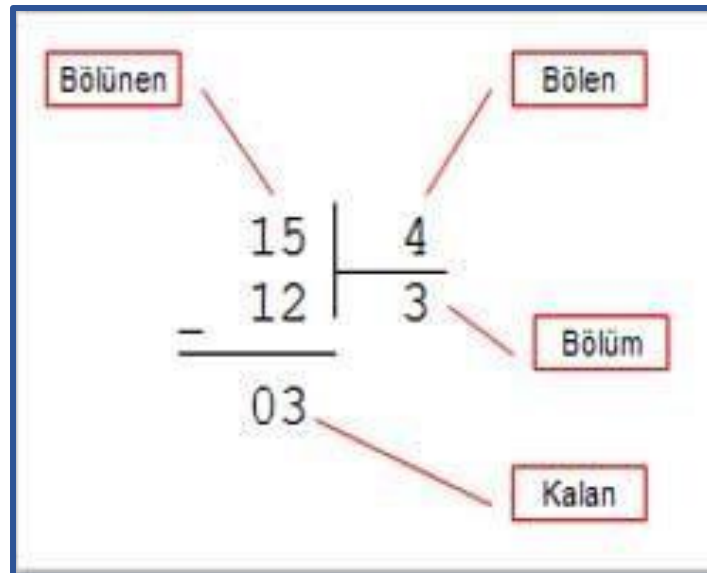
$$8 + 2 * (14 - 6 / 2) - 12 =$$



Modulus %

Modulus islemi bir bolme isleminde kalan sayiyi bize verir

```
public static void main(String[] args) {  
    int a=15 % 4;  
    System.out.println(a);  
}
```





Modulus %

Soru) Kullanıcıdan 4 basamaklı bir sayı alın ve rakamlar toplamını bulup yazdırın

İpucu 1:

Sayı % 10 => Bize son basamağı verir

$$538 \% 10 = 8$$

İpucu 2:

Int Sayı /10 => Bize son basamak hariç sayıyı verir

int sayı=538;

sayı = sayı / 10 =>

sayı'ya 53 değerini atar



Wrapper Class

Java **primitive** data turleri ile **methodlari** kullanabilmemiz icin **Wrapper class**'lari olusturmudur.

Character,Byte,Integer,Short,Float,Double primitive data turleri icin olusturulan wrapper class'lardir.

```
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



Önceki Dersten Aklımızda Kalanlar

- 1. Wrapper Class :** primitive data türlerine ait method yoktur. Java primitive data türündeki variable'ların Wrapper Class'larına yaparak o türler için de hazır bazı method'lar oluşturmıştır.
- 2. Pre Increment- Post Increment :** bu ikisi de ++, veya – için geçerlidir. Java yukarıdan aşağıya, soldan sağa doğru gider. Eğer bir satırda birden fazla işlem varsa, önce hangisini yapacağını bilmesi gerekir
`int sayi2 =sayi1++;` önce sayi1'in eski değerini sayi2'ye atar, sonra sayi1'i 1 artırır
`sout(++sayi1);` önce sayi1'in değerini 1 artırıp, sonra yeni değerini yazdırır
- 3 Modulus %a bir bölme işleminde kalan'ı verir (matematikteki mod işlemi gibidir)**
Bizim en çok kullandığımız yerler
 - bir sayının tek mi çift mi olduğunu bulmak
 - bir sayının verilen bir sayıya tam bölünüp bölünemediğine bakmaz
 - %10 yaparak sayının birler basamagini almak
- 4 bir sayının rakamları toplamını bulmak için için, basamak sayısı miktarınca şu işlemler tekrar edilir**
 - %10 ile birler basamagini bulmak
 - bulunan basamaktaki rakamı rakamlar toplamına eklemek
 - toplama eklediğimiz rakamdan kurtulmak için sayi/10 yapmak



Socrative Quiz

- 1) <https://b.socrative.com/login/student/> adresine gidin
- 2) Room Name **JAVA123** yazın
- 3) Isminizi yazın
- 4) **Done** butonuna basın



Concatenation / (String Datalari Birlestirme)

Birden çok String'i +isareti ile topladığınızda Java bu String değişkenleri birleştirerek yeni bir String oluşturur

```
String a = "Hello";  
String b = "World";  
System.out.println(a+b);  
System.out.println(a+" "+b);
```

→ HelloWorld
→ Hello World

Not : Eger matematiksel bir islemin icinde String kullanilirsa, matematikteki oncelikler dikkate alinarak islem yapilir. Sira String ile toplamaya geldiginde toplama yerine Concatenation uygulanir

```
String a = "Hello";  
int b = 2;  
int c = 3;  
  
System.out.println(a+b+c);  
System.out.println(c+b+a);  
System.out.println(a+(b+c));  
System.out.println(a+b*c);
```

→ Hello23
→ 5Hello
→ Hello5
→ Hello6



Concatenation

Soru 1) Asagida verilen variable'leri kullanarak istenen sonuclari yazdiran programlari yaziniz.

Variables

```
String str1= "Java";  
String str2= "Guzel";  
int sayi1=5;  
int sayi2=4;
```

Istenen Yazilar

- 1) Java Guzel 54
- 2) Java 5 Guzel
- 3) Java 94
- 4) Java 19
- 5) 54 Guzel



Relational Operators /(Karsilastirma Operatorleri)

= Assignment (Atama yapar) operatoru

int num1=3;

num1 degiskenine 3 degerini atar

String str1="Ali" +"" + "Can";

str1'e Ali Can degeri atar

c =c+5;

c'nin degerini 5 artirir ve son degeri c'ye atar

==Cift esittir isareti / karsilastirma (Comperison) operatoru

boolean sonuc1 =5+2 ==7;

sonuc1 degeri **true** olur

boolean sonuc2 =5*2 ==15;

sonuc2 degeri **false** olur



Relational Operators /(Karsilastirma Operatorleri)

\neq Esit degildir isareti

boolean sonuc1= 5+2 \neq 7;

sonuc1 degeri **false** olur

System.out.println(5*2 \neq 15);

true yazdirir

>Buyuktur , **\geq** Buyuk veya esittir

boolean sonuc1=5+2 \geq 7;

sonuc1degeri **true** olur

System.out.println(5*2>15);

false yazdirir

<Kucuktur , **\leq** Kucuk veya esittir

boolean sonuc1= 5+2 <7;

sonuc1 degeri **false** olur

System.out.println(5*2<15);

true yazdirir



Conditional Operators / (Sart Operatorleri)

&& AND (ve) isareti

&& isareti ile birlestirilen tum ifadeler dogru ise sonuc **true** olur.
Diger tum durumlarda **false** doner. (**&&** operatoru mukemmeliyettir)

```
boolean sonuc1=(5+2 ==7) && (4+3 !=5);  
System.out.println((5*2 !=15)&& (5>7));
```

sonuc1degeri **true** olur
false yazdirir

|| OR (veya) isareti

|| isareti ile birlestirilen tum ifadeler yanlis ise sonuc **false** olur.
Diger tum durumlarda **true** doner. (**||** operatoru iyimserdir)

```
boolean sonuc1=(5+2 ==7) || (4+3 !=5);  
System.out.println((5*2 ==15) || (5>7));
```

sonuc1degeri **true** olur
false yazdirir



& 1e && Arasindaki Fark

& isareti kullanildiginda Java isaretin iki yanindaki mantiksal ifadelerin ikisini de kontrol eder. Bu islem kodumuzu yavaslatir

40<30 & 50==50 & 60>50

ilk karsilastirma yanlis olmasina ragmen Java tum karsilastirmalari kontrol etmeye devam eder.

&& isareti kullanildiginda ise Java en bastan kontrol etmeye baslar, mantiksal ifadelerin birinde yanlisi bulursa sonrakileri kontrol etme ihtiyaci duymaz. Bu islem kodumuzu hizlandirir

40<30 && 50==50 && 60>50

ilk karsilastirma yanlis oldugunu gorunce Java diger karsilastirmalari kontrol etmeden alt satira gecir.



If Statements / (If cumleleri)

Eger hava guzel olursa piknige gidecegiz. (**guzel olmazsa karar yok**)

Eger (hava guzel olursa) {**piknige gideriz**} **her durumda alt satira gecer**

If (boolean sart) {**sart saglanirsa istenen kod**} **her durumda alt satira gecer**

```
public static void main(String[] args) {  
  
    int a = 2;  
    int b = 3;  
  
    if (a>b) {  
        System.out.println(a+b);  
    }  
    if (a==b) {  
        System.out.println(a*b);  
    }  
}
```



If Statements / (If cumleleri)

Not : If statement birden fazla olursa hepsi birbirinden bagimsiz olur. If cumlelerini birbirine baglamayi da ogrenecegiz.

Eger hava guzel olursa piknige gidecegiz. (guzel olmazsa karar yok)

Eger Ali ararsa ona kizacagim. (aramazsa karar yok)

Eger aksam mac varsa onu izleriz. (mac yoksa karar yok)

```
8      int a=10;
9      int b=8;
10
11     if (a==b) {
12         System.out.println("iki sayi esit");
13     }
14
15     if (a+b<100) {
16         System.out.println("sayilarin toplami yuzden kucuk");
17     }
18
19     if (a*b>1000) {
20         System.out.println("sayilarin carpimi bin'den buyuk");
21     }
```




If Statements / (If cumleleri)

Soru 1)Kullanıcıdan bir tamsayı isteyin ve sayının tek veya çift olduğunu yazdırın

Soru 2) Kullanıcıdan gün isimlerinden birinin ilk harfini isteyin ve o harfle başlayan gün isimlerini yazdırın

Ornek: ilkHarf=P output ="Pazar, Pazartesi veya Persembe"

ilkHarf=S output ="Salı"

*****Buyuk kucuk harf problem olmamasi icin toUpperCase methodunu kullanin**

Soru 3) Kullanıcıdan gün ismini alın ve hafta içi veya hafta sonu olduğunu yazdırın

Ornek: gün=Pazar output ="Hafta sonu"

gün=Salı output ="Hafta içi"

*****String icin equals method'unu kullanin**

Soru 4) Kullanıcıdan dikdörtgenin kenar uzunluklarını isteyin ve dikdörtgenin kare olup olmadığını yazdırın

Soru 5) Kullanıcıdan bir gün alın eğer gün "Cuma" ise ekrana "Muslumanlar icin kutsal gün" yazdırın. "Cumartesi" ise ekrana "Yahudiler icin kutsal gün" yazdırın. "Pazar" ise ekrana "Hristiyanlar icin kutsal gün" yazdırın



If Else Statements

Eger hava guzel olursa piknige gideriz, yoksa evde otururuz.

Eger (hava guzel olursa) {piknige gideriz} **yoksa** {evde otururuz}

If (boolean sart) {sart saglanirsa istenen kod} **else** {sart saglanmazsa istenen kod}

```
public static void main(String[] args) {  
  
    int a = 2;  
    int b = 3;  
  
    if (a>=b) {  
        System.out.println(a+b);  
    } else {  
        System.out.println(a*b);  
    }  
}
```



If Else Statements

Sorular

- Soru 1) Kullanıcıdan dikdörtgenin kenar uzunluklarını isteyin ve dikdörtgenin kare olup olmadığını yazdırın**
- Soru 2) Kullanıcıdan bir karakter girmesini isteyin ve girilen karakterin harf olup olmadığını yazdırın**
- Soru 3) Kullanıcıya yaşını sorun, eğer yaş 65'den küçük ise "emekli olamazsın, çalışmalısın", 65'e eşit veya büyükse "Emekli olabilirsiniz" yazdırın**
- Soru 4) Kullanıcıdan bir üçgenin üç kenar uzunluğunu alın eğer üç kenar uzunluğu birbirine eşit ise ekrana "Eşkenar üçgen" yazdırın. Diğer durumlarda ekrana "Eşkenar değil" yazdırın.**



If Else If ... Statements

Eger soruyu biliyorsa Ali soruyu cozsun , o bilmiyorsa Veli biliyorsa Veli cozsun, o da bilmiyorsa Ayse biliyorsa, Ayse cozsun, o da bilmiyorsa Fatma biliyorsa, Fatma cozsun, o da bilmiyorsa kim isterse o cozsun.

Eger soruyu biliyorsa Ali soruyu cozsun , **o bilmiyorsa Veli biliyorsa** Veli cozsun, **o da bilmiyorsa Ayse biliyorsa**, Ayse cozsun, **o da bilmiyorsa Fatma biliyorsa**, Fatma cozsun, **o da bilmiyorsa** kim isterse o cozsun.

If (sart) {**sart saglanirsa istenen kod**} **else if** {sart saglanmazsa istenen kod}
else if {sart saglanmazsa istenen kod} **else if** (kac tane durum varsa else if ...)
else {sart saglanmazsa istenen kod}



If Else If ...Statements

- Soru 5) Kullanıcıdan gun ismini yazmasını isteyin. Girilen isim geçerli bir gün ise gün isminin 1,2 ve 3.harflerini ilk harf büyük diğer ikisi küçük olarak yazdırın, gün ismi geçerli değilse "Geçerli gün ismi giriniz" yazdırın**
- Soru 6) Kullanıcıdan iki sayı isteyin, sayıların ikisi de pozitif ise sayıların toplamını yazdırın, sayıların ikisi de negatif ise sayıların çarpımını yazdırın, sayıların ikisi farklı işaretlere sahipse "farklı işaretlerde sayılarla işlem yapamazsın" yazdırın, sayılardan sifira eşit olan varsa "sifır çarpmaya göre yutan elemandır" yazdırın.**
- Soru 7) Kullanıcıdan 100 üzerinden notunu isteyin. Not'u harf sistemine çevirip yazdırın. 50'den küçükse "D", 50-60 arası "C", 60-80 arası "B", 80'nin üzerinde ise "A"**
- Soru 8) Kullanıcıdan maaş için bir teklif isteyin ve aşağıdaki değerlere göre cevap azdırın.**
Teklif 80.000'in üzerinde ise "Kabul ediyorum",
60 – 80.000 arasında ise "Konusabiliriz",
60.000'nin altında ise "Maalesef Kabul edemem" yazdırın



Nested If Else Statements

Eger calisan kadinsa 60 yasından büyük olduğunda emekli olabilir, calisan erkekse 65 yasından büyükse emekli olabilir

Eger (calisan kadinsa) {Kadin yasini kontrol et} ,
yoksa {erkek yasini kontrol et}

```
If (calisan kadinsa)  
    {if (yas>60){emekli olabilirsiniz} else {emekli olamazsiniz}}  
else  
    {if (yas>65){emekli olabilirsiniz} else {emekli olamazsiniz}}
```



If Else Statements

Soru 11) Nested If kullanarak asagidaki soruyu cozen kodu yaziniz.

Kullanıcıdan bir sifre girmesini isteyin

Eger ilk harf buyuk harf ise "A" olup olmadigini kontrol edin. İlk harf A ise "Gecerli Sifre" degilse "Gecersiz Sifre" yazdirin.

Eger ilk harf kucuk harf ise "z" olup olmadigini kontrol edin. İlk harf z ise "Gecerli Sifre" degilse "Gecersiz Sifre" yazdirin.

Soru12) Kullanıcıdan 4 basamakli bir sayi girmesini isteyin. Girdiği sayi 5'e bölünüyorsa son rakamını kontrol edin. Son rakamı 0 ise ekrana "5'e bölünen çift sayı" yazdırın. Son rakamı 0 değil ise "5'e bölünen tek sayı" yazdırın. Girdiği password 5'e bölünmüyorsa ekrana "Tekrar deneyin" yazdırın.



If Else If Statements

Soru 13) Interview Question

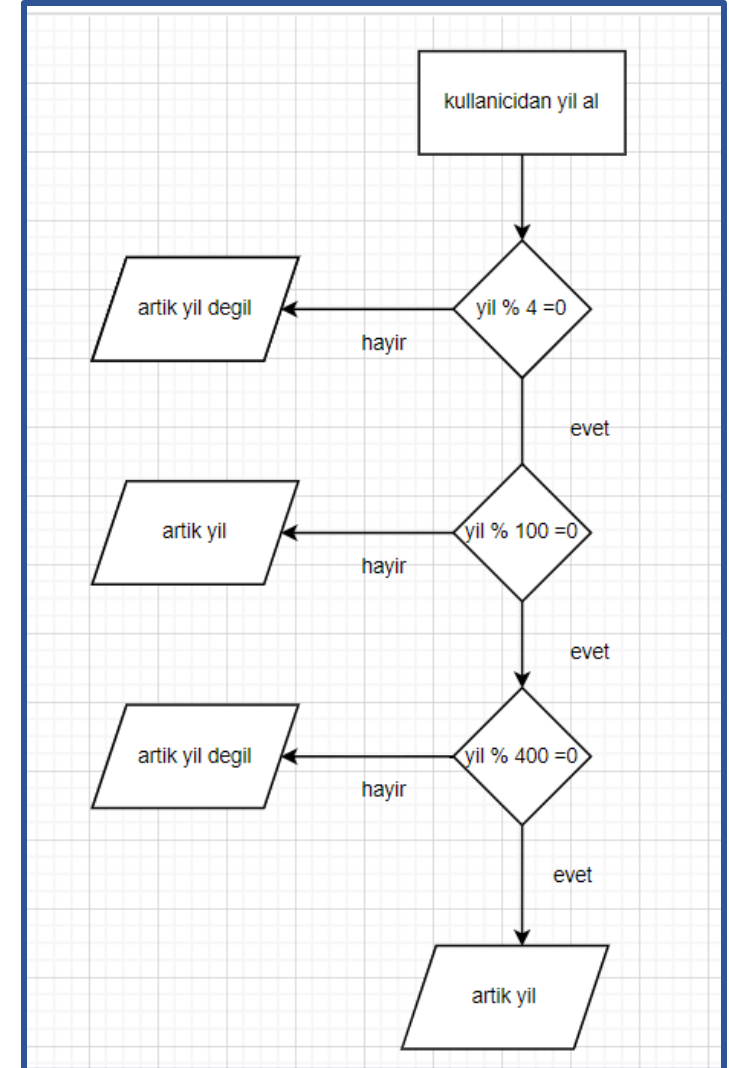
Kullanıcıdan artık yıl olup olmadığını kontrol etmek için yıl girmesini isteyin.

Kural 1: 4 ile bölünemeyen yıllar artık yıl değildir

Kural 2: 4 ile bölünüp 100 ile bölünemeyen yıllar artık yıldır

Kural 3: 4'ün katı olmasına rağmen 100 ile bölünebilen yıllardan sadece 400'ün katı olan yıllar artık yıldır

<https://app.diagrams.net/>





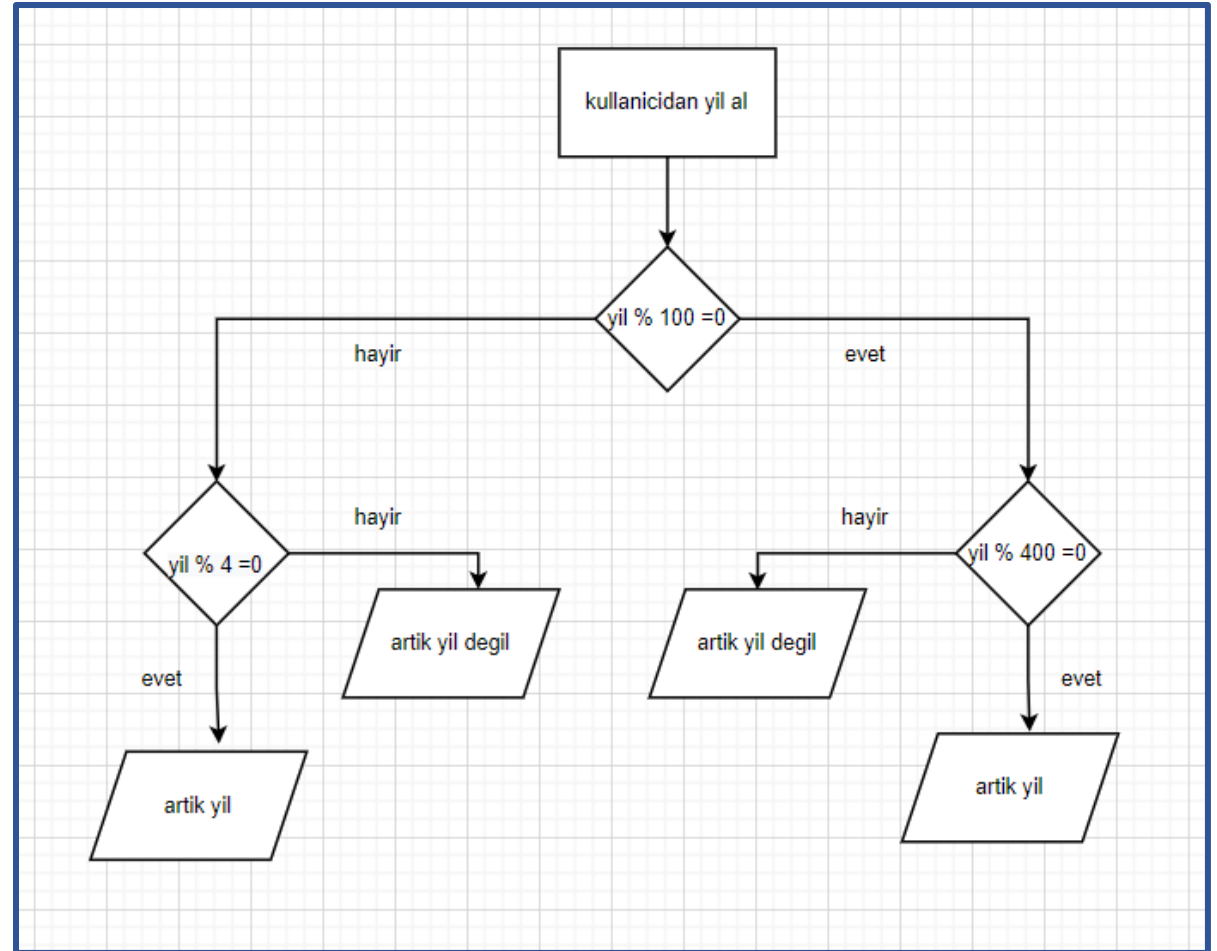
Nested If Else Statements

Soru 10) Interview Question

Kullanıcıdan artık yıl olup olmadığını kontrol etmek için yıl girmesini isteyin.

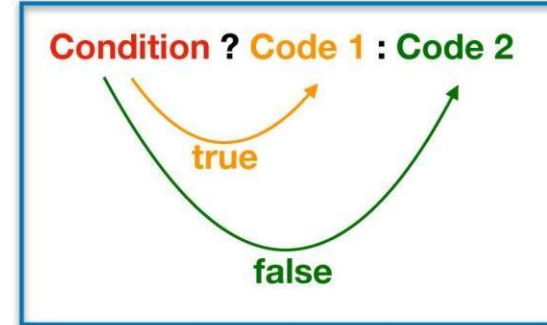
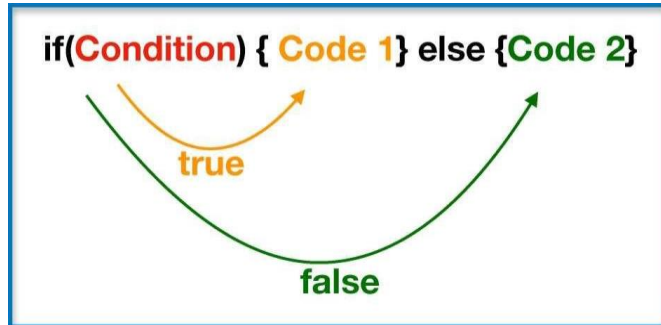
Kural 1: 4 ile bölünemeyen yıllar artık yıl değildir

Kural 2: 4'ün katı olmasına rağmen 100 ile bölünebilen yıllardan sadece 400'ün katı olan yıllar artık yıldır





Ternary Operator



Not1: Ternary işlemi If Statement ile yapacağımız işlemleri basit olarak yapmamızı sağlar

Not2: Ternary işlemi bize bir sonuç döneceği için, bu işlemi bir variable'a atamalıyız.

```
public static void main(String[] args) {  
    int x=10;  
  
    (x/2==0) ? "cift sayi" : "tek sayi";  
}
```

```
public static void main(String[] args) {  
    int x=10;  
  
    String sonuc = (x/2==0) ? "cift sayi" : "tek sayi";  
    System.out.println(sonuc);  
}
```



Ternary Operator

Ekranda Ne Goruruz ?

Soru1: `int y =112;`

```
System.out.println( (y >5) ? ("Inek") : ("Koyun") );
```

Soru2 : `int y =112;`

```
System.out.println((y <91) ? 9 : 11);
```

Soru3: `int y =1;`

```
int z =1;
```

```
int a =y<10 ? y++ : z++;
```

```
System.out.println(y +"," +z +"," +a);
```



Ternary Operator

Soru1) Kullanıcıdan iki sayı alın ve büyük olmayan sayıyı yazdırın

Soru2) Kullanıcıdan bir tamsayı alın ve sayının tek veya çift olduğunu yazdırın

Soru3) Kullanıcıdan bir sayı alın ve sayının mutlak değerini yazdırın

Soru4) Kullanıcıdan bir sayı alın. Sayı pozitifse "Sayı pozitif" yazdırın, negatifse sayının karesini yazdırın



Nested Ternary

Condition ? (Kod 1) : (Kod 2) ;

Condition1? Durum1 : Durum2

Condition2? Durum1 : Durum2

Soru1 : Kullanıcıdan bir tamsayı alın ve sayı 10'dan küçükse "Rakam" , 100'den küçükse "iki basamaklı sayı" değilse "uc basamaklı veya daha büyük sayı" yazdırın

Soru2 : Kullanıcıdan bir harf isteyin küçük harf ise consola "Küçük Harf", büyük harfse consola "Büyük Harf" yoksa "girdiğiniz karakter harf değil" yazdırın.



Nested Ternary

Ekranda Ne Goruruz ?

Soru1: `int y =8;`

`(y >5)? (y<10? 2*y : 3*y) : (y>10? 2+y : 3+y);`

Soru2: `int y =12;`

`(y >5)? (y<10? 2*y : 3*y) : (y>10? 2+y : 3+y);`

Soru3: `int y =5;`

`(y >5)? (y<10? 2*y : 3*y) : (y>10? 2+y : 3+y);`

Soru4) Kullanıcıdan dikdortgenin uzunlugunu ve genisligini alin, girilen degerlere gore dikdorgenin kare olup olmadigini yazdirin.

Soru5) Kullanıcıdan bir sayi alin ve sayi 3 basamakli ise "uc basamakli sayi", yoksa "Uc basamakli degil" yazdirin



Switch Statement

If else ile cozdugumuz sorularda **kontrol etmemiz gereken** sart sayisi cok oldugunda **switch Statement** kullanilir.

```
public static void main(String[] args) {  
    int sayi = 3;  
  
    switch(sayi) {  
        case 1 :  
            System.out.println("sayı = 1");  
            break;  
        case 2 :  
            System.out.println("sayı = 2");  
            break;  
        case 3 :  
            System.out.println("sayı = 3");  
            break;  
        case 4 :  
            System.out.println("sayı = 4");  
            break;  
        default :  
            System.out.println("sayı bunlardan biri değil");  
    }  
}
```



Switch Statement

break komutu yapacagimiz islem bittiginde switch statement'in sonuna gitmemizi saglar.

Java istenen case'e gittikten sonra **break** komutunu gorene kadar tum case'leri calistirir.

default komutu basta tanimlanan degisken icin hic bir case calismazsa calistirmek isedigimiz kodlari yazdigimiz bolumdur.

(If else statements da en sonda yazdigimiz else gibi calisir)

Switch Statement'da long,double,float ve boolean **kullanilamaz**



Switch Statement

Soru1 : Kullanıcıdan haftanın kacinci gunu oldugunu sorun ve gun ismini yazdirin

Soru2 : Kullanıcıdan kacinci ay oldugunu sorun ve ay ismini yazdirin

Soru3 : Kullanıcıdan bir sayi girmesini isteyin

Girilen sayi

10 ise "İki basamakli en kucuk sayi

100 ise "uc basamakli en kucuk sayi"

1000 ise "dort basamakli en kucuk sayi"

diger durumlarda "Girdigin sayiyi degistir" yazdirin

Soru4 : Kullanıcıdan SDET kisaltmasindaki harflerden birini yazmasini isteyin.

Kullanici S girerse "Software"

D girerse "Developer"

E girerse "Engineer"

T girerse "In Testing" yazdirin

Soru5 : Kullanıcıdan gun ismini alip haftaidi veya hafta sonu yazdiralim



String Manipulation / Methods

1 concatenation

Birden fazla String'i birleştirerek tek bir String haline getirmek için kullanılır.

İki şekilde kullanılır.

i) +(toplama) isareti ile

```
public static void main(String[] args) {  
  
    String isim= "Ali";  
    String soyisim="Can";  
  
    System.out.println(isim + " " + soyisim);  
}
```

Output:

Ali Can

ii) concat() methodu kullanarak

```
public static void main(String[] args) {  
  
    String isim= "Ali";  
    String soyisim="Can";  
  
    System.out.println(isim.concat(soyisim));  
}
```

Output:

AliCan



String Manipulation / Methods

1- charAt()

Istenen indexdeki karakteri (char) dondurur. Index 0'dan baslar, maximum index (String'in uzunlugu - 1) dir.

```
public static void main(String[] args) {  
    String isim= "Techproeducation";  
    System.out.println(isim.charAt(3));  
}
```

Output:

h

Eger method'da index olarak maximum indexden buyuk bir sayi kullanilirs Java hata verir (**StringIndexOutOfBoundsException**).

```
public static void main(String[] args) {  
    String isim= "Techproeducation";  
    System.out.println(isim.charAt(20));  
}
```

```
Exception in thread "main" java.lang.StringIndexOutOfBoundsException: String index out of range: 20  
    at java.lang.String.charAt(Unknown Source)  
    at _00_anlik.asd.main(asd.java:11)
```



String Manipulation / Methods

3 toUpperCase()

4 toLowerCase()

Girilen String degiskendeki tum harfleri istenen bicime cevirir.

```
public static void main(String[] args) {  
  
    String isim= "TechProEducation";  
  
    System.out.println(isim.toLowerCase());  
    System.out.println(isim.toUpperCase());  
}
```

Output :

techproeducation
TECHPROEDUCATION

NOT : toLowerCase(Locale locale)

Girilen String degiskendeki tum harfleri istenen local dilde istenen bicime cevirir.

```
public static void main(String[] args) {  
  
    String isim= "TECHPROEDUCATION";  
  
    System.out.println(isim.toLowerCase(Locale.forLanguageTag("tr")));  
}
```

Output :

techproeducation



String Manipulation / Methods

5-equals

Verilen iki String'in iceriginin birbirine esit olup olmadigini kontrol eder.

Eger verilen Stringlerdeki tum karakterler (bosluk, buyuk harf, kucuk harf, ozel karakter -) tamamen ayni ise **TRUE** doner, aksi durumda (bir karakter bile farkli olsa) **FALSE** doner.

```
public static void main(String[] args) {  
  
    String isim1= "Ali Can";  
    String isim2= "Ali Can";  
  
    System.out.println(isim1.equals(isim2));  
}
```

Output:

true



String Manipulation / Methods

equals Vs =

(Interview Sorusu)

equals() methodu verilen iki String'in iceriginin birbirine esit olup olmadigini kontrol eder.

= karsilastirma operatoru ise verilen iki String objesinin degerinin yaninda reference(adres)'lerine da bakar,

Ayni degere sahip olsa da farkli iki objeyi **=** ile karsilastirdigimizda sonuc **FALSE** olur.

```
public static void main(String[] args) {  
  
    String isim1= "Ali Can";  
    String isim2= isim1+"";  
  
    System.out.println(isim1==isim2);  
  
    System.out.println(isim1.equals(isim2));  
}
```

Output:
false
true



String Manipulation / Methods

6-equalsIgnoreCase

Verilen iki String degiskeni BUYUK HARF / kucuk harf farki gozetmeksizin karsilastirir.

Buyuk /kucuk harf farklilikigi disinda herhangi bir karakter farklilikigi oldugunda equals methodunda oldugu gibi FALSE dondurur.

```
public static void main(String[] args) {  
  
    String isim1= "Ali Can";  
    String isim2= "ali can";  
  
    System.out.println(isim1.equalsIgnoreCase(isim2));  
}
```

Output:

true



String Manipulation / Methods

7-length()

Verilen String'deki karakter sayisini dondurur.

```
public static void main(String[] args) {  
    String isim= "Ali Can";  
    System.out.println(isim.length());  
}
```

Output :

7

```
public static void main(String[] args) {  
    String isim= "";  
    System.out.println(isim.length());  
}
```

Output :

0

```
public static void main(String[] args) {  
    String isim= null;  
    System.out.println(isim.length());  
}
```

Exception in thread "main" [java.lang.NullPointerException](#)
at _00_anlik.asd.main([asd.java:11](#))



String Manipulation / Methods

8-indexOf()

Verilen String'de istenen karakterin kullanıldığı ilk index'i döndürür.

1) char'ın index'i sorgulanabilir

2) Parametre String olabilir

3) Olmayan karakter
sorgulanırsa

4) Parametre kelime olabilir

5) Belli bir index'ten sonrası
sorgulanabilir

```
String str= "Calisirsaniz, Java ogrenmek cok kolay";
```

```
System.out.println(str.indexOf('a'));
```

```
System.out.println(str.indexOf("a"));
```

```
System.out.println(str.indexOf("t"));
```

```
System.out.println(str.indexOf("Java"));
```

```
System.out.println(str.indexOf('a',11));
```

Output : **1**

: **1**

: **-1**

: **14**

: **15**



String Manipulation / Methods



String Manipulation / Methods