



BATCH : Batch 59
LESSON : Java 01
DATE : 19.02.2022
SUBJECT : Genel Hatırlatmalar
Java Giriş



techproeducation



techproeducation



techproeducation



techproeducation



techproedu

GOOGLE CLASSROOM da bugünkü ATTENDANCE bölümünü doldurmayı unutmayalım !!



Genel Hatırlatmalar



1. Derslere Hazırlanın ve Zamanında Katılın
2. Dersi Dikkatli Dinleyin
3. Derste Aktif Olun
4. Anlamadiklerinizi Sorun
5. Ödevlerinizi Yapın (Kod yazma araba kullanma gibidir)
6. Her Dersten Sonra Tekrar Yapın



Genel Hatırlatmalar

7. Basari = Egitim + Calismak
8. Grup calismalari yapin, En iyi ogrenme yontemi ogretmektir
9. Mentoring toplantilarini kacirmayin
10. Maillerinizi gunluk kontrol edin
11. Yoklama yapiliyor zooma isminizle girin
12. Teknik destek slack @technical support
13. Ders esnasinda canli destek
Free : Nur, Zafer , Y.Selim
Batch 59 : Elif, Merve, Feyza, Yusuf, Kenan
14. Customer service +1 917 768 74 66

**"TEACHERS
CAN OPEN
THE DOOR,
BUT YOU
MUST ENTER
IT YOURSELF."**

~ CHINESE PROVERB



Gorulecek Dersler

Automation Engineer:

Java	Selenium Grid
Selenium	Git, GitHub
SDLC	HTML & CSS
API	Bootstrap
SQL	Java Script
Jenkins	Lambda
JDBC	Project

Java Developer

Core Java	UML Diagram
Advance Java	Multi Thread
Oracle SQL	Hibernate
JDBC	MongoDB
HTML5 & CSS	SpringMVC
Bootstrap	Restful API
JavaScript	Micro Services with Spring Boot
React.js	Git-GitHub
SDLC	
Market Session	

Mobile Developer

Core Java	Git-GitHub
Oracle SQL	Bootstrap
SDLC	React.js
HTML5 & CSS	JavaScript
Advance Java	React Native
	Project



Mentoring

Mentoring toplantıları her hafta team tarafından ortak belirlenen gün ve saatte düzenli şekilde yapılmaktadır.

- ✓ Mentoring faaliyetleri STUDENT COACHING (öğrenci danışmanlığı) olarak yapılmaktadır.
- ✓ Mentoring faaliyetlerinde...
 - Haftanın görülen derslerin değerlendirmesi...
 - Derslerle ilgili döküman desteğinin sağlanması....
 - Ödev proje vs çalışmaların takip edilmesi...
 - Team work'lerin takip edilmesi...
 - FlipGrid çalışmalarının takip edilmesi...
 - Java verbal çalışmalarının takip edilmesi...
 - Java coding çalışmalarının takip edilmesi...
 - Interview çalışmalarının takip edilmesi...

DÜZENLİ OLARAK YAPILMAKTADIR....



Ders Isleyisi - Bilmeniz Gerekenler

1. Maillerinizi gunluk kontrol edin
2. Dersleri zoom'dan izliyoruz ama mesajlasma icin slack kullaniyoruz



- Iki slack kanalimiz var
- Direk mesaj
- Kod paylasma (**snippet**)
- Mesaj silme ve edit
- Pin yapma



3. Google Classroom
 - Tum ders notlari, zoom linki ve videolar Google Classroom'dan paylasilacak
 - Maillerinize davetiye gonderildi
 - Youtube videolari



Ders Isleyisi - Bilmeniz Gerekenler



1-Ders esnasında öğrencilerin dikkatini dağıtacak paylaşımlar yapmayın

2-Ders esnasında ders ve konu dışında paylaşım yapmayınız.

3-Diyaloglarınızda asgari nezaket ve saygı kurallarına azami dikkat ediniz.

4-Ders esnasında ders hocasına direct mesaj yazmayınız

5-Derste code paylaşırken SNIPPET ve screenshot kullanmaya dikkat ediniz

6-Dersi iyi takip ediniz, öncesinden sorulmuş ve cevaplanmış soruyu tekrar sormamaya azami gayret gösteriniz.

7-CODE ve SYNTAX hatalarınız için MENTOR'lerimiz, KURULUM hatalarınız için TECHNICAL SUPPORT yardımcı olacaktır.

8-CODE ve SYNTAX hatalarınızı DM olarak değil benzer hataları alanların da yararlanması için öğrenci yardımlaşma channel'den paylaşınız.



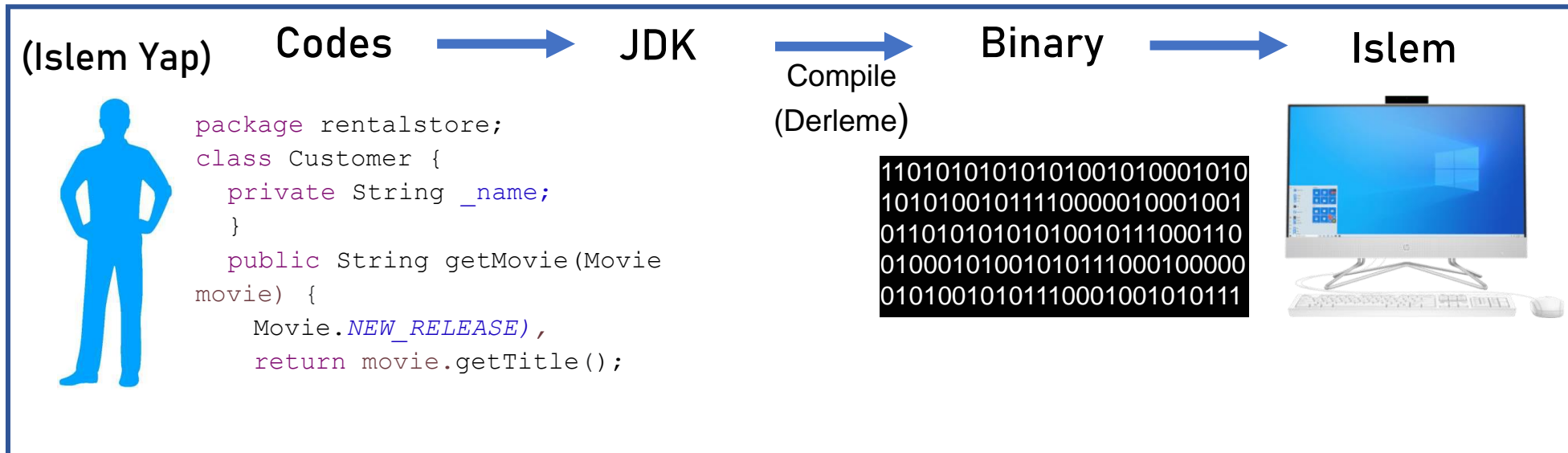
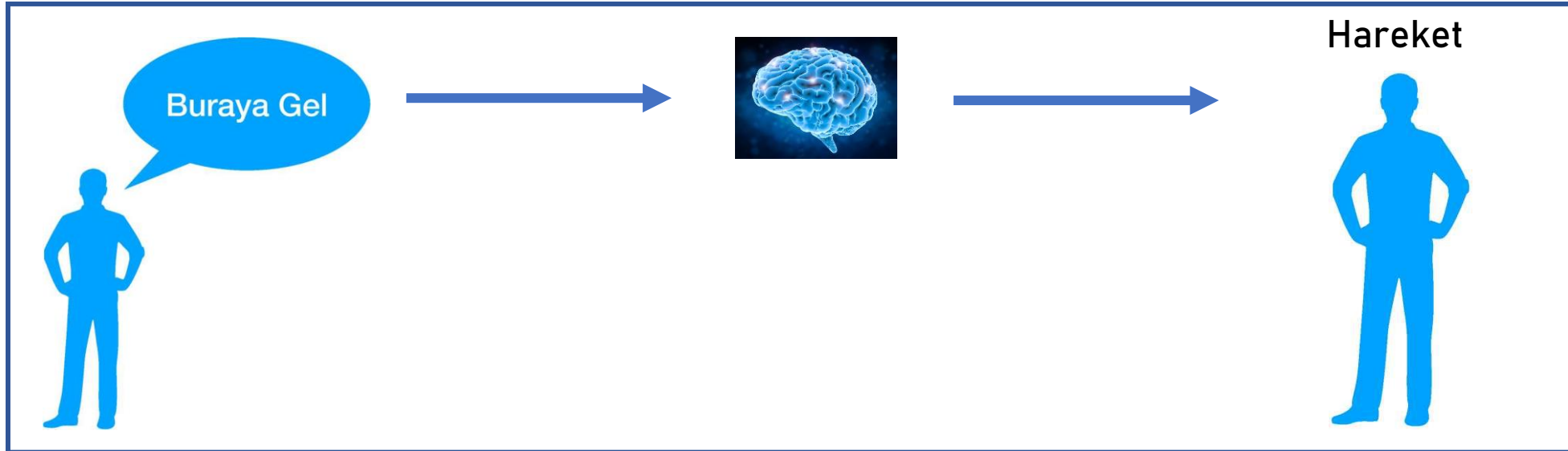
Ders Isleyisi - Bilmeniz Gerekenler

1. Ders tam zamaninda baslar.
2. Dersin basinda 10 dakika bir onceki gunun kisa tekrari yapilir
3. Her konu bittiginde ertesi gun kisa tekrardan sonra Socrative testi yapilir (10 -15 dk) sonra o sorular cozulerek konu tekrari yapilir





Programlama Dili Nedir ?





Nicin Java ?

1- Ogrenmesi kolay

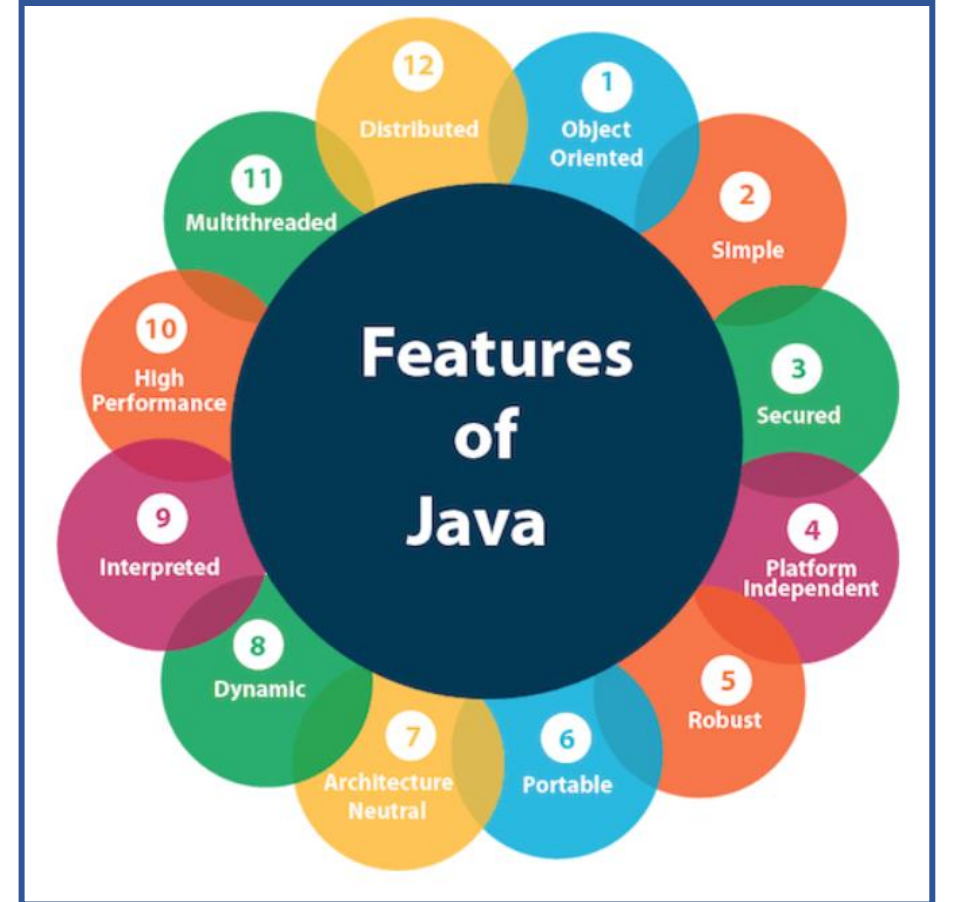
2- Dunyada en cok kullanılan programlama dili

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)

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



Application (Urun)

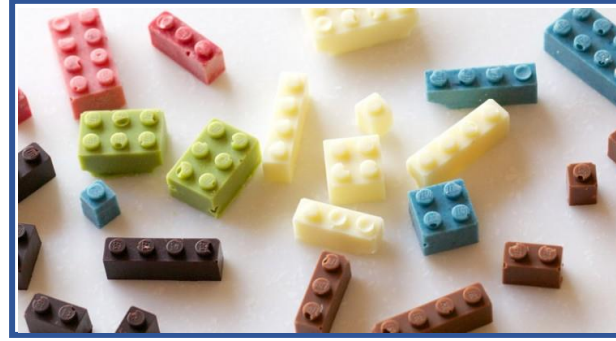


Bir Object Nasıl Olusturulur?



Class(Object Kalibi)

Field
(Variables) Method
(Functions)



Object



Birden fazla Obje birlestirilir



Application



Object Nasıl Kullanilir ?



Ogretmen



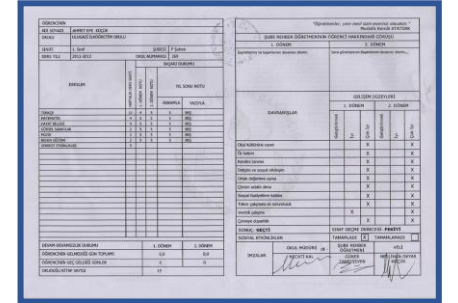
Ogrenci

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 bulunduгу class'da kullanilabilir
protected : Sedece icinde bulunduгу class ve child class'lardan kullanilir
- 2 **int** : Return Type, methodun ne urettigini ve bize dondurdugunu belirtir
- 3 **myFirstMethod** :Olusturdugumuz method'un ismidir. Isim mutlaka kucuk harfle baslar, birden fazla kelimedenden olusursa sonraki kelimelerin ilk harfleri buyuk harf yazilir (Camel Case)
- 4 **()** **parantez**: Methodlarda isimden sonra parantez kullanilir ve gerektiginde parantez icinde parametre yazilir.
- 5 **Body (Method Body)** : { } arasinda kalan kodlarimizi yazdigimiz bolumdur



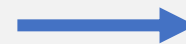
Main Method

```
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 Cumesi (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 calistirilmayan, amaci kodlarin aciklanmasi veya bir konuda bilgi vermek olan cumlelerdir

➤ Genelde iki kullanim vardir

1) Tek satirlik comment

2) Cok satirlik comment

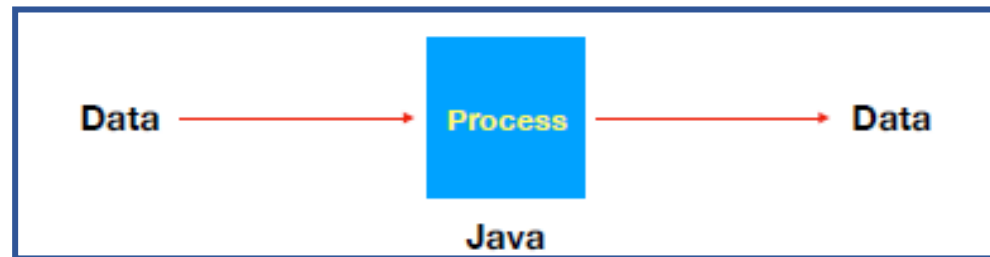


Data Nedir?

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.

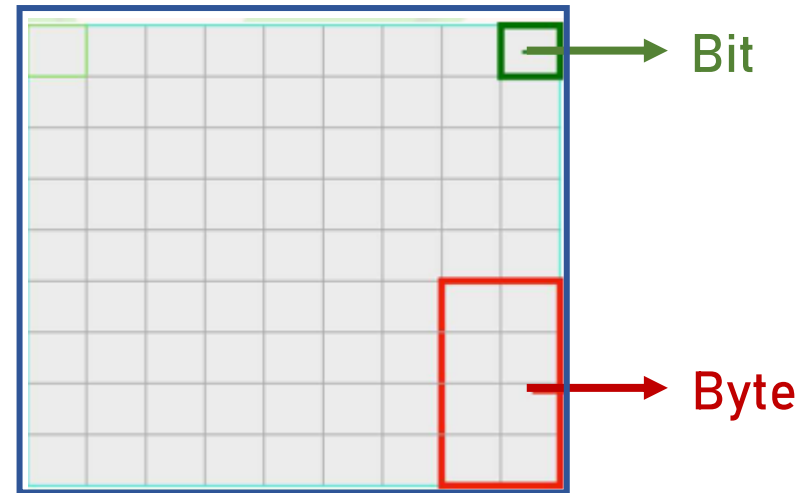
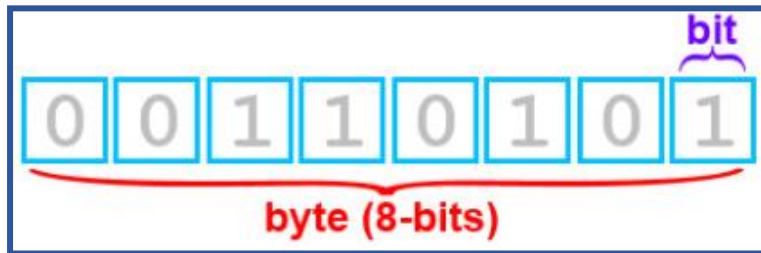




Bit

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

Note: 8 bit =1 byte



Memory
(Hafıza)



BATCH : Batch 60
LESSON : Java 02
DATE : 19.02.2022
SUBJECT : **J**ava **G**iris
Variables



techproeducation



techproeducation



techproeducation



techproeducation



techproedu



IntelliJ Kullanım

1- Proje oluşturma

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

2- Package (paket) oluşturma

src dosyasına sağ click -- New -- Package -- day01variables -- finish

3- Class oluşturma

day01variables dosyasına sağ click -- New -- Class -- C01_Variables01 -- finish

4- Main method oluşturma

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



Variables (Degisken) Olusturma

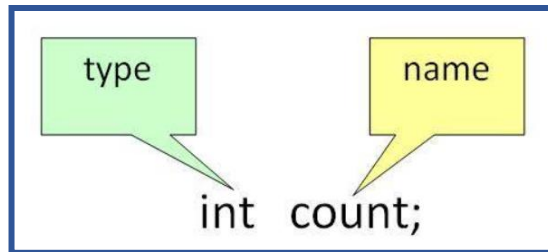
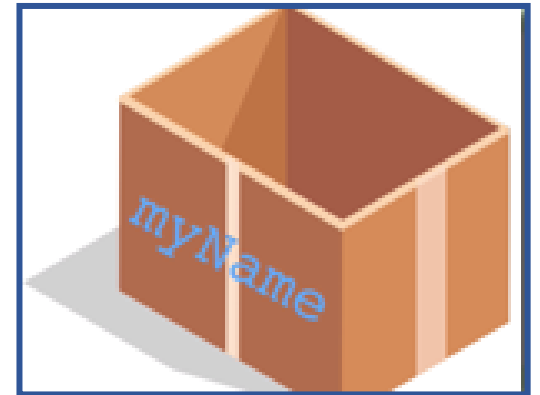
Declaration

Variable bellekte (**memory**) ayrılmış olan alanın (**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

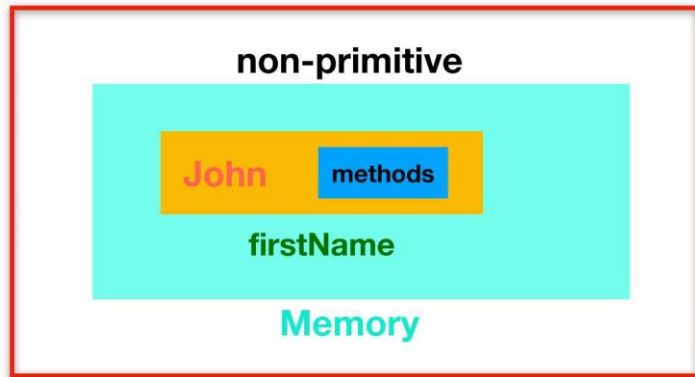
```
double doubleVar2 = -2.12345679078000000000123
```



Non-Primitive Data Type

String Data Type:

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



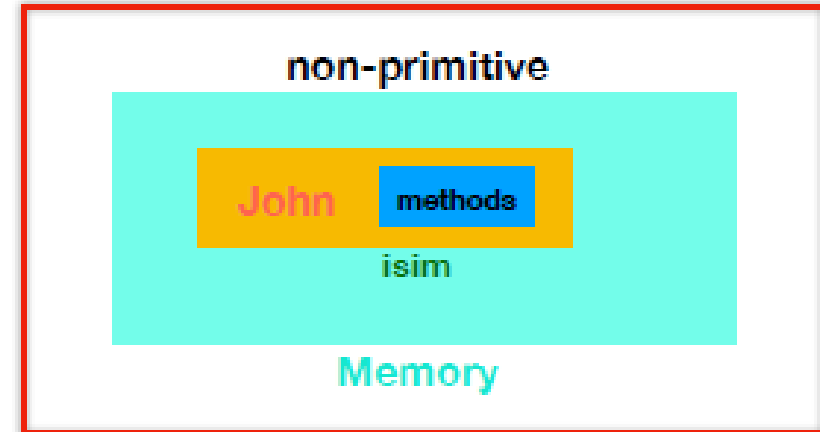
```
String okulAdi = "Yildiz Koleji, Cankaya Ankara #";  
String telNo   = "5321234567";  
String ilkHarf = "A";
```

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.



BATCH : Batch 60
LESSON : Java 03
DATE : 19.02.2022
SUBJECT : Kullanıcıdan Deger Alma
Data Casting
Increment/ Decrement



techproeducation



techproeducation



techproeducation



techproeducation



techproedu



Önceki Dersten Aklımızda Kalanlar

1. OOP concept : nesne tabanlı programlama demektir, biz olusturdugumuz class'lar sayesinde objeler uretebiliriz ve bu objeler i birlestirerek kompleks uygulamalar gelistirebiliriz(lego gibi)
2. Clas hangi bolumlerden olusur ?
 - class declaration : keyword ler sayesinde class'ı kimlerin kullanabilecegini gorebiliriz, class ismi buyuk harfle baslar ve CamelCase seklinde yazilir
 - { } curly braces / suslu parantez : Classin nerede baslayip bitttigini gosterir
 - Class Body : curly braces arasinda kalan ve kodlarimizi yazdigimiz bolumdur
- 3- Class icerisinde neler olur ?
 - Main method :arabanin motoru gibidir, Java kodlarimiz calistirmaya main method'dan baslar
 - normal method'lar : method'lar bizim adimiza belirledigimiz islemleri yapip, islem sonunda da istedigimiz sonucu bize dondururler
 - variable : bizim icin degerleri saklayan konteynir'lardir



Önceki Dersten Aklımızda Kalanlar

- 4- variable : bizim istedigimiz deger koyabilmemiz icin hafiza da ayrilan bolumun adidir.
Ornegin bir oyunda level bilgisi icin bir variable tanimlarsak, oyunun hangi asamasinda olursa olsun level variable'ina baktigimizda icinde level degerimizi gorebiliriz
Biz programimiz icerisinde ne zaman variable ismini yazsak, java o variable yerine en son atanan degeri kullanir.
- 5- Data turleri
 - primitive data turleri : boolean, char, byte, short, int, long, float, double
 - non-primitive data turleri : String (ilerde pek cok cesidini gorecegiz, biz de istersek non-primitive data turu olusturabiliriz)
- 6- iki data turu arasindaki farklar
 - p'ler sadece depolama yapar, np'ler ise hem depolama yapar hem de kendilerine ozel methodlar sayesinde istedigimiz degisimleri yaparlar
 - p data turlerinin isimleri kucuk harfle baslar, np data turlerinin isimleri buyuk harfle baslar
 - p'ler 8 tanedir ve biz yeni p data turu uretemeyiz, ancak np'leri java da uretebilir biz de uretebiliriz, dolayisiyla np data turu sayisi sinirlendirilamaz



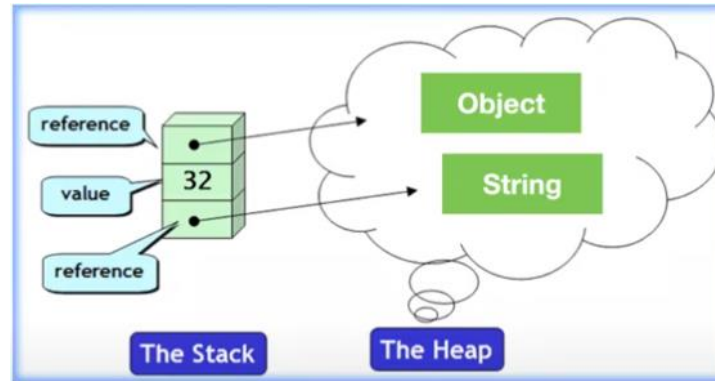
Variable ve Method'lar Nasıl Adlandırılır

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

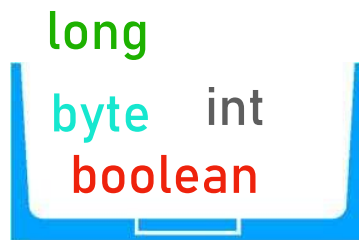
1- **Stack Memory** : primitive data tiplerine ait degerleri ve Non-primitive datalara (Object) ait referanslari(adres) barindirir

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

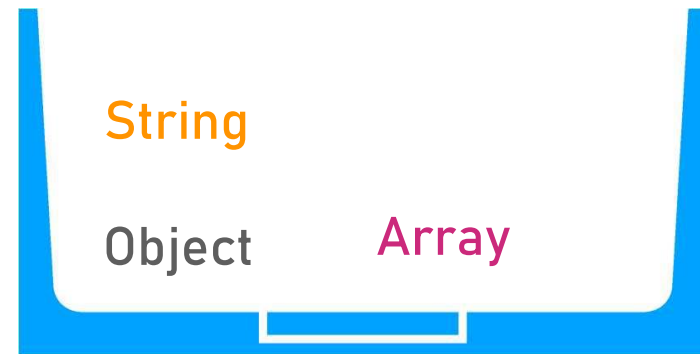


Memory (Hafiza) Kullanımı

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
seklinde yazdirin
- 3- Iki farkli tamsayi data turunde 2 variable olusturun bunlarin toplamini yazdirin
- 4- Bir tamsayi ve bir ondalikli variable olusturun ve bunlarin toplamini yazdirin
- 5 - char data turunde bir variable olusturun ve yazdirin
- 6- Bir tamsayi, bir de char degisken olusturun ve bunlarin 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	ì	173	¡
142	Ä	174	«
143	Å	175	»
144	É	176	
145	æ	177	
146	Æ	178	
147	ô	179	
148	ö	180	
149	ø	181	À
150	ù	182	Á
151	û	183	Â
152	ÿ	184	Ã
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	ô
194	Ł	226	Ô
195	ł	227	Ò
196	Ł	228	ö
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 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 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 hesaplayip yazdırın

Soru 3) Kullanıcıdan yarıçap isteyip cemberin cevresini ve dairenin alanini hesaplayip yazdırın

Soru 4) Kullanıcıdan dikdörtgenler prizmasının uzun, kısa kenarlarını ve yüksekliğini isteyip prizmanın hacmini hesaplayip 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.



BATCH : Batch 59-60

LESSON : Java 04

DATE : 23.02.2022

SUBJECT : Data Casting

Increment / Decrement

Matematiksel Operatorler



techproeducation



techproeducation



techproeducation



techproeducation



techproedu



Önceki Dersten Aklımızda Kalanlar

1. Scanner class'ı kullanıcidan bilgi almak için kullanılır.
2. Uc adımda kullanıcidan degeri aliriz
 - Scanner olusturma, parameter olarak System.in yazmaliyiz
 - Kullaniciya ne girecegini soyleyen bir mesaj yazdirma
 - Olusturdugumuz scan objesi ve girilecek dataya uygun next method'u ile kullanıcinin girdigi degeri alip, gelen dataya uygun data turunde olusturdugumuz variable'a atama yapariz
- 3- Kullanıcidan aldigimiz deger metin ise next() veya nextLine() method'u kullanilir. Next method'u sadece ilk bosluga kadar olan metni alirken, nextLine tum satiri alir
- 4- Java da kullanılan 2 tur hafiza vardır.
 - stack- primitive dataturundeki variable'ların aldigi degerler 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 , olusturulduđu data tipinden farklı bir data turunden deđer atanmasına 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 sıralaması şu şekildeydi

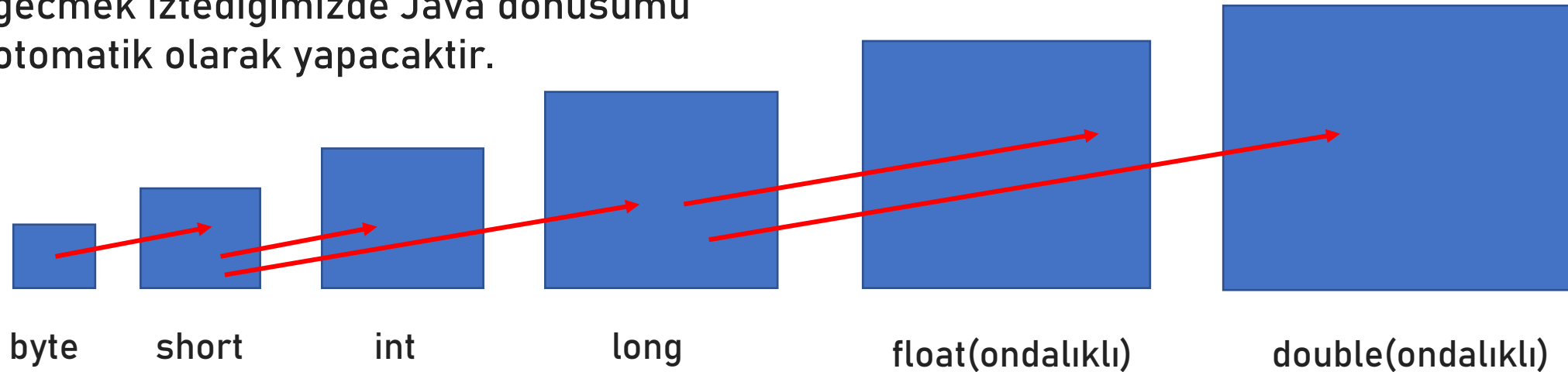
byte < short < int < long < float(ondalıklı) < double(ondalıklı)



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



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 -- ;
```

?



BATCH : Batch 59-60

LESSON : Java 05

DATE : 24.02.2022

SUBJECT : Pre & Post Increment
Matematiksel Operatorler
Modulus



techproeducation



techproeducation



techproeducation



techproeducation



techproedu



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 1 artirip azaltacaksak bunu kullaniriz



Pre-Increment & Post Increment

- Pre-Increment ve Post Increment operatorlerinin her ikisi de artirma islemi icin kullanilir
- Pre-Increment isleminde variable statement'da kullanilmadan once artirilir veya azaltilir

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

Output : 16

- Post Increment isleminde variable statement'da kullanilir, sonra artirilir veya azaltilir

```
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 :

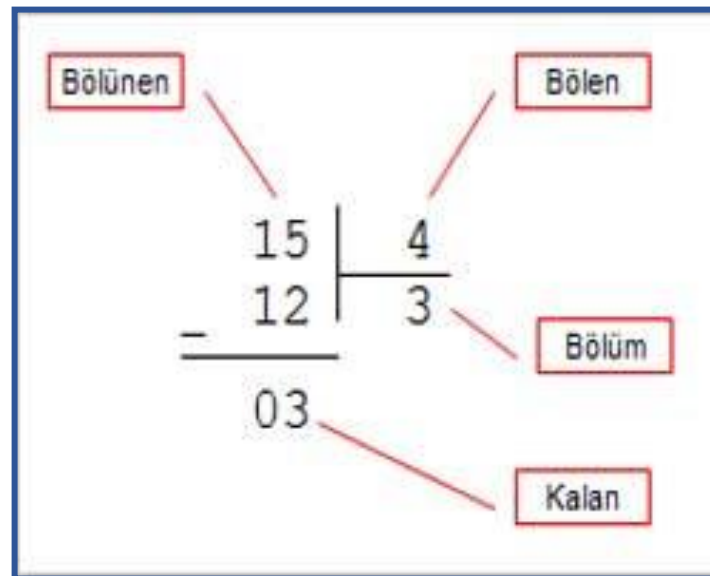
$$8 + 2 * (14 - 6 / 2) - 12 =$$



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



BATCH : Batch 59-60

LESSON : Java 06

DATE : 25.02.2022

SUBJECT : Concatenation

Relational Operators

Conditional Operators



techproeducation



techproeducation



techproeducation



techproeducation



techproedu



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 artırır, sonra yeni değerini yazdırır
- 3- Modulus % : 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 bakmak
 - %10 yaparak sayının birler basamağını almak
- 4- bir sayının rakamları toplamını bulmak için için, basamak sayısı miktarınca şu işlemler tekrar edilir
 - %10 ile birler basamağını 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 **BULUTLUOZ** yazın
- 3) Isminizi yazın
- 4) **Done** butonuna basın

Sure : 15 Dakika



Concatenation / (String Datalari Birlestirme)

Birden cok String'i + isareti ile topladiginizda Java bu String degiskenleri birlestirerek yeni bir String olusturur

```
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'lari 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)

!= Esit degildir isareti

boolean sonuc1= 5+2 != 7;

sonuc1 degeri **false** olur

System.out.println(5*2 != 15);

true yazdirir

> Buyuktur , **>=** Buyuk veya esittir

boolean sonuc1= 5+2 >= 7;

sonuc1 degeri **true** olur

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

false yazdirir

< Kucuktur , **<=** 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 mukemmeliyetcidir)

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

sonuc1 degeri **true** olur
false yazdirir

|| OR (veya) isareti

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

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

sonuc1 degeri **true** olur
false yazdirir



& ile && 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 gecer.



BATCH : Batch 59-60
LESSON : Java 07
DATE : 26.02.2022
SUBJECT : **If** Statements



techproeducation



techproeducation



techproeducation



techproeducation



techproedu



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"

***** Büyük küçük harf problem olmaması için toUpperCase methodunu kullanın**

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 için equals method'unu kullanın**

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

Soru 5) Kullanıcıdan bir gün alın eğer gün "Cuma" ise ekrana "Müslümanlar için kutsal gün" yazdırın. "Cumartesi" ise ekrana "Yahudiler için kutsal gün" yazdırın. "Pazar" ise ekrana "Hristiyanlar için 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 olabilirsin" 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 sadece eşit olan varsa “sıfır 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 oldugunda 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. Ilk harf A ise “Gecerli Sifre” degilse “Gecersiz Sifre” yazdirin.

Eger ilk harf kucuk harf ise “z” olup olmadigini kontrol edin. Ilk 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

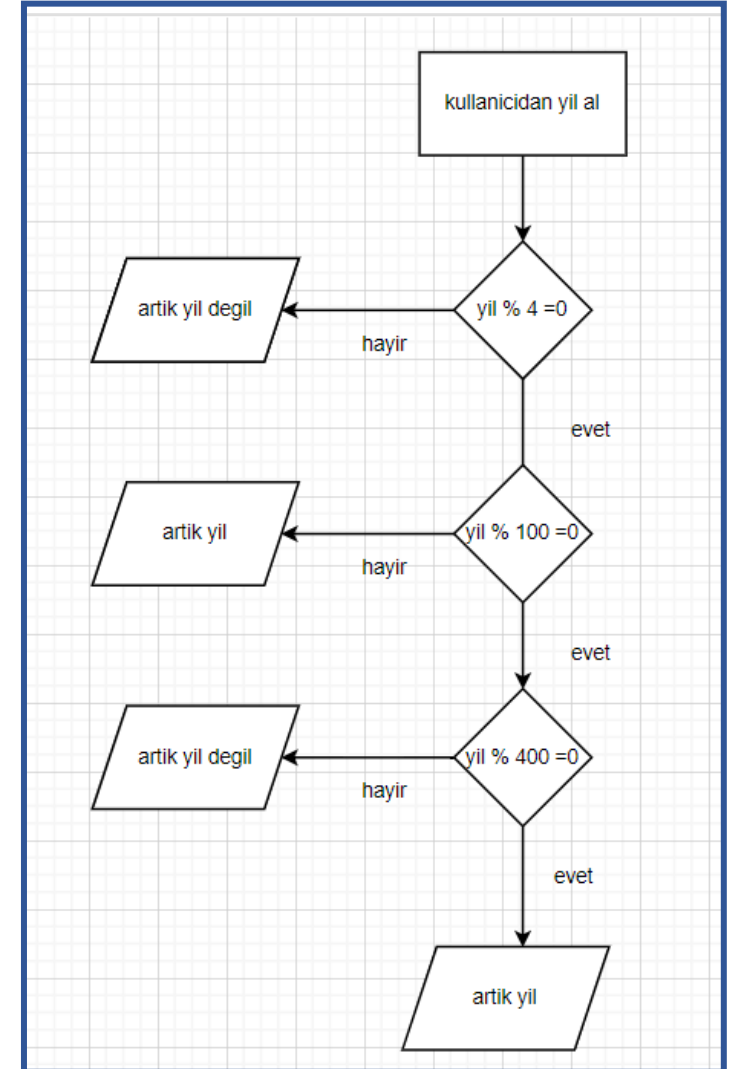
Kullanıcıdan artık yıl olup olmadığını kontrol etmek için yıl girmesini isteyin.

Kural 1: 4 ile bölünemeyen yıllar artık yıl değildir

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

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

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





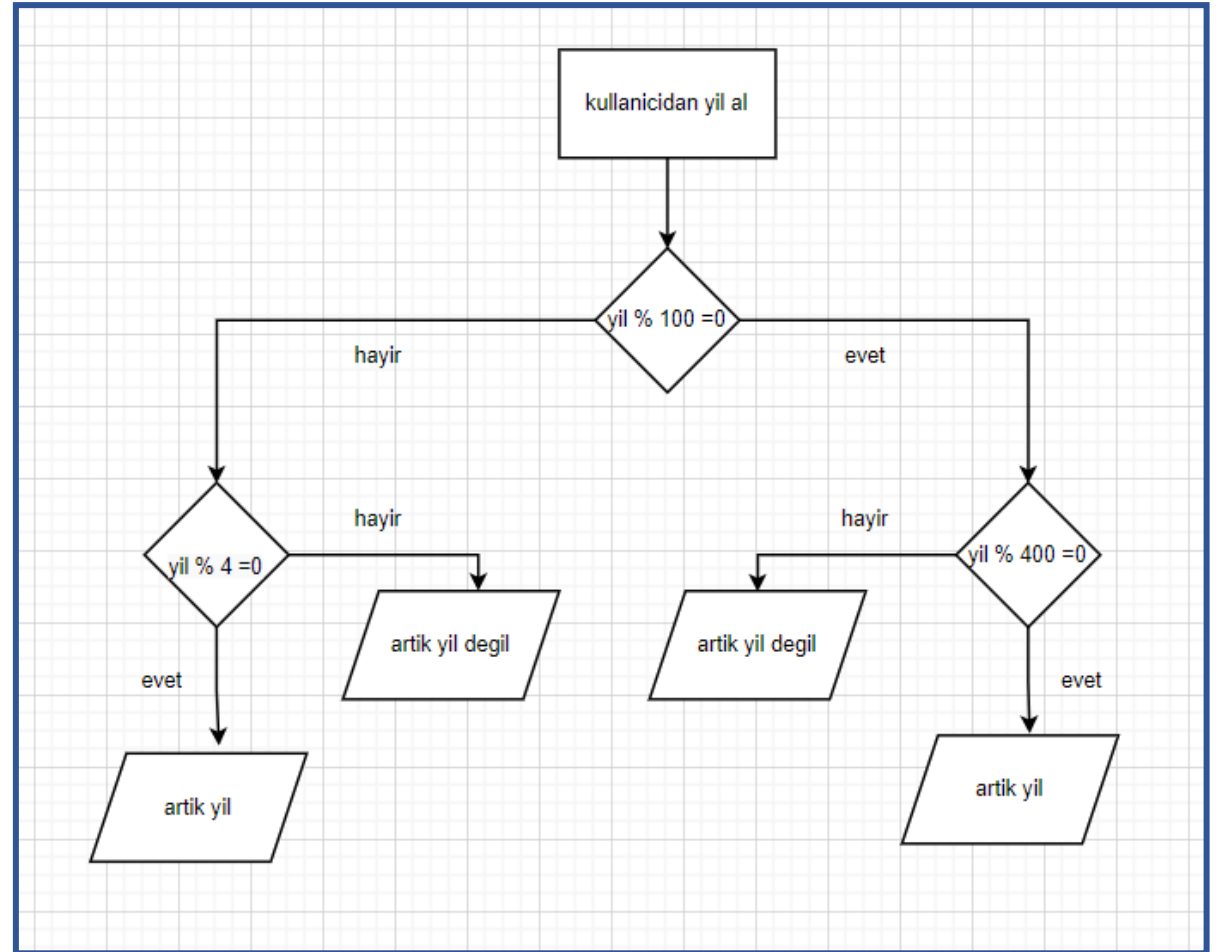
Nested If Else Statements

Soru 10) Interview Question

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

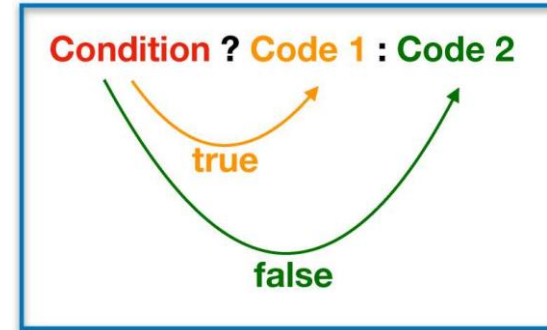
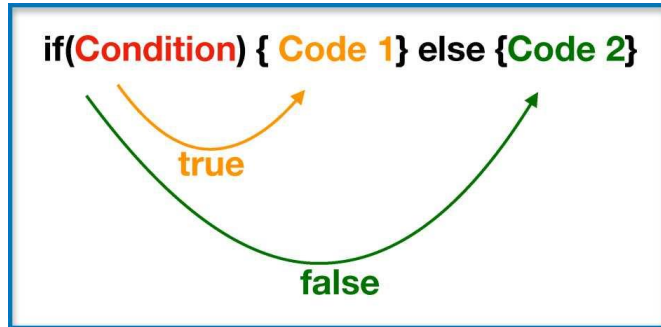
Kural 1: 4 ile bölünemeyen yıllar artık yıl değildir

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





Ternary Operator



Not1 : Ternary islemi If Statement ile yapacagimiz islemleri basit olarak yapmamizi saglar

Not2 : Ternary islemi bize bir sonuc donecegi icin, bu islemi bir variable'a atamaliyiz.

```
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 haftaici 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()

İstenen indexdeki karakteri (char) döndürür. Index 0'dan başlar, maximum index (String'in uzunluğu - 1) dir.

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

Output :

h

Eğer method'da index olarak maximum indexden büyük bir sayı kullanılırsa 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 farkliligi disinda herhangi bir karakter farkliligi 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 kullanildiği ilk index'i döndürür.

- 1) char'in 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