

23 EKIM 2021
DERS 1

Genel Hatırlatmalar
Java Giriş

Mehmet BULUTLUOZ
Elk.Elektronik Yuk.Muh.

Genel Hatırlatmalar



1. Derslere Hazirlanın ve Zamanında Katılın
2. Dersi Dikkatli Dinleyin
3. Derste Aktif Olun
4. Anlamadıklarınızı 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 toplantılarini kacirmayin
10. Maillerinizi gunluk kontrol edin
11. Yoklama yapiliyor zooma isminizle girin
<https://class.techproed.com/>
12. Teknik destek slack @technical support
13. Ders esnasında canlı destek
Haluk Bilgin, Fatih Kilic, Ebubekir Sahin
14. Customer service +1 917 768 74 66

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

~ CHINESE PROVERB

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....

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

Ders İsleyisi Bilmeniz Gerekenler

1. Maillerinizi günlük kontrol edin

2. Dersleri zoom'dan izliyoruz ama mesajlaşma için slack kullanıyoruz



- İki slack kanalımız var
- Direk mesaj
- Kod paylaşma (**snippet**)
- Mesaj silme ve edit
- Pin yapma



Google Classroom

3. Google Clasroom

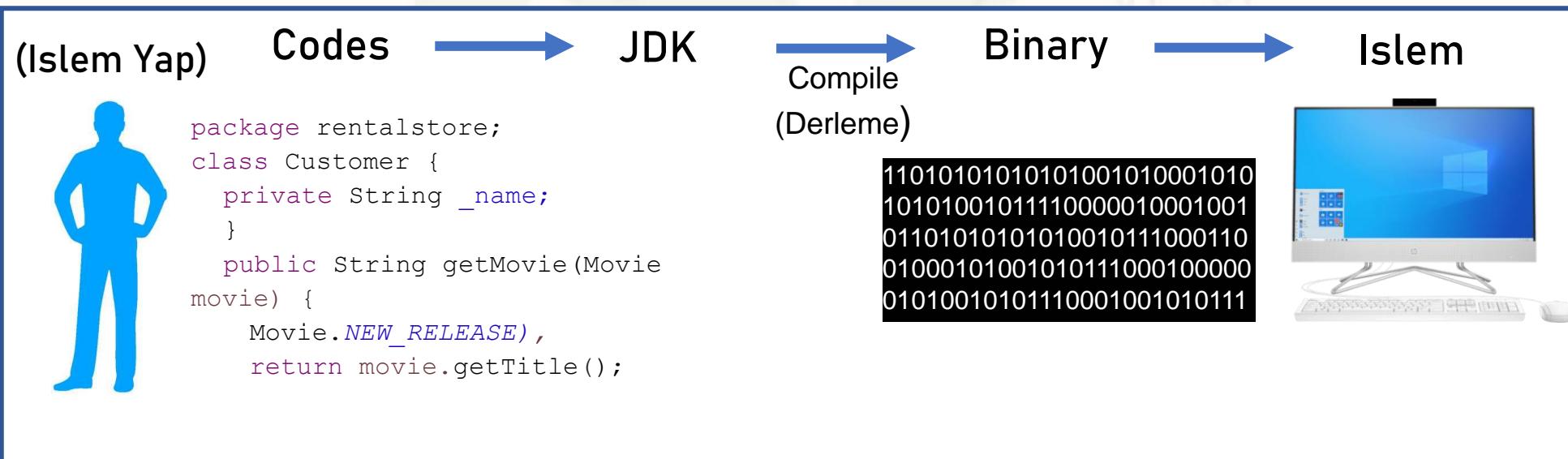
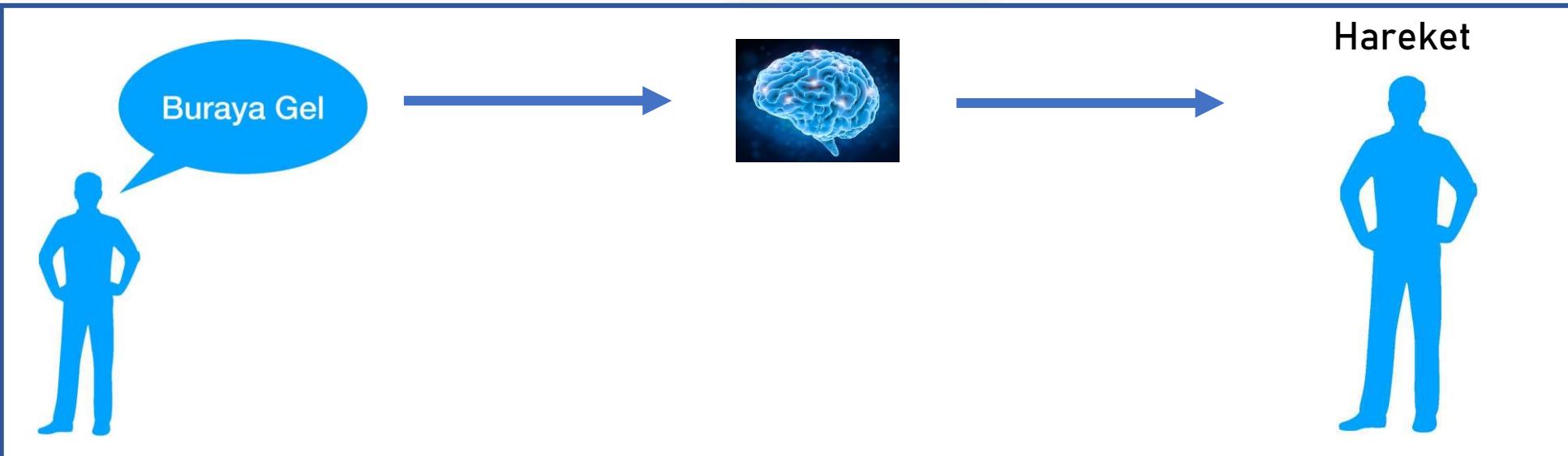
- Tüm ders notları, zoom linki ve videolar Google Classroom'dan paylaşılacak
- Maillerinize davetiye gönderildi

Ders Isleyisi Bilmeniz Gerekenler

1. Ders tam zamanında baslar.
2. Dersin basında 10 dakika bir önceki günün kısa tekrarı yapılır
3. Her konu bittiğinde ertesi gün kısa tekrardan sonra Socrative testi yapılır (10 -15 dk) sonra o sorular cozulerek konu tekrarı yapılır



Programlama Dili Nedir ?



Nicin Java ?

1- Öğrenmesi kolay

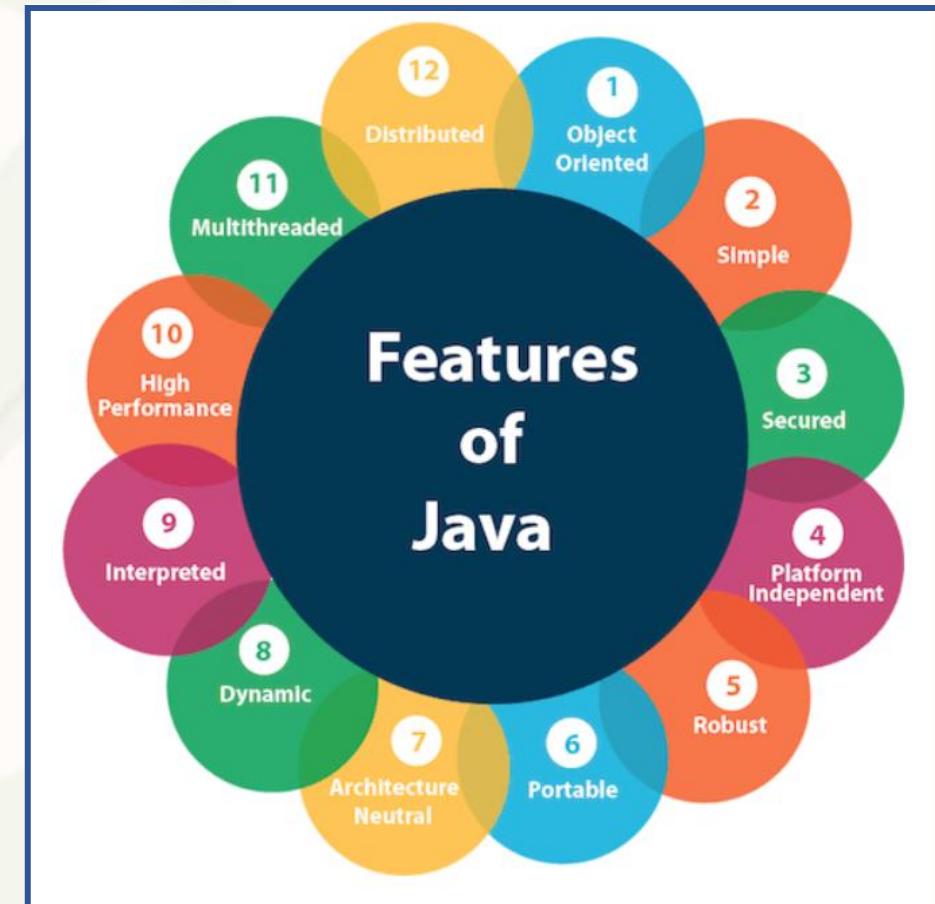
2- Dünyada en çok kullanılan programlama dili

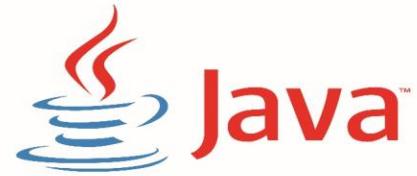
Sun'a göre 3 milyar cihaz Java kullanıyor. Şu anda Java'nın kullanıldığı birçok cihaz var.

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)



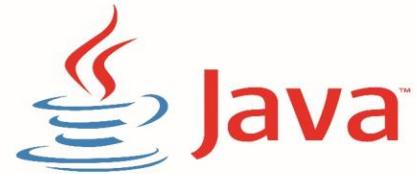
Application (Urun)

1- Feature (Fields veya Variables)

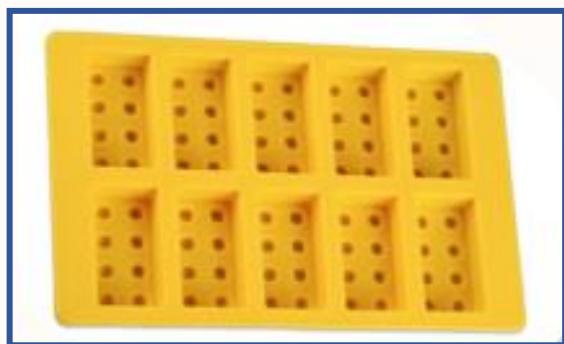
Pasif ozellik (renk,sekil,isim)

2- Functionality (Method)

Aktif ozellik (tasima,degistirme)



Bir Object Nasil Olusturulur?



Class(Object Kalibi)

Field Method
(Variables) (Functions)



Object

Birden fazla Obje birlestirilir

Application





Object Nasıl Kullanılır ?



Ogretmen

Dersler

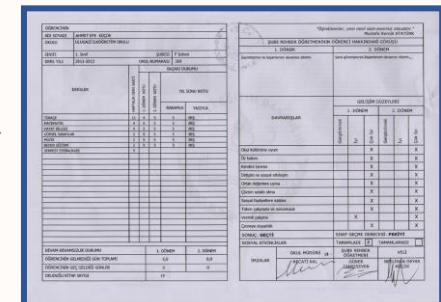
09:00	TÜRKÇE-1
09:30	MATEMATİK-1
10:00	TÜRKÇE-2
10:30	MATEMATİK-2
11:00	TÜRKÇE-3
11:30	MATEMATİK-3
12:00	TÜRKÇE-4
12:30	MATEMATİK-4
13:00	İYEP TÜRKÇE



Personel



Ogrenci



Notlar

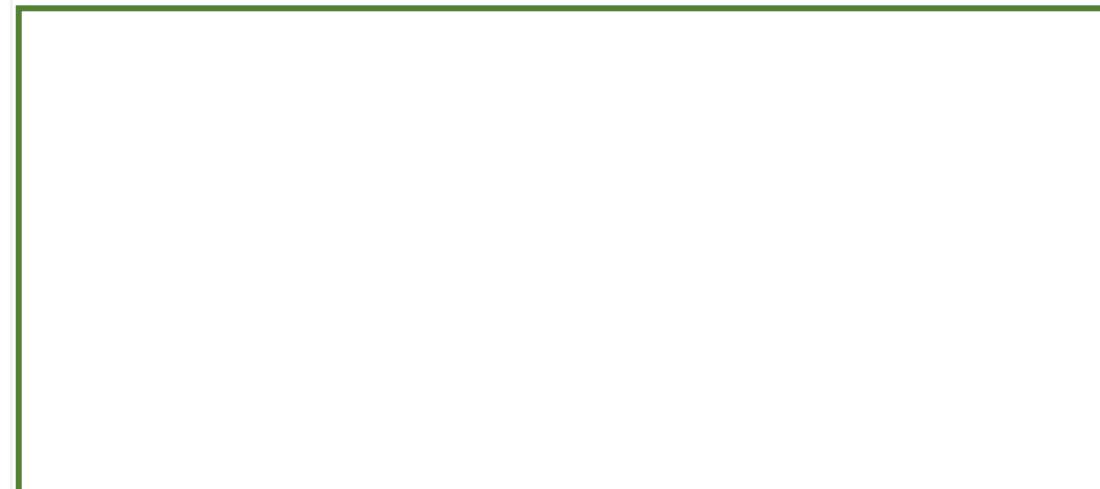


Bir Class Hangi Bolumlerden Olusur?

```
public class C2_MethodCreation2 {
```

1

2



2

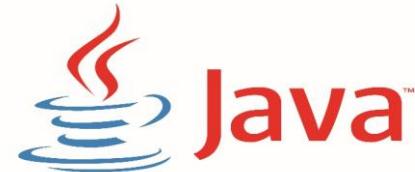
```
}
```

// Class sonu

1 – Class Declaration

2 – Curly braces : Suslu parantez

3 – Class Body : Suslu parantezler arasında kalan ve kodlarimizi yazdigimiz bolum



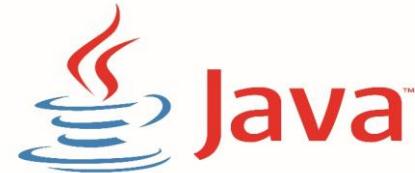
Bir Class'in Icinde Neler Bulunur?

```
public class C2_MethodCreation2 {  
  
    private double ortalama; 1  
    public int sonuc;  
  
    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);  
    }  
} // Class sonu
```

1 – Field / Variables

2 – Main Method

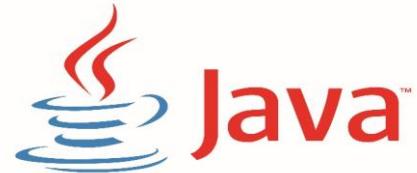
3 – Method



Class Olustururken (Declaration) Kullanilan Keyword'ler Nelerdir?

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

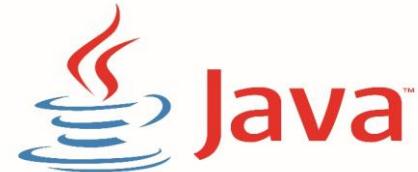
- 1 **public** : Access Modifier (Erisim duzenleyici) : class'a kimlerin erisebilecegini belirler. Public olursa her yerden erisilebilir
default : Sadece bulundugu Package'den kullanilabilir
- 2 **class** : Yazdigimiz kodun class oldugunu belirtir
- 3 **MyFirstClass** : Olusturdugumuz class'in ismidir.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



Method Oluştururken Kullanılan Keyword'ler Nelerdir?

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

- 1 **public** : Access Modifier (Erisim duzenleyici):method'a kimlerin erisebilecegini belirler
private: Sadece bulunduğu class'da kullanilabilir
protected : Sadece icinde bulunduğu 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) : **{ }** arasında kalan kodlarimizi yazdigimiz bolumdur



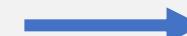
Main Method

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



- main method, java'nin calismaya basladigi giristir. (**Entry Point**)
- main method olusturulurken yazilmasi gereken syntax (kod dizimi) degistirilemez
- Parantez icinde yazilan (`String[] args`) java'nin calismasi icin gerekli olan parametreleri barindirir ve olmasi sarttir.

Araba

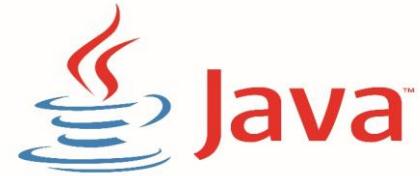


Motor

Java Project



Main Method



Yorum Cumlesi (Comment) Nasil Eklenir ?

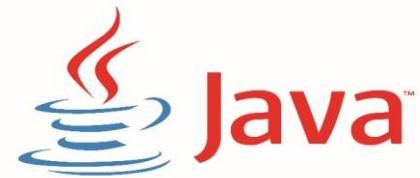
```
public class Example {  
    // Bir satiri comment haline getirmek icin // kullanilir  
    String isim = "Mehmet";  
  
    /*  
     * Eger birden fazla  
     * satiri yorum haline  
     * getirmek istiyorsak  
     * kullanilir  
     */  
    int sayi=10;  
    double not=75.70;  
}  
  
boolean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

➤ **Comments** : Java tarafindan calistirilmayan, amaci kodların aciklanması veya bir konuda bilgi vermek olan cümlelerdir

➤ Genelde iki kullanim vardir

1) Tek satirlik comment

2) Cok satirlik comment

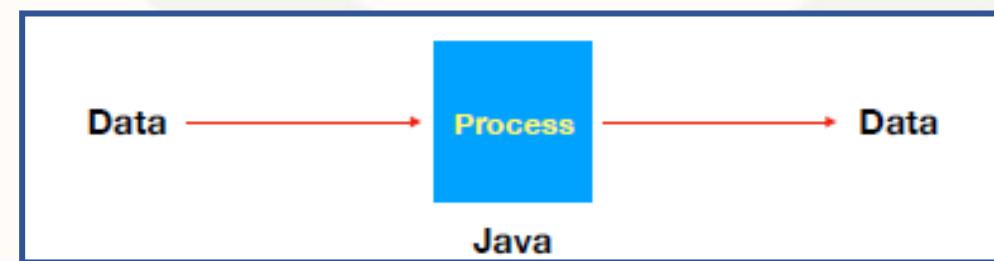


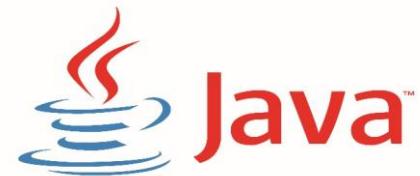
Data

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
01100011011011110110110101110000011101010111010001100101011100100010000001101000000 101

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

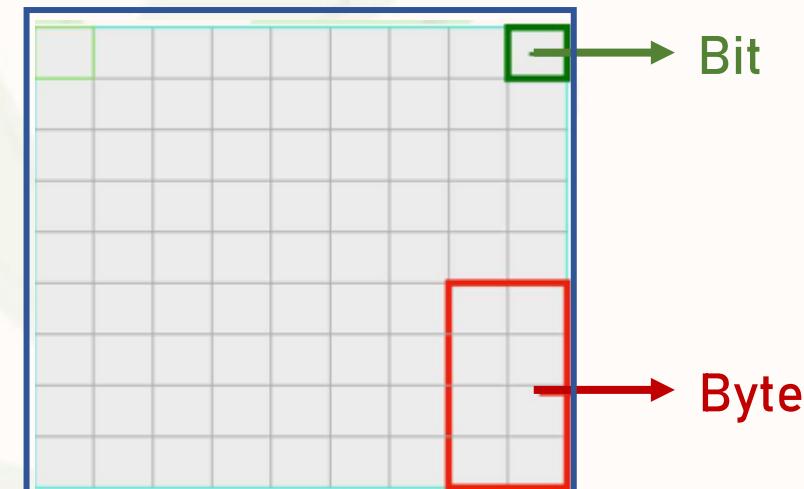
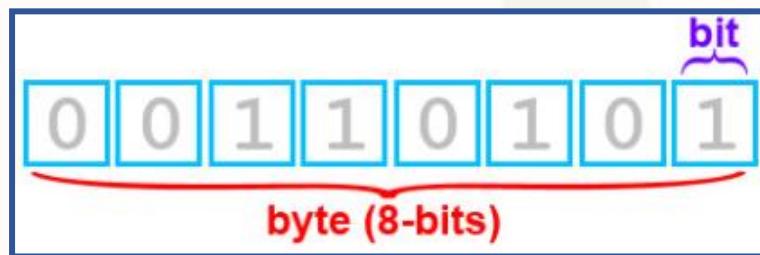


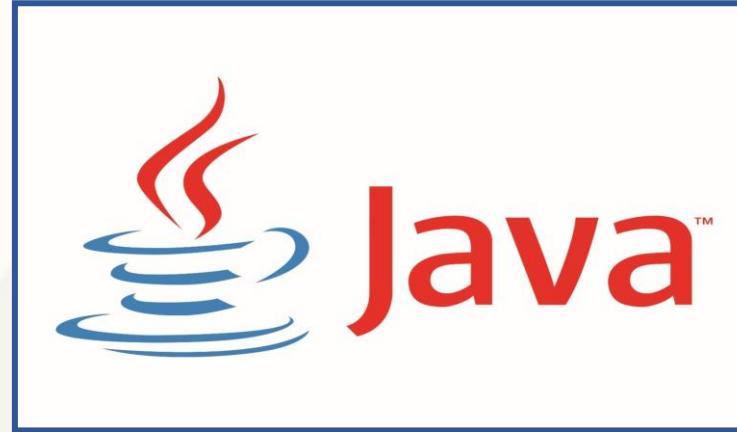


Bit

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

Note: 8 bit =1 byte

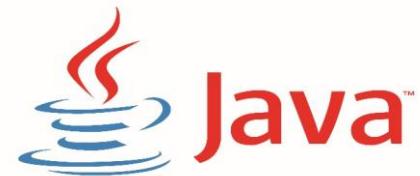




23 EKIM 2021
DERS 2

Java Giriş
Variables

Mehmet BULUTLUOZ
Elk.Elektronik Yuk.Muh.



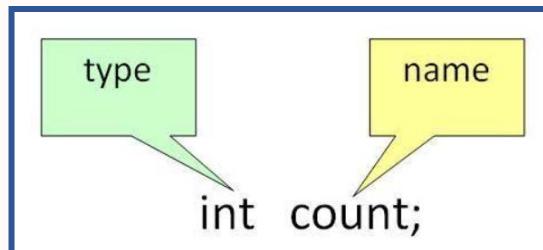
Variables (Degisken) Olusturma Declaration

Variable bellekte (**memory**) ayrılmış olan alanın (**reserved area**) adıdır.

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

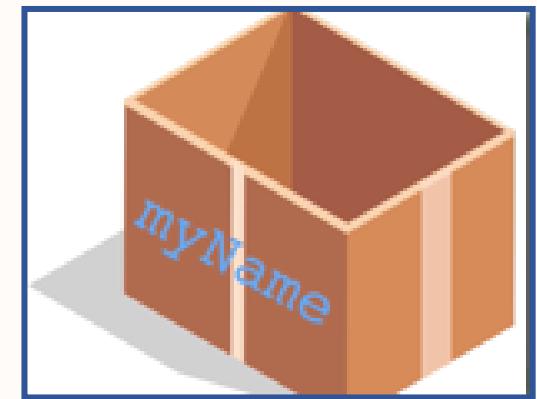
Bir değişkende saklanan değer, program yürütülürken değiştirilebilir.

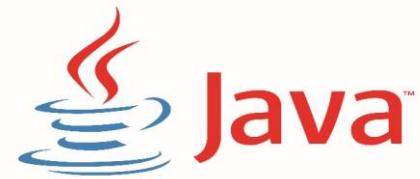
Java'da, tüm değişkenler kullanılmadan önce deklare edilmelidir (**variable declaration**)



Variable declaration için iki seyi belirtmemiz gerekiyor

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



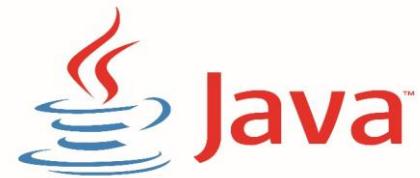


Variables Deger Atama Assignment

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";
```



Variables Deger Atama Assignment

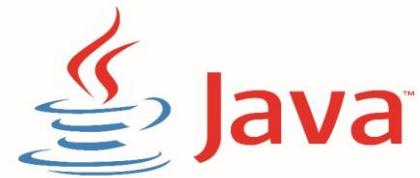
2- İlk önce declaration, daha sonra atama yapılabilir.

```
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 değeri tutar, öncekini siler.

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

```
8  
9  int level=5, yas=20, maas=10000;  
10
```

Eclips Kullanim

1- Proje olusturma

File -- New -- Project -- (Java Project) Next -- java2021FallTr -- finish

2- Package (paket) olusturma

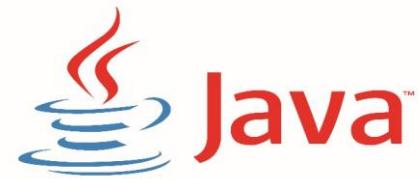
src dosyasina sag click -- New -- Package -- day01variables -- finish

3- Class olusturma

day01variables dosyasina sag click -- New -- Class -- C01_Variables01 -- finish

4- Main method olusturma

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

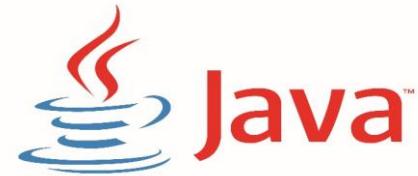


Data Types

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 gorecegimiz 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

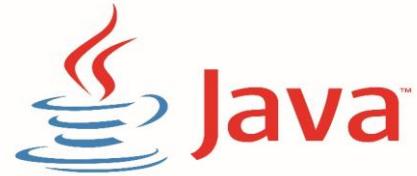
Harf, sayi veya simbol bakilmaksizin sadece 1 karakter kullanacak variable'larda kullanilir

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

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

```
char cymbol = '#';
```

Note: char degerlerini single quote arasina yazilir.



Primitive Data Types

3) **byte** Data Type: -128 den 127'e (dahil) tamsayilar icin kullanilabilir. Hafizada **8 bit** kullanir

```
byte age = 73;
```

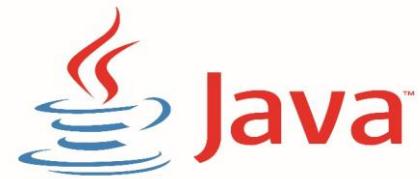
4) **short** Data Type: -32.768 den 32.767'e (dahil) tamsayilar icin kullanilabilir. Hafizada **16 bit** kullanir

```
short koyNufusu = 27,324;
```

5) **int** Data Type: -2.147.483.648 den 2.147.483.647'e (dahil) tamsayilar icin kullanilabilir. Hafizada **32 bit** kullanir

```
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) tamsayilar icin kullanilabilir. Hafizada **64 bit** kullanir



Primitive Data Types

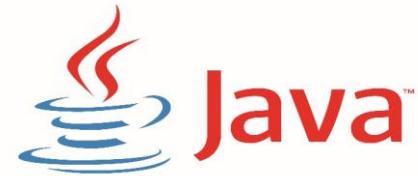
7) **float** Data Type: Kucuk ondalik sayilar icin kullanilabilir. Hafizada **64 bit** kullanir

```
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

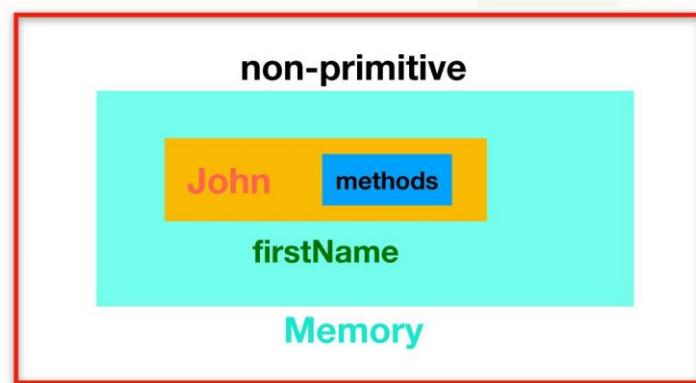
```
double doubleVar2 = -2.1234567907800000000123
```



Non-Primitive Data Type

String Data Type:

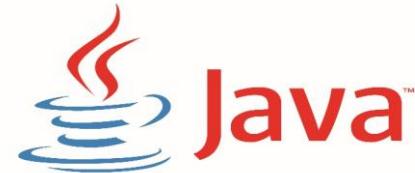
String pes pese dizilmis char'lardan olusur. Kelimeler, cumleler, matematiksel islem yapilmayacak sayisal degerler de String olarak tanimlanabilir



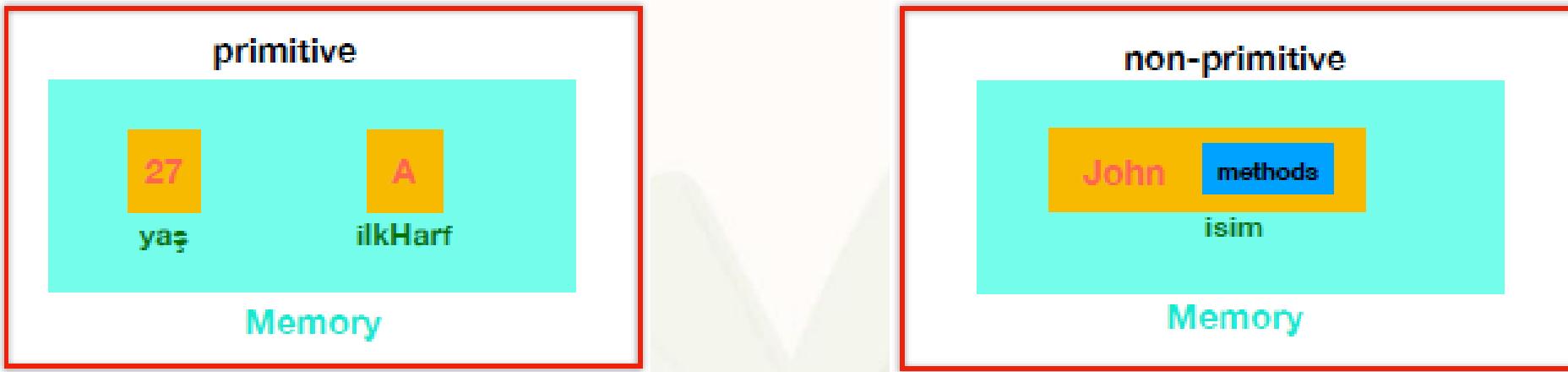
```
String okulAdi = "Yildiz Koleji, Cankaya Ankara #";  
String telNo      = "5321234567";  
String ilkHarf    = "A";
```

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

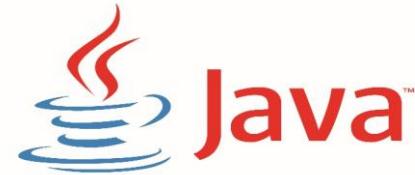
Note: Baska non-primitive data type'lar da var, daha sonra ogrenecegiz.



Primitive VS Non-Primitive Data Types

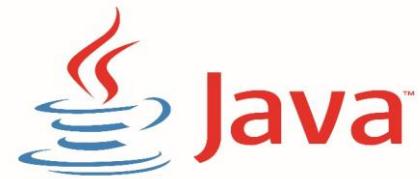


- 1) Primitive'ler sadece value icerir, non-primitive'ler value ve methodlar icerir.
- 2) Primitive'ler kucuk harf ile, non-primitive'ler buyuk harf ile baslar.
- 3) Primitive'leri Java olusturur biz primitive data turu olusturamayiz.
Non-primitive'leri biz de olusturabiliriz, Java da olusturabilir.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



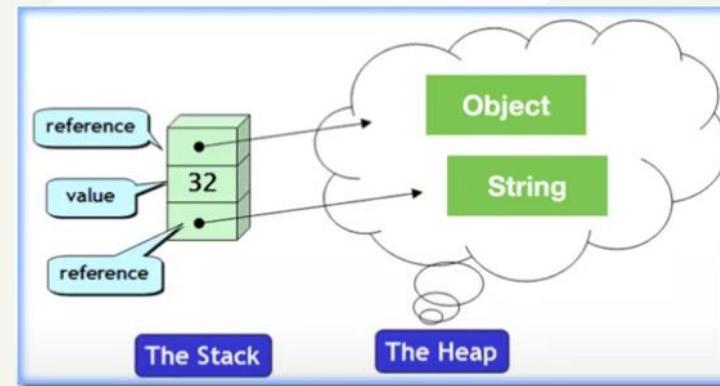
Variable ve Method'lar Nasıl Adlandırılır

1. Java variable isimleri **case sensitive** (Buyuk kucuk harfe duyarlidir)dir.
“money”, “Money” veya “MONEY” birbirinden farklidir
2. Java variable isimleri “harf”, “\$” veya “_” ile baslamlidir.
Fakat “\$” ve “_” ile baslamak tavsiye edilmez.
3. Java variable isimlerinde, ilk harften sonra sayi, “\$” ve “_” kullanilabilir.
4. Variable isimleri icin Java'ya ozel terimler (key word) kullanilamaz. (int, for, if, import vb.).
5. Variable isimleri kucuk harflerle baslar, camel case kullanilir
6. Variable isimleri 1'den fazla kelime iceriyorsa, ilk kelimededen sonraki her kelimenin ilk hafi buyuk harf ile baslamlidir. firstName, bigApple, ageJohnWalker gibi. Buna camelCase denir.



Memory (Hafiza) Kullanimi

Javada kullanılan iki hafıza vardır

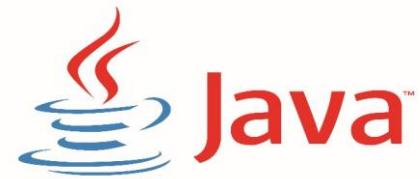


Stack => small

Heap => huge

1- **Stack Memory** : primitive data tiplerine ait değerleri ve Non-primitive datalara (Object) ait referansları(adres) barındırır

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



Memory (Hafiza) Kullanımı

reference of an Object

int



Stack

Object

byte

boolean

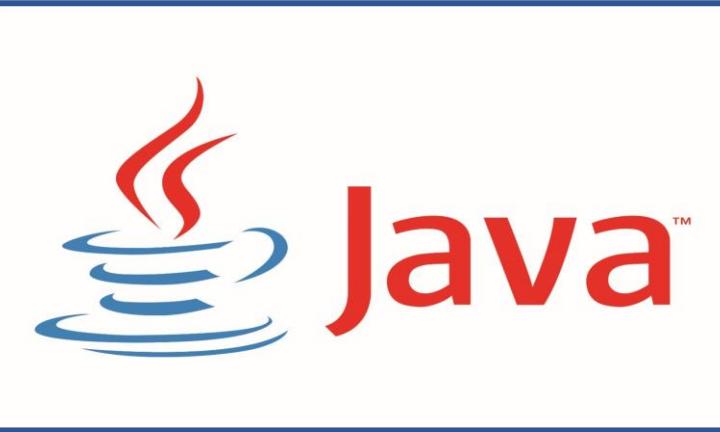
String

long

Array



Heap



24 EKİM 2021
DERS 3

Scanner
Data Casting

Mehmet BULUTLU0Z
Elk.Elektronik Muh.

Onceki Dersten Aklimizda Kalanlar

- 1) Data : bilgisayarda islenen(processed) veya depolanan (stored) herseye data denir
- 2) Compile : derleme, bilgiyi islemek demektir. Java'nin kullanicinin yazdigı kodları bilgisayarın anlayacağı binary kodlara cevirmesidir.
- 3) OOP Concept : Object oriented Programming , java'da Class'lari kullanarak objeler üretiriz, sonra bu objeleri kullanarak application'lar üretiriz. (Lego gibi)
- 4) Class'larda neler bulunur? : fields (pasif,hareketsiz) , methods (dinamik,hareketli)
- 5) Variable nedir ? : Data değerlerini saklamak (store) için kullanılan container'dır.
- 6) Variable neden kullanılır ? : Datayı program içerisinde kullanabilmek için variable'lara atarız. Programımız içinde ne zaman variable ismini yazsak, Java bize o variable'a atanmış son değeri getirir.

Onceki Dersten Aklimizda Kalanlar

7) Variable nasıl olusturulur? : Variable olusturmak icin declaration (tanimlama) yapilir.
Veri turu (data type) ve variable ismini yazilmalidir

8) Variable deger atama : Assignment denir. Variable isminin karsina = isareti
konularak istenen deger assign edilir.

9) Variable declaration ve assignment icin nelere dikkat etmeliyim ?

Declaration ve assignment sirasisiyla yapilmalidir.

Once declaration,sonra assignment olmali.

Istersek tek satirda ikisini birden yapabiliriz

int sayi = 20 ;

istersek de once declare edip sonra deger atayabiliriz

int sayi;

sayi = 10;

Onceki Dersten Aklimizda Kalanlar

10) Class olusturmak icin hangi keyword'lar kullanilir ? :

```
public class ClassIsmi { Class Body }
```

11) Method olusturmak icin hangi keyword'lar kullanilir ? :

```
public String methodIsmi (parametre) { method Body}
```

12) Main method Nedir ?

Java'nin kodlari calistirmaya basladigi giris noktasidir (Entry Point).

13) Main method olusturmak icin kullanilan syntax nedir ? :

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

Onceki Dersten Aklimizda Kalanlar

14) Java'da kac cesit data type vardir ? Java'da temel iki data tipi vardir.

- primitive (ikel) data types : boolean (true /false) ,
char (tek karakter),
byte, short, int, long(tam sayilar),
float, double (ondalikli sayilar)
- non-primitive (object): String (simdilik)

15) Java'da kac cesit hafiza vardır ? :

stack- (small) primitive datalar ve non-primitive data tiplerinin reference'larinin
store edildigi hafizadir.

Heap memory (huge) : non-primitive datalarin store edildigi hafizadir.

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 bunların toplamini yazdirin

4- Bir tamsayi ve bir ondalikli variable olusturun ve bunların toplamini yazdirin

5 - char data turunde bir variable olusturun ve yazdirin

6- Bir tamsayi, bir de char degisken olusturun ve bunların toplamini yazdirin.

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

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 acknow.l.)	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)						

Kullanicidan 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 faydalı olacaktir.

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

Kullanicuya girmesini istedigimiz degerler icin aciklayici bilgi vermeliyiz.

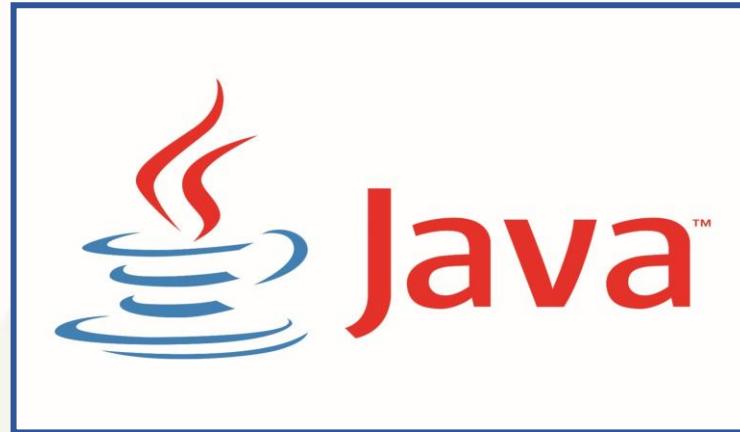
Burada aciklama olarak ne yazdirlsa kodumuz calisir, hatta birsey yazdirmasak da calisir ancak kullanici kendisinden ne istedigimizi bilmezse deger girmesi gerektigini veya ne tur bilgi girmesi gerektigini bilemez

Kullanicidan 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



25 EKİM 2021
DERS 4

Data Casting
Increment-Decrement

Mehmet BULUTLU0Z
Elk.Elektronik Muh.

Onceki Dersten Aklimizda Kalanlar

1) Scanner (Kullanicidan deger alma) : 3 adimda kullanicidan deger alabiliriz.

- Scanner scan = new Scanner(System.in);

scan bu objenin ismidir, baska isim de verebiliriz ama bunu tercih ediyoruz.

- Kullaniciya bir mesaj veriyoruz (Kullaniciya ondan ne bekledigimizi soyluyoruz)

- scan objesi ile ilgili method'u kullanarak kullanicinin girdigi degeri alip, olusturdugumuz uygun bir variable'a assign ediyoruz

- String degiskenler icin next() ve nextLine() method'lari kullanilabilir. Next() sadece bir kelime alir (ilk space'e kadar olan kisim), nextLine() ise satirin tamamini alir

2) Scanner objesi olusturdugumuzda Java itiraz edip altini kirmizi cizer. (Cunku class'imizda scanner ile ilgili bilgiler yoktur) Java.util kutuphanesinden Scanner class'ini import edince bu sikayet ortadan kalkar.

Kullanicidan Deger Alma Sorular

- Soru 1)** Kullanicidan iki tamsayi alip bu sayilarin toplam,fark ve carpimlarini yazdirin
- Soru 2)** Kullanicidan karenin bir kenar uzunlugunu alin ve karenin cevresini ve alanini hesaplayip yazdirin
- Soru 3)** Kullanicidan yaricap isteyip cemberin cevresini ve dairenin alanini hesaplayip yazdirin
- Soru 4)** Kullanicidan dikdortgenler prizmasinin uzun, kisa kenarlarini ve yuksekligini isteyip prizmanın hacmini hesaplayip yazdirin
- Soru 5)** Kullanicidan ismini ve soyismini isteyip asagidaki sekilde yazdirin
- Isminiz : Mehmet
- Soyisminiz : Bulut
- Kursumuza katiliminiz alinmistir,tesekkur ederiz
- Soru 6)** Kullanicidan ismini ve soyismini alip aralarinda bir bosluk olusturarak asagidaki sekilde yazdirin
- Isim – soyisim : Mehmet Bulutluoz
- Soru 7)** Kullanicidan ismini alip isminin bas harfini yazdirin.

Data Casting

Veri Sınıfı Değiştirme

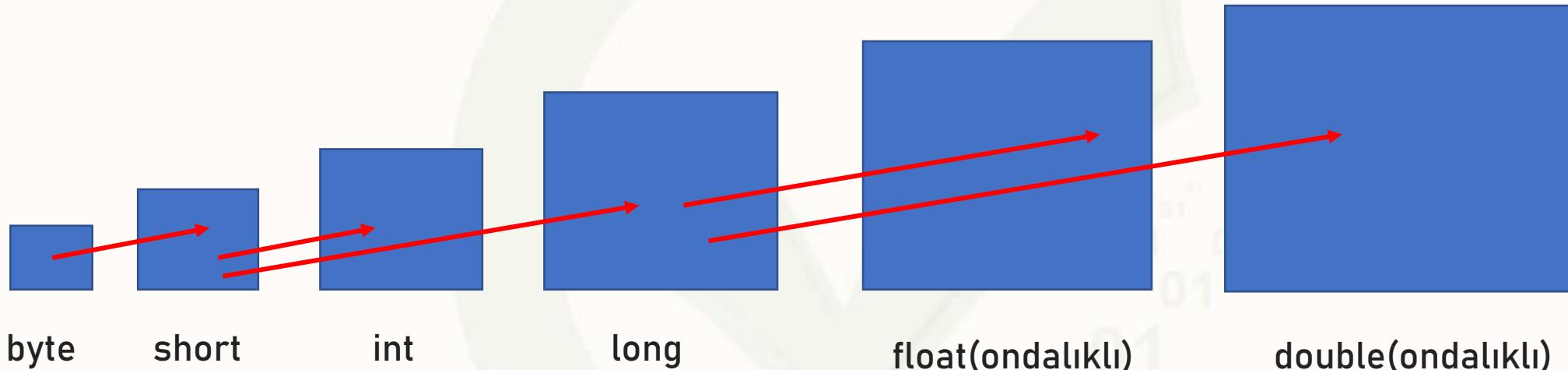
- Java'da kod yazarken bir veri tipinden diğer bir veri tipine aktarım yapmamız gerekebilir.
- Veri tiplerinde bir variable'a , olusturuldugu data tipinden farkli bir data turunden deger atanmasina Data Casting denir.
- Data casting yaparken aklımızdan cikarmamamız gereken konu data tiplerinin sınırlarıdır. Data tipinin sınırlarını aşan data casting islemlerinde hata almamamız için dikkat etmemiz gereken bazı durumlar olacaktır.
- Hatırlayacağımız şekilde Java'da sayılarla ilgili data tiplerinin sıralaması su sekildeydi

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

Data Casting

1) Auto Widening (Otomatik Genisletme)

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



Orn : **byte num1 = 12;**

short num2 = num1; // yazdırırsak 12 olarak yazdırır

int num3 = num2; // yazdırırsak 12 olarak yazdırır

float num4=num3; // yazdırırsak 12.0 olarak yazdırır

double num5=num4; // yazdırırsak 12.0 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 sayıya bolun ve sonucu yazdirin
- Soru 7)** Farklı data tipleri ile işlem 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 - - ;
```

?

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 azaltilar

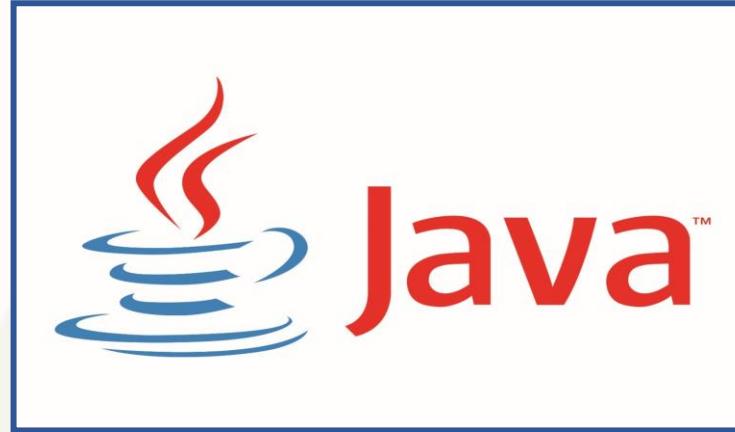
```
public static void main(String[] args) {  
    int a=15;  
    int b=++a;  
    System.out.println(b);  
}
```

Output : 16

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

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

Output : 15



26 EKİM 2021
DERS 5

**Matematiksel Operatorler
Modulus**

Mehmet BULUTLU0Z
Elk.Elektronik Muh.

Javada Matematiksel Operatorler

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

Ornek 1:

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

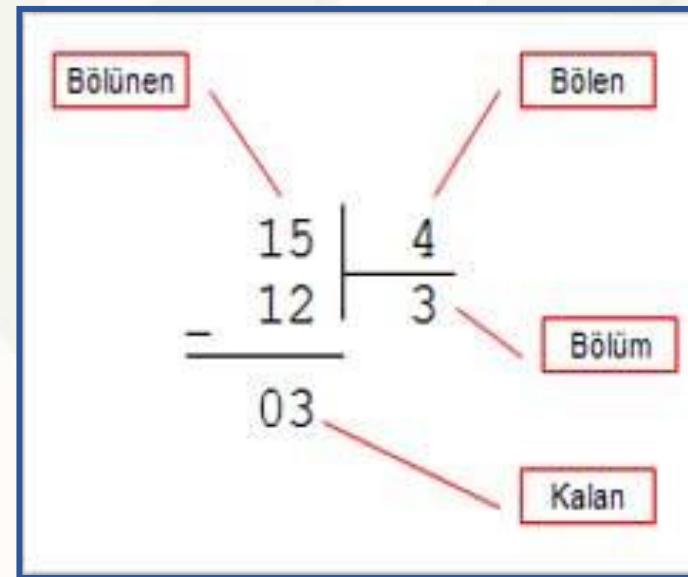
Ornek 2 :

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

Modulus %

Modulus işlemi bir bolme işleminden kalan sayiyi bize verir

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



Modulus %

Soru 1) Kullanicidan 4 basamakli bir sayı alın ve rakamlar toplamini bulup yazdırın

Ipucu 1:

Sayı % 10 => Bize son basamagi verir

$$538 \% 10 = 8$$

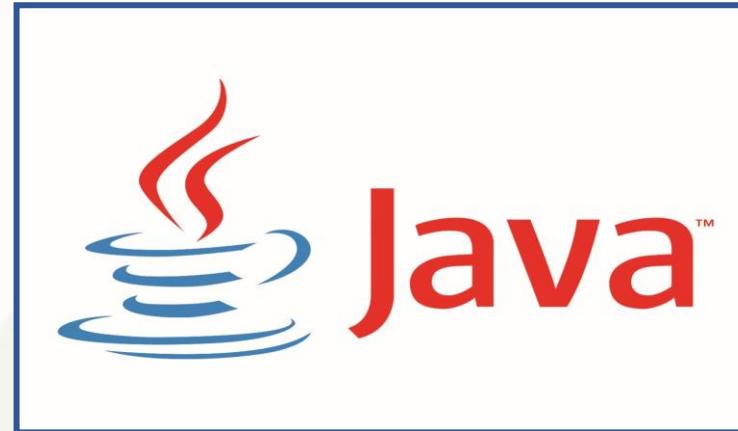
Ipucu 2:

Int Sayı /10 => Bize son basamak haric sayiyi verir

int sayı=538;

sayı = sayı / 10 =>

sayı'ya 53 degerini atar



30 EKİM 2021
DERS 6

Wrapper Class
Concatenation
Relational Operators

Mehmet BULUTLU0Z
Elk.Elektronik Muh.

Onceki Dersten Aklimizda Kalanlar

1- matematiksel operatorler : kodlarimizi yazarken dogru sekilde yazmazsak yanlis sonuclara ulasabiliriz.

“Java” + 30 + 20 => Java3020

“Java” + (30 + 20) => Java50

20+30+ “Java” => 50Java

”+20+30+ “Java” => 2030Java

A- us ve parantez : yazim sekline gore us veya parantez once olabilir

B- carpma ve bolme

C- toplama ve cikarma

Egit oncelikli islemlerde once soldaki islem yapilir

2- Modulus : bir bolme islemindeki kalan degerini bize verir

sayi % 2 ye bakariz sonuc sifirsa sayi cift, sonuc 1 ise sayi tek

sayi % 10 sayinin birler basamagini verir

sayi/10 (sayi tam sayi olmak uzere) : birler basamagini yok edip kalan kismi bize verir

627 % 10 => 7

627 / 10 => 62

Wrapper Class

Java **primitive** data turleri ile **methodlari** kullanabilmemiz icin **Wrapper class'lar**i olusturmustur.

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

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 alınarak islem yapılır. Sira String ile toplamaya geldiginde toplama yerine Concatenation uygulanır

```
String a = "Hello";
int b = 2;
int c = 3;
```

System.out.println(a+b+c); → Hello23
System.out.println(c+b+a); → 5Hello
System.out.println(a+(b+c)); → Hello5
System.out.println(a+b*c); → Hello6

Concatenation (String Datalari Birlestirme)

Soru 1) Asagida verilen variable'lari kullanarak istenen sonucları yazdırın programları yazınız.

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

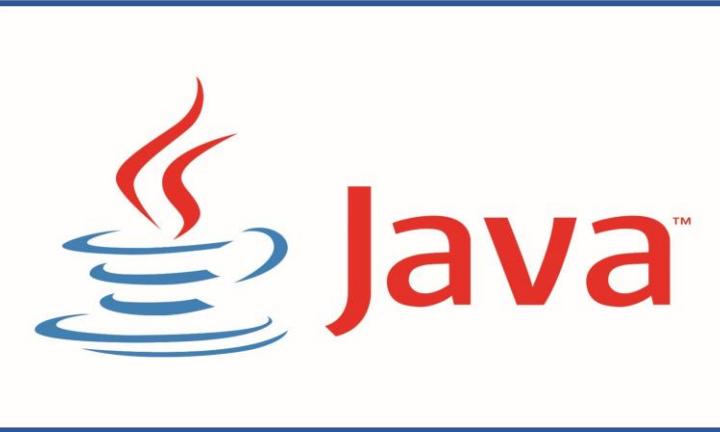
|| OR (veya) isareti

|| 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



30 EKİM 2021
DERS 7

If Statements
If Else Statements

Mehmet BULUTLU0Z
Elk.Elektronik Muh.

If Statements (If cümleleri)

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

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

If (boolean şart) {şart saglanırsa 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
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 (Sorular)

Soru 1) Kullanicidan bir tamsayi isteyin ve sayinin tek veya cift oldugunu yazdirin

Soru 2) Kullanicidan gun isimlerinden birinin ilk harfini isteyin ve o harfle baslayan gun isimlerini yazdirin

Ornek: ilkHarf=P output = “Pazar, Pazartesi veya Persembe”
ilkHarf=S output = “Sali”

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

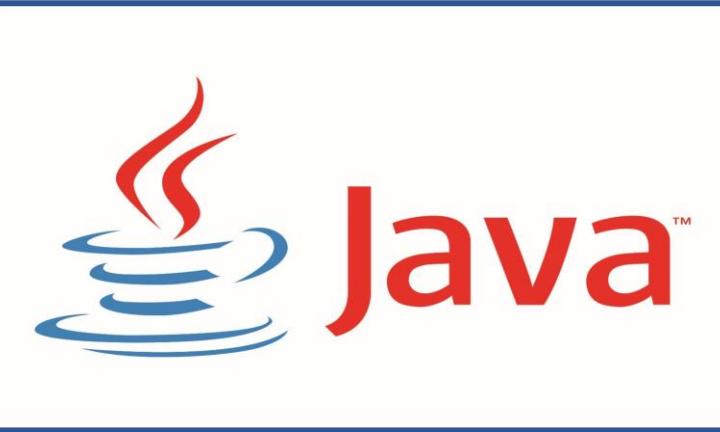
Soru 3) Kullanicidan gun ismini alin ve haftaici veya hafta sonu oldugunu yazdirin

Ornek: gun=Pazar output = “Hafta sonu”
gun=Sali output = “Hafta ici”

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

Soru 4) Kullanicidan dikdortgenin kenar uzunluklarini isteyin ve dikdortgenin kare olup olmadigini yazdirin

Soru 5) Kullanicidan bir gun alin eger gun “Cuma” ise ekranra “Muslimanlar icin kutsal gun” yazdirin. “Cumartesi” ise ekranra “Yahudiler icin kutsal gun” yazdirin. “Pazar” ise ekranra “Hiristiyanlar icin kutsal gun” yazdirin



31 EKİM 2021
DERS 8

If Else Statements

Mehmet BULUTLU0Z
Elk.Elektronik Muh.

Onceki Dersten Aklimizda Kalanlar

1- Wrapper class : Java primate data turleri ile method'lar kullanabilmemiz icin her primitive data turune bir tane non-primitive data turu hazirlamistir ve bunlara ozel olarak wrapper class denir

2- Mantiksal karsilastirmalar && ve OR

- && mukemmeliyetcidir, sonucun true olmasi icin hepsinin true olmasi gerekir, matematikteki carpma gibidir, bir tane sifir sonucu sifir yapmaya yeter, && bir tane false sonucu false yapmaya yeter

- OR ise iyimserdir, sonucun false olmasi icin tum bilesenlerin false olmasi gerekir. Matematikteki toplama gibidir, 1 tane bile 1 olursa toplam sifir olmaz, OR icin de 1 tane bile TRUE olursa sonuc false olmaz

3- = isareti assignment isaretidir ve karsilastirma yapmaz,

- Java = isareti gordugunde bir atama oldugunu anlar ve oncelikle esitligin sag tarafindaki degeri hesaplar, esitligin sag tarafindaki islem bitince, bulunan sonucu esitligin solundaki variable'a assign eder

- eger karsilastirma yapmak istersek != kullanmalıyiz. Java == gordugunde bu cift esittir isaretinin saginda ve solundaki ifadelerin esit olup olmadigina bakar, esit ise true, esit degilse false dondurur.

- eger esit olmadiklarini control etmek istiyorsak != isareti kullanilir. != isareti mantiksal bir ifadenin onune konulursa tersine cevirir. !true=>false gibi..

- String ile esitligi control etmek istedigimizde == yerine equals() kullanilir

If Else Statements

Eger hava guzel olursa piknige gideriz, yoksa evde otururuz.

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

If (boolean şart) {şart saglanırsa istenen kod} else {şart 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) Kullanicidan dikdortgenin kenar uzunluklarini isteyin ve dikdortgenin kare olup olmadigini yazdirin
- Soru 2) Kullanicidan bir karakter girmesini isteyin ve girilen karakterin harf olup olmadigini yazdirin
- Soru 3) Kullaniciya yasini sorun, eger yas 65'den kucuk ise "emekli olamazsin, calismalisin", 65'e esit veya buyukse "Emekli olabilirsin" yazdirin
- Soru 4) Kullanicidan bir ucgenin uc kenar uzunlugunu alin eger uc kenar uzunlugu birbirine esit ise ekran'a "Eskenar ucgen" yazdirin. Diger durumlarda ekran'a "Eskenar degil" yazdirin.

& 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.

If Else If ... Statements

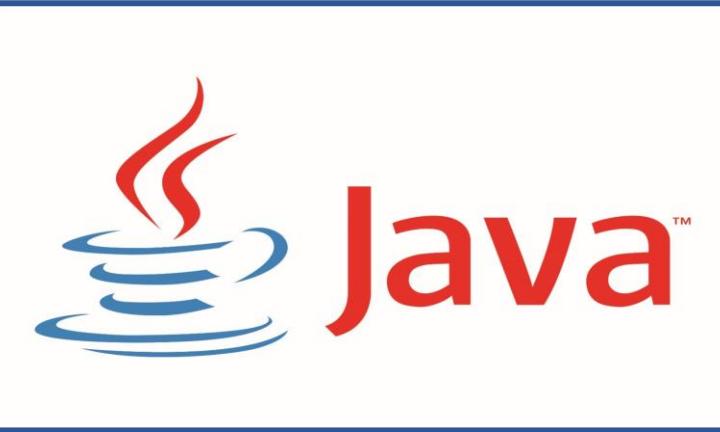
Eger soruyu biliyorsa Ali soruyu cozsun , o bilmiyorsa Veli biliyorsa Veli cozsun,
o da bilmiyorsa Ayse biliyorsa, Ayse cozsun, o da bilmiyorsa Fatma biliyorsa,
Fatma cozsun, o da bilmiyorsa kim isterse o cozsun.

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 (Sorular)

- Soru 5) Kullanicidan gun ismini yazmasini isteyin. Girilen isim gecerli bir gun ise gun isminin 1.,2. ve 3.harflerini ilk harf buyuk diger ikisi kucuk olarak yazdirin, gun ismi gecerli degilse "Gecerli gun ismi giriniz" yazdirin
- Soru 6) Kullanicidan iki sayi isteyin, sayilarin ikisi de pozitif ise sayilarin toplamini yazdirin, sayilarin ikisi de negative ise sayilarin carpimini yazdirin, sayilarin ikisi farkli isaretlere sahipse "farkli isaretlerde sayilarla islem yapamazsin" yazdirin, sayilardan sifira esit olan varsa "sifir carpmaya gore yutan elemandir" yazdirin.
- Soru 7) Kullanicidan 100 uzerinden notunu isteyin. Not'u harf sistemecevirip yazdirin.
50'den kucukse "D", 50-60 arasi "C", 60-80 arasi "B", 80'nin uzerinde ise "A"
- Soru 8) Kullanicidan maas icin bir teklif isteyin ve asagidaki degerlere gore cevap azdirin.
Teklif 80.000'in uzerinde ise "Kabul ediyorum" ,
60 – 80.000 arasinda ise "Konusabiliriz",
60.000'nin altinda ise "Maalesef Kabul edemem" yazdirin



1 KASIM 2021
DERS 9

Nested If Else Statements
Ternary

Mehmet BULUTLU0Z
Elk.Elektronik Muh.

Onceki Dersten Aklimizda Kalanlar

1- basit if cumleleri sadece bir şart ve bir sonuctan oluşur, şart gerçekleşirse if body çalışır, şartın gerçekleşmemesi durumunda if body'si çalışmaz, kod bir alttan devam eder.

Basit if cumleleri art arda yazılısa da bağımsız olarak çalışır. Örneğin 5 basit if cümlesi alt alta yazılısa hepsinin if body'si devreye girebileceği gibi, hiçbir devreye girmeyebilir

2- If else : eğer bir şart cümlesinin sonuçları sadece 2 durum ise, başka hiç bir ihtimal yoksa if else kullanırız. If içine yazılan koşul true ise if body'si , koşul false ise else body'si devreye girer.

if ve else body'leri birbirini tamamladığı için sadce ve sadece iki body'den biri çalışır

3- If Else If ... : Eğer bir soru için sınırlı sayıda durum mevcut ise (örneğin not durumuna göre A,B,C,D not alınması gibi) arkaya arkaya if else cümleleri eklenebilir.

if else if şartları arkaya arkaya geldiğinden dolayı birbirini bütünlüler, yani ilk if de şart olarak yazdığımız durumu bir daha sorgulamamıza gerek kalmaz.

4-Nested If cümleleri : bazen sorguladığımız durum birden fazla değişkene bağlı olabilir (cinsiyet ve yaşı' gore emekliliğin değişmesi gibi)

Bu durumda ilk önce bir değişkene göre if else if ... yapısı oluşturulur, sonra uygun bölmelere ikinci değişkene göre nested if else if .. Cumleleri oluşturulur.

Nested If Else Statements

Eger calisan kadinsa 60 yasindan buyuk oldugunda emekli olabilir, calisan erkekse 65 yasindan buyukse emekli olabilir

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

```
If (calisan kadinsa)
    {if (yas>60) {emekli olabilirsin} else {emekli olamazsin}}
else
    {if (yas>65) {emekli olabilirsin} else {emekli olamazsin}}
```

If Else Statements

Sorular

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

Kullanicidan 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 basamaklı bir sayı girmesini isteyin. Girdiği sayı 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 (Sorular)

Soru 9) Interview Question

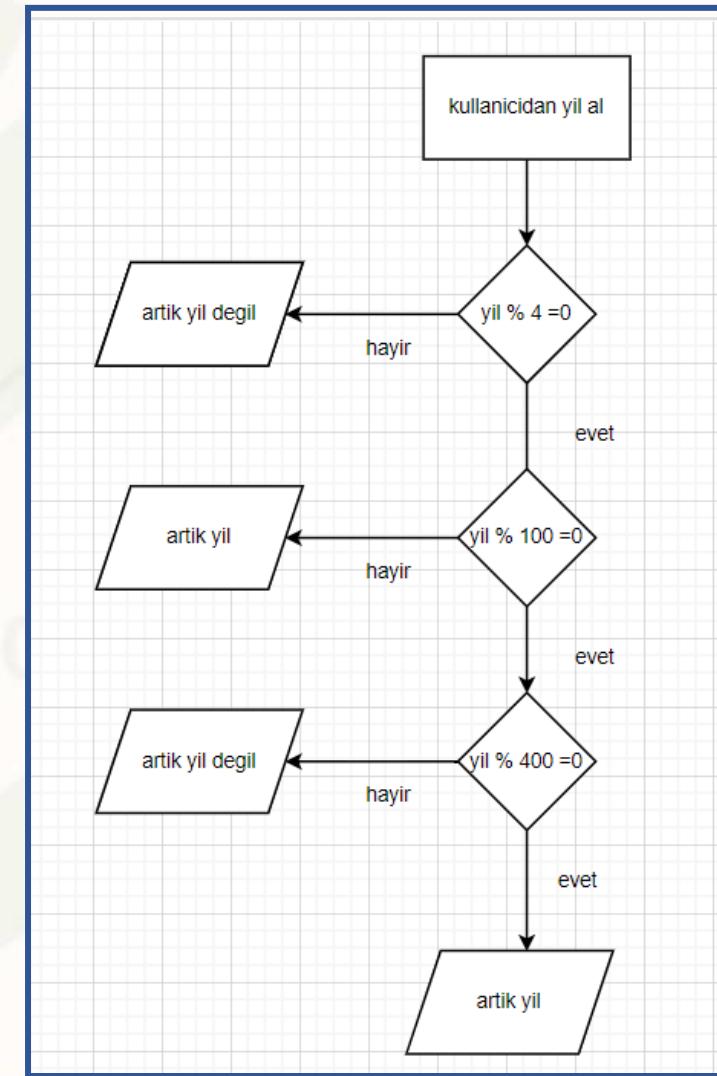
Kullanicidan artik yil olup olmadigini kontrol etmek icin yil girmesini isteyin.

Kural 1: 4 ile bolunemeyen yillar artik yil degildir

Kural 2: 4 ile bolunup 100 ile bolunemeyen yillar artik yildir

Kural 3: 4'un kati olmasina ragmen 100 ile bolunebilen yillardan sadece 400'un kati olan yillar artik yildir

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



Nested If Else Statements

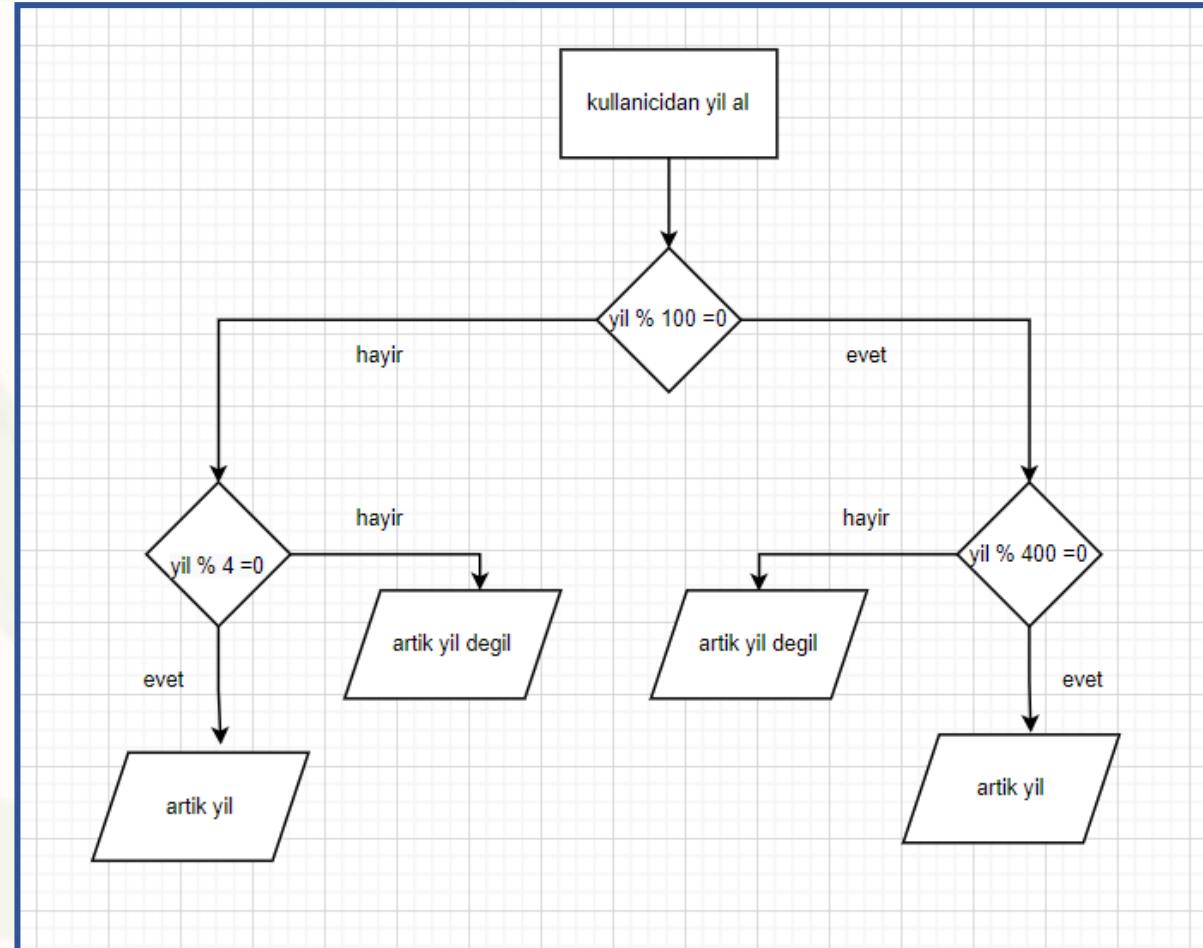
Sorular

Soru 10) Interview Question

Kullanicidan artik yil olup olmadigini kontrol etmek icin yil girmesini isteyin.

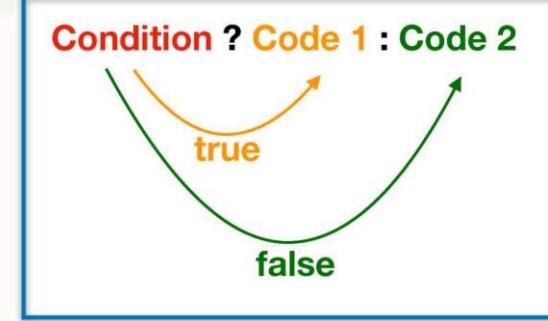
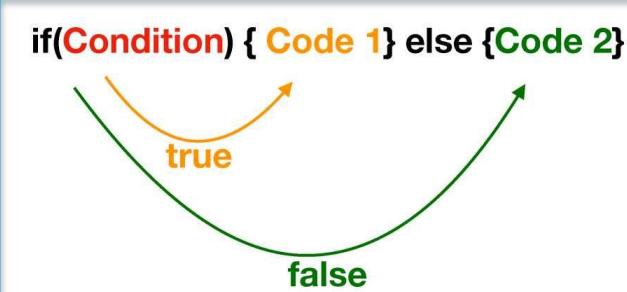
Kural 1: 4 ile bolunemeyen yillar artik yil degildir

Kural 2: 4'un kati olmasina ragmen 100 ile bolunebilen yillardan sadece 400'un kati olan yillar artik yildir



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

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;

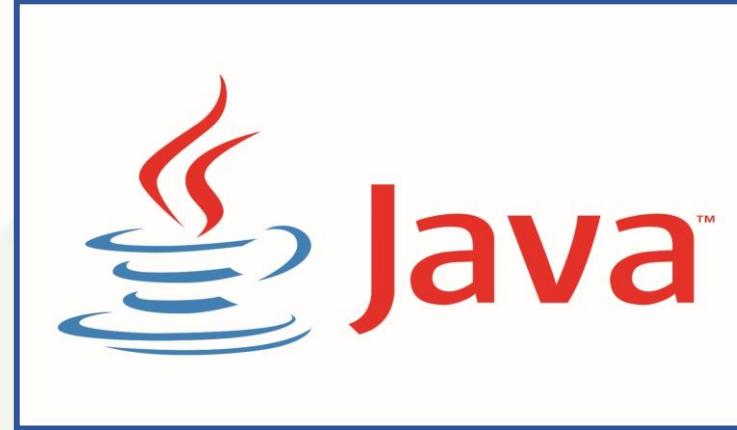
```
System.out.println((y < 91) ? 9 : 11);
```

Soru3 : int y = 1;

```
int z = 1;
```

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

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



2 KASIM 2021
DERS 10

Ternary
Switch Case

Mehmet BULUTLU0Z
Elk.Elektronik Muh.

Onceki Dersten Aklimizda Kalanlar

1) Ternary : If-Else ile yazabildigimiz bazi kodlari ternary ile daha basit olarak yazabiliriz.

(Condition) ? (condition true ise) : (condition false ise)

Condition true veya false oldugunda calisacak kisim basit sonuc veya islemler olmalıdır.

Her iki durumda da java bize sonuc dondurdugu icin ya ternary'i uygun bir variable'a atamali veya direkt sysout icine yazarak ekraninda cikti olarak gormeliyiz

eger true veya false oldugunda donecek sonuclarin data turleri farkli ise atama yapamayacagimiz icin sadece sysout opsiyonu kullanilabilir

Ternary Operator

Sorular

Soru1) Kullanicidan iki sayı alın ve büyük olmayan sayiyi yazdırın

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

Soru3) Kullanicidan bir sayı alın ve sayının mutlak değerini yazdırın

Soru4) Kullanicidan 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 : Kullanicidan bir tamsayi alin ve sayi 10'dan kucukse "Rakam" , 100'den kucukse "iki basamaklı sayı" degilse "uc basamaklı veya daha buyuk sayı" yazdirin

Soru2 : Kullanicidan bir harf isteyin kucuk harf ise consola "Kucuk Harf" , buyuk harfse consola "Buyuk Harf" yoksa "girdiginiz karakter harf degil" yazdirin.

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

Soru5) Kullanicidan 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 şart sayısı çok olduğunda switch Statement kullanılır.

```
public static void main(String[] args) {
    int sayı = 3;

    switch(sayı) {
        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 calistirmak 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 Sorular

Soru1 : Kullanicidan haftanin kacinci gunu oldugunu sorun ve gun ismini yazdirin

Soru2 : Kullanicidan kacinci ay oldugunu sorun ve ay ismini yazdirin

Soru3 : Kullanicidan bir sayi girmesini isteyin

Girilen sayi

10 ise “iki basamakli en kucuk sayı”

100 ise “uc basamakli en kucuk sayı”

1000 ise “dort basamakli en kucuk sayı”

diger durumlarda “Girdigin sayiyi degistir” yazdirin

Soru4 : Kullanicidan SDET kisaltmasindaki harflerden birini yazmasini isteyin.

Kullanici S girerse “Software”

D girerse “Developer”

E girerse “Engineer”

T girerse “In Testing” yazdirin

Soru5 : Kullanicidan 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) işaretleri 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 kullanilirsa 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

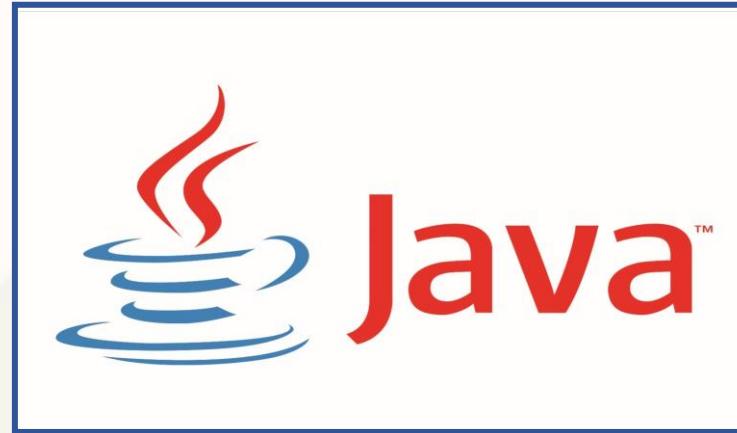
NOT : toLowerCase(Locale locale)

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



6 KASIM 2021
DERS 11

String Manipulations

Mehmet BULUTLUOZ
Elk.Elektronik Yuk.Muh.

Onceki Dersten Aklimizda Kalanlar

- 1) Nested ternary : bazen kodumuz basit olmakla beraber iki tercihten fazlasini gerektirebilir. Bu durumda tek ternary ile kodu yazmamız mümkün olmaz, ilk ternary de false kismina yeni bir ternary yazarak ihtimalleri 3'e çıkarabiliriz.

Eger true yerine de nested ternary yazarsak ihtimalleri 4 'e çıkarabiliriz.

Ancak ternary'yi tercih etme sebebimiz basit yapida olmasidir, karmaşık durumlarda ternary yerine if-else tercih edilmelidir

- 2) Switch -case : Eger if-else if ... cozebilecegimiz bir sorunda tercih sayisi çok ise if-else 'leri cogaltmak kodumuzu karmaşık hale getirebilir.

Bu durumda if-else yerine switch-case tercih edilir.

Switch parantezinin içine Boolean bir ifade yazamyız, aynı şekilde parantez içine long, double ve float da yazılmaz.

Switch parantezinin içine yazdigim degree gore Java uyan case'e gider ve break gorene kadar kodu calistirmaya devam eder.

Eger birden fazla case icin calisacak kod aynı ise break yazmadan case'leri alt alta siralayabiliriz.

Eger yazilan case'ler disinda kalan tum durumlar icin ortak bir kod varsa, default : yazip ilgili kodumuzu ekleyebiliriz.

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)'larine 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 farkliliği disinda herhangi bir karakter farkliliği 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

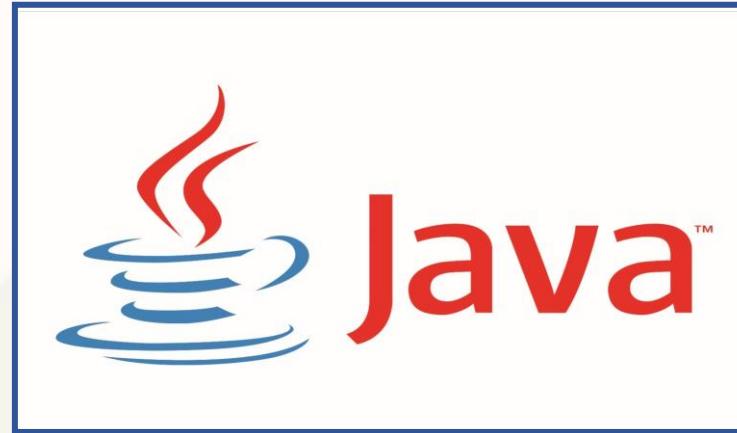
8-indexOf()

Verilen String'de istenen karakterin kullanildigi ilk index'i dondurur.

- 1) char'in index'i sorgulanabilir
- 2) Parametre String olabilir
- 3) Olmayan karakter sorgulanirsa
- 4) Parametre kelime olabilir
- 5) Belli bir index'ten sonrasi 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



6 KASIM 2021
DERS 12

String Manipulations

Mehmet BULUTLUOZ
Elk.Elektronik Yuk.Muh.

String Manipulation / Methods

indexOf() Sorular

Soru 1) Kullanicidan bir cümle ve bir harf isteyin, harfin cümlede var olup olmadığını yazdirin

Soru 2) Kullanicidan bir cümle ve bir kelime isteyin, kelimenin cümledeki kullanımına bakarak aşağıdaki 3 cümleden uygun olanı yazdirin

- Girilen kelime cümlede kullanilmamis.
- Girilen kelime cümlede 1 kere kullanilmis.
- Girilen kelime cümlede 1'den fazla kullanilmis.

String Manipulation / Methods

9-lastIndexOf()

Verilen String'de istenen karakterin kullanildigi son index'i dondurur.

- 1) char'in son index'i sorgulanabilir
- 2) Parametre String olabilir
- 3) Olmayan karakter sorgulanirsa
- 4) Parametre kelime olabilir
- 5) Belli bir index'ten oncesi sorgulanabilir

```
String str= "Calisirsaniz, Java ogrenmek cok kolay";  
  
System.out.println(str.lastIndexOf('a'));  
System.out.println(str.lastIndexOf("a"));  
System.out.println(str.lastIndexOf("t"));  
System.out.println(str.lastIndexOf("Java"));  
System.out.println(str.lastIndexOf('a',11));
```

: 35 : 35 : -1 : 14 : 8

String Manipulation / Methods

lastIndexOf() Sorular

Soru 1) Kullanicidan bir cümle ve bir harf isteyin, harfin cümlede var olup olmadığını yazdirin

Soru 2) Kullanicidan bir cümle ve bir kelime isteyin, kelimenin cümledeki kullanımına bakarak asagidaki 3 cümleden uygun olanı yazdirin

- Girilen kelime cümlede kullanilmamis.
- Girilen kelime cümlede 1 kere kullanilmis.
- Girilen kelime cümlede 1'den fazla kullanilmis.

String Manipulation / Methods

10-contains()

Verilen String'in istenen karakter(ler)i icerip icermedigini kontrol eder. Iceriyorsa TRUE, icermiyorsa FALSE dondurur.

- 1) Parametre String olmalidir
- 2) Olmayan karakter sorgulanirsa
- 3) Parametre kelime olabilir

```
public static void main(String[] args) {  
  
    String str= "Calisirsaniz, Java ogrenmek cok kolay";  
  
    System.out.println(str.contains("a"));           true  
  
    System.out.println(str.contains("t"));           false  
  
    System.out.println(str.contains("Java"));         true
```

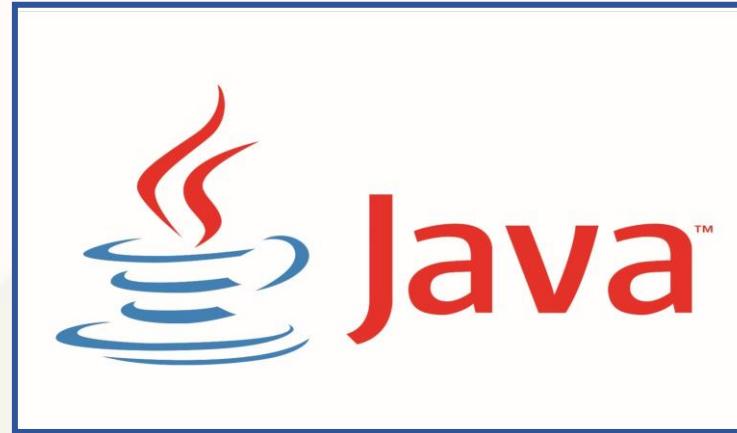
NOT contains() methodu char icin kullanilamaz, String kullanmak zorunludur.

String Manipulation / Methods

contains() sorular

Soru 1) Kullanicidan email adresini girmesini isteyin, mail @gmail.com icermiyorsa “lutfen gmail adresi giriniz”, @gmail.com ile bitiyorsa “Email adresiniz kaydedildi ” , @gmail.com ile bitmiyorsa lutfen yazimi kontol edin yazdirin

Soru 2) Kullanicidan bir cümle isteyin. Cümle “buyuk” kelimesi iceriyorsa tum cumleyi buyuk harf olarak, “kucuk” kelimesi iceriyorsa tum cumleyi kucuk harf olarak yazdirin, iki kelimeyi de icermiyorsa “Cumle kucuk yada buyuk kelimesi icermiyor” yazdirin.



7 KASIM 2021
DERS 13

String Manipulations

Mehmet BULUTLUOZ
Elk.Elektronik Yuk.Muh.

Onceki Dersten Aklimizda Kalanlar

- 1) str.indexOf(str2) : str icinde str2 nin ilk kullanildigi yerin index'ini verir.
str2 yerine char da kullanilabilir.
eger str.indexOf(str2, int a) a.index'ten sonra str2 arar
eger bir str2 icin ilk kullanilan index'i bulduysak ve ikinci bir kullanım var mi diye control etmek istiyorsak str.indexOf(str2, ilk index+1) seklinde kullanabiliriz, boylece java ilk kullanım index'inden sonra ikinci bir kullanım var mi diye kontrol etmis olur
- 2) str1.equals(str2) str1 ve str2'nin iceriginin ayni olup olmadigini kontrol eder, ikisinin içerikleri esit ise true, esit degilse false doner.
== ile equals() nun temel farki , == hem degree hem referansa bakar dolayisiyla ancak ayni objeyi kendisiyle karsilastirdigimizda true doner, equal() ise referans'a degil sadece icerige bakar, case sensitive olarak ayni icerige sahiplerse true doner
- 3) equalsIgnoreCase() kullanildiginda ise case sensitive olmadan içeriklerin ayni olup olmadigina bakar
- 4) str1.lastIndexOf(str2) aramaya sondan baslayarak, basa dogru gelir ve buldugu ilk str2'nin index'ini verir, digger ozellikleri indexOf(0'a benzer
- 5) str1.contains(str2) : str1'in icinde str2 varsa true, yoksa false doner. char kullanilamaz
*** indexOf ve lastIndexOf bize int index dondurur, equals() ve contains() ise Boolean bir sonuc dondurur

String Manipulation / Methods

11-endsWith()

Verilen String'in istenen karakter(ler) ile bitip bitmediğini kontrol eder. İstenen karakter(ler) ile bitiyorsa TRUE, yoksa FALSE dondurur.

- 1) Parametre String olmalıdır
- 2) Yanlış karakter sorgulanırsa
- 3) Parametre kelime olabilir

```
String str= "Calisirsaniz, Java ogrenmek cok kolay";  
  
System.out.println(str.endsWith("y"));  
  
System.out.println(str.endsWith("t"));  
  
System.out.println(str.endsWith("olay"));
```

true
false
true

String Manipulation / Methods

12-startsWith()

Verilen String'in istenen karakter(ler) ile baslayip baslamadigini kontrol eder. Istenen karakter(ler) ile basliyorsa TRUE, yoksa FALSE dondurur.

- 1) Parametre String olmalidir
- 2) Parametre kelime olabilir
- 3) Belirli karakterden sonrasi olabilir

```
String str= "Calisirsaniz, Java ogrenmek cok kolay";  
  
System.out.println(str.startsWith("C")); true  
  
System.out.println(str.startsWith("Calis")); true  
  
System.out.println(str.startsWith("s",4)); true  
  
System.out.println(str.startsWith("Java",14)); true
```

String Manipulation / Methods

13-isEmpty()

Verilen String'in uzunluğu 0(sıfır) ise (Hicbir karakter içermiyorsa) TRUE, yoksa FALSE döndürür.

```
String str= "Calisirsaniz, Java ogrenmek cok kolay";  
  
System.out.println(str.isEmpty());  
  
String str2="";  
  
System.out.println(str2.isEmpty());  
  
String str3=null;  
  
System.out.println(str3.isEmpty());
```

false

true

Hata verir

```
Exception in thread "main" java.lang.NullPointerException  
at _00_anlik.asd.main(asd.java:19)
```

String Manipulation / Methods

14- replace()

Verilen String'deki istenen karakter(ler)i istenen yeni karakter(ler) ile degistirir.

```
String str= "Java ogrenmek cok kolay";  
  
System.out.println(str.replace("a", "x"));  
  
System.out.println(str.replace("Java", "x"));  
  
System.out.println(str.replace("a", "xxx"));  
  
System.out.println(str.replace("a", ""));  
  
System.out.println(str.replace('a', 'x'));
```

Jxvx ogrenmek cok kolxy
x ogrenmek cok kolay
Jxxxxxx ogrenmek cok kolxxxxy
Jv ogrenmek cok koly
Jxvx ogrenmek cok kolxy

NOT : replace() methodu char icin de kullanilabilir

String Manipulation / Methods

15- replaceAll()

replace() methodu ile benzer olarak verilen String'deki istenene karakter(ler)i istenen yeni karakter(ler) ile degistirir. Aralarindaki farklar

- replace() methodunda char kullanilabilir, replaceAll()'da char kullanilamaz
- replaceAll() methodunda Regular Expressions kullanilabilir

\s : bosluk (space)

\S : bosluk disindaki tum karakterler

\w : harfler ve rakamlar (a-z , A-Z, 0-9)

\W : harfler ve rakamlar disindaki tum karakterler

\d : rakamlar (0-9)

\D : rakamlar disindaki tum karakterler

String Manipulation / Methods

replaceAll()

```
public static void main(String[] args) {  
  
    String str= "Java'da rakamlar 1234567890";  
  
    System.out.println(str.replaceAll("a", "*"));  
  
    System.out.println(str.replaceAll("\s", "*"));  
  
    System.out.println(str.replaceAll("\S", "*"));  
  
    System.out.println(str.replaceAll("\w", "*"));  
  
    System.out.println(str.replaceAll("\W", "*"));  
  
    System.out.println(str.replaceAll("\d", "*"));  
  
    System.out.println(str.replaceAll("\D", "*"));  
}
```

J*v*d* r*k*ml*r 123456789
Java'da*rakamlar*1234567890
***** ***** *****
***** ***** *****
Java*da*rakamlar*1234567890
Java'da rakamlar *****
*****1234567890

String Manipulation / Methods

16- replaceFirst()

Verilen String'deki istenen karakter(ler)in ilkini, istenen yeni karakter(ler) ile degistirir

```
public static void main(String[] args) {  
  
    String str= "Java'da rakamlar 1234567890";  
  
    System.out.println(str.replaceFirst("a", "*"));  
  
    System.out.println(str.replaceFirst("lar", "*"));  
  
    System.out.println(str.replaceFirst("\s", "*"));  
  
    System.out.println(str.replaceFirst("\D", "*"));
```

J*va'da rakamlar 1234567890
Java'da rakam* 1234567890
Java'da*rakamlar 1234567890
*ava'da rakamlar 1234567890

String Manipulation / Methods

17- substring()

Index kullanarak verilen String'in istenen parcasini almamizi saglar.

- Parametre olarak 1 sayi girilirse, girilen index'den String'in sonuna kadar bolumu
- Parametre olarak 2 sayi girilirse, girilen 1.sayidaki indexden (inclusive) baslayip, 2.sayiya kadar (exclusive) karakteri bize dondurur

```
public static void main(String[] args) {  
  
    String str= "Java OOP konsepti kullanir";  
  
    System.out.println(str.substring(0));  
  
    System.out.println(str.substring(10));  
  
    System.out.println(str.substring(26));  
  
    System.out.println(str.substring(29));
```

Java OOP konsepti kullanir

onsepti kullanir

Hata verir

```
Exception in thread "main" java.lang.StringIndexOutOfBoundsException: String index out of range: -3  
at java.lang.String.substring(Unknown Source)  
at _00_anlik.asd.main(asd.java:17)
```

String Manipulation / Methods

substring()

```
public static void main(String[] args) {  
  
    String str= "Java OOP konsepti kullanır";  
  
    System.out.println(str.substring(5,11));  
  
    System.out.println(str.substring(3,4));  
  
    System.out.println(str.substring(8,8));  
  
    System.out.println(str.substring(8,2));
```

OOP ko

a

Hata verir

```
Exception in thread "main" java.lang.StringIndexOutOfBoundsException: String index out of range: -6  
at java.lang.String.substring(Unknown Source)  
at _00_anlik.asd.main(asd.java:17)
```

Not : Java'da iki tur hata mesaji aliriz

- 1- Compile Time Error (CTE) : Kodumuzu yazarken kod altinin kirmizi çizgi olmasi
- 2- Run Time Error (RTE) : Kod calistirildiginda (Execute) karsilastigimiz hatalar

String Manipulation / Methods

18- trim()

Istedigimiz String'in basinda veya sonunda var olan bosluk / "space" leri temizler

```
String str = " Java ogrenmek cok guzel. ";
System.out.println(str);
System.out.println(str.length());
System.out.println(str.trim());
System.out.println(str.trim().length());
```

| Java ogrenmek cok guzel. |

28

|Java ogrenmek cok guzel.|

24

String Manipulation / Methods

Soru 1) String methodlarini kullanarak “ Java ogrenmek123 Cok guzel@ ” String'ini “Java ogrenmek cok guzel.” sekline getirin.

Soru 2) String seklinde verlen asagidaki fiyatlarin toplamini bulunuz

String str1 = “\$13.99”

String str2 = “\$10.55”

ipucu : Double.parseDouble() methodunu kullanabilirsiniz.

Soru 3) Kullanicidan isim isteyin. Eger

- isim “a” harfi iceriyorsa “Girdiginiz isim a harfi iceriyor”
- isim “Z” harfi iceriyorsa “Girdiginiz isim Z harfi iceriyor”
- ikisi de yoksa “Girdiginiz isim a veya Z harfi icermiyor” yazdirin

Soru 4) Kullanicidan isim ve soyismini isteyin ve hangisinin daha uzun oldugunu yazdirin.

Soru 5) Kullanicidan 4 harfli bir kelime isteyin ve girilen kelimeyi tersten yazdirin.

String Manipulation / Methods

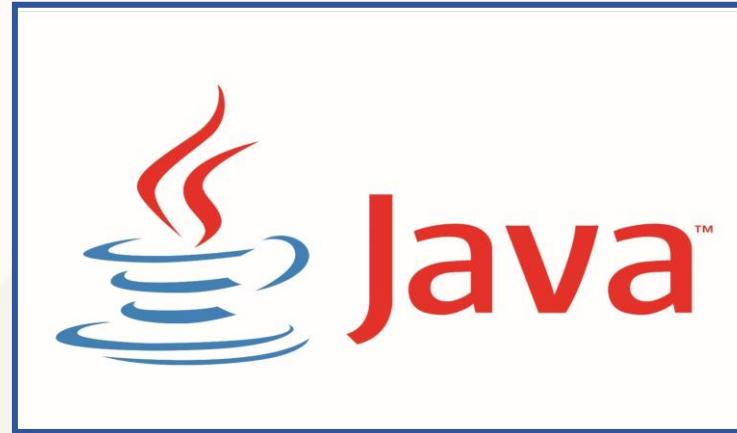
Soru 6) Kullanicidan bir sifre girmesini isteyin. Asagidaki şartlari sagliyorsa "Sifre basari ile tanimlandi", şartlari saglamazsa "Islem basarisiz,Lutfen yeni bir sifre girin" yazdirin

- Ilk harf buyuk harf olmali
- Son harf kucuk harf olmali
- Sifre bosluk icermemeli
- Sifre uzunlugu en az 8 karakter olmali

Soru 7) Kullanicidan ismini, soyismini ve kredi karti bilgisini isteyin ve asagidaki gibi yazdirin

isim-soyisim : M***** B*****

kart no : **** * **** * 1234



8 KASIM 2021
DERS 14

Method Creation
Method Call

Mehmet BULUTLUOZ
Elk.Elektronik Yuk.Muh.

Socrative Quiz

- 1) <https://www.socrative.com/> adresine gidin
- 2) **Login** butonuna basin
- 3) **Student Login** butonuna basin
- 4) Room Name **BULUTLUOZ** yazın
- 5) Isminizi yazın
- 6) **Done** butonuna basin

Sure : 15 Dakika

Method Creation

Method Olusturma

Method Olusutururken Kullanilan Keyword'ler Nelerdir?

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

- 1 **public** : Access Modifier (Erisim duzenleyici):method'a kimlerin erisebilecegini belirler
protected : Sadece icinde bulundugu package ve child class'lardan kullanilir
default : Sadece icinde oldugu package
private: Sadece bulundugu class'da kullanilabilir
- 2 **Int** : Return Type, methodun ne urettigini ve bize dondurdugunu belirtir
- 3 **myFirstMethod** :Olusturdugumuz method'un ismidir. Isim mutlaka kucuk harfle baslar, birden fazla kelimededen 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) : { } arasında kalan kodlarimizi yazdigimiz bolumdur

Method Creation

Method Olusturma

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

1 Access Modifier (Erisim duzenleyici):

public : methoda'a kimlerin erisebilecegini belirler

protected : Sadece icinde bulundugu package ve child class'lardan kullanilir

default : Sadece icinde bulundugu paket(package)'den kullanilir

private: Sadece bulundugu class'da kullanilabilir

Access Levels				
Modifier	Class	Package	Subclass	World
public	Y	Y	Y	Y
protected	Y	Y	Y	N
no modifier	Y	Y	N	N
private	Y	N	N	N

Method Creation

Method Oluşturma

2 static (İlleride detaylı anlatılacak)

Bir method oluştururken **static** kelimesinin kullanılması mecburi degildir.

Main method'umuz static oldugu icin main method'dan cagiracagimiz tum method'lari static yapmamız gereklidir

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

Method Creation

Method Oluşturma

- 3 **int (Return Type)** : methodun ne urettigini ve bize ne dondurdugunu belirtir.
- Return Type, primitive veya non-primitive tum data turlerinden olabilir
 - Eger method bir sey dondurmeyecekse (ornegin, sadece bir sey hesaplayip yazdiracaksa) return type olarak **void** secilir
 - Return Type olarak void disinda bir sey yazdiysak, methodun sonunda mutlaka **return** keyword kullanilmalidir
 - Return keyword'den sonra return type'a uygun bir **deger veya variable** yazilmalidir.
 - Return type'a sahip methodlar cagrildiklari satira, return keyword'den sonra yazilan deger veya variable'i dondururler.

```
public static void main(String[] args) {  
    int sonuc= topla(15,24);  
  
}  
  
public static int topla(int num1, int num2) {  
  
    return num1 + num2;  
}
```

Method Creation Method Olusturma

4 myFirstMethod :Olusturdugumuz method'un ismidir. Isim mutlaka kucuk harfle baslar, birden fazla kelimedenden olusursa sonraki kelimelerin ilk harfleri buyuk harf yazilir (Camel Case)

5 () parantez : Methodlarda isimden sonra parantez kullanilir ve gerektiginde parantez icinde parametre yazilir.

***** Eger bir Class'da ayni isme sahip birden fazla method olusturmamiz gerekirse parametreleri farkli yapmamiz gereklidir (Overloading)**

Method Creation

Method Olusturma

6 Body (Method Body) : { } arasında kalan kodlarımızı yazdigımız bolumdur

*** Method nerede olusturulmalidir ?

Method Class body'si icinde Main method disinda olusturulmalidir

```
public class asd {  
  
    public static void main(String[] args) {  
        toplama(5,4);  
  
    }  
  
    private static void toplama(int i, int j){  
        System.out.println(i+j);  
  
    }  
}
```

Method Call

Method Cagirma

Method olusturmak method'u calistirmak icin yeterli degildir.

Ihtiyac duyuldugunda daha onceden olusturulmus methodu calistirmak icin Method ismi (parametreler ile birlikte) yazilmalidir.

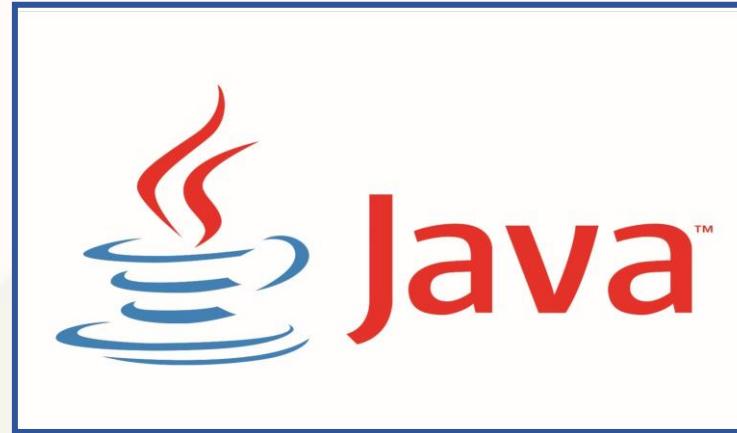
Bu isleme method cagirma denir

```
public class asd {  
    public static void main(String[] args) {  
        toplama(5,4);  
    }  
    private static void toplama(int i, int j) {  
        System.out.println(i+j);  
    }  
}
```

*** Method cagirirken parantez icine yazilan degerlere **Arguments (arguman)** denir.

*** Method cagirirken kullandigimiz argumanlar ile method parametrelerinin uyumlu olması gereklidir.

*** Sayi parametreleri icin char degerler de arguman olarak kullanilabilir



**9 KASIM 2021
DERS 15**

**Method Creation
Method Call**

Mehmet BULUTLUOZ
Elk.Elektronik Yuk.Muh.

Method Creation

Soru 1) Kullanicidan bir sayi alin. Bu sayinin tek mi cift mi oldugunu, sifirdan buyuk mu kucuk mu oldugunu, ayrica ve 100'den buyukse birler,onlar ve yuzler basamagindaki rakamlarin toplamini,100'den kucukse sadece 1'ler basamagini yazdiran 3 method olusturun.

Soru 2) Kullaniciya kac sayi toplamak istedigini sorun. Kullanici 2,3 veya 4 degerini girerse, kullanicidan bu sayilari girmesini isteyin ve sayilarin toplamini yazdirin. Kullanici toplamak istedigi sayi adedini 4'den buyuk girerse "Cok sayi girdiniz, ben toplayamam" yazdirin.

Soru 3) Email kontrolu yapan bir program yazin.Kullanicinin girdigi sifre

- @ isareti icermiyorsa gecersiz email yazdirin
- @gmail.com icermiyorsa "lutfen gmail adresinizi girin" yazdirin
- @gmail.com ile bitmiyorsa "Yazimda bir sorun var, maili kontrol ediniz"

Soru 4) Kullanicidan ismini, soyismini ve bosluk birakmadan 16 hane olarak kredi karti numarasini alin. Isim ve soyismi ilk harfleri buyuk diger harfler kucuk olacak sekilde, KK numarasini ise 4 rakamlik 4 blok ve aralarinda bosluk olacak sekilde duzelten 2 method yazin, ve programda kullanabilmek icin disenlenmis hallerini geri dondurun.

Method Overloading

Interview Sorusu

1) Overloading nedir ? Eger bir Class'da ismi ayni fakat parametreleri farkli olan methodlar olusturursak buna **Overloading** denir.

2) Overloading nasıl yapılır ? Java ayni isim ve ayni parametrelerle birden fazla method olusturulmasına izin vermez. Ayni isimle birden fazla method olusturmak isterseniz **method signature (metot imzası)**'nin degistirilmesi gereklidir

3) method signature (metot imzası) nasıl degistirilir?

Method signature'i degistirmek icin 3 yontem kullanilabilir

- parametrelerin data tipleri degistirilebilir
- parametrelerin sayisi degistirilebilir
- parametre sayisi ayni olmak zorunda ise farkli data tipindeki parametrelerin sirasi degistirilir

*** method'un return type'ini degistirmek, access modifier'ini degistirmek veya static kelimesi eklemek method signature'i degistirmez

For Loop

Belirli bir koşul sağlandığı sürece tekrarlanması gereken işler için kullanılan kod bloklarına LOOP(Dongu) denir. Tekrar sayısı belirli olan durumlarda for loop kullanılması tercih edilir.

The diagram illustrates the structure of a for loop. It features a green arrow pointing right labeled "Start", a red octagonal STOP sign, and two orange arrows pointing up and down labeled "Increasing or Decreasing the Value". The code structure is as follows:

```
for(Starting Value ; Ending Condition ; Increasing or Decreasing the Value) {  
}
```

```
for ( int i=4; i>1; i- - ) {  
    System.out.println( i );  
}
```

For Loop



- Eger **Ending Condition** hep **true** verirse loop sonsuz donguye girer
- Eger Loop'ta **Ending Condition** hic **true** olmazsa loop body hic devreye girmez
- loop'da artis degeri pozitif oldugu gibi negatif de olabilir (**i--** vb.)
- Artis degeri 1 olmak zorunda degil, farkli da olabilir (**i+=2** vb..)

For Loop

Soru 1) Ekrana 10 kez "Java guzeldir" yazdirin

Soru 2) 10 ile 30 arasindaki sayilari aralarinda virgule olarak ayni satirda yazdirin

Soru 3) 100'den baslayarak 50'ye kadar olan sayilari aralarinda virgule olarak ayni satirda yazdirin

Soru 4) Kullanicidan 100'den kucuk bir tamsayi isteyin. 1'den baslayarak girilen sayiya kadar 3'un kati olan sayilari yazdirin.

Soru 5) Kullanicidan 100'den kucuk bir tamsayi isteyin. 1'den baslayarak girilen sayiya kadar 3'un veya 5'in kati olan sayilari yazdirin.

Soru 6) Interview Question Kullanicidan 100'den kucuk bir tamsayi isteyin. 1'den baslayarak girilen sayiya kadar tum sayilari yazdirin. Ancak;

- Sayi 3'un kati ise sayi yerine "Java" yazdirin.
- Sayi 5'in kati ise sayi yerine "Guzeldir" yazdirin.
- Sayi hem 3'un hem 5'in kati ise sayi yerine "Java Guzeldir" yazdirin.

For Loop

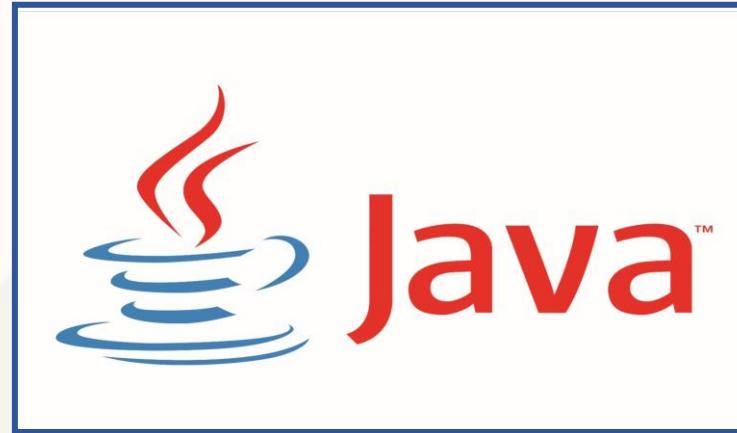
Soru 7) Interview Question Kullanicidan bir String isteyin ve Stringi tersten yazdirin.

Soru 8) Interview Question Kullanicidan bir String isteyin ve Stringi tersine ceviren bir program yazin.

Soru 9) Interview Question Kullanicidan bir String isteyin. Kullanicinin girdigi String'in palindrome olup olmadigini kontrol eden bir program yazin.

Soru 10) Kullanicidan iki sayi isteyin. Girilen sayilar ve aralarindaki tum tamsayilari toplayip, sonucu yazdiran bir program yaziniz

Soru 11) Interview Question Kullanicidan 10'dan kucuk bir tamsayi isteyin ve girilen sayinin faktoryel'ini bulun. ($5!=5*4*3*2*1$)



14 TEMMUZ 2021
DERS 15

Nested For Loop
While Loop

Mehmet BULUTLUOZ
Elk.Elektronik Yuk.Muh.

Onceki Dersten Aklimizda Kalanlar

- 1- For loop : baslangic ve bitisi belli olan tekrarli kod calismalarini yapmak icin loop kullaniriz.
- 2- Loop yazarken karar vermemiz gereken 3 sey
 - i) Baslangic degeri
 - ii) bitirmek icin kosulumuz nedir
 - iii) her adimda degiskeni nasil degistirecegiz
- 3- Eger baslangic degerini artis veya azalis yaparak degistirdigimizde bitis kosuluna yaklasmiyorsak sonsuz loop olusur,
- 4- Eger baslangic degeri bitis kosulunu saglamazsa, for loop calisir ama loop body'si hic devreye girmez, dolayisiyla loop anlamsiz olur...
- 5- loop daki degisken artabildigi gibi, soruya gore azalabilir

Nested For Loop

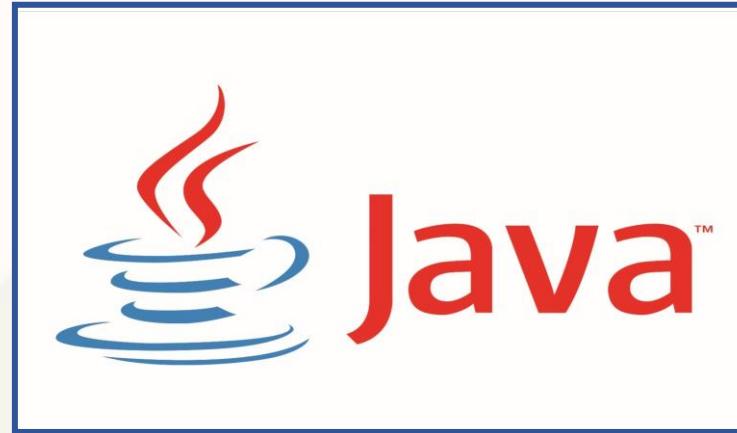
Bazen tek bir loop ile istedigimiz sonuclarla ulasamayiz.

Ozellikle iki boyutlu sekiller cizdirmek veya carpim tablosu gibi sayı ikilileri olusturmak icin nested loop kullanmamiz gereklidir.

*	1	2	3	4
**	2	4	6	8
***	3	6	9	12

```
for (int i = 1; i <= 4; i++) {  
  
    for (int j = 1; j <= 4; j++) {  
  
        System.out.print("(" + i + "," + j + ") ");  
    }  
  
    System.out.println();  
}
```

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4)



15 TEMMUZ 2021
DERS 16

Nested For Loop
While Loop

Mehmet BULUTLUOZ
Elk.Elektronik Yuk.Muh.

Nested For Loop

Soru 12) Kullanicidan pozitif bir rakam girmesini isteyin ve girilen rakama gore asagidaki sekli cizdirin

```
*  
* *  
* * *  
* * * *
```

Soru 13) Kullanicidan pozitif bir rakam girmesini isteyin ve girilen rakama gore carpim tablosu olusturun. Ornek,kullanici 3 girerse,

```
1 2 3  
2 4 6  
3 6 9
```

While Loop

`while(condition) {` **Code** `}`

true

After running the code check the condition again

`while(condition) {` **Code** `}`

false

Break the loop and proceed to the next line

```
int i = 0;  
while (i < 5) {  
    System.out.println(i);  
    i++;  
}
```

While Loop

Soru 1) While loop kullanarak 3 den 13 e kadar tum tek tamsayıları ekrana yazdiriniz.

Soru 2) For loop ve while Loop kullanarak 3 basamaklı sayılarından 15, 20 ve 90'na tam bolunebilen sayıları yazdirin.

Soru 3) Kullanicidan baslangic ve bitis degerlerini alin. Baslangic degeri ve bitis degeri dahil aradalarindaki tum cift tamsayıları while loop kullanarak ekrana yazdiriniz.

Soru 4) Kullanicidan baslangic ve bitis haflerini alin ve baslangic harfinden baslayip bitis harfinde biten tum harfleri buyuk harf olarak ekrana yazdirin. Kullanicinin hata yapmadigini farz edin.

While Loop

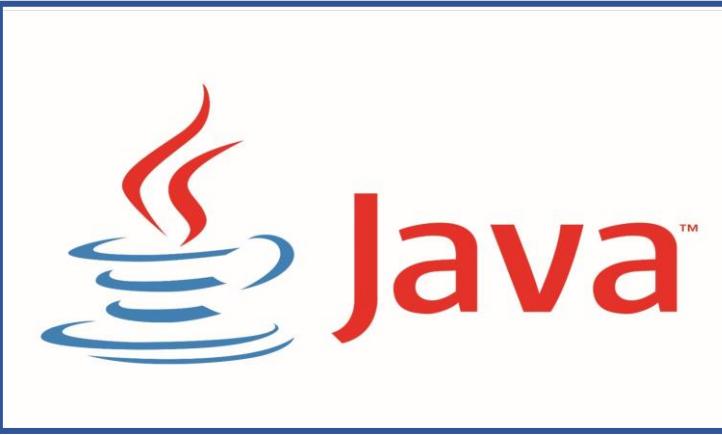
Soru 5) Kullanicidan bir rakam alin ve bu rakam icin carpim tablosunu ekrana yazdirin. Kullanicinin hata yapmadigini farz edin.

Ornegin kullanici 3 girerse;

$3 \times 1 = 3$ $3 \times 2 = 6$ $3 \times 3 = 9$ $3 \times 4 = 12$ $3 \times 5 = 15$ $3 \times 6 = 18$ $3 \times 7 = 21$ $3 \times 8 = 24$ $3 \times 9 = 27$ $3 \times 10 = 30$

Soru 6) Kullanicidan bir sayi alin ve bu sayiyi tam bolen sayilari ve toplam kac tane olduklarini ekranada yazdirin

Soru 7) Kullanicidan bir sayi alin ve bu sayinin rakamlari toplamini yazdirin



16 TEMMUZ 2021
DERS 18

**DoWhile Loop
Scope**

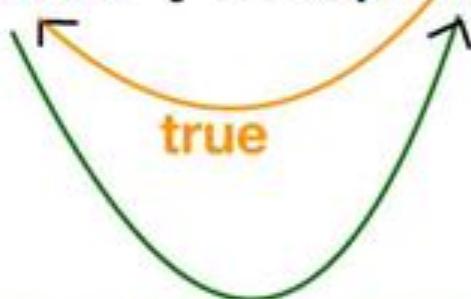
Mehmet BULUTLUOZ
Elk.Elektronik Yuk.Muh.

Onceki Dersten Aklimizda Kalanlar

- 1- Nested For Loop : iki boyutlu sekiller veya iki degisenli denklemler kullanmamiz gerektiginde nested for loop'lara ihtiyacimiz olur
- 2- Outer loop'un her bir dongusu icin inner loop bastan sona calisir. Inner bittikten sonra bir alt satira gecip orada islem varsa devam eder..
- 3- While loop , sart olarak yazdigimiz boolean ifade true oldugu muddetce while loop body'si tekrar tekrar calisir. Sart false oldugunda loop sona erer ve sonraki satirdan kod calismaya devam eder
- 4- Daha once for loop da kullandigimiz baslangic degeri ve artis sekli while loop'un syntax'inde yer almaz ama loop'un calismasi icin gereklidir. Bunun icin biz her while loop icin loop baslamadan once baslangic degeri olusturacak variable'l olusturmali ve loop'un icinde de degisim seklini ve miktarini yazmaliyiz.
- 5- For loop Vs While Loop :
adim sayisi veya baslangic ve bitis degerleri belli ise for loop daha Pratik olabilir.
Ancak adim sayisi ongoreulemiyorsa, loop'un bitisi bir adima degil de sarta bagliysa bu durumda while loop daha avantajli olmaktadır.
Ornegin kullanidan degerler aliyorsak ve kullaniciya istedigi kadar deger grime hakki veriyorsak for loop kullanmamiz mumkun degildir.

Do While Loop

do { **Code** } while(**condition**)



Run the code then check the condition

do { **Code** } while(**condition**)

Run the code then check the condition

false

Break the loop and proceed to the next line

```
public static void main(String[] args) {  
    int i = 0;  
  
    do {  
        System.out.println(i);  
        i++;  
    }  
    while (i<5);  
}
```

Do While Loop Vs While Loop

```
public static void main(String[] args) {  
  
    int i = 10;  
  
    do {  
        System.out.println(i);  
        i++;  
    }  
    while (i<5);  
}
```

```
public static void main(String[] args) {  
  
    int i = 10;  
  
    while (i<5){  
        System.out.println(i);  
        i++;  
    }  
}
```

Fark : While Loop, dongunun başlangıcında kosulu kontrol eder ve kosul sağlanırsa body icindeki kodları çalıştırır.
Do-while loop'ta ise , kosul body içerisindeki kodlar 1 kere çalıştırıldıktan sonra kontrol edilir.

Sonuc : Bir while loop'daki kosul yanlışsa, loop hiç çalışmaz 'do-wile' loop'ta ise , kosul yanlışsa kodlar 1 kere çalışır

Do While Loop

Soru 1) 9 den 190 e kadar 7 nin katı olan tüm tamsayıları ekrana yazdırınız.

Soru 2) 'm' harfinden başlayarak 'c' harfine kadar tüm harfleri yazdırın.

Soru 3) Kullanıcıdan toplamak üzere pozitif sayılar isteyin, işlemi bitirmek için 0'a basmasını söyleyin.

Kullanıcı 0'a bastığında toplam kaç pozitif sayı girdigini ve girdiği pozitif sayıların toplamının kaç olduğunu yazdırın.

Soru 4) Kullanıcıdan toplamak üzere pozitif sayılar isteyin, işlemi bitirmek için 0'a basmasını söyleyin.

Kullanıcı yanlışlıkla negative sayı girerse o sayıyı dikkate almayın ve “Negatif sayı giremezsiniz” yazdırıp basa donun

Kullanıcı 0'a bastığında toplam kaç pozitif sayı girdigini, yanlışlıkla kaç negative sayı girdigini ve girdiği pozitif sayıların toplamının kaç olduğunu yazdırın.

Do While Loop

Soru 5) Kullanicidan bir sifre girmesini isteyin. Girilen sifreyi asagidaki şartlara gore kontrol edin ve sifredeki hatalari yazdirin.

Kullanici gecerli bir sifre girinceye kadar bu islemi tekrar edin ve gecerli sifre girdiginde "Sifreniz Kabul edilmistir" yazdirin.

- Sifre kucuk harf icermelidir
- Sifre buyuk harf icermelidir
- Sifre ozel karakter icermelidir
- Sifre en az 8 karakter olmalidir.

Soru 6) Kullanicidan toplamak icin sayi isteyin ve toplam 500'e ulasincaya kadar devam istemeyi ettirin. Toplam 500'e ulastiginda veya gectiginde toplami ve kac sayi girildigini yazdirin

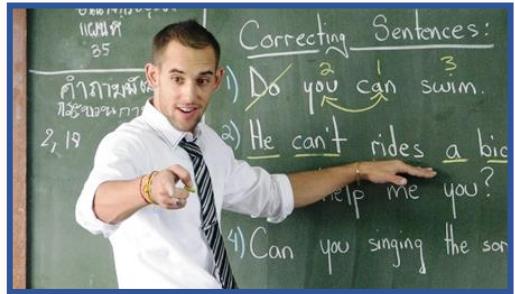
Scope

Instance, Class ve Local Variables





Object Nasıl Kullanılır ?



Ogretmen

Dersler

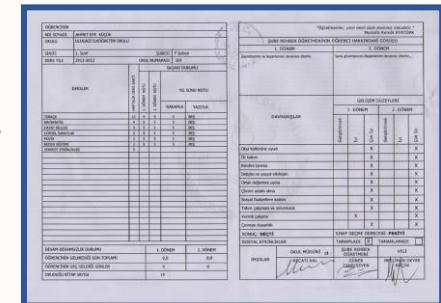
09:00	TÜRKÇE-1
09:30	MATEMATİK-1
10:00	TÜRKÇE-2
10:30	MATEMATİK-2
11:00	TÜRKÇE-3
11:30	MATEMATİK-3
12:00	TÜRKÇE-4
12:30	MATEMATİK-4
13:00	İYEP TÜRKÇE



Personel



Ogrenci



Notlar

Scope

Instance, Class ve Local Variables

Scope (Kapsam)

- Bir Class icerisinde olusturulan variable'lar icin Scope, o variable'a nereden, nasil ulasabilecegini ve nerede gecerli oldugunu ifade eder.
- Scope'a uymayan bir kullanimda Java Compile Time Error verir.
- Java'da olusturulan variable'lar icin 4 Scope mevcuttur
 - 1) Instance (Object) Variables // ogretmenin adi gibi, ogrencinin notu gibi
 - 2) Static (Class) Variables // okul adi, adresi gibi
 - 3) Local (Method) Variables
 - 4) Loop Variables

Scope

Instance, Class ve Local Variables

Instance (Object) Variable

Class'in içinde ancak main method'un
disında olmalıdır

Static olmamalıdır

Oluşturulması yeterlidir, değer
atanması şart değildir.

```
public class Example {  
  
    int sayi;  
  
    public static void main(String[] args) {  
        //  
    }  
}
```

Default Value

Eğer instance bir variable oluşturur ama değer atamazsanız, Java otomatik olarak
default değerleri assign eder. (String için null, sayısal data türleri 0, boolean false)

Scope

Instance, Class ve Local Variables

Instance (Object) Variable

class icerisinde veya baska class'larda direkt kullanilamaz, kullanmak istedigimizde MUTLAKA object olusturmali ve object uzerinden ulasilmalidir.

```
public class Example {  
  
    int sayi;  
    char ilkHarf;  
    String isim;  
    boolean ogrenciMi;  
  
    public static void main(String[] args) {  
  
        Example ex1=new Example();  
        System.out.println(ex1.sayi);  
        System.out.println(ex1.ilkHarf);  
        System.out.println(ex1.isim);  
        System.out.println(ex1.ogrenciMi);  
    }  
}
```

Outputs

0
null
false

Ornek :

Bir okul uygulamasi yaptigimizi dusundugumuzde, ogretmenIsmi, ogrenciIsmi, matematiNotu gibi degiskenler bir kisi ile ilişkilendirilmedikce anlamlı olmaz

Scope

Instance, Class ve Local Variables

Class (static) Variable

Class'in icinde ancak main method'un disinda olmalidir.

Static olmalidir

Olusturulmasi yeterlidir, deger atanmasi şart degildir.

```
public class Example {  
    static int sayi;  
  
    public static void main(String[] args) {  
    }  
}
```

Scope

Instance, Class ve Local Variables

Class (static) Variable, class içerisinde direkt kullanılır, başka class'larda kullanmak istedigimizde object olusturmaya ihtiyac duymadan classIsmi.variableIsmi ile variable'a ulaşabilir ve kalıcı olarak degistirebiliriz.

```
public class Example {  
  
    static int okulId;  
    static String okulIsmi;  
    static boolean acikMi;  
  
    public static void main(String[] args) {  
  
        System.out.println(okulId);  
        System.out.println(okulIsmi);  
        System.out.println(acikMi);  
    }  
}
```

Outputs

0
null
false

Ornek : Bir okul uygulaması yaptığımızı düşünün okulları, okullar, açık mı gibi değişkenler bir kişiyi değil okulla ilgili herkesi ilgilendirir ve bir kişi okul ismini veya okul telefon numarasını değiştirdiğinde okulla ilgili herkes için okul ismi değişir.

Scope Instance Vs Class Variables

Instance (Object) Variable, class içerisinde veya başka class'larda direk kullanılamaz, kullanmak istedigimizde MUTLAKA object olusturmali ve object uzerinden ulasılabilir.

Class (static) Variable, class içerisinde direkt kullanılabılır, başka class'larda kullanmak istedigimizde object olusturmaya ihtiyac duymadan classIsimi.variableIsimi ile variable'a ulasabilir ve kalıcı olarak degistirebiliriz.

Static variable'lar herkes için ortaktır (okul ismi gibi) , instance variable'lar ise objeye bağlıdır (matematikNotu, ogrenciIsmi gibi)

Static variable yetkisi olan herkes tarafından degistirilebilir ve bu degisim her obje için gecerlidir. Instance variable da yetkisi olan herkes tarafından degistirilebilir ancak yapılan degisiklik sadece o obje ile ilgilidir, geneli kapsamaz.

Scope

Instance, Class ve Local Variables

Local Variable

- Herhangi bir method içerisinde oluşturulan variable'lardır (main method dahil).
- Sadece o method içerisinde geçerlidir.
- Baska methodlarda da kullanılacak variable'lari, local oluşturmak yerine **class level**'da oluşturmak gereklidir.
- Class level'da oluşturulacak variable, main method'da kullanılacaksa static olarak oluşturulmalıdır. Bu durumda bu variable kullanacak, diğer method'lar da static olmalıdır.

```
public class Example {  
  
    public static void main(String[] args) {  
  
        int sayi;  
  
    }  
  
    public void add() {  
  
        String isim;  
    }  
}
```

Scope

Instance, Class ve Local Variables

Local Variable

- Java local variable'lara default değer atamaz.
- Sadece olusturdugunuzda Java şikayet etmez. (variable olusturuldu method icerisinde değer atanacak diye bekler.)
- Olusturulan local variable'lara değer atamadan kullanmaya calisirsaniz Java şikayet eder(CTE)

```
5 public class Example {  
6  
7  
8    public static void main(String[] args) {  
9  
10       int sayi;  
11       sayi++;  
12  
13  
14  
15  
16       public void add() {  
17  
18           String isim;  
19           System.out.println(isim);  
20  
21 }
```

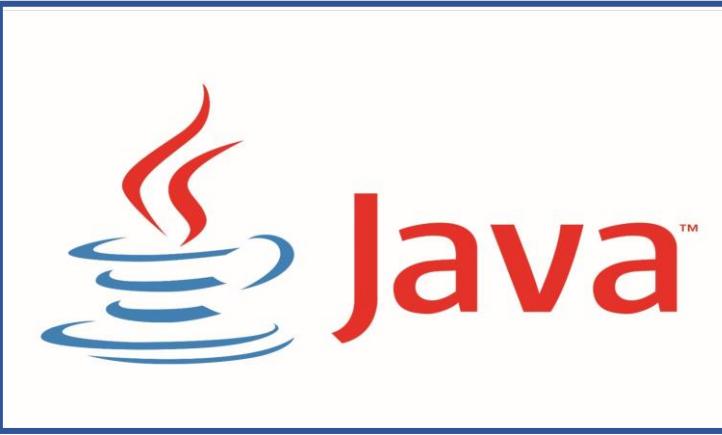
Scope

Instance, Class ve Local Variables

Loop Variables

- Bir loop icinde olusturulan variable'lar sadece o loop icerisinde gecerlidir.
- Loop icerisinde olusturulan variable'lara loop disindan ulasilmaz ve loop disinda kullanilamaz.
- Loop icerisinde olusturulan local variable'lari disarida kullanmaya calisirsaniz Java sikayet eder(CTE)

```
public static void main(String[] args) {  
    for (int i = 0; i < 10; i++) {  
        int sayi=10;  
        System.out.println(sayı);  
    }  
    System.out.println(sayı);  
}
```



17 TEMMUZ 2021
DERS 19

Arrays

Mehmet BULUTLUOZ
Elk.Elektronik Yuk.Muh.

Onceki Dersten Aklimizda Kalanlar

- 1- Scope : Olusturdugumuz variable'larin gecerli oldugu bolgedir. Bu konuda ogrenmemiz gereken sey hangi variable'a nereden ve nasil ulasabilecegimizdir.
- 2- Java'da bilmemiz gereken 4 scope vardir. Bunlardan ilk ikisi class level'da olusturulan variable'lar icin gecerlidir.
 - i) instance variables (Object variables): Objelere bagli olarak degisir (ogretmen ismi veya not gibi)
 - class level'da olusturulmalidir.
 - static olmamasi gerekir
 - static olmadigi icin static alanlardan direk kullanilamaz (object olusturularak kullanilabilir)
 - olusturmak yeterlidir, deger atamak mecburi degildir. (deger atanmazsa default degeri alir)
 - ii) static variables (Class variable) : Tum objeler icin ortaktir. (okul ismi gibi) Eger static variable degistirilirse tum objeler etkilenir.
 - class level'da olusturulmalidir. (local'de static variable olusturulamaz)
 - static kelimesi kullanilarak tanimlanmalidir
 - Class icerisinde static olan veya olmayan heryer'den erisilebilir.
 - olusturmak yeterlidir, deger atamak mecburi degildir. (deger atanmazsa default degeri alir)

Onceki Dersten Aklimizda Kalanlar

- 1- Scope : Olusturdugumuz variable'larin gecerli oldugu bolgedir. Bu konuda ogrenmemiz gereken sey hangi variable'a nereden ve nasil ulasabilecegimizdir.
- 2- Java'da bilmemiz gereken 4 scope vardir. Bunlardan ilk ikisi class level'da olusturulan variable'lar icin gecerlidir.
 - i) instance variables (Object variables): Objelere bagli olarak degisir (ogretmen ismi veya not gibi)
 - class level'da olusturulmalidir.
 - static olmamasi gerekir
 - static olmadigi icin static alanlardan direk kullanilamaz (object olusturularak kullanilabilir)
 - olusturmak yeterlidir, deger atamak mecburi degildir. (deger atanmazsa default degeri alir)
 - ii) static variables (Class variable) : Tum objeler icin ortaktir. (okul ismi gibi) Eger static variable degistirilirse tum objeler etkilenir.
 - class level'da olusturulmalidir. (local'de static variable olusturulamaz)
 - static kelimesi kullanilarak tanimlanmalidir
 - Class icerisinde static olan veya olmayan heryer'den erisilebilir.
 - olusturmak yeterlidir, deger atamak mecburi degildir. (deger atanmazsa default degeri alir)

Onceki Dersten Aklimizda Kalanlar

3- diger 2 scope class level'da degildir

iii) local variable : herhangi bir method icerisinde olusturulan variable'lardir ve sadece olusturulduklari method icerisinde direkt kullanilabilirler.

- olusturmak icin declaration yeterlidir, ancak kullanmak icin declaration yetmez, mutlaka oncelikle deger atamasi (assignment) yapmak gereklidir.

- static olarak tanimlanamaz

- assignment yapmadan kullanmaya kalkarsak java CTE verir

iv) loop variable (bazi kaynaklarda buna da local variable denir)

- sadece olusturulduklari loop icerisinde gecerlidirler.

- loop'un disinda erisilemez ve kullanilamazlar.

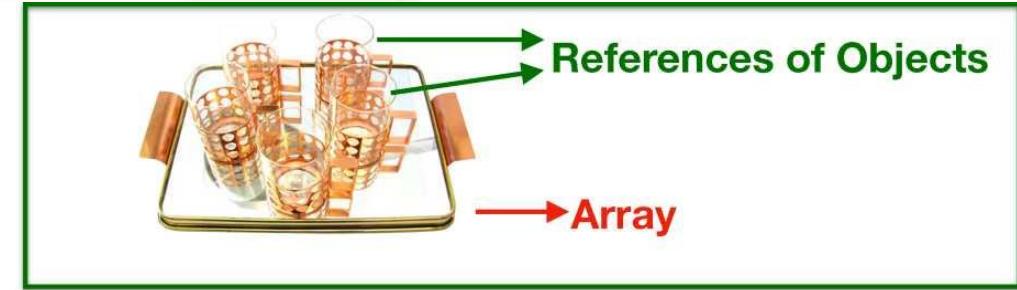
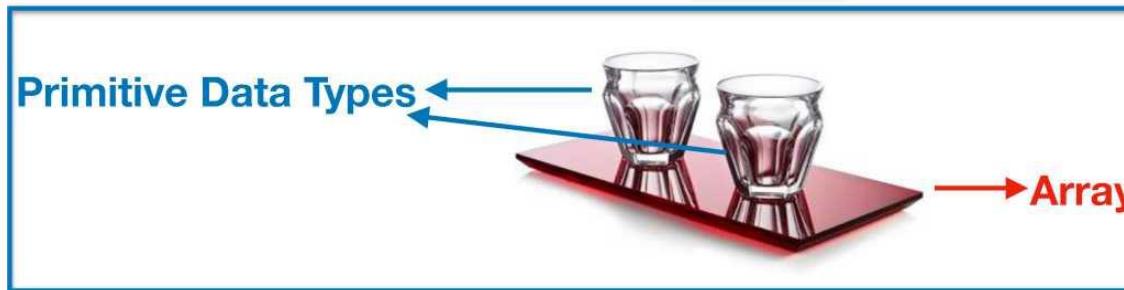
Scope Instance, Class ve Local Variables

```
public class MyClass{  
    int num1;  
    String name = "Ali";  
    public static void main(String args){  
        add();  
        product (5);  
    }  
    public static add(){  
        num1++;  
        int num2 = 6;  
        char letter;  
        System.out.println("Do addition ");  
    }  
    public product(int num3){  
        name = "Veli";  
        num2++;  
        System.out.println(num3 * num3);  
    } }
```

- 1) Hangileri instance variable'dir ?
- 2) Hangileri local variable'dir?
- 3) num1 icin default value nedir ?
- 4) Java hangi satirlarin altini kirmizi cizer?
- 5) Kac satir compile time error verir?

Arrays

Arrays birden fazla variable depolamak için kullanılabilen object (non-primitive data)'lerdir.



- 1) Arrays'de sadece primitive datalar veya non-primitive datalara ait referans'lar depolanabilir
- 2) Arrays içindeki tüm variable'lar aynı data type'inde olmalıdır.

Arrays

- 3) Bir Array olusturmadan once o Array'in icine kac variable koyacagimiza karar vermeliyiz.
- 4) Bir Array icine koyabilecegimiz variable sayisina o Array'in "length" i denir. O Array icine length'den fazla variable koyamayiz.



Maximum capacity (length) = 2



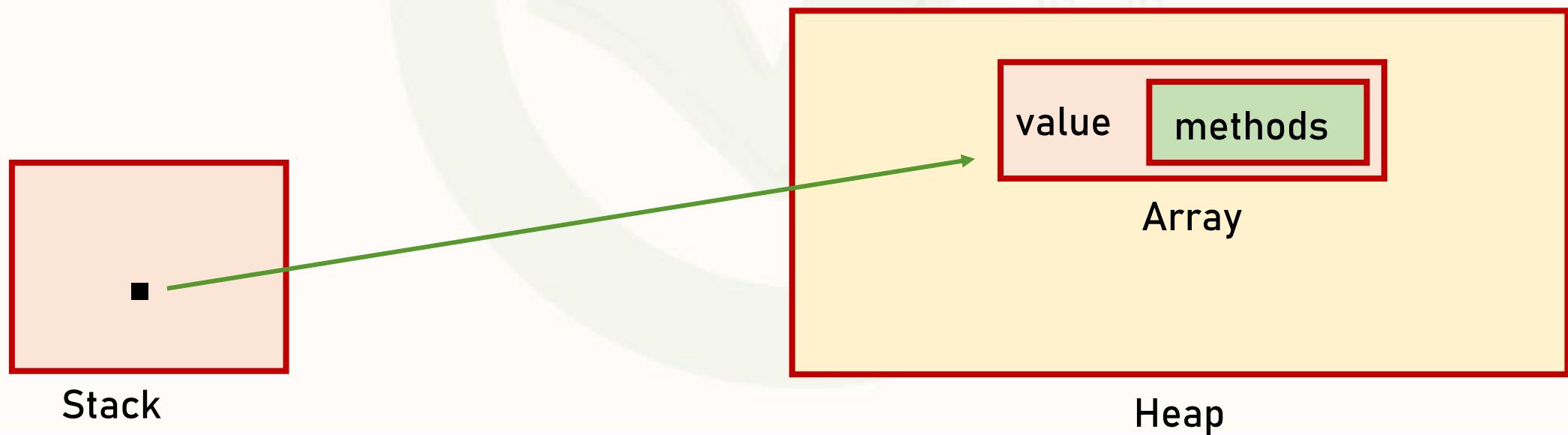
Maximum capacity (length) = 5

Arrays

5) Array'ler object (non-primitive) 'tir. Bu yuzden

- Heap Memory'de depolanirlar.
- Value ile birlikte method'lara da sahiptirler
- runtime'da olusturulurlar.

Bir Array declare edildiginde stack memory'de referans olusturulur ama Array henuz olusturulmamistir.



Arrays

6) Bir Array nasıl declare edilir?

Array declare etmek için iki yol vardır :

- int myArray[]; // Bu daha çok kullanılır
- int [] myArray;

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

7) Bir Array nasıl oluşturulur

```
int myArray[ ] = new int[6];
```

- Yukarıdaki kod length'i 6 olan bir array oluşturur.
- Biz array'e eleman eklemezsek Java elemanlar için data type'ına uygun default değerler atar.
- Eğer yukarıdaki array'i yazdırırsanız ekranda {0, 0, 0, 0, 0, 0} gorursunuz

NOT : Array oluştururken length'i yazmazsanız compile time error alırsınız.

Arrays

8) Array'e degerler nasil atanir

```
int myArray[ ] = new int[3];
```

```
myArray[0] = 9;  
myArray[1] = 10;  
myArray[2] = 11;
```

Once olusturup, sonra istedigimiz indexler icin deger atayabiliriz

Veya

```
int myArray[ ] = {9, 10, 11};
```

Olusturma ve tum indexler icin deger atamayı tek satirda yapariz.

Soru 1: Elemanlari “Ali” , “Veli”, “Ayse” ve “Fatma” olan bir array olusturun ve bu array'i yazdirin.

Arrays

9) Array'in elemanlarina nasil ulasilir ve nasil update edilir ?

```
int myArray[ ] = {9, 10, 11};
```

Array elemanlarina index'ler kullanilarak ulasilir.

myArray[0] ==> 9,

myArray[1] ==> 10,

myArray[2] ==> 11,

NOT 1: "n" array'in length'i olmak uzere myArray[n-1] son elemani gosterir

NOT 2 : Bir Array'de olmayan index'i kullanmak isterseniz
“**ArraysIndexOutOfBoundsException**” alirsiniz.

Soru 2: Soru 1'deki elemanlardan “Ali” yerine “Can”, “Ayse” yerine “Gul” atayin.

Arrays

10) Bir Array'in uzunlugu nasil bulunur?

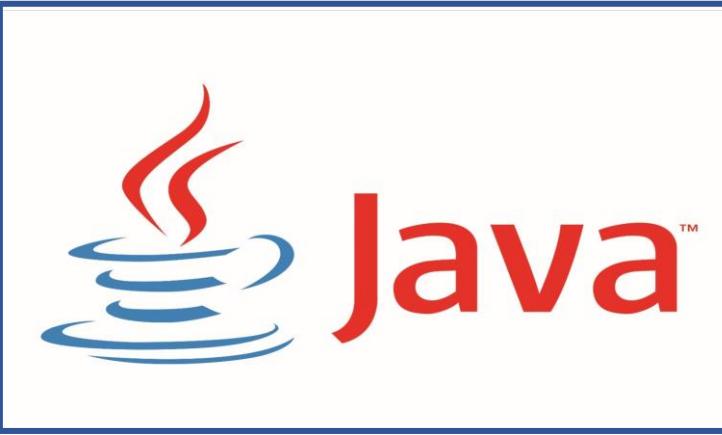
```
int myArray[] = {9, 10, 11};
```

```
int size = myArray.length;
```

NOT : String ve Array icin length method'larinda dikkatli olmak gerekir.

Strings ==> **length()**

Arrays ==> **length**



19 TEMMUZ 2021
DERS 20

Arrays

Mehmet BULUTLUOZ
Elk.Elektronik Yuk.Muh.

Onceki Dersten Aklimizda Kalanlar

- 1- Array (dizi) : birden çok datayı store etmek için kullandığımız objelerdir. (non-primitive)
- 2- NP olduğu için heap memory'de run time'da oluşturulur
- 3- Array oluşturmadan önce maximum element sayısına (length) karar vermemiz ve koyacağımız elementlerin data turunu declare etmemiz gereklidir. String arr [] = new String[5];
- 4- Array'ı oluştururken içine koyacağımız elementlerin data turunu declare ettigimiz için, o data turundan başka turde data koymayız
- 5- Array'in kapasitesi sonradan degistirilemez.. Declare edilirken length yazılmazsa CTE verir
- 6- Array'ler sadece primitive data turundekidataları veya non-primitive data turundeki dataların referanlarını store edebilirler
- 7- Array'ı direkt yazdırmak istersek Java array'ı değil referansını yazdırır. Array'deki elementleri yazdırma için for loop kullanabiliriz veya tüm array'ı yazdırma için Arrays class'ından toString method'unu kullanabiliriz
- 8- Array'ı oluşturduğumuzda değer atamazsa Java default değerler assign eder
- 9- Arraydeki tüm elementlere index ile ulaşabilir veya update edebiliriz
- 10-

Arrays

11) Bir Array'in tum eleamanlari nasil yazdirilir?

```
int myArray[ ] = {9, 10, 11};
```

```
for(int i=0; i<size; i++) {  
    System.out.println(myArray[i]);  
}
```

```
System.out.println(Arrays.toString(myArray));
```

Soru 1: Verilen 3 elemanli bir array'in tum elemanlarini bir soldaki konuma tasiyacak bir program yazin. Ornek; array [1,2, 3] ise output [2, 3, 1] olacak.

Soru 2: Verilen bir array'in tum elemanlarini toplayan bir program yazalim.

Arrays

12) Bir Array'in tum elemanlari nasil siralanir?

```
int myArray[ ] = {9, 15, 11};  
Arrays.sort(myArray);
```

Siralama buyukten kucuge nasil yapilir ?

- Once sort methodu kullanilir
- Sonra siralamayi ters cevirmek icin loop kullanilir

Arrays

13) Bir Array'de istenen bir elemanın varlığı nasıl kontrol edilir?

`binarySearch()` method'u belli bir elemanın bir array'de olup olmadığını kontrol etmek için kullanılır.

Ancak, `binarySearch()` methodunu kullanmadan önce mutlaka `sort()` methodu kullanılmalıdır.

```
int[ ] numbers = { 2, 8, 6, 4 };
Arrays.sort(numbers);
System.out.println( Arrays. binarySearch(numbers, 2)); //===== 0
System.out.println( Arrays. binarySearch(numbers, 4)); //===== 1
```

Eğer bir eleman array'de yoksa output negatif olur.

- 1) 0 eleman var olsaydı sıra numarası kaç olurdu, buluruz.
- 2) Bulduğumuz sıra numarasının negatif hali, `binarySearch()`'un outputu olur.

```
System.out.println( Arrays. binarySearch(numbers, 1)); // ===== -1
System.out.println( Arrays. binarySearch(numbers, 3)); // ===== -2
System.out.println( Arrays. binarySearch(numbers, 9)); // ===== -5
```

Arrays

Output nedir ?

```
int[ ] numbers = { 2, 1, 7, 6 };
Arrays.sort(numbers);
System.out.println(Arrays.binarySearch(numbers, 2));
System.out.println(Arrays.binarySearch(numbers, 7));
System.out.println(Arrays.binarySearch(numbers, 3));
System.out.println(Arrays.binarySearch(numbers, 9));
```

→ 1
→ 3
→ -3
→ -5

```
String[ ] letters = { "A", "N", "F", "C" };
Arrays.sort(letters);
System.out.println(Arrays.binarySearch(letters, "A"));
System.out.println(Arrays.binarySearch(letters, "C"));
System.out.println(Arrays.binarySearch(letters, "E"));
System.out.println(Arrays.binarySearch(letters, "G"));
```

→ 0
→ 1
→ -3
→ -4

Arrays

14) İki array'in eşit olup olmadığı nasıl kontrol edilir?

`equals()` method'u değerleri ve indexleri birlikte kontrol edip, boolean bir değer return eder.

```
int arr1[] = {2, 1, 7, 6};  
int arr2[] = {7, 1, 6, 2};  
System.out.println(Arrays.equals(arr1, arr2)); → false  
  
int arr3[] = {3, 2, 7, 8, 11};  
int arr4[] = {7, 3, 8, 2, 12};  
Arrays.sort(arr3);  
Arrays.sort(arr4);  
System.out.println(Arrays.equals(arr3, arr4)); → false  
  
int arr5[] = {4, 2, 6, 8, 11};  
int arr6[] = {11, 4, 8, 2, 6};  
Arrays.sort(arr5);  
Arrays.sort(arr6);  
System.out.println(Arrays.equals(arr5, arr6)); → true
```

Arrays

16) Bir String nasıl array'e cevrilir ?

`split()` method'u String'e ait bir method'dur ve belirledigimiz ayırac'a göre String'i parçalara ayırip bir Array'e çevirir.

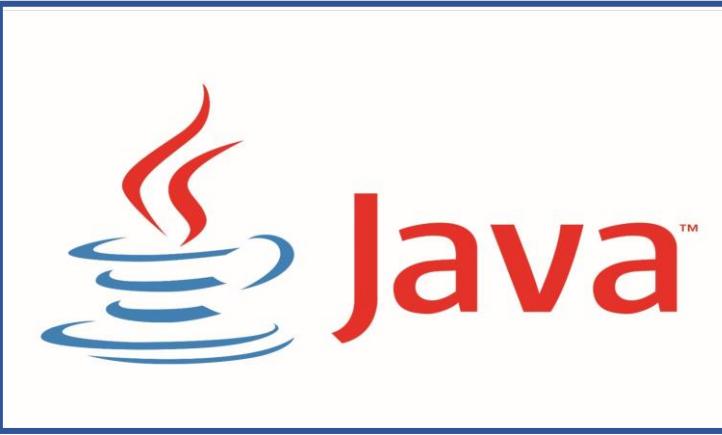
```
String str = "Java ogrenmek, IT alaninda yer edinmek demektir.";  
  
String arr1[] = str.split(",");  
System.out.println(Arrays.toString(arr1));  
                                → [Java ogrenmek, IT alaninda yer edinmek demektir.]  
String arr2[] = str.split(" ");  
System.out.println(Arrays.toString(arr2));  
                                → [Java, ogrenmek,, IT, alaninda, yer, edinmek, demektir.]  
  
String arr3[] = str.split("");  
System.out.println(Arrays.toString(arr3));  
  
[J, a, v, a, , o, g, r, e, n, m, e, k, , , I, T, , a, l, a, n, i, n, d,  
a, , y, e, r, , e, d, i, n, m, e, k, , d, e, m, e, k, t, i, r, .]
```

Arrays

What is the result of the following?

```
int[] random = { 6, -4, 12, 0, -10 };
int x = 12;
int y = Arrays.binarySearch(random, x);
System.out.println(y);
```

- A.** 2
- B.** 4
- C.** 6
- D.** The result is undefined.
- E.** An exception is thrown.
- F.** The code does not compile.



23 TEMMUZ 2021
DERS 21

Multi Dimensional Arrays

Mehmet BULUTLUOZ
Elk.Elektronik Yuk.Muh.

Multi Dimensional Arrays (Cok Katli Array'ler)

Eger bir Array ic ice Array'lerden olusuyorsa buna Multi Dimensional Array denir

```
int[][][] arrr = { { {1,2},{3,4},{5,6} }, { {7,8},{9,1},{2,3} } }
```

child arrays
parent array *parent array*
grand parent array

Multi Dimensional Arrays (Cok Katli Array'ler)

Array'i tanimlarken (declaration), her bir kat icin bir [] kullanilir.

```
Int arr[ ][ ] = { {1,2} , {3,4}, {5,6}};
```

```
int arr[][]= new int [3][2];  
  
arr[0][0]=1;  
arr[0][1]=2;  
  
arr[1][0]=3;  
arr[1][1]=4;  
  
arr[2][0]=5;  
arr[2][1]=6;  
  
System.out.println(Arrays.toString(arr[0]));  
System.out.println(Arrays.toString(arr[1]));  
System.out.println(Arrays.toString(arr[2]));  
  
System.out.println(arr[0][1]);  
System.out.println(arr[2][0]);  
  
System.out.println(Arrays.toString(arr));
```

Multi Dimensional Array olusturma

Array icindeki elemanlara deger atama

Inner Array'leri yazdirma

Belirli bir elemani yazdirma

[[I@15db9742, [I@6d06d69c, [I@7852e922]

Multi Dimensional Arrays

(Cok Katli Array'ler)

Multi Dimensional Array'in tum elemanlari nasıl yazdirilir ?

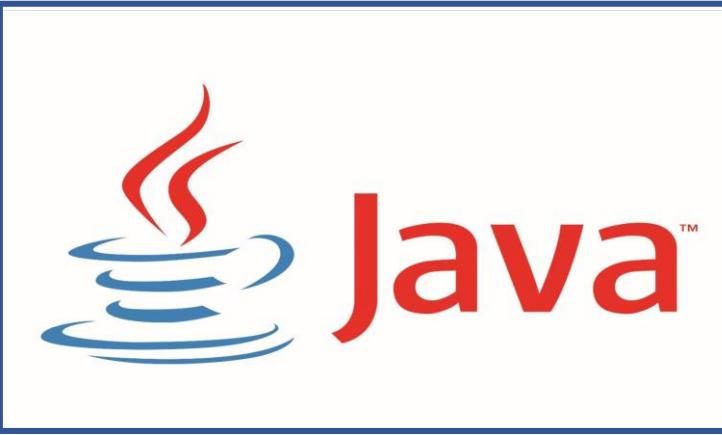
```
public static void main(String[] args) {  
  
    int arr[][] = { {1,2} , {3,4}, {5,6}};  
  
    for (int i = 0; i < arr.length; i++) {  
  
        for (int j = 0; j < arr[i].length; j++) {  
  
            System.out.print(arr[i][j] + " ");  
        }  
    }  
  
    System.out.println(Arrays.deepToString(arr));  
}
```

→ Nested For Loop kullanilabilir

→ Arrays Class'indan method kullanilabilir

Multi Dimensional Arrays (Cok Katli Array'ler)

- Soru 1)** Asagidaki multi dimensional array'in tum elemanlarinin carpimini ekrana yazdiran bir method yaziniz. { {1,2,3}, {4,5,6} }
- Soru 2)** Asagidaki multi dimensional array'in ic array'lerindeki son elemanların carpimini ekrana yazdiran bir program yaziniz { {1,2,3}, {4,5}, {6} }
- Soru 3)** Asagidaki multi dimensional array'lerin ic array'lerinde aynı index'e sahip elemanların toplamini ekrana yazdiran bir program yaziniz. (Zor soru) arr1 = { {1,2}, {3,4,5}, {6} } ve arr2 = { {7,8,9}, {10,11}, {12} }
- Soru 4)** Asagidaki multi dimensional array'in ic array'lerindeki tum elemanların toplamini birer birer bulan ve herbir sonucu yeni bir array'in elemani yapan ve yeni array'i ekrana yazdiran bir program yaziniz { {1,2,3}, {4,5}, {6,7} }
- Ornek; { {1,2,3}, {4,5}, {6,7} } ==> 1+2+3=6 4+5=9 6+7=13 ==> output: {6, 9, 13}
- Soru 5)** Kullanicidan bir cümle isteyin ve cumledeki kelime sayisini yazdirin
- Soru 6)** Verilen bir Array'den isten degere esit olan elamanlari kalsdirip, kalanlari yeni bir Array olarak yazdiran bir method yaziniz



24 TEMMUZ 2021
DERS 23

ArrayLists

Mehmet BULUTLUOZ
Elk.Elektronik Yuk.Muh.

Onceki Dersten Aklimizda Kalanlar

- 1- Multi Dimensional Arrays : çok katlı array, eğer bir array'in içinde element olarak yine array'ler bulunuyorsa buna MDA denir.
- 2- distaki array outer, içteki array'ler ise inner array olarak adlandırılır.
- 3- MDA kaç katlı ise declaration sırasında o kadar [] yazmamız gereklidir
- 4- int arr[][] = new int [3][2]; bu declaration'da ilk [3] outer array'in içinde kaç tane inner array olduğunu gösterir, ikinci [2] inner array'lerin uzunluğunu verir. Bu şekilde bir declaration ile MDA oluşturulursa inner array'ler farklı uzunlukta olamaz, hepsinin uzunluğunun 2 olması şarttır... [[0,0],[0,0],[0,0]]
- 5- Alternatif olarak tüm array elementlerini yazarak MDA oluşturulabilir.
int arr [][] = {{4,1}, {5,1,3,6},{2,7,8,9,12}} bu durumda inner array'lerin uzunlukları aynı olmak zorunda değildir. arr[1] sadece 1 kosalı parantez kullanırsak outer array'in 1 index'indeki inner array'ını ifade eder. arr[2][1] → outer array'in 2 index'indeki inner array'in 1 index'indeki elementini verir = 7
- 6- MDA'de her bir elemente ulaşmak veya kullanmak istiyorsak nested for loop kullanmalısınız
- 7- MDA direkt yazdırmak istersek Arrays.deepToString(arr) kullanılabilir. Eğer deepToString kullanılmaz, yerine toString kullanılırsa inner array'lerin referanslarını yazdırır

Socrative Quiz

- 1) <https://www.socrative.com/> adresine gidin
- 2) **Login** butonuna basin
- 3) **Student Login** butonuna basin (veya
<https://b.socrative.com/login/student/>)
- 4) Room Name **BULUTLUOZ** yazın
- 5) Isminizi yazın
- 6) **Done** butonuna basin

Sure : 15 Dakika

ArrayList

ArrayList nedir?

ArrayList length'i esnek olan bir Array'dir

ArrayList'e neden ihtiyac duyuyoruz?

- Biz array olustururken length'in en basta belirlemek zorundayız ve daha sonra length'ini degistirememiz.Bu durum bizim esnek calismamiza engel olur.
- Bir array'in uzunlugunu degistirmek istedigimizde yeni bir array olusturmamız gereklidir, ArrayList de gerekmeyez.
- Bir array'den bir eleman silmek istedigimizde yeni bir array olusturmamız gereklidir, ArrayList de gerekmeyez.

ArrayList

ArrayList olusturma

```
ArrayList<String> list1 = new ArrayList<String>();
```

```
ArrayList<String> list2 = new ArrayList<>();
```

```
List<String> list3 = new ArrayList<>(); En çok bu kullanılır
```

```
ArrayList<String> list4 = new List<>();
```

Compile Time Error verir, eşitliğin sağ tarafında ArrayList kullanmak zorundayız

ArrayList'i nasıl yazdırırız?

ArrayList'i ekrana yazdirmak çok kolaydır.

```
System.out.println(list3);
```

ArrayList Method'lari

1) add()

add() method ArrayList'e eleman eklemek icin kullanilir

Ornek :

```
List<String> hayvan = new ArrayList<>();
```

A) add() method'u index olmadan calisabilir

```
hayvan.add("kedi"); // [kedi]
```

```
hayvan.add("yilan"); // [kedi, yilan]
```

B) add() method'u index ile de calisabilir

```
hayvan.add(1, "kartal"); // [kedi, kartal, yilan]
```

```
hayvan.add(0, "sinek"); // [sinek, kedi, kartal, yilan]
```

```
hayvan.add(1, "aslan"); // [sinek, aslan, kedi, kartal, yilan]
```

```
System.out.println(hayvan); // [sinek, aslan, kedi, kartal, yilan]
```

ArrayList Method'lari

2) size()

size() method ArrayList'de kaç eleman olduğunu gösterir.

Ornek :

```
List<String> hayvan = new ArrayList<>();
```

```
System.out.println(hayvan.size()); // 0
```

```
hayvan.add("kedi"); // [kedi]
```

```
hayvan.add("yilan"); // [kedi, yilan]
```

```
System.out.println(hayvan.size()); // 2
```

3) isEmpty()

isEmpty() method'u ArrayList boş ise true, boş değilse false dondurur

ArrayList Method'lari

4) remove()

remove() method'u ArrayList'den belli bir elemani silmek icin kullanilir.

A) remove(index) kullanarak. Size'dan buyuk index yazilrsa exception verir.
Index'li remove() methodu ArrayList'de verilen index'deki elemani siler.

```
List<String> hayvan = new ArrayList<>();  
hayvan.add("kedi"); // [kedi]  
hayvan.add("yilan"); // [kedi, yilan]  
hayvan.remove(1); // index'i 1 olan elemani siler  
System.out.println(hayvan); // [kedi]
```

NOT: remove(index) method'u silinen elemani dondurur. Yani method'u
System.out.println() icinde kullanırsak silinen elemani ekrana yazdırır.

```
System.out.println(hayvan.remove(1)); //yilan
```

ArrayList Method'lari

B) `remove("eleman")` index'i degil elemani kullanırsak kullandığımız elemanın ilk kullanıldığı yeri bulur ve siler.

```
List<String> hayvan = new ArrayList<>();  
    hayvan.add("kedi"); // [kedi]  
    hayvan.add("yilan"); // [kedi, yilan]  
    hayvan.add("kedi"); // [kedi, yilan, kedi]  
    hayvan.remove("kedi");  
    System.out.println(hayvan); // [yilan, kedi]
```

Not: Index'siz remove() method'u true veya false dondurur.

```
System.out.println(hayvan.remove("kedi")); //true yani kedi eleman olarak vardı ve sildim
```

```
System.out.println(hayvan.remove("tavsan")); // false yani tavsan eleman olarak yoktu  
dolayısıyla silemedim
```

ArrayList Method'lari

5) set()

set() methodu ArrayList'de var olan bir elemani degistirmeye yarar

```
List<String> hayvan = new ArrayList<>();
```

```
hayvan.add("kedi"); // [kedi]
```

```
hayvan.add("yilan"); // [kedi, yilan]
```

```
hayvan.set(1, "tavsan");
```

```
System.out.println(hayvan); // [kedi, tavsan]
```

NOT: set() method'u add() method'u yerine kullanilamaz .
Olmayan bir index ile set() kullanilirsa exception verir.

```
hayvan.set(2, "aslan"); // IndexOutOfBoundsException
```

ArrayList Method'lari

6) get(index)

get() methodu ArrayList'deki istenen indexdeki elemani dondurur.

```
List<String> hayvan = new ArrayList<>();  
hayvan.add("kedi"); // [kedi]  
hayvan.add("yilan"); // [kedi, yilan]
```

```
System.out.println(hayvan.get(0)); // kedi
```

```
System.out.println(hayvan.get(1)); // yilan
```

ArrayList Method'lari

7) contains()

contains() methodu ArrayList'de bir elemanın var olup olmadığını kontrol eder. Eleman varsa true, yoksa false return eder.

```
List<String> hayvan = new ArrayList<>();  
hayvan.add("kedi"); // [kedi]  
hayvan.add("yilan"); // [kedi, yilan]
```

```
System.out.println(hayvan.contains("kedi")); // true
```

```
System.out.println(hayvan.contains("tavsan")); // false
```

ArrayList Method'lari

- 8) **Collections.sort()** : sort() methodu ArrayList'deki elemanlari kucukten buyuge veya alfabetik siraya gore dizer.

```
List<String> hayvan = new ArrayList<>();  
hayvan.add("yilan"); // [yilan]  
hayvan.add("kedi"); // [yilan, kedi]  
hayvan.add("tavsan"); // [yilan, kedi, tavsan]
```

```
System.out.println(hayvan); // [yilan, kedi, tavsan]
```

```
Collections.sort(hayvan);  
System.out.println(hayvan); // [kedi, tavsan, yilan]
```

ArrayList Method'lari

9) equals()

equals() methodu iki listteki ayni indexteki elemanların ayni olup olmadığını kontrol eder. Aynı indexteki tüm elemanlar aynı ise true return eder, farklı ise false return eder

```
List<String> first = new ArrayList<>();  
List<String> second = new ArrayList<>();  
System.out.println(first.equals(second)); // true  
  
first.add("a"); // [a]  
System.out.println(first.equals(second)); // false  
  
second.add("a"); // [a]  
System.out.println(first.equals(second)); // true  
  
first.add("b"); // [a,b]  
second.add(0,"b"); // [b,a]  
System.out.println(first.equals(second)); // false
```

ArrayList Method'lari

10) clear()

clear() methodu ArrayList'teki tum elemanlari siler.
Return type'i void'dir, hic bir sey donmez

```
List<String> hayvan = new ArrayList<>();
hayvan.add("yilan"); // [yilan]
hayvan.add("kedi"); // [yilan, kedi]

System.out.println(hayvan.isEmpty()); // false
System.out.println(hayvan.size()); // 2

hayvan.clear();

System.out.println(hayvan.isEmpty()); // true
System.out.println(hayvan.size()); // 0
```

Sorular

ArrayList

- 1) Elemanlari A, C, E, ve F olan bir String ArrayList olusturup ekrana yazdiriniz.
- 2) indexsiz **add()** methodunu kullanarak, B'yi ekleyiniz.
index'li **add()** methodunu kullanarak, L'yi 1 numarali index'e ekleyiniz.
ArrayList'i ekrana yazdiriniz, list goyle olmali; A, L, C, E, F, B.
- 3) **set()** methodu kullanarak, E'yi D yapiniz.
ArrayList'i ekrana yazdiriniz, list goyle olmali; A, L, C, D, F, B.
- 4) **remove()** methodu kullanarak, F'yi siliniz.
ArrayList'i ekrana yazdiriniz, list goyle olmali; A, L, C, D, B.
- 5) **sort()** methodu kullanarak, elemanlari alfabetik siraya diziniz.
ArrayList'i ekrana yazdiriniz, list goyle olmali; A, B, C, D, L.
- 6) **contains()** methodu kullanarak, L'nin list'de var oldugunu ve M'nin list'de var
olmadigini dogrulayiniz.
- 7) **size()** methodu kullanarak, list'in kag eleman oldugunu ekrana yazdiriniz.
- 8) **clear()** methodu kullanarak, list'deki tum elemanlari siliniz.
- 9) **isEmpty()** methodu kullanarak, list'deki tum elemanların silindigini dogrulayiniz

Array'i ArrayList'e Cevirmek

```
String [ ] arr = {"tavsan", "serce"};
```

```
List<String> list = Arrays.asList(arr);
```

Uzunlugu degistirilemeyen bir list'e cevirir. Yani; yeni olusturulan listte add(), remove() ve clear() methodlarini kullanamazsiniz. Exception

```
System.out.println(list.size()); // 2
```

```
System.out.println(list); // [tavsan, serce]
```

NOT: Eger array'deki bir elemani degistirirseniz list'teki eleman da otomatik olarak degisir. Listteki bir elemani degistirirseniz array de otomatik olarak degisir.

```
list.set(1, "test"); // [tavsan, test]
```

```
arr[0] = "new"; // [new, test]
```

```
System.out.println(Arrays.toString(arr)); // [new, test]
```

```
System.out.println(list); // [new, test]
```

ArrayList'i Array'e Cevirmek

```
List<String> list = new ArrayList<>();  
list.add("tavsan");  
list.add("horoz");  
System.out.println(list); // [tavsan,horoz]
```

1.yontem

```
String arr[ ] = list.toArray(new String[0]);
```

```
System.out.println(arr.length); // 2  
System.out.println(Arrays.toString(arr)); // [tavsan,horoz]
```

2.yontem

```
Object arr[ ] = list.toArray();
```

```
System.out.println(arr.length); // 2  
System.out.println(Arrays.toString(arr)); // [tavsan,horoz]
```

For Each Loop Enhanced (Gelistirilmis) For Loop

Faydalari:

Kodun daha okunabilir olmasini saglar. Hata yapma ihtimalini azaltir.

```
public static void main(String args[ ]){  
  
    int arr[ ]={12,13,14,44};  
  
    for( int i: arr) {  
        System.out.print(i + " ");  
    }  
}
```

```
public static void main(String args[ ]){  
    ArrayList<String> list=new  
    ArrayList<String>();  
    list.add("Ali");  
    list.add("Veli");  
    list.add("Can");  
  
    for( String s : list) {  
        System.out.print(s + " ");  
    }  
}
```

For Each Loop

Soru 1:

Bir integer array olusturunuz ve bu array'deki tum sayilarin carpimini For-each loop kullanarak bulunuz. Sonucu ekrana yazdiriniz.

Soru 2:

Bir integer list olusturunuz ve bu list'deki tum sayilarin karesinin toplamini For-each loop kullanarak bulunuz. Sonucu ekrana yazdiriniz.

Soru 3:

iki String array olusturunuz ve bu array'lerdeki ortak elemanlari For-each loop kullanarak bulunuz. Sonucu ekrana yazdiriniz.

Ortak eleman yoksa ekrana "Ortak eleman yok" yazdiriniz

Soru 4:

Bir String olusturunuz, bu String'deki character'leri for-each loop kullanarak yazdiriniz. *ipucu: split()*

Constructor

(Java object'leri nasıl oluşturur ?)