



13 SUBAT 2021  
DERS 1

Genel Hatırlatmalar  
Java Giriş

Mehmet BULUTLUOZ  
Elk.Elektronik Muh.

---

---

## Genel Hatırlatmalar



1. Derse Katılım
2. Dersi Dikkatli Dinleme
3. Derste Aktif Olma
4. Anlamadıklarını Sor
5. Ödev Yapma (Kod yazma araba kullanma gibidir)
6. Her Dersten Sonra Tekrar

TECHPROED

---

## Genel Hatırlatmalar

- 6. Basari = Egitim + Calisma
- 7. Grup calismalari yapin, En iyi ogrenme yontemi ogretmektir
- 8. Mentoring toplantilarini kacirmayin
- 9. Maillerinizi gunluk kontrol edin
- 10. Yoklama yapiliyor zooma isminizle girin
- 11. Customer service Selehattin Yasar +1 917 768 74 66
- 12. Teknik destek Nazmi Mert Slack @Nazmi Mert

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

~ CHINESE PROVERB

TECHPROED

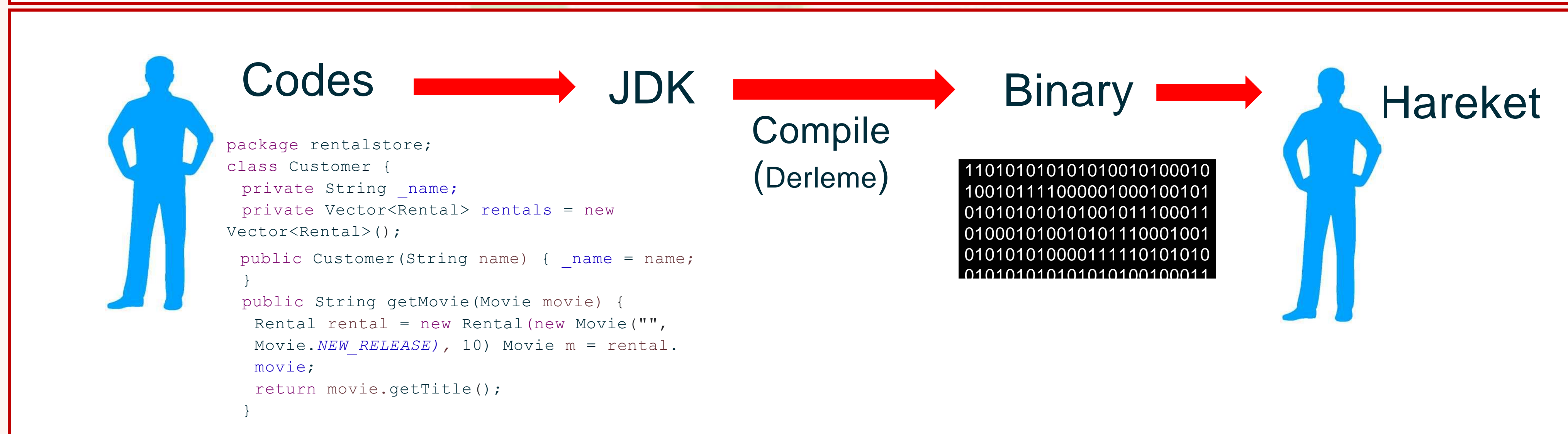
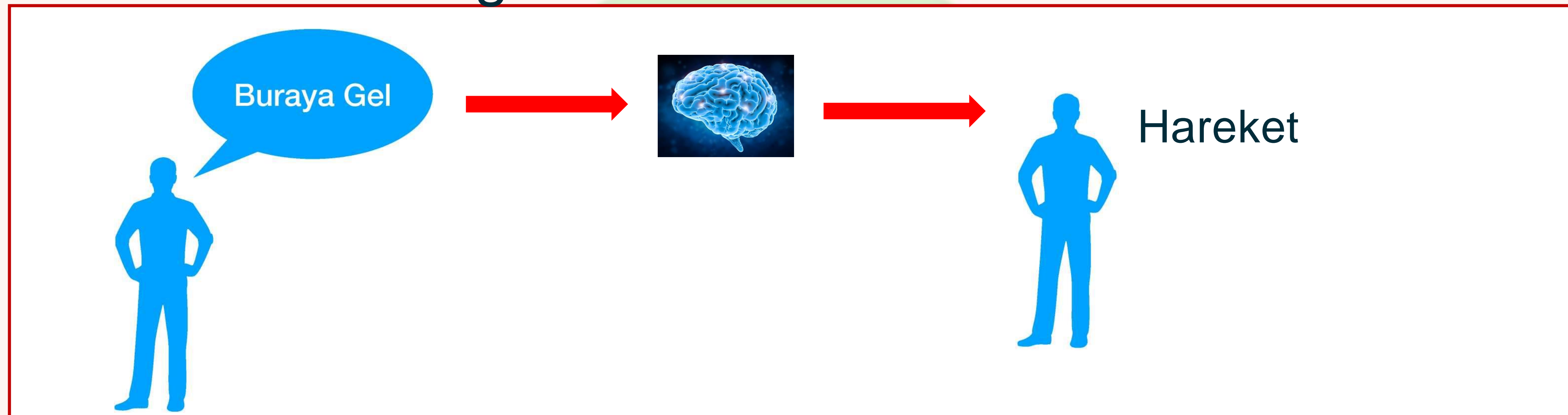
# Ders İsteyisi Bilmeniz Gerekenler



1. Mailleri kontrol edin
2. Mesajlaşma için slack kullanalım
  - İki slack kanalımız var
  - Direk mesaj
  - Kod paylaşma (**snippet**)
  - Mesaj silme ve edit
  - Pin yapma
3. Google Classroom
  - Tüm ders notları, zoom linki ve videolar Google Classroom'dan paylaşılabilecek
  - Maillerinize davetiye gönderildi



# Programlama Dili Nedir ?



## Nicin Java ?

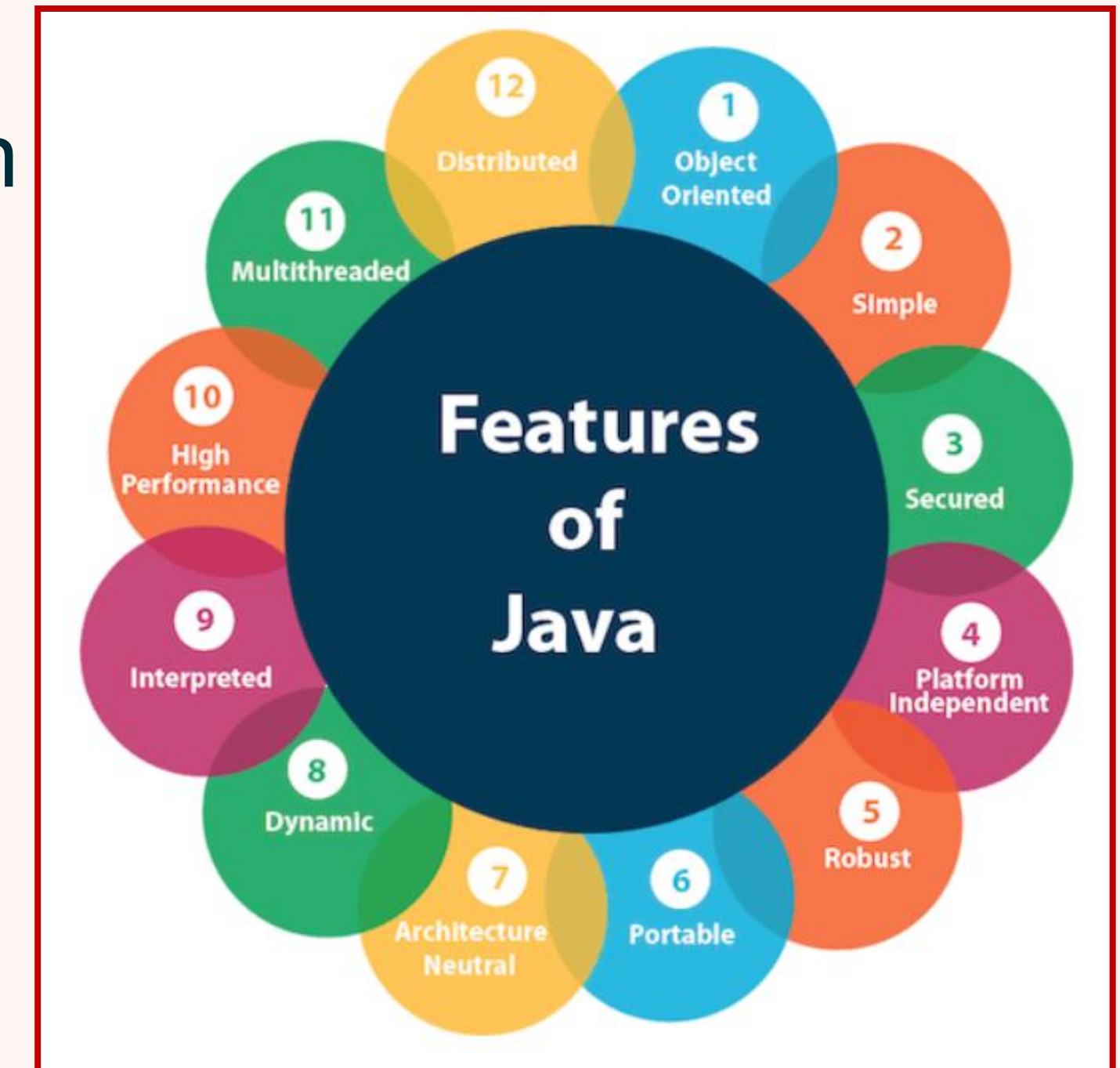
1. Ogrenmesi kolay
2. Dunyada en cok kullanılan programlama dili

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

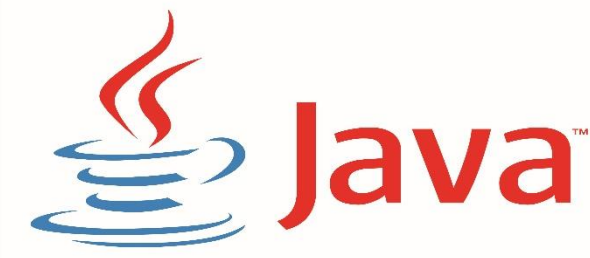
Bunlardan bazıları şu şekildedir:

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

3. Java “Object Oriented Programming (OOP)” Language’ dir.







# Object Oriented Programming Nedir?



Objects



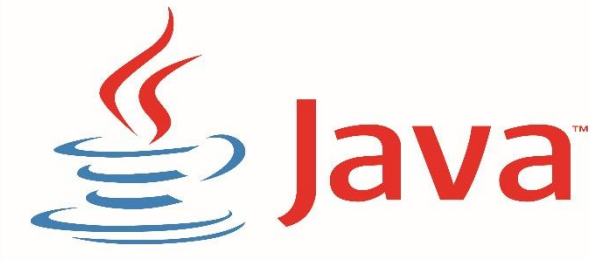
Application  
(Urun)

1- Feature (Fields veya Variables)

Pasif ozellik (renk,sekil,isim)

2- Functionality (Method)

Aktif ozellik (tasima,degistirme)



## Bir Object Nasıl Olusturulur?



Class(Object Kalibi)

Field  
(Variables)      Method  
(Functions)



Object



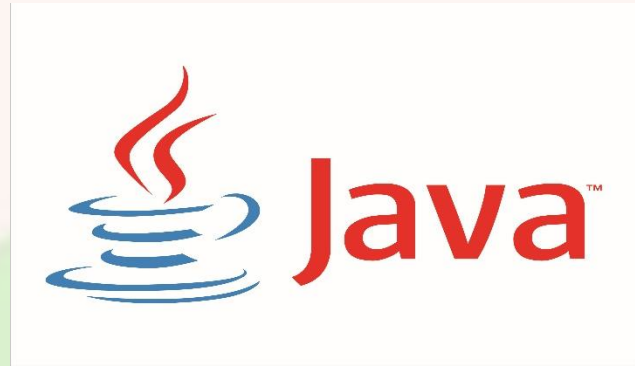
Birden fazla Obje birlestirilir



Application

TECHPROED





# Object Nasıl Kullanilir ?



# Ogretmen



# Ogrenci



# Personel

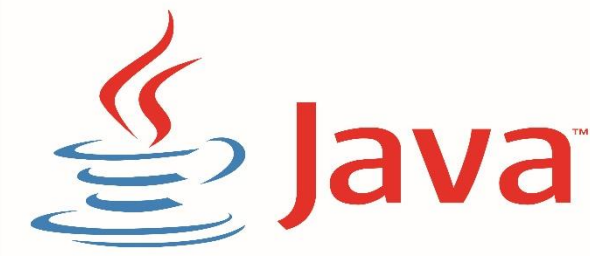
09:00	TÜRKÇE-1
09:30	MATEMATİK-1
10:00	TÜRKÇE-2
10:30	MATEMATİK-2
11:00	TÜRKÇE-3
11:30	MATEMATİK-3
12:00	TÜRKÇE-4
12:30	MATEMATİK-4
13:00	İYEP TÜRKÇE

# Ders

[illegible]

# Not





## Bir Class Hangi Bolumlerden Olusur?

```
public class BeveragePage {
```

```
    private String beveragePageTitleXpath = "//h1[text() = 'Beverage']";  
    private String buyNowButtonXpath = "//a[contains(text(), 'Buy Now')]";  
    private String searchBarXpath = "//input[@id='searchByCode']";
```

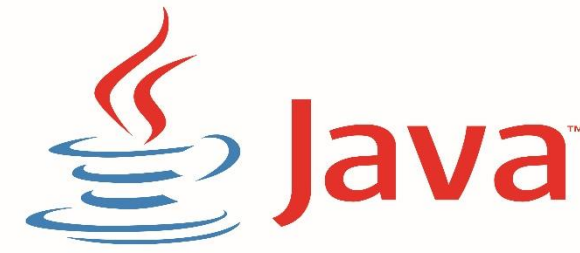
→ **Field / Variables**

```
    @Step("Verify to be on beverage page")  
    public boolean isOnBeveragePage() {  
        return $x(beveragePageTitleXpath).shouldHave(Condition.text("Beverage")).isDisplayed();  
    }
```

→ **Method**

```
    @Step("Select a specific product")  
    public GuestSelectionPage selectAProductByCode(ProductDetails productDetails) {  
        $x(searchBarXpath).sendKeys(productDetails.getVariantDetails().getVariantCode());  
        $x(searchBarXpath).pressEnter();  
        $x(buyNowButtonXpath).click();  
        return new GuestSelectionPage();  
    }
```

→ **Method**



## Class Olusutururken Kullanilan Keyword'ler Nelerdir?

```
public class MyFirstClass { }
```

1            2            3            4

1 **public** : Access Modifier (Erisim duzenleyici) : class'a kimlerin erisebilecegini belirler

**Default** : Sadece bulunduгу Package'den kullanilabilir

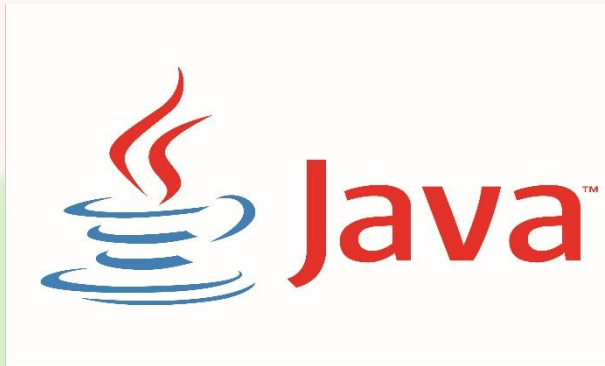
2 **class** : Yazdigimiz kodun class oldugunu belirtir

3 **MyFirstClass** : Olusturdugumuz class'in ismidir. Class'a istedigimiz ismi verebiliriz ancak isim verirken genelde class'da yapilan 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)** : { } arasinda kalan kodlarimizi yazdigimiz bolumdur



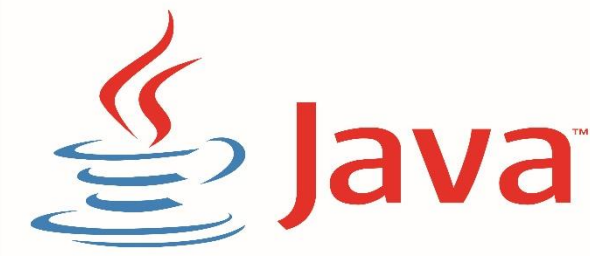


## Method Nasıl Olusuturulur ?

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

- 1 **public** : Access Modifier (Erisim duzenleyici):methoda'a kimlerin erisebilecegini belirler  
**Private**: Sadece bulunduğu class'da kullanılabilir  
**Protected** : Sadece 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)** : { } arasinda kalan kodlarimizi yazdigimiz bolumdur





## Main Method

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



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

Araba



Motor

Java Project



Main Method



# Yorum Cümlesi (Comment) Nasıl Eklenir ?

- **Comments** : Java tarafından çalıştırılmayan, amacı kodların açıklanması veya bir konuda bilgi vermek olan cümlelerdir
- Genelde iki kullanım vardır



1) Tek satırlık comment :

// (**double slash**) dan sonra yazılan satır çalıştırılmaz  
// toplamak istediğim sayıları gireceğim vb..

2) Çok satırlık comment

/\* ile \*/ arasına yazılan tüm satırlar Java tarafından comment kabul edilir



/\*

java tarafından çalıştırılmayacak  
satırları buraya yazabilirsiniz

\*/

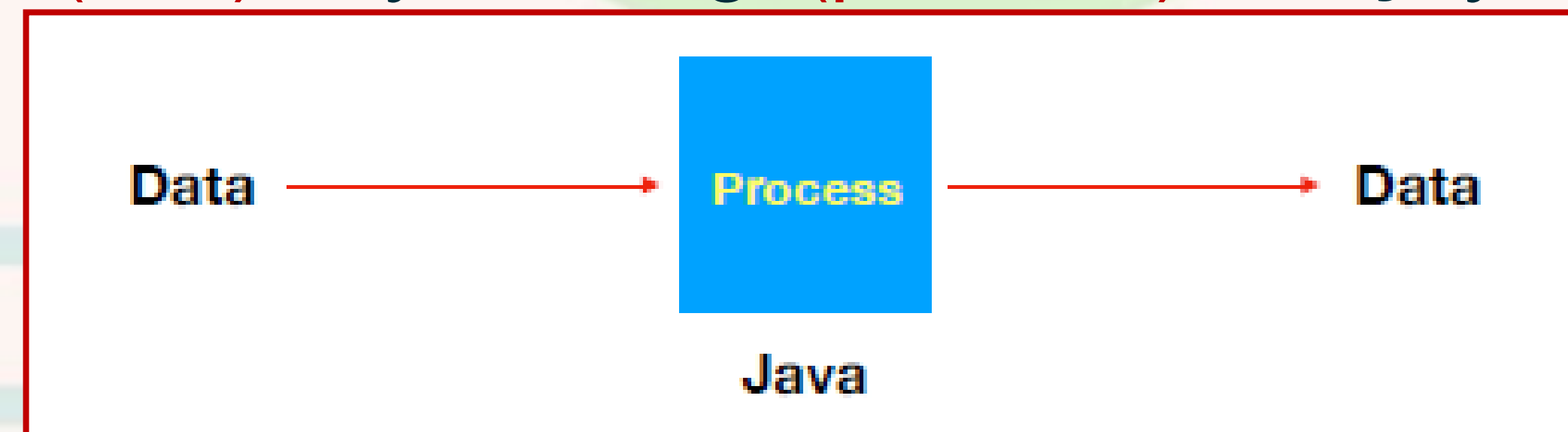


# Data

**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

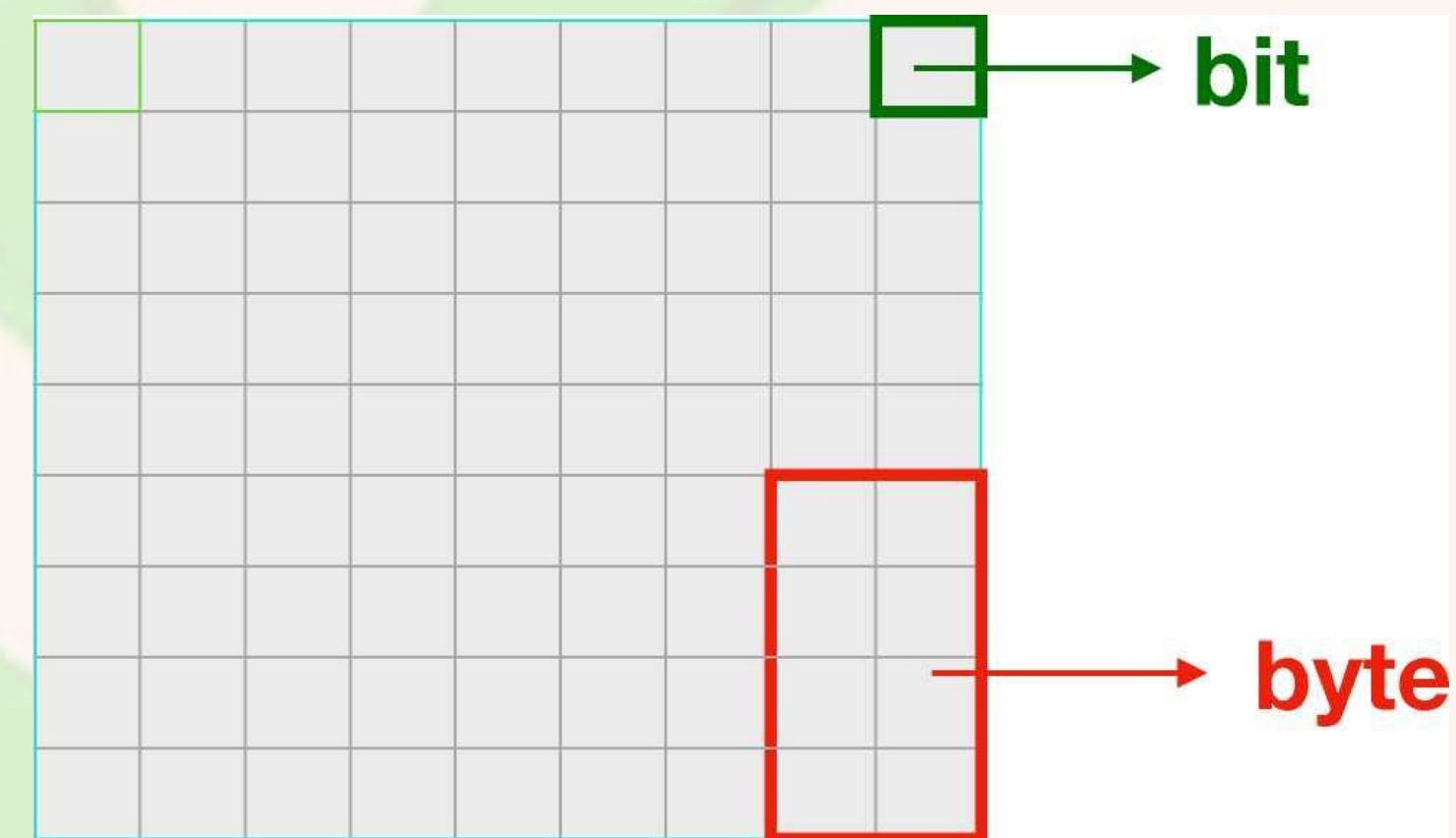
Java'nin kullandığı (**use**) veya ürettiği (**produce**) her şey data'dır.



# Bit

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

Note: 8 bit =1 byte

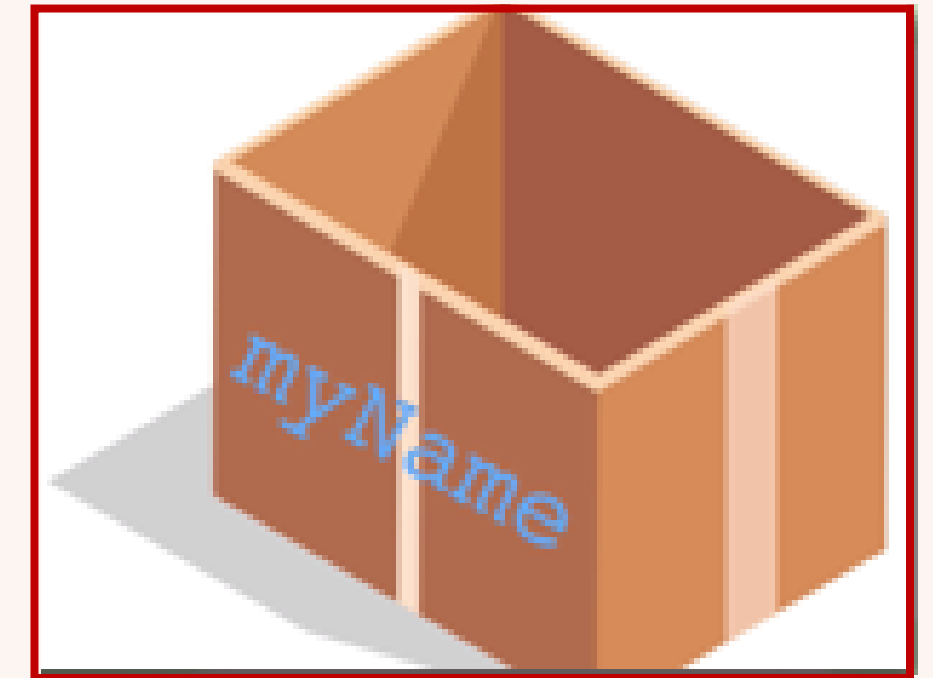


Memory

# Variables Olusturma

## Declaration

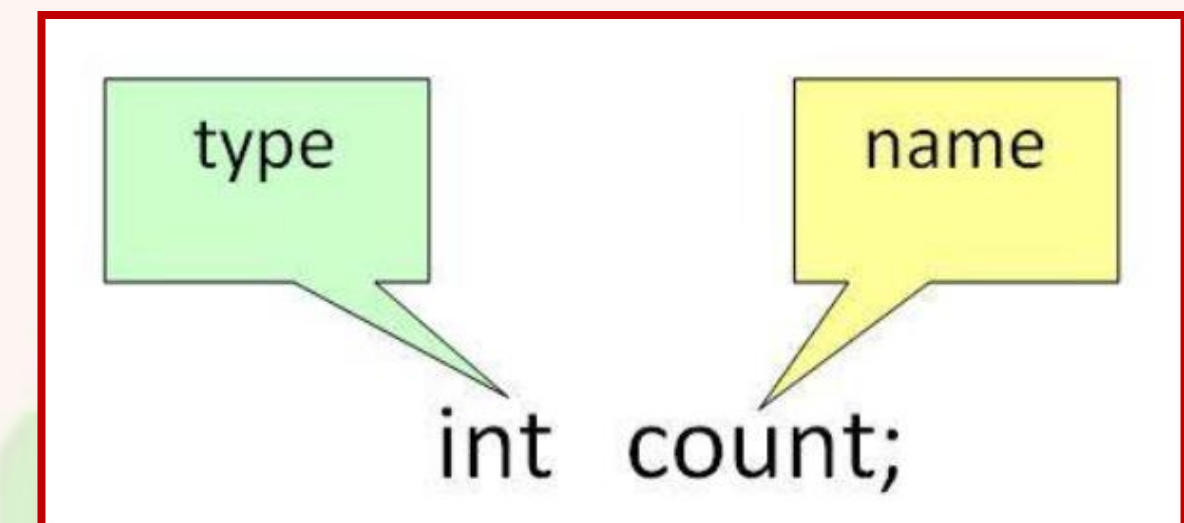
**Variable** bellekte (**memory**) ayrılmis olan alanin (**reserved area**) adidir. Variable icinde deger saklayan bir konteynirdir (**container**). Bir deęişkende saklanan deęer, program yürütölürken deęiştirilebilir.



- **Variable Olusturma (Declaration) :**  
Java'da, tüm deęişkenler kullanılmadan önce bildirilmelidir.

Data Tipi (Type) VariableAdi (Name);

```
int yas ;  
char ilkHarf ;
```





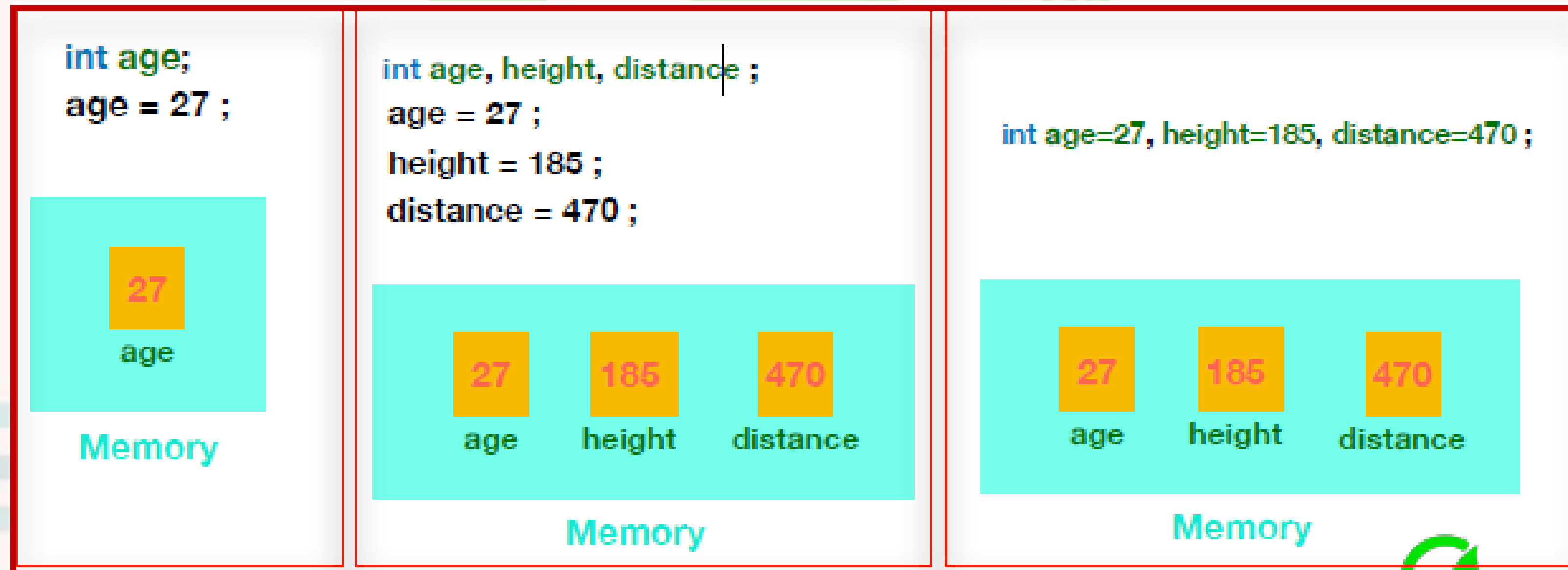
# Variables Deger Atama Assignment

Data Tipi (Type) VariableAdi (Name);

```
int yas=27 ;  
char ilkHarf='A' ;
```



- Not : Declaration ve Assignment asagidaki gibi farkli sekillerde yapilabilir



---

# Primitive Data Type

Java'da iki data tipi kullanılmaktadır

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

## Primitive Data Types

1) **boolean** Data Type: true and false. **1 bit.**

boolean isExpensive = true;

boolean isCold = false;

2) **char** Data Type: single character. **16 bit**

char letter = 'a';

char digit = '3';

***Note:** char degerlerini single quote arasina koyun*

---

---

# Primitive Data Types

3) **byte** Data Type: -128 den 127'e (dahil). **8 bit**

```
byte age = 73;
```

4) **short** Data Type: -32,768 den 32,767'e (dahil = inclusive). **16 bit**

```
short ilceNufusu = 27,324;
```

5) **int** Data Type: -2,147,483,648 den 2,147,483,647'e (dahil = inclusive). **32 bit**

```
int profitOfApple = $1,342,345,000;
```

---



---

# Primitive Data Types

6) **long** Data Type: -9,223,372,036,854,755,808 den ,223,372,036,854,755,807'e **64 bit**

```
long profitOfApple = $1,342,345,000;
```

7) **double** Data Type: Buyuk ondalik sayi. **64 bit** double

```
double doubleVar2 = -2.12345679078000000000123
```

8) **float** Data Type: Kucuk ondalik sayi. **32 bit**

```
float floatVar2 = -2.1 2 3 4 5 6 f;
```

**Note:** float sayilarin sonunda "f" yazilmalidir, yazilmazsa java sayiyi double kabul eder

---

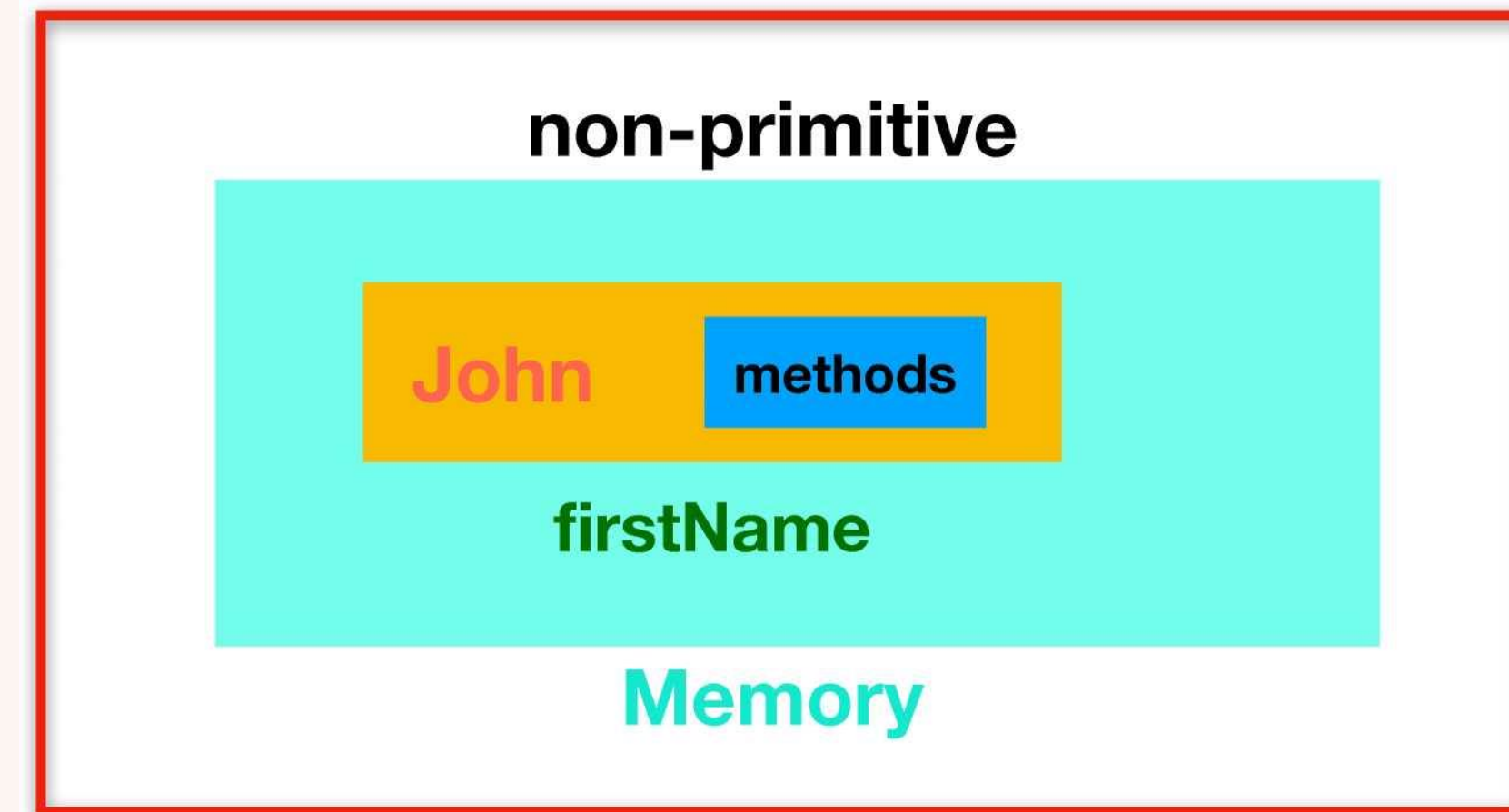
# ASCII Table

[illegible]

# Non-Primitive Data Type

## String Data Type:

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



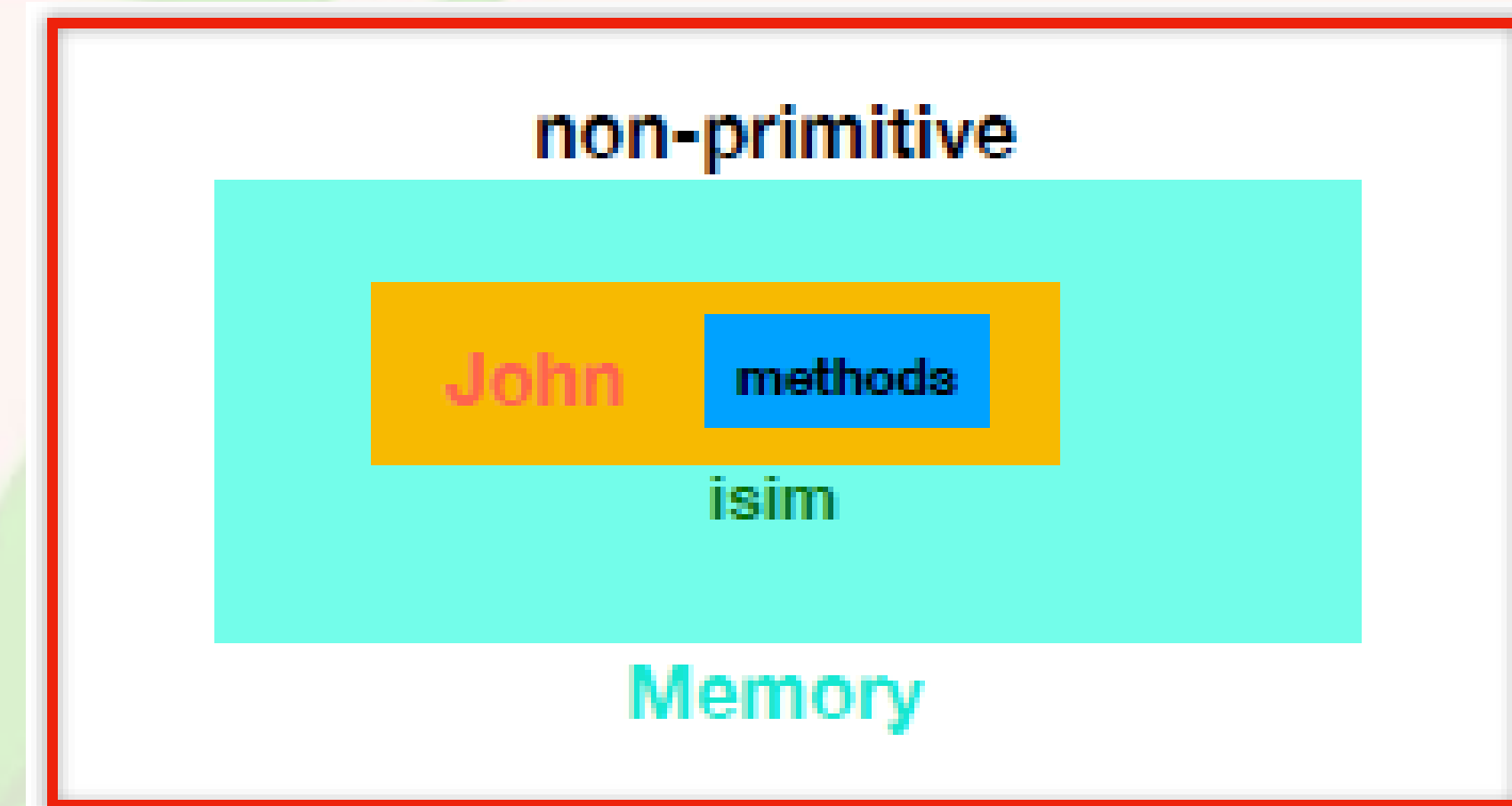
```
String okulAdi = "Yildiz Koleji";  
String telNo = "5321234567";  
String ilkHarf = "A";
```

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

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



# Primitive VS Non-Primitive Data Types



- 1) Primitive'ler sadece value içerir, non-primitive'ler value and methodlar içerir.
- 2) Primitive'ler küçük harf ile, non-primitive'ler büyük harf ile baslar.
- 3) Primitive'leri Java oluşturur biz primitive olusturamayiz.  
Non-primitive'leri biz de olusturabiliriz, Java da olusturabilir. Mesela String'i Java olusturmudur.
- 4) Primitive'lerin buyuklukleri data type'ing gore degisir, non-primitive'lerin hepsi ayni buyukluktedir.

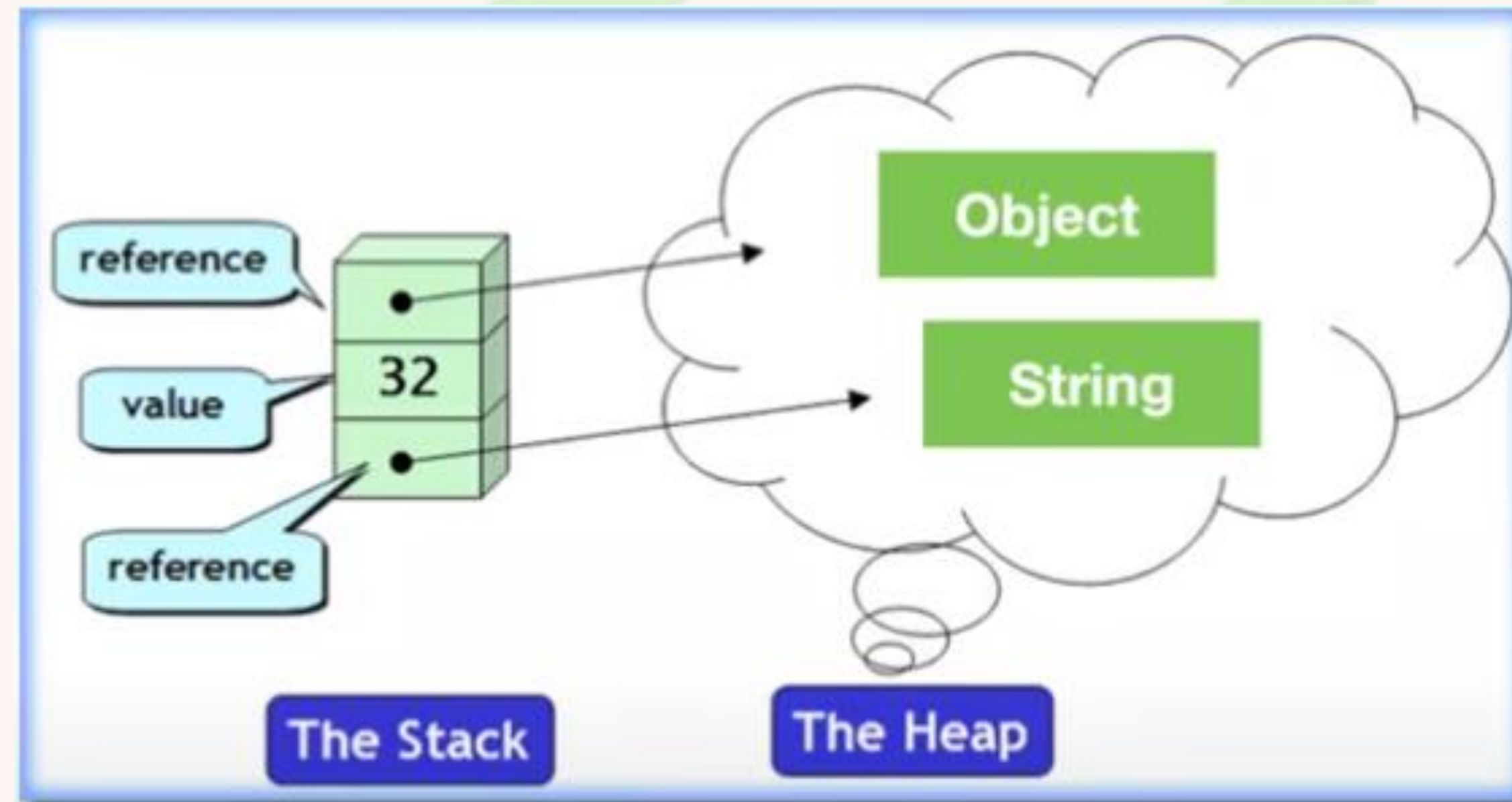
---

## Variable ve Method'lar Nasıl Adlandırılır

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

# Memory (Hafıza) Kullanımı

Javada kullanılan iki hafıza vardır



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

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

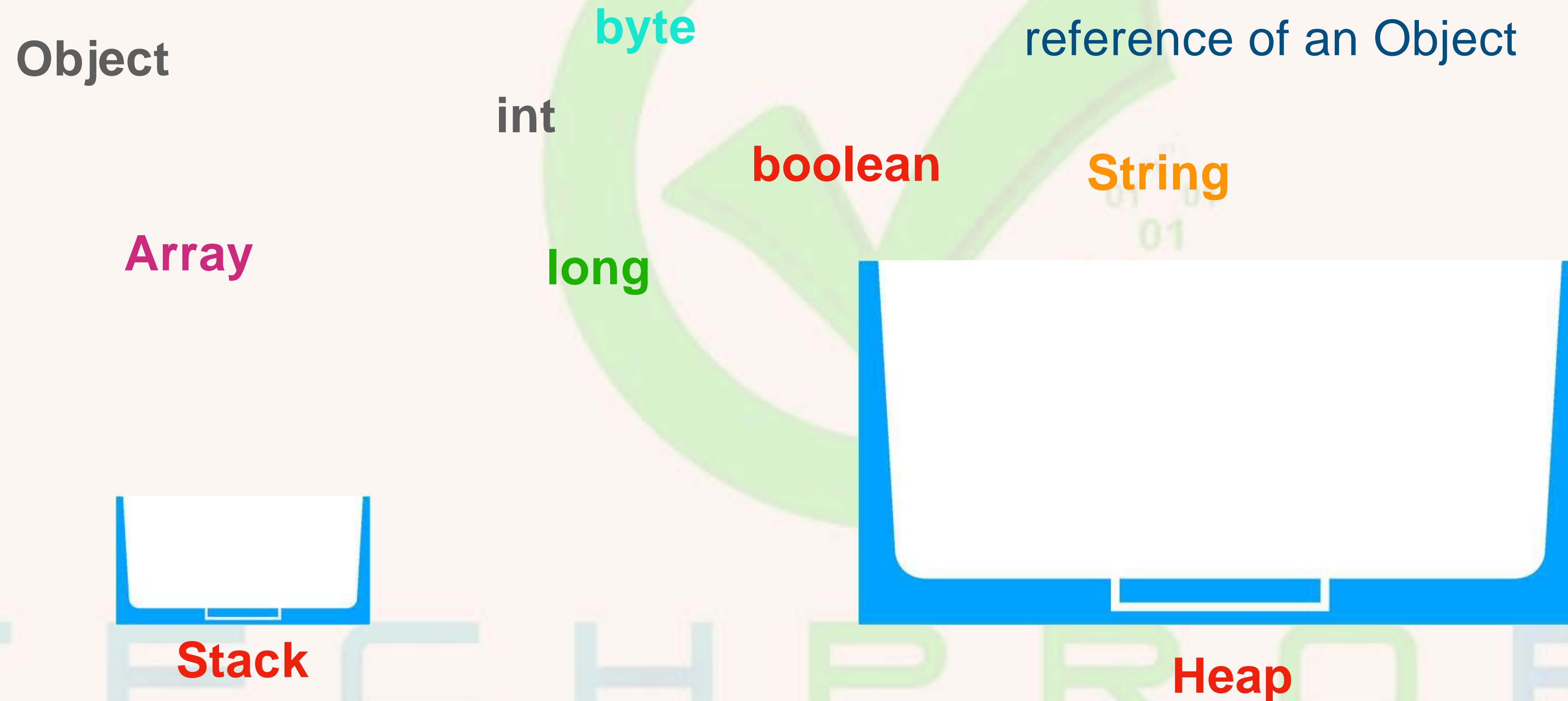
Stack => small    Heap => huge



---

# Memory (Hafıza) Kullanımı

Ornek : Yazili data tiplerini olmasi gereken Memory'icine yerlestirin



---

# Eclips Kullanim

## 1- Proje olusturma

File -- New -- Project -- (Java Project) Next -- java2021turkce -- finish

## 2- Package (paket) olusturma

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

## 3- Class olusturma

day01variables dosyasina sag click -- New -- Class -- Variables01 --

**public static void main(String[] args)** kutusunu click yapin -- finish

Orn 1: Farkli veri tiplerinde variable olusturalim ve yazdiralim

Orn 2: Aralarina + koyarak 2 degiskeni yazdiralim

---

---

# Kullanıcıdan Deger Alma

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

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

Bu tur isimlendirmelerde genel kurallara uymamiz kodumuzun anlasilabilir olmasi acisindan faydali olacaktir.

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

Kullaniciya girmesini istedigimiz degerler icin aciklayici bilgi vermeliyiz.

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

---



# Kullanıcıdan Deger Alma

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

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

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

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

---

## Kullanıcıdan Deger Alma

**Soru 1)** Kullanıcıdan iki tamsayı alıp bu sayıların toplam,fark ve carpimlerini yazdırın

**Soru 2)** Kullanıcıdan karenin bir kenar uzunlugunu alin ve karenin cevresini ve alanini hesaplayip yazdırin

**Soru 3)** Kullanıcıdan yarıcap isteyip cemberin cevresini ve dairenin alanini hesaplayip yazdırin

**Soru 4)** Kullanıcıdan dikdortgenler prizmasının uzun, kısa kenarlarını ve yuksekligini isteyip prizmanın hacmini hesaplayip yazdırin

**Soru 5)** Kullanıcıdan ismini ve soyismini isteyip asagidaki sekilde yazdırin

Isminiz : Mehmet

Soyisminiz : Bulutluoz

Kursumuza katiliminiz alinmistir,tesekkur ederiz

**Soru 6)** Kullanıcıdan ismini ve soyismini alıp aralarında bir bosluk olusturarak asagidaki sekilde yazdırin

Isim – soyisim : Mehmet Bulutluoz

**Soru 7)** Kullanıcıdan ismini alıp isminin bas harfini yazdırin.

---

---

# Data Casting

## Veri Sinifi Degistirme

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

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

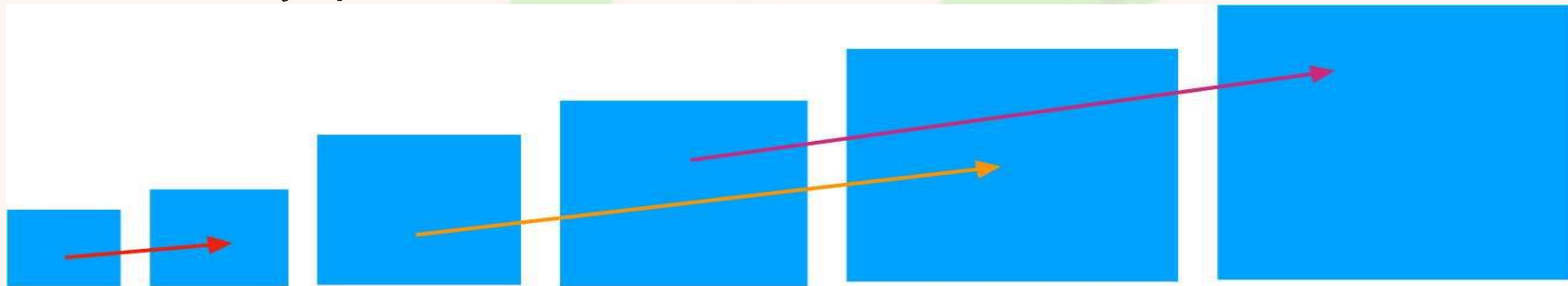
---



# Data Casting

## 1) Auto Widening (Otomatik Genisletme)

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



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

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)

- Genis veri tipinden daha dar bir veri tipine gecmek iztedigimizde 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

```
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  
    }  
}
```

---

## Data Casting

**Soru 1 )** byte veri tipinde bir degisken olusturun, short,int,float ve double data tiplerinde birer degisken olusturup adim adim widening yapin ve yazdirin

**Soru 2 )** int veri turunde bir degisken olusturun ve adim adim narrowing yapin ve yazdirin

**Soru 3 )** Float data turunde bir variable olusturun ve yazdirin

**Soru 4 )** double 255.36 sayisini int'a ve sonra da olusturdugunuz int sayiyi byte'a cevirip yazdirin

**Soru 5 )** int 2 sayiyi birbirine boldurun ve sonucu yazdirin

**Soru 6 )** int bir sayiyi double bir sayiya bolun ve sonucu yazdirin

**Soru 7 )** Farkli data tipleri ile islem yapip, sonuclarini yazdiralim

TECHPROED

---



## Bir Variable'in Degeri Nasil Artirilir (Increment)?

```
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++ ;
```

**= ?**

## Bir Variable'in Degeri Nasıl Azaltılır (Decrement)?

```
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 azaltilir

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

Output : 16

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

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

Output : 15



# Javada Matematiksel Operatorler

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

Ornek 1

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

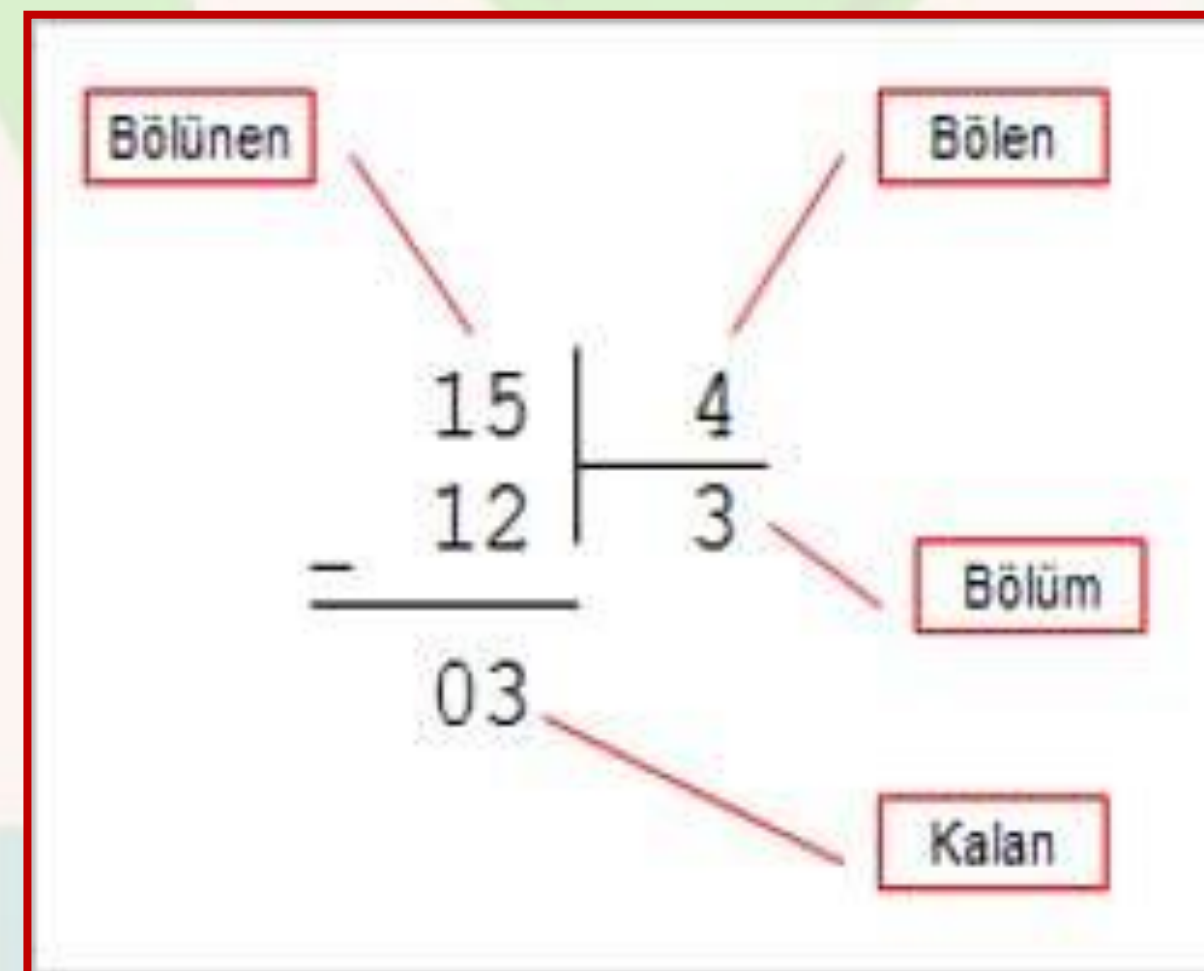
Ornek 2

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

# Modulus %

Modulus islemi bir bolme isleminde kalan sayiyi bize verir

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



## Modulus %

**Soru 1)** Kullanıcıdan 4 basamaklı bir sayı alın ve rakamlar toplamını bulup yazdırın

**İpucu 1:** Sayı % 10 => Bize son basamagi verir

$$538 \% 10 = 8$$

**İpucu 2:** Int Sayı /10 => Bize son basamak haric sayiyi verir

```
int sayi=538;
```

**sayi=sayı / 10** => sayiya 53 degerini atar



# Wrapper Class

Java primitive data turleri ile methodlari kullanabilmemiz icin Wrapper class'lari olusturmudur. Character,Byte,Integer,Short,Float,Double primitive data turleri icin olusturulan wrapper class'lardir.

Data Tipleri Icin Max. ve Min. Degerleri Bulma

```
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);  
}
```

int Data Type: -2,147,483,648 den 2,147,483,647'e

short Data Type: -32,768 den 32,767'e

byte Data Type: -128 den 127'e (dahil).

# Concatenaion

## (String Datalari Birlestirme)

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

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

HelloWorld

Hello World

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

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

---

# Concatenaion (String Datalari Birlestirme)

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

Variables

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

Istenen yazilar

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

TECHPROED

---



# Operator Signs (Operator Isaretleri)

1 ) **=** Assignment (Atama yapar)

**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

2) **==** Esittir isareti / karsilastirma operatoru

**boolean sonuc1= 5+2 == 7;** sonuc1 degeri **true** olur

**boolean sonuc2= 5\*2 == 15;** sonuc2 degeri **false** olur

---

## Operator Signs (Operator Isaretleri)

3) **!=** Esit degildir isareti

**boolean sonuc1= 5+2 != 7;** sonuc1 degeri **false** olur

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

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

**boolean sonuc1= 5+2 >= 7;** sonuc1 degeri **true** olur

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

5 ) **<** Kucuktur , **<=** Kucuk veya esittir

**boolean sonuc1= 5+2 < 7;** sonuc1 degeri **false** olur

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

---

## Operator Signs (Operator Isaretleri)

### 6) && 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) ;` sonuc1 degeri **true** olur

`System.out.println((5*2 != 15) && (5>7));` **false** yazdirir

### 7) || 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) ;` sonuc1 degeri **true** olur

`System.out.println((5*2 == 15) && (5>7));` **false** yazdirir

**Not :** ==, !=, <, <=, >, >=, AND ve OR operatorleri Comperison (Karsilastirma) operatorleridir



# If Statements (If cumleleri)

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

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

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

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

---

## If Statements

### Sorular

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

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

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

ilkHarf=S output = "Sali"

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

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

Ornek gun=Pazar output = "Hafta sonu"

gun=Sali output = "Hafta içi"

\*\*\* String için equals method'unu kullanın

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

---

# If Else Statements

Eger hava guzel olursa piknige gideriz, yoksa evde otururuz.

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

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

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



---

## If Else Statements Sorular

- Soru 1) Kullanıcıdan dikdörtgenin kenar uzunluklarını isteyin ve dikdörtgenin kare olup olmadığını yazdırın
- Soru 2) Kullanıcıdan bir karakter girmesini isteyin ve girilen karakterin harf olup olmadığını yazdırın
- Soru 3) Kullanıcıya yaşını sorun, eğer yaş 65'den küçük ise "emekli olamazsın, çalışmalısın", 65'den büyükse "Emekli olabilirsiniz" yazdırın
- Soru 4) Kullanıcıya yaşını sorun, eğer yaş 18'den küçük ise "çalışmak için küçüksün", yaş 18-65 arası ise "çalışmalısın", 65'den büyükse "Emekli olabilirsiniz" yazdırın.

---

## If Else Statements

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

**Eger** (soruyu biliyorsa) {Ali soruyu cozsun} , **o bilmiyorsa** {Veli cozsun},  
**o da bilmiyorsa** {Ayse cozsun}, **o da bilmiyorsa** {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 Statements

### Sorular

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



# If Else Statements Sorular

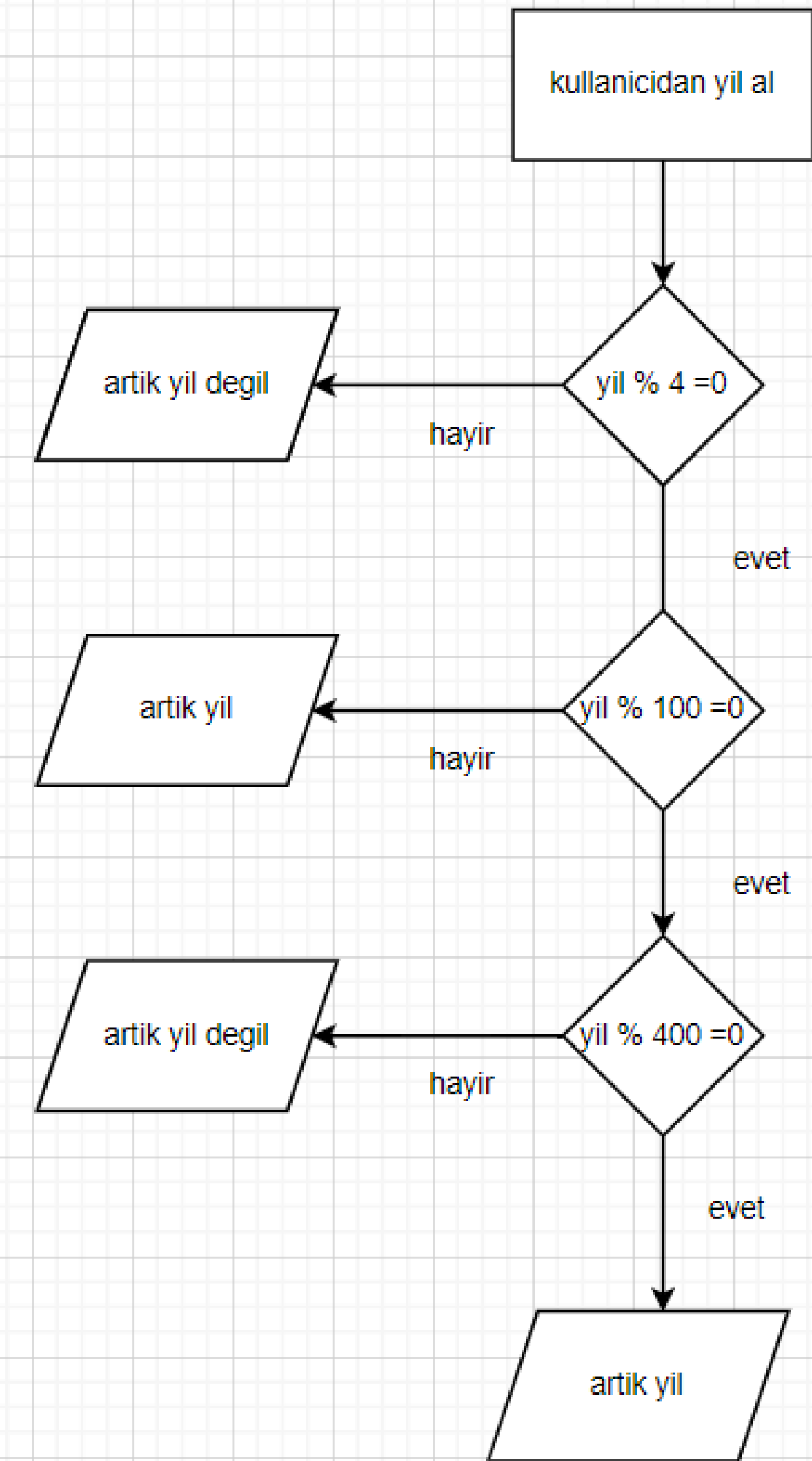
## Soru 9) Interview Question

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

Kural 1: 4 ile bölünemeyen yıllar artık yıl değildir

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

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



---

## Nested If Else Statements

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

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

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

TECHAPROED

---

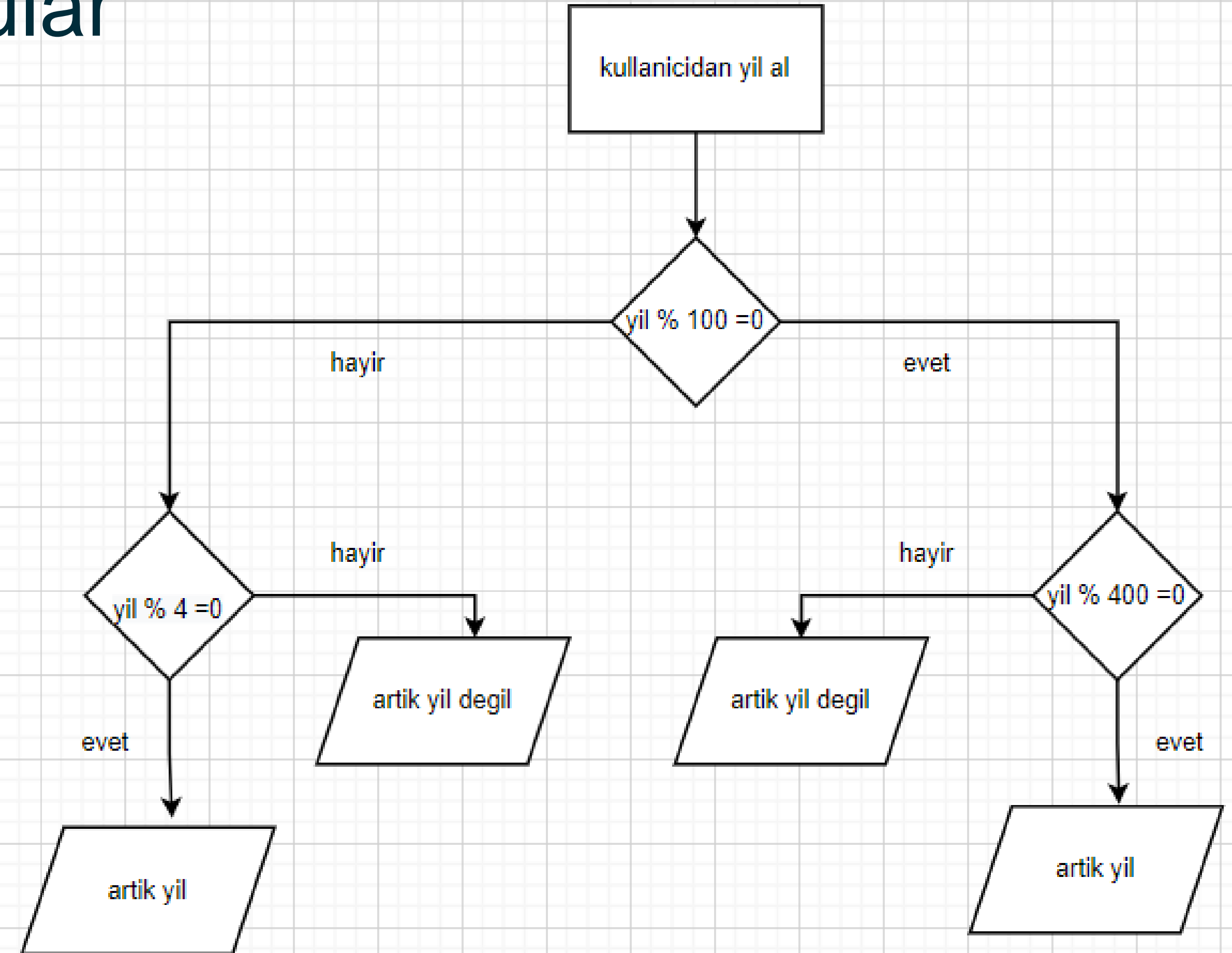
# Nested If Statements Sorular

## Soru 10) Interview Question

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

Kural 1: 4 ile bölünemeyen yıllar artık yıl değildir

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



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



---

## If Else Statements

### Sorular

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

Kullanıcıdan bir sifre girmesini isteyin

Eger ilk harf buyuk harf ise “A” olup olmadigini kontrol edin. Ilk harf A ise “Gecerli Sifre” degilse “Gecersiz Sifre” yazdirin.

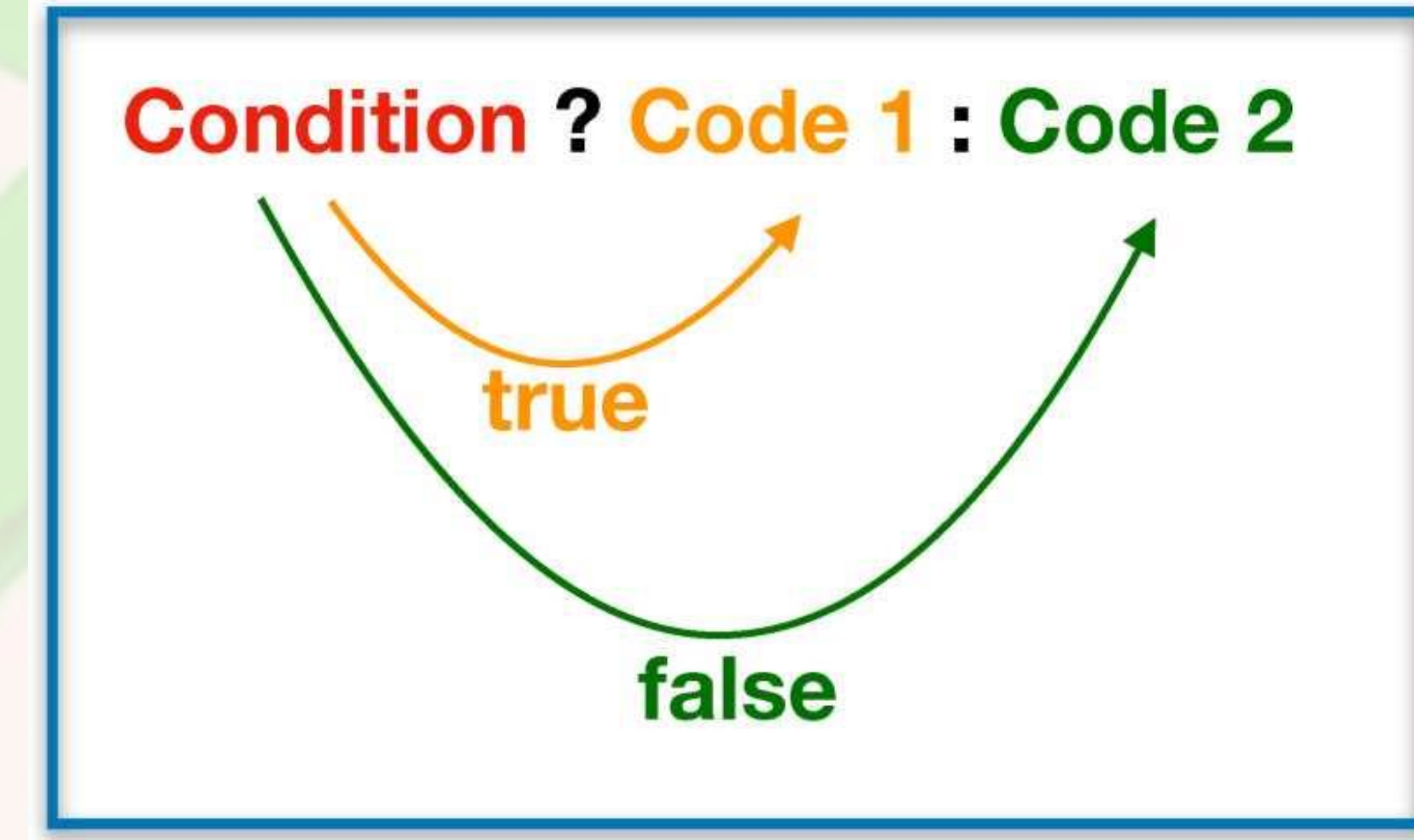
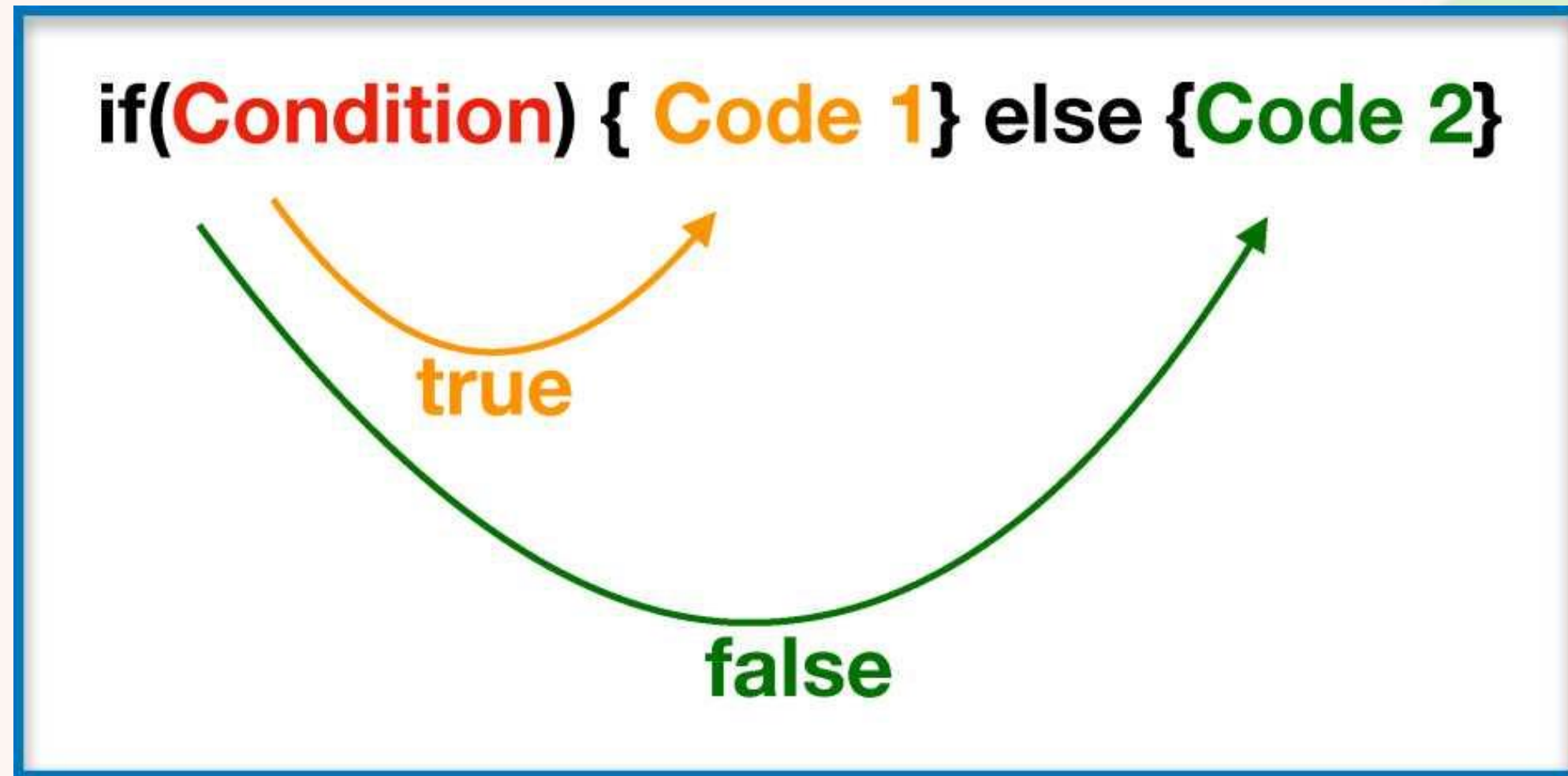
Eger ilk harf kucuk harf ise “z” olup olmadigini kontrol edin. Ilk harf z ise “Gecerli Sifre” degilse “Gecersiz Sifre” yazdirin.

Soru 12) Kullanıcıdan 4 basamakli bir sayi girmesini isteyin. Girdiği sayi 5’e bölünüyorsa son rakamını kontrol edin. Son rakamı 0 ise ekrana “5’e bölünen çift sayı” yazdırın. Son rakamı 0 değil ise “5’e bölünen tek sayı” yazdırın. Girdiği password 5’e bölünmüyorsa ekrana “Tekrar deneyin” yazdırın.

TECHPROED

---

# 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

## Sorular

- Soru1** ) Kullanıcıdan iki sayı alın ve büyük olan sayıyı yazdırın
- Soru2** ) Kullanıcıdan bir tamsayı alın ve sayının tek veya çift olduğunu yazdırın
- Soru3** ) Kullanıcıdan bir sayı alın ve sayının mutlak değerini yazdırın
- Soru4** ) Kullanıcıdan dikdörtgenin uzunluğunu ve genişliğini alın, girilen değerlere göre dikdörtgenin kare olup olmadığını yazdırın.
- Soru5** ) Kullanıcıdan bir sayı alın ve sayı 3 basamaklı ise “uc basamaklı sayı”, yoksa “Uc basamaklı değil” yazdırın
- Soru6** ) Kullanıcıdan bir sayı alın. Sayı pozitifse “Sayı pozitif” yazdırın, negatifse sayının karesini yazdırın



---

## Nested Ternary

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

Condition1 ? Durum1 : Durum2

Condition1 ? Durum1 : Durum2

**Soru1** : Kullanıcıdan bir tamsayı alın ve sayı 10'dan küçükse "Rakam" değilse "Rakam değil" yazdırın

**Soru2** : Artık yıl sorusunu nested ternary ile çözün

TECHPROED

---

# Switch Statement

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

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

---

# Switch Statement

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

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

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

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

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

---



---

# Switch Statement

## Sorular

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

**Soru2** : Kullanıcıdan kacinci ay oldugunu sorun ve ay ismini yazdirin

**Soru3** : Kullanıcıdan bir sayi girmesini isteyin

Girilen sayi 10 ise “iki basamakli en kucuk sayi”

100 ise “uc basamakli en kucuk sayi”

1000 ise “dort basamakli en kucuk sayi”

diger durumlarda “Girdigin sayiyi degistir” yazdirin

**Soru4** : Kullanıcıdan SDET kisaltmasindaki harflerden birini yazmasini isteyin.

Kullanici S girerse “Software”

D girerse “Developer”

E girerse “Engineer”

T girerse “In Testing” yazdirin

---

# String Manipulation / Methods

## 1- concatenation

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

i) + (toplama) isareti ile

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

Output :  
Ali Can

ii) concat() methodu kullanarak

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

Output :  
AliCan

# String Manipulation / Methods

## 2-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 büyük bir sayi kullanilirs Java hata verir (**StringIndexOutOfBoundsException**).

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

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



# String Manipulation / Methods

3-toUpperCase()

4-toLowerCase()

Girilen String degiskendeki tum harfleri istenen bicime ceviriir.

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

NOT : toLowerCase(Locale locale)

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

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

Output :  
techproeducation

---

# String Manipulation / Methods

## 5-equals

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

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

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

Output :  
true

TECHPROED

---

# String Manipulation / Methods

**equals == karsilastirmasi (Interview Sorusu)**

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

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

TECH PRO



---

# String Manipulation / Methods

## 6-equalsIgnoreCase

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

Buyuk / kucuk harf farkliliklari disinda herhangi bir karakter farkliliklari 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](#))