



## Genel Hatırlatmalar



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



## Genel Hatırlatmalar

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

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

~ CHINESE PROVERB



# Gorulecek Dersler

## Automation Engineer:

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

## Java Developer

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

## Mobile Developer

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



# Mentoring

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

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

DÜZENLİ OLARAK YAPILMAKTADIR....



## Ders Isleyisi - Bilmeniz Gerekenler

1. Maillerinizi gunluk kontrol edin

2. Dersleri zoom'dan izliyoruz ama mesajlasma icin slack kullaniyoruz



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



3. Google Classroom

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



## Ders Isleyisi - Bilmeniz Gerekenler



1-Ders esnasında öğrencilerin dikkatini dağıtacak paylaşımlar yapmayın

2-Ders esnasında ders ve konu dışında paylaşım yapmayınız.

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

4-Ders esnasında ders hocasına direct mesaj yazmayınız

5-Derste code paylaşırken SNIPPET ve screenshot kullanmaya dikkat ediniz

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

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

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



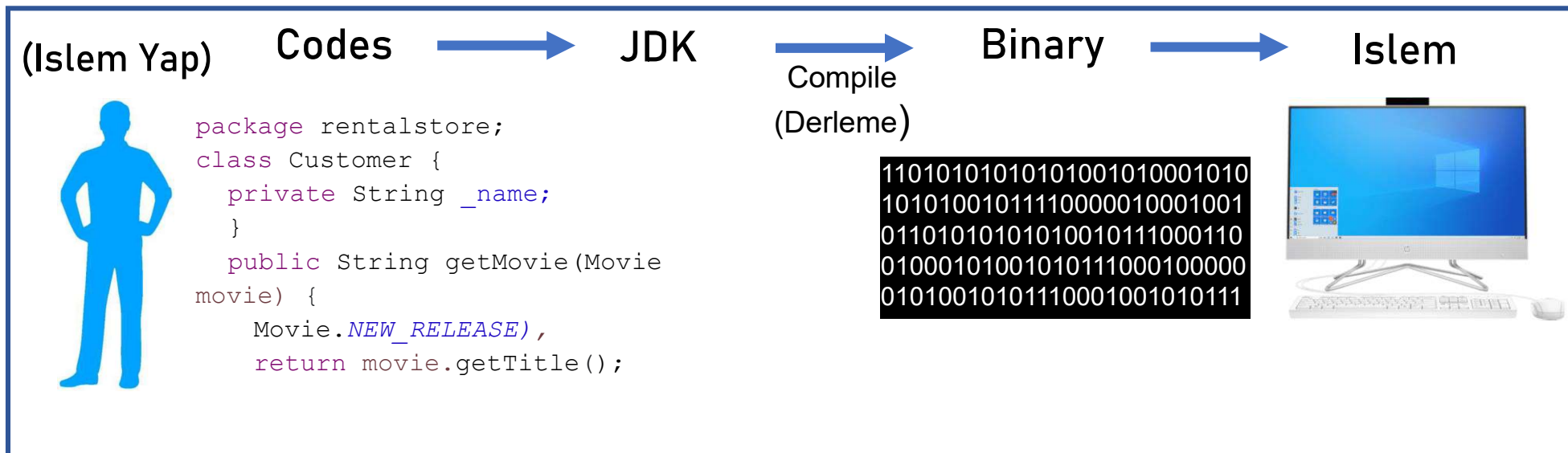
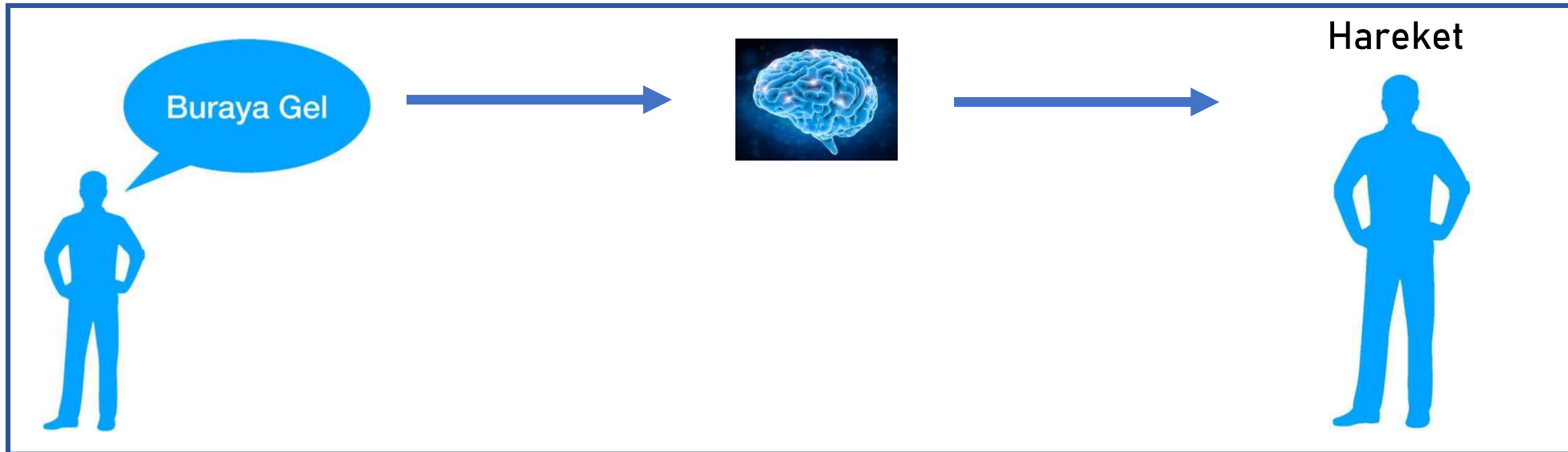
## Ders Isleyisi - Bilmeniz Gerekenler

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





# Programlama Dili Nedir ?



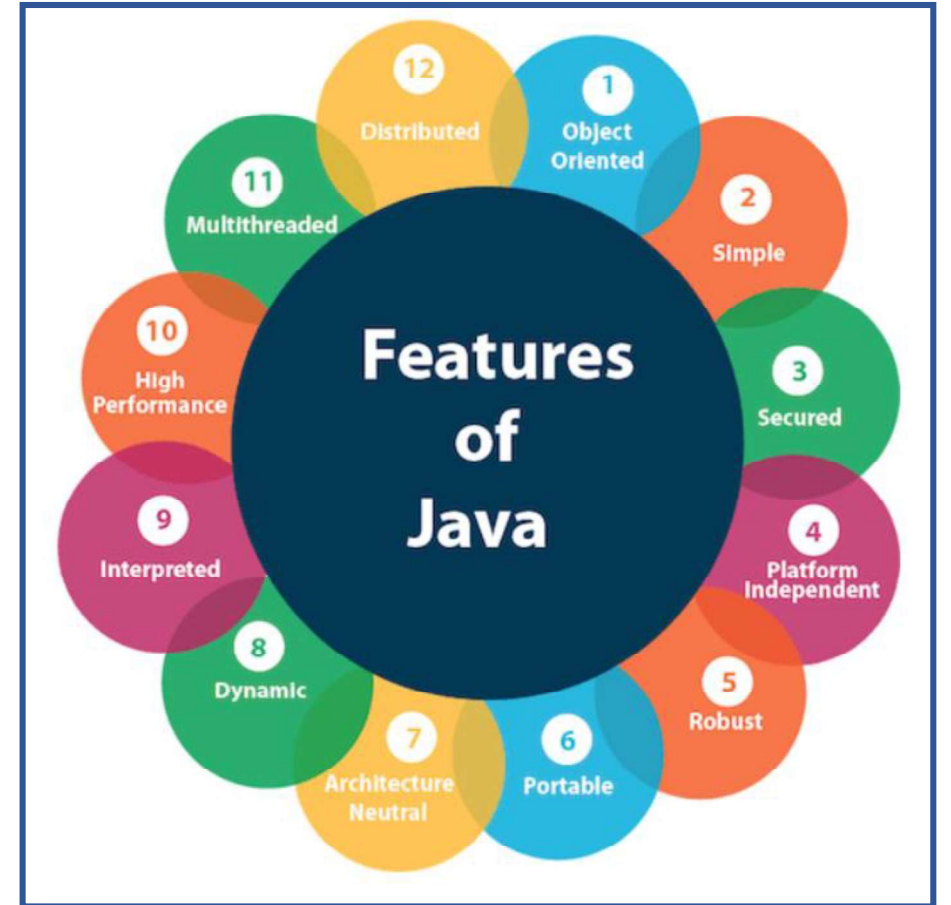




## Nicin Java ?

- 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)

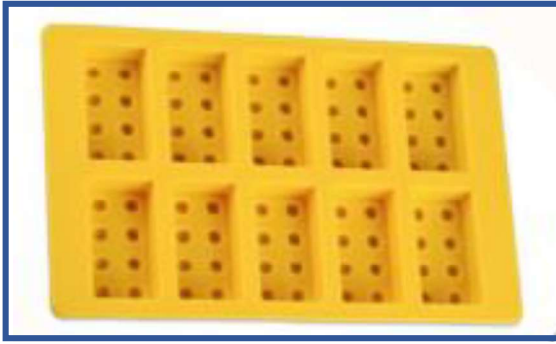
- 1- Feature (Fields veya Variables)  
Pasif ozellik (renk,sekil,isim)
- 2- Functionality (Method)  
Aktif ozellik (tasima,degistirme)



Application (Urun)

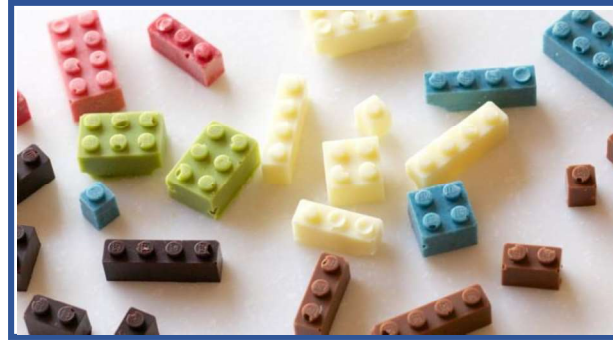


# Bir Object Nasıl Olusturulur?



Class(Object Kalibi)

Field  
(Variables)      Method  
(Functions)



Object



Birden fazla Obje birlestirilir



Application



# Object Nasıl Kullanilir ?



Ogretmen



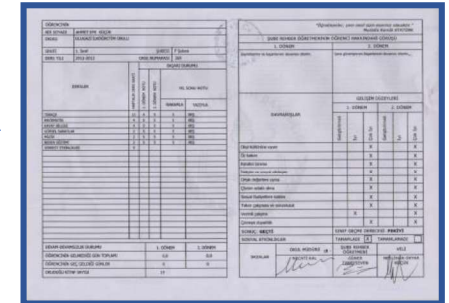
Ogrenci

Dersler

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



Personel



Notlar



## Bir Class Hangi Bolumlerden Olusur?

```
public class C2_MethodCreation2 {
```

1

2

3

2

```
} // Class sonu
```

1 – Class Declaration

2 – Curly braces : Suslu parantez

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





## Bir Class'ın İçinde Neler Bulunur?

```
public class C2_MethodCreation2 {
```

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

1

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

2

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

3

```
} // Class sonu
```

1 – Field / Variables

2 – Main Method

3 – Method



## Class Olustururken (Declaration) Kullanilan Keyword'ler

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

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



## Method Olusutururken Kullanilan Keyword'ler

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

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





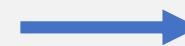
# Main Method

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



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

Araba



Motor

Java Project



Main Method



## Yorum Cümlesi (Comment) Nasıl Eklenir ?

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

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

➤ Genelde iki kullanım vardır

1) Tek satirlik comment

2) Çok satirlik comment



# Data Nedir?

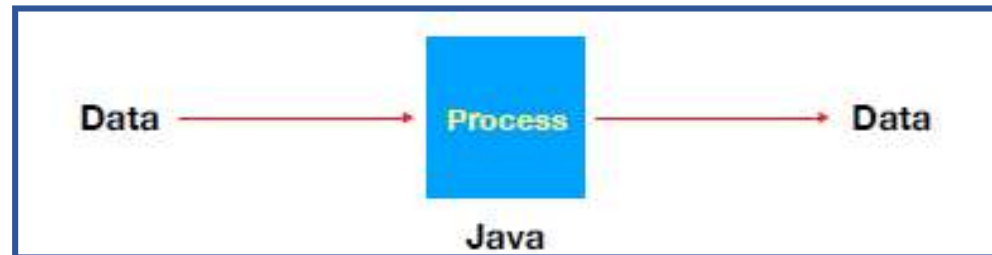
**Data** bilgisayar tarafından işlenen (**processed**) veya depolanan (**stored**) bilgidir.

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

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

011000110110111101101101110000011101010111010001100101011100100010000001101000000  
101

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

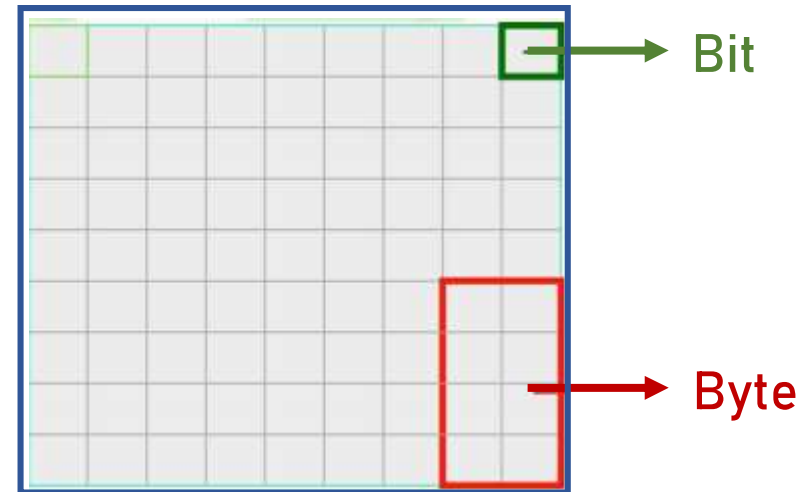
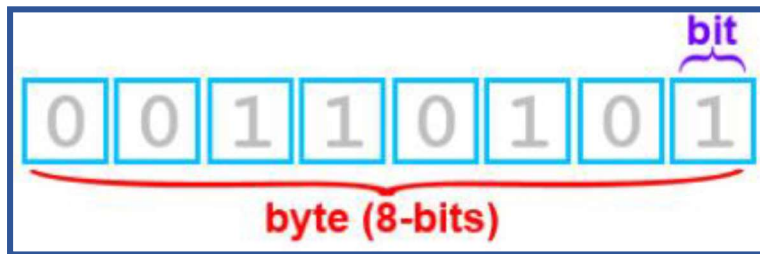




# Bit

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

Note: 8 bit =1 byte



Memory  
(Hafıza)



# IntelliJ Kullanım

## 1- Proje oluşturma

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

## 2- Package (paket) oluşturma

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

## 3- Class oluşturma

day01variables dosyasına sağ click -- New -- Class -- C01\_Variables01 -- finish

## 4- Main method oluşturma

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