



CLARUSWAY

WAY TO REINVENT YOURSELF



clarusway



clarusway



clarusway



clarusway



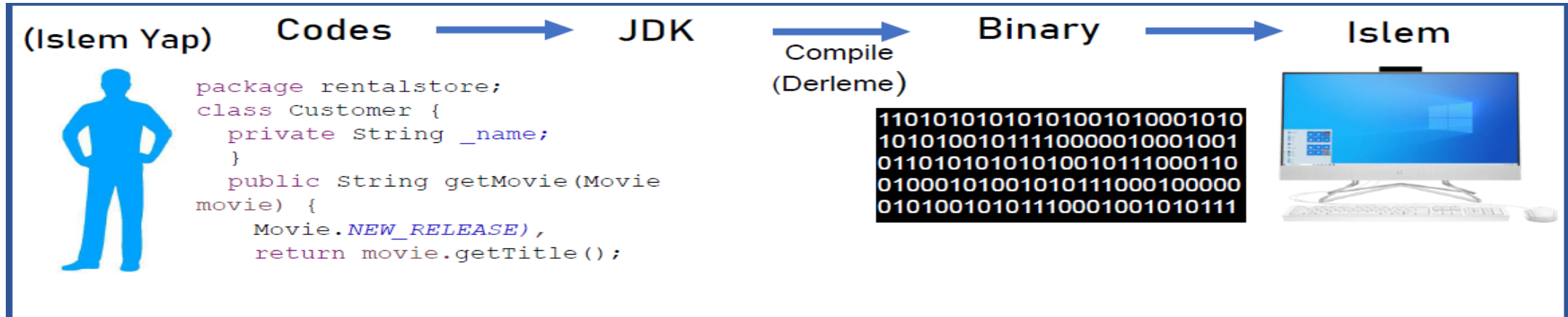
clarusway

1

JAVA Variables

Lesson:
JAVA Chapter 01

Programming Language (Programlama Dili)



- Programlama Dili,
 - istenilen hesaplamaları yapmak için, elde edilen veriyi saklamak için ve girdi/çıkıtlarına veri gönderme/alma gibi işlemleri yapmak için kullanılan dildir.
 - Doğal dillerde olduğu gibi programlama dillerinde de belirli bir yazım kuralı (sentaks) vardır.
 - Programlama dilleri ile sadece bilgisayarlar üzerinde çalışan uygulamalar değil, işlemcisi ve belleği bulunan diğer elektronik cihazlarda çalışan uygulamalar da yazılır.



Niçin JAVA?

1- Öğrenmesi kolaydır.

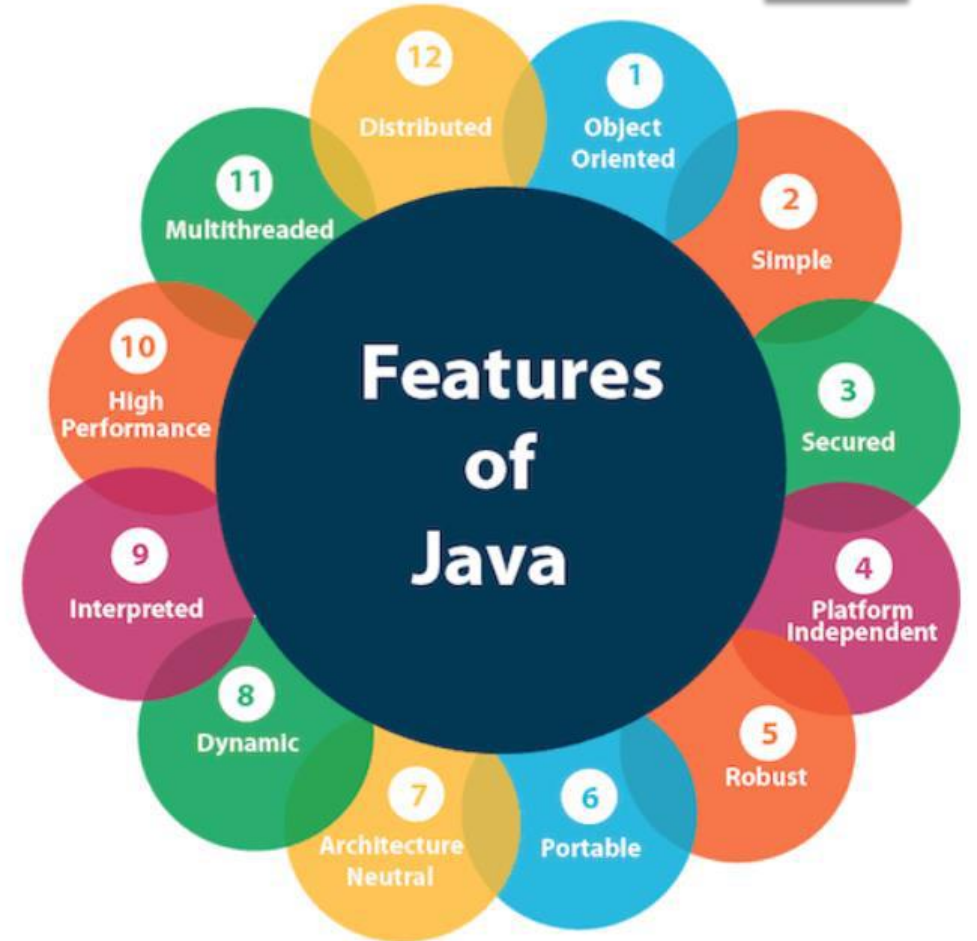
2- Güvenlidir. Oracle gibi güçlü bir şirketin bünyesinde olan JAVA kullanıcı ve uygulama sahipleri için güvenlidir.

Sun'a göre 3 milyar cihaz Java kullanıyor. ABD'deki kişisel PC'lerin %89'u şirket PC'lerini %97'si Java bulunuyor.

Bunlardan bazıları şu şekildedir:

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

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



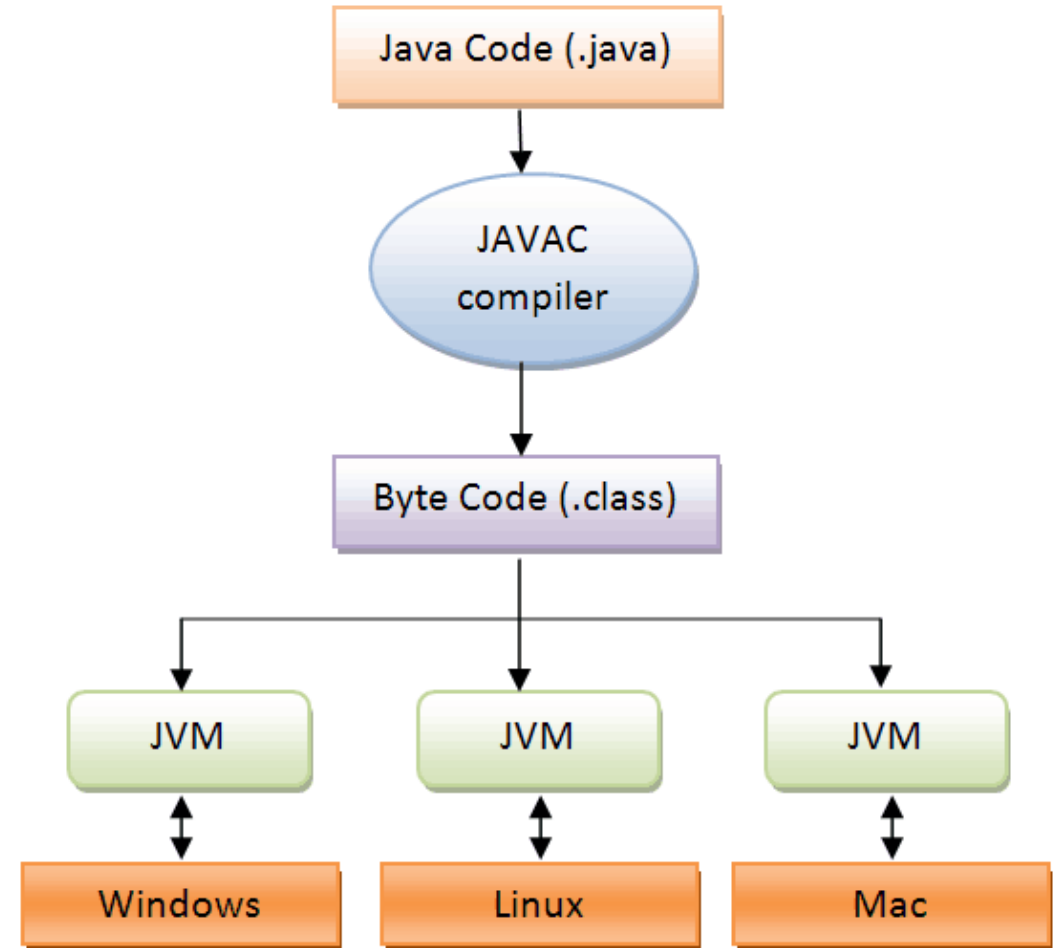
Niçin JAVA?

4- Platform Independent.

"Write once run everywhere" bir kere yaz her yerde kullan.

5- Ücretsizdir. Java ile yazılan bazı uygulamalar..

- Acrobat Reader
- Google Earth
- Amazon
- Netflix
- Uber
- Tinder
- Mat Lab
- E-Devlet



Niçin JAVA?

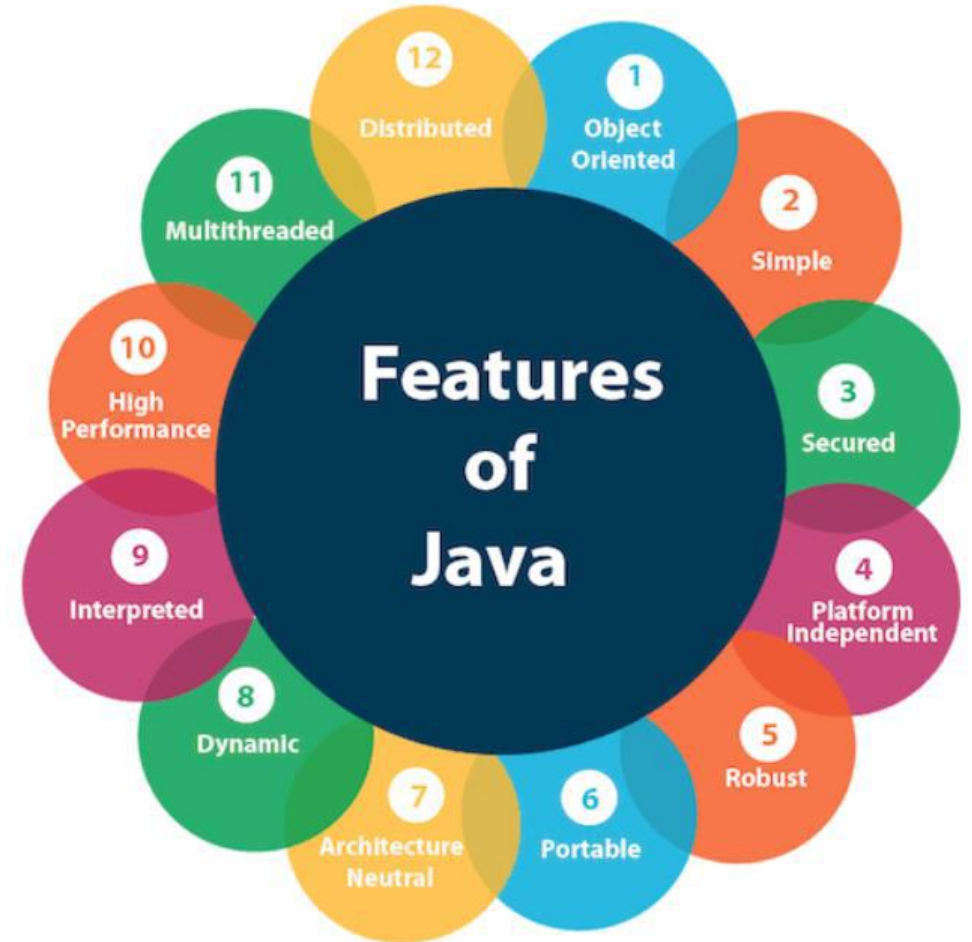
6- Java çok zengin API setleri barındırır. Standart JDK, XML'in ayrıştırılmasından saat dilimlerinin arasındaki çeviriye kadar 200'den fazla yerleşik paket bulundurur.

7- Geniş tool ekosistem;

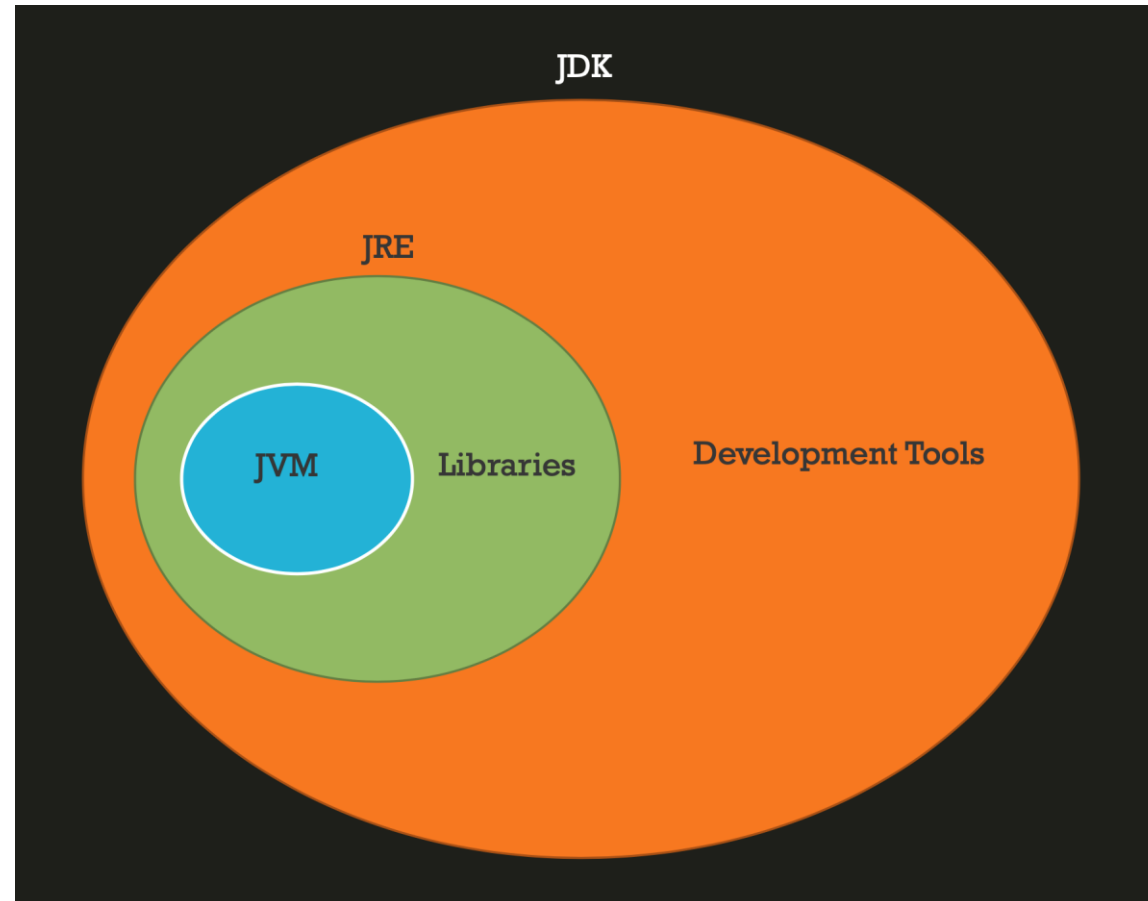
- Gradle
- Maven
- Jenkins

8- Functional Programming. Lambda ile uzun ve complex kodları işlevleri daha kısa kodlarla kolayca yazabilme imkanı.

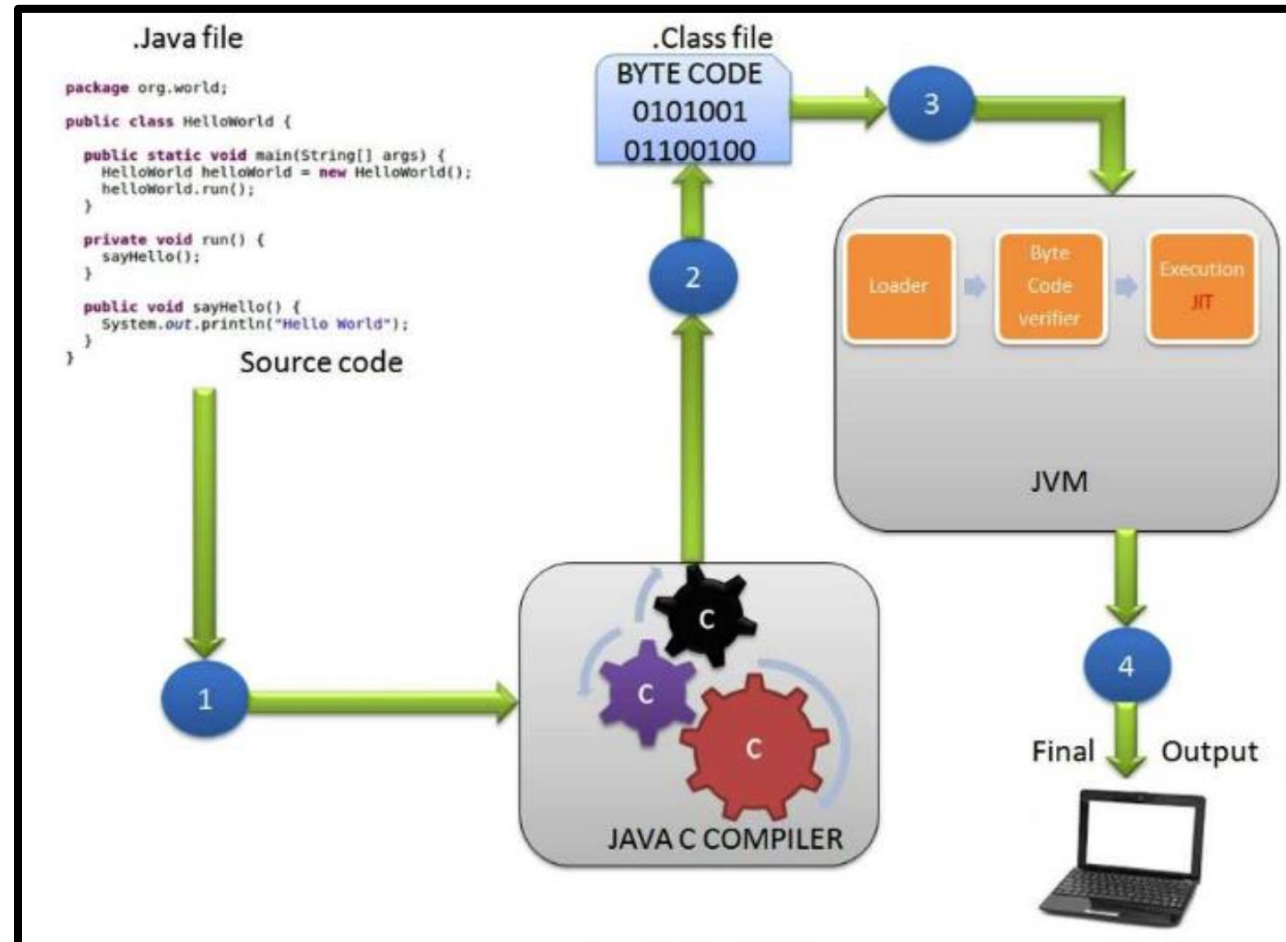
9- Birinci derecede ANDROID desteği.



JDK (Java Development Kit)



JAVA Nasıl Çalışır?



Object Oriented Programming (OOP)



Object (Nesne)



Application (Ürün)

1- Feature (Fields, Variables)

Pasif özellik (isim, meslek, maaş)

2- Functionality (Method)

Aktif özellik (hesaplama, update etme)



Bir Object Nasıl Oluşturulur?



Class (Object Kalıbı)



Object



Birden fazla obje birleştirilir.



Application

Field
(Variables)

Method
(Functions)



Object (Nesne)



Öğretmen



Öğrenci



Karne

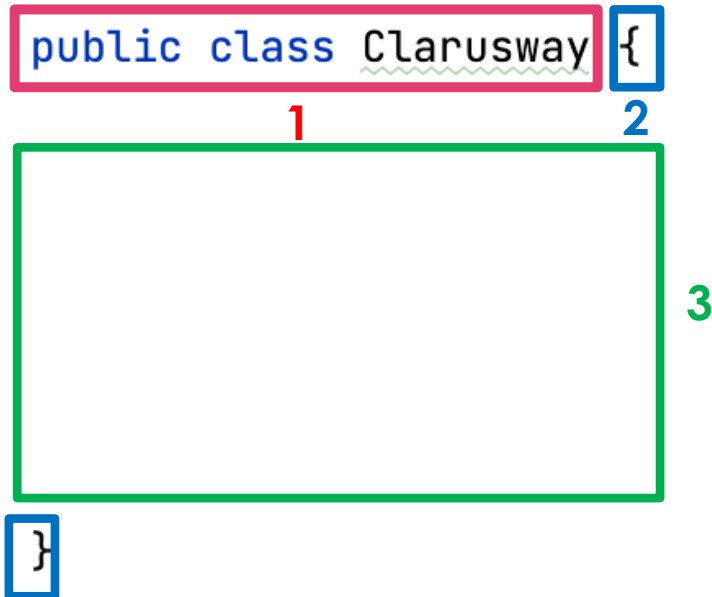


Çalışan

ÖĞRETMEN KATILIMLARI (İLKOKUL VE ORTAOKUL) HAFTALIK DERS ÇİZELGESİ		SINIFLAR						
DERSLER		1	2	3	4	5	6	7
ÖĞRETMENLER	Okuma	10	10	10	10	10	10	10
	Yazma	10	10	10	10	10	10	10
	Okuma ve Yazma	10	10	10	10	10	10	10
	Okuma ve Yazma	10	10	10	10	10	10	10
	Okuma ve Yazma	10	10	10	10	10	10	10
	Okuma ve Yazma	10	10	10	10	10	10	10
	Okuma ve Yazma	10	10	10	10	10	10	10
	Okuma ve Yazma	10	10	10	10	10	10	10
	Okuma ve Yazma	10	10	10	10	10	10	10
	Okuma ve Yazma	10	10	10	10	10	10	10
ÖĞRENCİLER	Okuma	10	10	10	10	10	10	10
	Yazma	10	10	10	10	10	10	10
	Okuma ve Yazma	10	10	10	10	10	10	10
	Okuma ve Yazma	10	10	10	10	10	10	10
	Okuma ve Yazma	10	10	10	10	10	10	10
	Okuma ve Yazma	10	10	10	10	10	10	10
	Okuma ve Yazma	10	10	10	10	10	10	10
	Okuma ve Yazma	10	10	10	10	10	10	10
	Okuma ve Yazma	10	10	10	10	10	10	10
	Okuma ve Yazma	10	10	10	10	10	10	10

Dersler

Class Bölümleri



1- Class Declaration->Class tanımlama

2- Curly Braces-> Süslü Parantez

3- Class Body->Süslü parantezler arasında kalan ve kodların yazıldığı alan.



Bir Class içindeki Elementler

```
1  
2  
3 ▶ public class ClarusWay //class
```

```
4     private int maas; //field / variable
```

```
5     private double maasOrtalama; //field / variable
```

```
6  
7 ▶     public static void main(String[] args) //main method
```

```
8         maasOrtalamaHesapla(8000, 13000); //method call
```

```
9     } //main method sonu
```

```
10  
11     1 usage
```

```
12     private static void maasOrtalamaHesapla(int maas1, int maas2) //method
```

```
13         System.out.println("maas ortalamanız : " + (maas1 + maas2) / 2);
```

```
14     } //method sonu
```

```
15 } //class sonu
```



Class Declaration (Sınıf Tanımlama) Keyword'leri

```
public class MyFirstClass { }
```

1 2 3 4

- 1 **public** - > Access Modifier (Erişim Belirleyici : Class'a kimlerin erişebileceğini belirler).
public : Class'a hiçbir kısıtlama olmadan herkesin erişebileceğini tanımlar.
- 2 **class** : Yazılan code bloğunun class olduğunu tanımlar
- 3 **MyFirstClass** : Tanımlanan class'a verilen isimdir. Class'a istenilen ismi verebilir ancak verilen isim class'ın aksiyonuna uygun olmasına dikkat edilir.
Class ismi mutlaka büyük harfle başlamalı, eğer class name birden fazla kelimeden oluşuyorsa sonraki kelimelerin ilk harfleri de büyük yazılmalı. (CamelCase)
- 4 **Body (Class Body) : { }** Arasında kalan kodların create edilen alandır.



Method Declaration (Method Tanımlama) Keyword'leri

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

- 1 Access Modifier (Erişim Düzenleyici) : Method'a kimlerin erişebileceğini belirler.
public : Methoda herhangi bir kısıtlama olmadan istenilen yerden erişilebilir.
private : Sadece methodun tanımlandığı class'da erişilebilir.
protected : Sadece methodun tanımlandığı class ve child class'lardan erişilebilir.
- 2 **int** : Return Type: method aksiyonu sonucu çıktının data type'nı ve döndürdüğünü belirtir.
- 3 **myFirstMethod** : Tanımlana method'un ismidir. Method ismi mutlaka küçük harfle başlar, eğer method name birden fazla kelimeden oluşursa sonraki kelimelerin ilk harfleri büyük yazılır. (camelCase)
- 4 **() parantez** : Methodlarda isimden sonra parantez kullanılır ve gerektiğinde parantez içinde parametre yazılır.
- 5 Body (Method Body) : { } Arasında kalan code create edilen bölümdür.



İsimlendirme Kuralları (Naming Convention)

Camel Case(CamelCase)

Camel case çok kullanılan isimlendirme kurallarından biridir. Kelimeler bitişik ve ilk harfleri büyük yazılır. İlk kelimenin ilk harfinin küçük yazıldığı olabilir.

Örnek: **HelloWorld**

Snake Case(Snake_Case)

Upper snake case ve lower snake case olarak ikiye ayırabiliriz. Kelimeler alt tire ile birbirine bağlanır.

Upper snake case örnek: **Hello_World**

Lower snake case örnek: **hello_World**

Screaming snake case(SCREAMING_SNAKE_CASE)

Bütün harfleri büyük yazılır. Genellikle sabit isimlendirmede kullanılır.

Screaming snake case örnek: **HELLO_WORLD**



İsimlendirme Kuralları (Naming Convention)

Java dilinde isimlendirmeler için bazı standartlar oluşturulmuştur.

Class: Sınıf için upper camel case kullanılır.

Örnek: HelloWorld

Methodlar: Methodlar için lower camel case kullanılır.

Örnek: helloWorld

Variables: Değişkenler için lower camel case kullanılır.

Örnek: helloWorld

Sabitler: Sabitler için screaming snake case kullanılır.

Örnek: HELLO_WORLD



İsmlendirme Kuralları (Naming Convention)

1. JAVA variable isimleri **case sensitive** (Büyük küçük harfe duyarlıdır)
“money”, “Money” veya “MONEY” birbirinden farklı derlenir.
2. JAVA variable isimleri “harf”, “\$” veya “_” ile başlamalıdır. Fakat “\$” ve “_” ile başlamak tavsiye edilmez.
3. JAVA variable isimlerinde, ilk harften sonra sayı, “\$” ve “_” kullanılabilir.
4. Variable isimleri için JAVA'ya özel terimler (key word) kullanılamaz. (int, for, if, import vb).
5. Variable isimleri küçük harflerle başlar, camel case kullanılır.



Main Method

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



* **TRICK :)** Main Method, JAVA'da code'ların run edildiği giriştir. (Entry Point)

* **TRICK :)** Main Method tanımlanırken yazılması gereken Syntax (Kod Dizimi) değiştirilemez.

* **TRICK :)** Parametre parantezi içerisindeki (String[] args) ifadesi JAVA'nın derlemesi için gerekli olan parametreleridir ve olması şarttır.

Araba → Motor

Java Project → Main Method



Comment Line (Yorum Satırı)

```
public class ClarusWay {//class
```

```
// tek bir satırı yoruma(comment) almak için // ifadesi kullanılır  
String bootcamp = "clarusway";//bootcamp ismi clarusway olarak cretae edildi.
```

```
/*  
Birden çok satır yoruma (multi line comment) almak için  
bu şekilde yapılır.  
çalışmasını istemediğini code blok'larını da yoruma almak için  
int age=48;  
String name="Hardy";  
*/
```

```
boolean itManager = true;
```

```
}//class sonu
```

Comments Line (Yorum Satırı): JAVA tarafından derlenip run edilmeyen, kodları açıklamak ve daha okunabilir hale getirmek veya bilgilendirme için kullanılan ifadelerdir. Alternatif kodu test ederken yürütmeyi önlemek içinde kullanılabilir.

Genelde iki kullanım vardır.

- 1) **Tek Satırlık Comment** (Single-line comments)-> //
- 2) **Çok Satırlık Comment** (Multi-line comments)-> /* ... */



Data Nedir?

20

* **TRICK :**) JAVA'nın kullanılan (**use**) veya üretilen (**produce**) her enformasyon parçacığı **data**'dır.

* **TRICK :** JAVA'nın kullanılan (**use**) veya üretilen (**produce**) her enformasyon parçasığı **data**'dır.



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

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

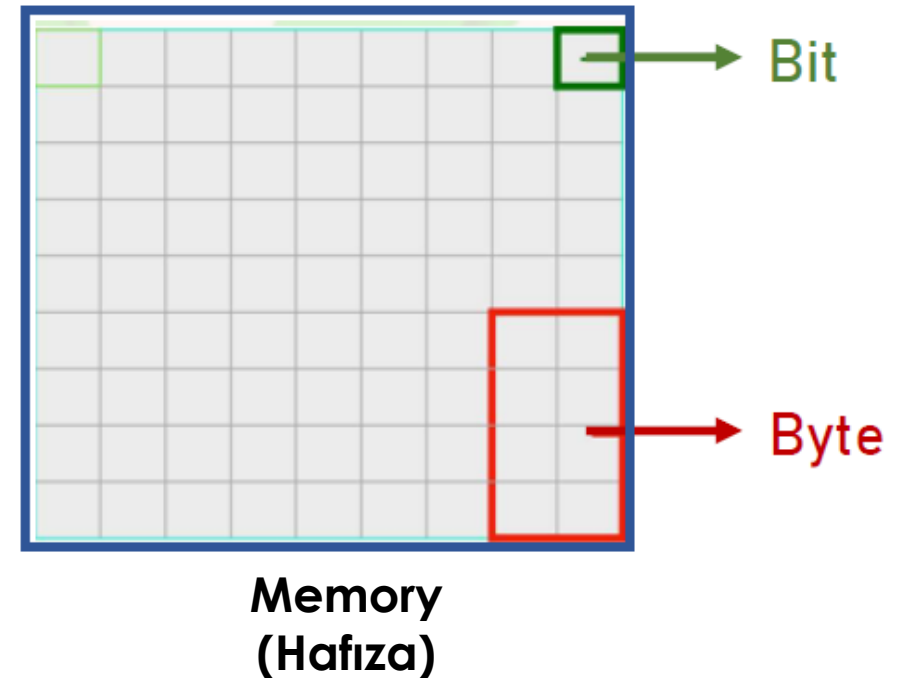
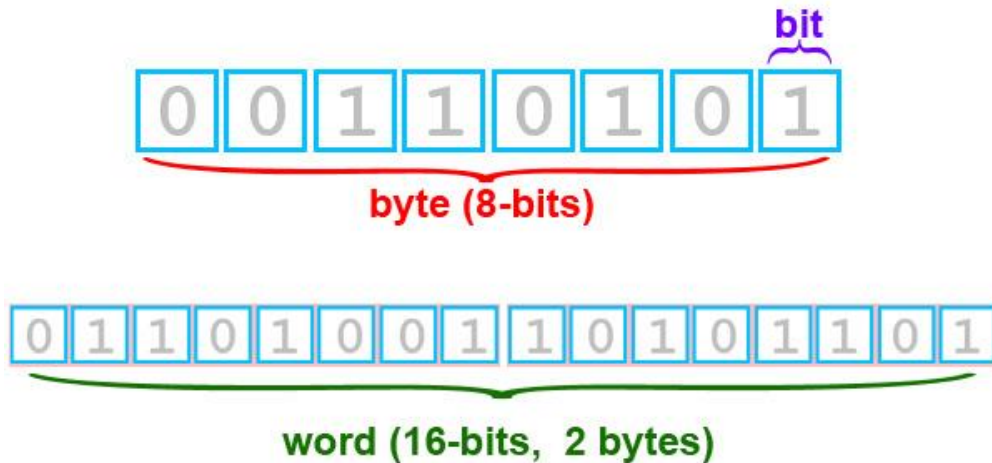
01100011011011110110110101110000011101010111010001100101011100100010000001101000000
101



Bit Nedir?

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

* **TRICK :) 8 bit = 1 byte**



Variables Declaration (Değişken Tanımlama)

Declaration (Tanımlama)

Variable bellekte (**memory**) ayrılmış olan alanın (**Reserved Area**) adıdır.

*** Variable içinde değer saklayan bir kutudur. (**Container**).

*** Bir variable'da saklanan değer, program yürütülürken değiştirilebilir.

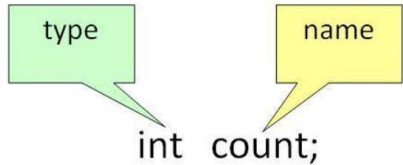
*** **JAVA'da tüm variable'lar kullanılmadan önce declare edilmelidir. (Variable Declaration)**

* **TRICK :) Variable declaration için...**

1- Data type (Data Türü)

2- Variable Name (Değişken İsmi)

Mutlaka yapılmalı...



Variables Assignment (Değişkene Değer Atama)

Tanımlanan bir variable'a değer atamaya **assignment** denir.

1- Bir variable'a değer ataması yapılırken data type uygun değer atanmalıdır değilse JAVA derleme hatası verir.

```
public class ClarusWay {//class

    String bootcamp = "clarusway";//bootcamp ismi clarusway olarak create edildi.
    String instructor = Hardy;
    int age = 48;
    double salary = "12.500";
    boolean itManager = true;
    int years=false;
    double avarage=12.5

//class sonu
```



Variables Assignment (Değişkene Değer Atama)

2- Variable'a ilk önce declaration (tanımlama), daha sonra assignment (atama) yapılabilir.

```
public class ClarusWay {//class

    public static void main(String[] args) {//main method

        String bootcamp;
        int age;
        double salary;
        boolean itManager;

        bootcamp = "clarusway";
        age = 48;
        salary = 12.500;
        itManager = true;
    }//main method sonu

}//class sonu
```

3- Variable declaration yapıldıktan sonra, birden fazla atama yapılabilir. Variable'ın son değeri update edilip önceki değer silinir.

```
public class ClarusWay {//class

    public static void main(String[] args) {//main method

        double salary = 12500;
        salary = 13500;
        salary = 15000;
    }//main method sonu

}//class sonu
```



Variables Assignment (Değişkenlere Değer Atama)

4- Aynı data türünde birden fazla variable tek komutla declare edilebilir.

```
String meslek, isim, memleket;  
memleket = "İstanbul";  
meslek = "java developer";  
isim = "Hardy";
```

5- Aynı data türünde birden fazla variable tek komutla declare edilip değer atanabilir.

```
String meslek = "java developer", isim = "Hardy", memleket = "İstanbul";
```



THANKS!

Any questions?

Garry T. - Full Stack Automation Engineer

Haluk B. - JAVA Backend Developer

