

Java'ya Giriş

Ahmet Emre Aladağ

aladagemre@gmail.com

<http://www.emrealadag.com>

- Çalışma Yapısı
- Netbeans IDE
- Merhaba Dünya
- Veri yapıları, Arrayler
- I/O İşlemleri
- Class'lar

Genel Yapısı

- <http://java.sun.com/javase/downloads>
- Kaynak kod: Dosya.java
- Derleme işlemi: `javac Dosya.java`
- Çalıştırma: `java Dosya`
- Byte Code
- Object Oriented
- java.sun.com => Tutorial
- Google => Java for C++ programmers, Java Crash Course, vs.

Merhaba Dünya

Merhaba.java

```
class Merhaba {  
    public static void main(String args[]){  
        System.out.println("Merhaba Dunya");  
    }  
}
```

C:\> javac Merhaba.java

C:\> java Merhaba

Merhaba Dünya

Veri Yapıları

int

float

double

long

String

char

byte

Arrayler: int[], float[], double[], long[], String[] vs.

boolean : true/false

null (değer)

Metodlar (Fonksiyonlar)

```
public int ikiKere(int sayi){  
    return sayi*2;  
}  
public void merhabaDiyelim(){  
    System.out.println("Merhaba " + "Dünya");  
}  
public void selam(String isim){  
    System.out.printf("Selam %s\n", isim);  
}  
public static void main(String args[]){  
    ...  
}
```

Scanner (cin)

```
import java.util.Scanner;
public class Tahsildar {
    public static void main(String args[]){
        Scanner memur = new Scanner(System.in);
        int borc=100, miktar, paraUstu;

        System.out.println("Lutfen odemenizi yapiniz:");
        miktar = memur.nextInt();
        if (miktar >=borc){
            paraUstu = borc – miktar;
            System.out.printf("Teşekkürler, para üstünüz: %d\n");
        }
        else {
            System.out.println("Yetersiz ödeme.");
        }
    }
}
```

Arrayler

```
int i;  
int[] yaslar = new int[10];
```

```
String[] isimler;  
isimler = new String[10];
```

```
yaslar[0] = 20;  
yaslar[1] = 24;  
isimler[0] = "Ahmet";  
isimler[1] = "Mehmet";
```

```
for (int i=0; i < 2; i++){  
    System.out.printf("%s (%d)\n", isimler[i], yaslar[i]);  
}
```

Arrayler / For

```
int[] arananSayilar = {4,5};
int[] dizi = new int[10];
for (int i=0; i<10; i++){
    dizi[i] = i*2;
}
//Arayalim
for (int i=0; i<10; i++){
    for (int j=0; j<2;j++){
        if (dizi[i]==arananSayilar[j]){
            System.out.println("Bulduk: " + dizi[i]);
        }
    }
}
```


Class

Human.java

```
public class Human{  
    private int age;  
    private String name;  
    private Human mother;  
  
    public Human(String name){  
        this.name = name;  
        System.out.println(name+ “ has born”);  
    }  
    public int getAge(){  
        return age;  
    }  
    public void setAge(int newAge){  
        age = newAge;  
    }  
}
```

.....

```
public void setMother(Human mother){
    this.mother = mother;
}
public Human getMother(){
    return mother;
}
public void setName(String name){
    this.name = name
}
public String getName(){
    return this.name;
}
public void talk(String sentence){
    System.out.println(name + " says:" + sentence);
}
/* now lets create an instance.
in the following slide. */
```

```
public static void main(String args[]){  
    Human bebek = new Human("Bob");  
    Human anne = new Human("Jenny");  
    bebek.setAge(0);  
    bebek.setMother(anne);  
  
    bebek.talk("Ingaaa");  
    bebek.talk(bebek.getMother().getName() + " neredeeeeee?!");  
}  
}
```

Output:

Ingaaa

Bob says: Jenny neredeeeeee?!

İsimlendirme

- Class isimleri büyük harfle başlamak zorundadır.
- “Genelde” her java dosyasında bir class bulunur.
- Dosya ismi içindeki public class ile aynı isme sahip olmalı:

```
public class Test { ... }
```


Test.java
- Metodlar ve değişkenler mixedCase yazılır.
- Birbirini kullanan classlar ya aynı klasörde olmalı, ya da biri diğerini import etmelidir.

Class (Sınıf) – Object (nesne)

- ClassTuru nesne = new ClassTuru();
Human insan;
insan = new Human(“Marty”, “McFly”,
new Human(“Clint”, “Eastwood”));
- nesne.metod();
insan.talk(“Geleceğe hoşgeldiniz”);
int yas = insan.getAge();

main(String args[])

- Ana programın yazıldığı metoddur.. Genelde yeni sınıflar oluşturulup etkileşimleri sağlanır.
- String args[] parametresi mutlaka olmalıdır.
- public static void main(String args[])
- Herhangi bir class içerisine yazılabilir. Eğer o class'ın dosyası çalıştırılırsa bu main çalışır.

package

- Aynı klasörde yer alan dosyaları birbirine bağlayan isimdir.
- Her class'ın bir package içerisinde olması gerekmez.
- Kod dosyaları farklı klasörlere yayılan geniş bir yazılımda kullanılması düşünülebilir.
- package paketismi; şeklinde dosyanın en başında tanımlanır.

import

- import, başka Class'ları kullanabilmek için kullanılan “#include” benzeri bir keyworddür.
- import paket1.paket2.paket3.ClassIsmi
- import java.util.Scanner;
- Import finans.banka.MevduatHesabi;
- Import finans.banka.YatirimHesabi;