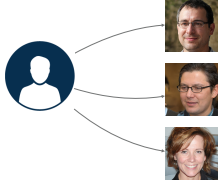


NESNEYE YÖNELİK PROGRAMLAMA

Sınıflar ve Nesneler

SINIF VE NESNE KAVRAMLARI



- Sınıf
 - Nesneye yönelik programlama
 - Struct benzeri
- Nesne
 - Örnek

[*https://thispersondoesnotexist.com/](https://thispersondoesnotexist.com/)

YAPILAR

- Yapı tanımı
 - Özellikler
 - Kullanıcı tanımlı değişkenler
 - Metotlar
 - Yazdırma
 - Java

YAPILAR - SINIFLAR

```
class kisi{
    public String ad;
    public int yas;
}

public class Main {
    public static void main(String[] args){
        kisi k = new kisi();
        k.ad = "Emir";
        k.yas = 253;
        System.out.println(k);
    }
}
```



net.emirozturk.kisi@77459877

```
System.out.println(k.ad + " " + k.yas);
```

```
class kisi{
    public String ad;
    public int yas;
    public int no;
}
```

```
k.yas = 11121;
```

```
public class Main {
    public int sonIkiHane(kisi k){
        return k.no % 100;
    }
    public boolean ilkiBuyukMu(kisi k1,kisi k2){
        return sonIkiHane(k1)>sonIkiHane(k2);
    }

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

YAPILAR - SINIFLAR

- Yapı içeriğinin doğruluğu
- Yapıyı kullanan fonksiyonların üzerindeki değişiklik

YAPILAR - SINIFLAR

- Doğruluğun sağlanması
 - Erişim belirleyicileri
 - Erişim metotları
 - Yapı ile ilgili metotlar

SINIF BİLDİRİMİ

- Sınıflar
 - Sınıf ismi
 - Özellikler (değişkenler)
 - Metotlar (fonksiyonlar)

◦	kisi	
f	ad	String
f	yas	int
f	no	int
m	setYas(int)	void
m	getYas()	int
m	sonKiHane()	int

← Sınıf Adı

← Özellikler

← Metotlar

SINIF BİLDİRİMİ (JAVA)

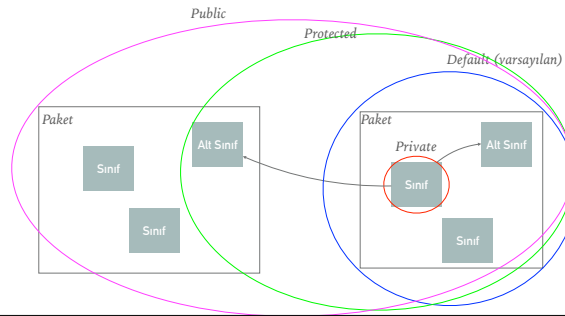
```
class SınıfAdı {  
  
    ErişimTürü DeğişkenTürü ad;  
    ErişimTürü DeğişkenTürü ad;  
    ErişimTürü DeğişkenTürü ad;  
  
    ErişimTürü DönüşTipli İsim(Tür ad){  
        //işlemler  
    }  
    ErişimTürü DönüşTipli İsim(){  
        //işlemler  
    }  
}
```

SINIF BİLDİRİMİ

◦	kisi	
f	ad	String
f	yas	int
f	no	int
m	setYas(int)	void
m	getYas()	int
m	sonKiHane()	int

```
class Kisi{  
    private String ad;  
    private int yas;  
    private int no;  
    public void setYas(int yas){  
        if(yas<150){  
            this.yas = yas;  
        }  
        else  
            yas = 0;  
    }  
    public int getYas(){  
        return yas;  
    }  
    public int sonKiHane(){  
        return no % 100;  
    }  
}
```

ERİŞİM BELİRLEYİCİLERİ



ERİŞİM BELİRLEYİCİLERİ

```
class kisi{
    public String ad;
    public int yas;
    public int no;
}

public class Main {
    public int sonKiHane(kisi k){
        return k.no % 100;
    }
    public boolean ilkiBuyukMu(kisi k1,kisi k2){
        return sonKiHane(k1)>sonKiHane(k2);
    }

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

```
class kisi{
    private String ad;
    private int yas;
    private int no;
    public void setYas(int yas){
        if(yas<150){
            this.yas = yas;
        }
        else
            yas = 0;
    }
    public int getYas(){
        return yas;
    }
    public int sonKiHane(){
        return no % 100;
    }
}
```

ERİŞİM DÜZEYLERİ

```
class kisi{
    private String ad;
    private int yas;
    private int no;
    public void setYas(int yas){
        if(yas<150)
            this.yas = yas;
        else
            yas = 0;
    }
    public int getYas(){
        return yas;
    }
    public int sonKiHane(){
        return no % 100;
    }
}
```

```
public class Main {
    public static void main(String[] args){
        kisi k = new kisi();
        k.yas = 30;
    }
}
```

```
public class Main {
    public static void main(String[] args){
        kisi k = new kisi();
        k.setYas(30);
    }
}
```


ELEMAN FONKSİYONLAR (METOTLAR)

- Sınıf içerisinde
- Tüm elemanlara erişim
- Aşırı yüklenebilirler

ELEMAN FONKSİYONLAR (METOTLAR)


```
class Kisi{
    public String ad;
    public int yas;
}

public class Main {
    public static void main(String[] args){
        Kisi k = new Kisi();
        k.ad = "Emir";
        k.yas = 253;
        System.out.println(k);
    }
}
```



```
class Kisi{
    private String ad;
    private int yas;
    private int no;
    public void setYas(int yas){
        if(yas<150) this.yas = yas;
        else yas = 0;
    }
    public int getYas(){ return yas; }
    public int sonUkSureti(){ return no % 100; }
}

public class Main {
    public static void main(String[] args){
        Kisi k = new Kisi();
        System.out.println(k.yasdir());
    }
}
```



```
System.out.println(k.ad + " " + k.yas);
```

YAPICI FONKSİYONLAR

- Bildirim anında çağırılır
- İlk değer atama
- İsim
- Dönüş türü
- Aşırı yüklenebilirler

YAPICI FONKSİYONLAR

```
class Kisi{
    private String ad;
    private int yas;
    private int no;

    public Kisi(){
    }

    public Kisi(String ad,int yas,int no){
        this.ad = ad;
        this.yas = yas;
        this.no = no;
    }

    public void setYas(int yas){
    }
}
```

```
public class Main {
    public static void main(String[] args){
        Kisi k = new Kisi();
        k.ad = "Emir";
        k.yas = 233;
        System.out.println(k);
    }
}
```

```
public class Main {
    public static void main(String[] args){
        Kisi k1 = new Kisi();
        Kisi k2 = new Kisi( ad: "Emir", yas: 255, no: 203022);
    }
}
```

YIKICI FONKSİYONLAR

- Yaşam döngüsü
- Bellek iadesi
- Bitirilmesi gereken işlemler
 - Dosyalar
 - Veritabanları
 - Ağ bağlantıları
- Non - deterministik
- Aşırı yüklenemezler
- Değer almaz / döndürmezler

ÇÖP TOPLAYICI (GARBAGE COLLECTOR)

- Garbage Collector
- Heap
- Nesiller
 - Eden
 - Survivor
 - Tenured
 - Permanent

*<https://medium.com/@tugrulbayrak/jvm-garbage-collector-nedir-96e76b6f6239>

*<https://stackoverflow.com/questions/2129044/java-heap-terminology-young-old-and-permanent-generations>