

Nesne Tabanlı Programlama I

Öğr. Gör. Dr. Aysun ALTIKARDEŞ

AMAÇ

- × Kalıtım
- × Çok Biçimlilik

üzerinde önemle durularak, temel programlama bilgilerinin kazandırılması amaçlanmıştır.

KALITIM(INHERITANCE)

* Kalıtım yeniden sınıflar oluşturmak ve düzenlemek için kullanılan temel bir nesne yönelimli tasarım tekniğidir.

Kalıtım bir yazılım geliştiricidir, varolan bir sınıftan yeni bir sınıf türetmeyi sağlar.

* Örnek vericek olursak :

* Güvercin Kus üst sınıfından türer.Bir güvercin kuşta olması gereken özellikleri alır(Uçma,kanat gibi).

* Kuşta Hayvan sınıfından türer Hayvanda olan özellikleri alır(beslenme, barınma gibi).

KALITIM'IN ÖZELLİKLERİ

- * Mevcut Sınıfın özellikleri alt sınıfa aktarılır.
- * Türeyen sınıfa yeni özellikler aktarılabilir.
- * Alt sınıflara inildikçe özellikleri daha da artar.
- * Bir sınıftan birçok sınıf türeyebilir.
- * Private değişkenlere alt sınıflar erişemez.
- * Her sınıf Object sınıfından türer.
- * Davranışlar alt sınıflarda değişebilir.
- * Örnek olarak Her hayvan beslenir.
- * Aslanlar ise et ile beslenir.

* Javada class isminin yanına extends getirilerek
* türetme yapılır.

* Kullanımı :

* [Erişim] class [Class ismi] extends [Türetme yapılacak Class]

SORULAR

Soru: Kalıtım yolu ile bir sınıftan, yeni sınıflar türetilebilmesinin, türetilen sınıflara etkisi nedir?

Cevap: Türetilen her bir sınıf, türediđi sınıfın özelliklerindeki devralır.

Buradan, türetilmiş bir sınıf içerisinde, türediđi sınıfa ait üyelere erişilebileceğiz. Bu erişimde bazı kuralları vardır. Erişim belirleyicilerinin etkisi veya aynı üyelerin kullanılışı gibi durumlar örnek olarak gösterilebilir

SORULAR

Soru: Neden bir sınıftan başka sınıflar türetiriz ki?

Cevap: Tüm sınıflarda ortak olan özellikleri tek bir sınıf içerisinde toplamak.

Kalıtımın bu türünde, her sınıf için ortak olan üyelerin tekrar yazılmasını engelleriz. Ve sınıflar arasında düzenli bir [hiyerarşi](#) yapısının oluşmasını sağlarız. Örneğin, Bizim aileyi bir temel sınıf olarak düşünebiliriz. Bu aileden aileye ait değişik kategorileri türetebiliriz.

Ailem

Erkekler

Dedem

Babam

Hakan UĞUR (Ben)

Kızlar

Babaannem

Annem

Klasik kişiler

Dedem

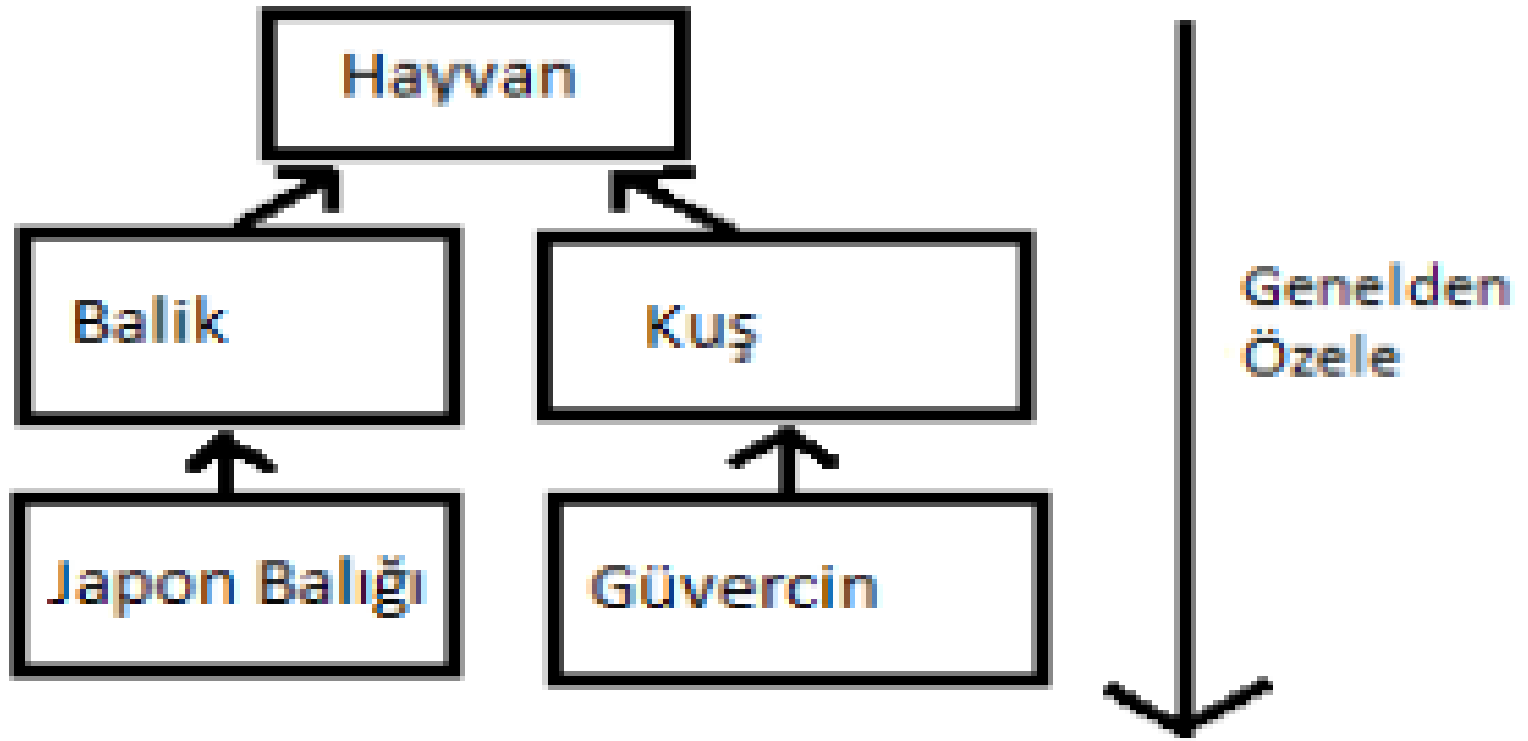
Babam

Sportik kişiler

Hakan UĞUR

İşte basit bir örnek. Aile sınıfı dışında başka örneklerde verebiliriz. Örneğin, Memeli hayvanlar gibi.

İki sınıf arasında kalıtım özelliğinin olduğunu anlayabilmek için **is-a** adı verilen bir ilişkinin kullanılır. Yani, **the cat is a mammiferous. Kedi bir memelidir.** Bu ilişkiyi yakaldıysak işte o zaman kalıtımdan söz edebiliyoruz. Yukarıdaki örnekte olduğu gibi.



KALITIM UYGULAMA

File Edit Run Window Help



Quick Access

package-info.java Hayvan.java Kus.java Balık.java Guvercin.java japonbaligi.java sonuc.java

```
package kalitim;  
  
public class Hayvan {  
    public void beslenme(){  
        System.out.println("hayvanlar beslenir...");  
    }  
    public void barınma(){  
        System.out.println("hayvanlar barınır...");  
    }  
    public void diski() {  
        System.out.println("hayvanlar diski bırakır...");  
    }  
    public void boyut(){  
        System.out.println("hayvanların boyutları vardır...");  
    }  
    public void solunum(){  
        System.out.println("hayvanlar solunum yapar...");  
    }  
}
```



package-info.java Hayvan.java Kus.java Balık.java Guvercin.java japonbalığı

```
package kalitim;
```

```
public class Kus extends Hayvan{  
    public void ucma(){  
        System.out.println("kuslar uçar..");  
    }  
    public void barınma() {  
        System.out.println("kuslar samanlardan yaptıkları"+"yuvalarda barınırlar");  
    }  
}
```

package-info.java Hayvan.java Kus.java Balık.java Guvercin.java japonbalığı.java sonuc.java

```
package kalıtım;
```

```
public class Balık extends Hayvan{  
    public void sudayuzer(){  
        System.out.println("balıklar suda yuzer");  
    }  
    public void solunum(){  
        System.out.println("balıklar solungaçları ile solunum yapar");  
    }  
}
```




package-info.java Hayvan.java Kus.java Balık.java Guvercin.java japonbalığı.java sonuc.java

```
package kalıtım;

public class japonbalığı extends Balık {
    public void solunum(){
        super.solunum();
    }
}
```



package-info.java Hayvan.java Kus.java Balık.java Guvercin.java japonbalığı.java sonuc.java

```
package kalitim;  
  
public class sonuc {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        Guvercin guvercin= new Guvercin();  
        Kus kus= new Kus();  
        guvercin.beslenme();  
        kus.beslenme();  
        System.out.println();  
        guvercin.barınma();  
        guvercin.boyut();  
        System.out.println();  
        kus.boyut();  
        kus.solunum();  
        System.out.println();  
        Balık balık= new Balık();  
        balık.sudayuzer();  
        balık.solunum();  
        System.out.println();  
        japonbalığı jp= new japonbalığı();  
        jp.solunum();  
    }  
}
```

Ekran Çıktısı

Güvercin buğday yer

Hayvanlar beslenir

Kuşlar Samanlardan Yaptıkları Yuvalarda Barınırlar

Güvercinin boyutları 20-30cm Arasıdır

Hayvanların Boyutları vardır

Hayvanlar Solunum yapar

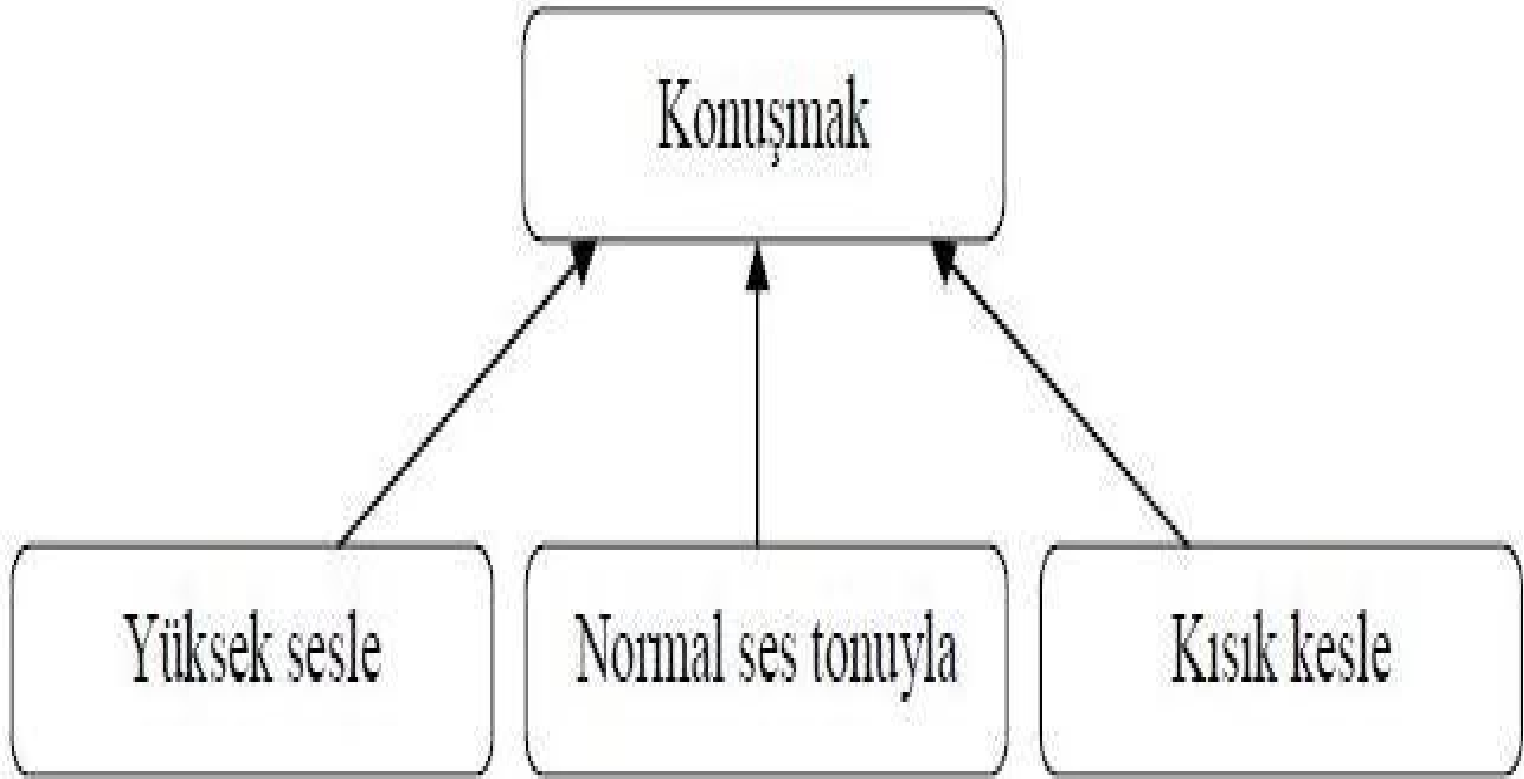
Balıklar suda yüzer

Balıklar Solungaçları ile Solunum Yapar

Balıklar Solungaçları ile Solunum Yapar

ÇOK BİÇİMLİLİK (POLYMORPHISM)

Çok biçimlilik, kalıtım ile iç içe geçmiş ilk bakışta karışık gibi görünen, hatta ne gerek var bunlara dedirten, bir yapıya sahiptir. En genel anlamda, oluşturduğumuz nesnelere gerektiğinde kılıktan kılığa girip başka bir nesneye gibi davranabilmesine polimorfizm diyebiliriz.



Çok biçimlilik ,bir metodun farklı nesnelere veya farklı ortamlarda farklı sonuçlar üretmesidir.

Mesela “konuşmak” insan sınıfına ait bir metottur.

ÇOK BİÇİMLİLİK UYGULAMA

Source Refactor Navigate Search Project Run Window Help



Arac.java Kamyon.java Otobus.java adiyok.java

```
public class Arac {  
  
    public String motorHacmi;  
    public String renk;  
    public String modelYili;  
  
    public Arac(String motorHacmi,String renk,String modelYili)  
  
    {  
        this.motorHacmi=motorHacmi;  
        this.modelYili=modelYili;  
        this.renk=renk;  
    }  
    public void EkranaYaz()  
  
    {  
        System.out.println("Motorlu Aracımız");  
  
        System.out.println(motorHacmi+" motor hacminde" +renk+ "renginde" +modelYili+ "modeldir");  
    }  
}
```

```
public class Kamyon extends Arac {  
  
    public String kasaHacmi;  
  
    public Kamyon(String motorHacmi, String renk, String modelYili,String kasaHacmi)  
    {  
        super(motorHacmi,renk,modelYili);  
  
        this.kasaHacmi=kasaHacmi;  
    }  
  
    public void KasayiBosalt()  
    {  
  
        System.out.println("Kasa boşaltılıyor...");  
  
    }  
  
    @Override  
  
    public void EkranaYaz()  
    {  
  
        System.out.println("Kamyonumuz");  
        System.out.println(motorHacmi+"motor hacminde"+renk+ "renginde"+modelYili+ "model"+kasaHacmi+" kasa hacmindedir");  
  
    }  
  
}
```

```
public class Otobus extends Arac {

public String yolcuSayisi;

public Otobus(String modelYili,String motorHacmi,String renk,String yolcuSayisi)
{
    super(modelYili,motorHacmi,renk);

    this.yolcuSayisi=yolcuSayisi;
}

public void OrtaKapiyiAc()
{

System.out.println("Orta Kapı Açıldı..");

}

@Override

public void EkranaYaz()
{

System.out.println("Otobüsümüz");

System.out.println(motorHacmi+" motor hacminde "+renk+" renginde "+modelYili+" model "+yolcuSayisi+" yolcu kapasitesindedir");

}
```



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

        Arac arac1=new Kamyon("5999","Beyaz","1999","10590");
        arac1.EkranaYaz();

        Arac arac2=new Otobus("5000","Mavi","2000","2005");
        arac2.EkranaYaz();

        Arac a= new Arac("1598", "Beyaz", "2010");
        a.EkranaYaz();

        System.out.println("*****");

        Otobus o= new Otobus("6000", "Sarı", "2006", "45");
        o.EkranaYaz();
        o.OrtaKapiyiAc();

        Kamyon k= new Kamyon("5999", "Mavi", "2009", "10690");
        k.EkranaYaz();
        k.KasayiBosalt();
    }}
}
```