***INHERITANCE:***

ÖNEMLİ: Bir Class tan başka bir Class ı türetirken EXTENDS kelimesini kullanıyoruz:

= Hayvan Klasından Kopek türetiyorum.

**HAYVAN ÖRNEĞİ:**

Önce Hayvan Class ını oluşturuyoruz:



Şimdi de Hayvan dan Köpek türetelim: Dikkat: Köpekle ilgili bir constructor yazarken, onu türettiğimiz Hayvan Class ını dikkate almamız gerekiyor:

***SUPER CLASS:*** Bir Class hangi Class tan türetiliyorsa, o Class süper class olarak adlandırılır. Üst Class olduğunu belirtmek için “super” anahtar kelimesini kullanacağız.

Önce Kopek constructor ını yazacağız. İçine Hayvan constructor ının içindeki field ları koyacağız. Süslü parantez içinde this….. demeden, bir üst sınıftan alındığını belli etmek için “super” kelimesiyle atıf yapacağız. Parametreleri de türünü belirtmeden yalnızca isimlerini yerleştireceğiz:

metin içeren bir resim

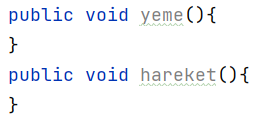
Açıklama otomatik olarak oluşturuldu

Köpeklerin hayvanlarla olan ortak özellikleri Hayvan sınıfından alındı. Şimdi Köpek sayfasındayız ve buraya extra sadece köpeklere özgü olan field ları yazalım:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şimdi de Köpeğin işlevlerini yazalım, diğer bir adla metotlarını oluşturalım:



Metodların içini daha sonra dolduracağız.

Şimdi Kopek constructor kodundayız.

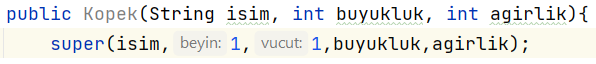


Burada iki konu öne çıkıyor:

1. Zaten Hayvan constructor kodunda belirlendiği için fieldları silebiliriz. Ancak silinen her bir field “super” satırında uyarı veriyor, zira orada değerini girmemizi istiyor. Örneğin beyin ve vücut u siliyoruz: Altta otomatik olarak değer girmemizi isteyecek, bir nevi default yapmış oluyoruz:



Şimdi iki alana da değer gireceğim:



1. Diğer konu ise, köpeklere özgü bu sayfada yazdığımız field ları, yine bu construcrot a yazmam gerekiyor: Dikkat, constructor da parantez içi ve süslü parantez içi kısımlarını şöyle dolduruyoruz: Tipik kod mantığı doğrultusunda;

Bu arada, bu köpeklere özgü field lar hiçbir yerde field lara atanmadığı için, super satırının altına this. ile field ları field lara atamamız gerekiyor. (yukarıda atanmış halde)

Şimdi Main a gidelim ve Hayvan Class ından bir hayvan nesnesi türetelim, sonra da Köpek Class ından kopek nesnesi türetelim:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şimdi bizden constructor ların içinin doldurmamız isteniyor (kırmızı alt çizgi):

metin içeren bir resim

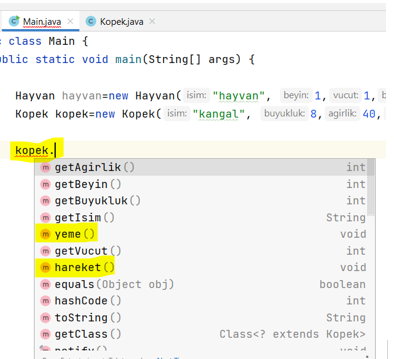
Açıklama otomatik olarak oluşturuldu

Bu arada yeme ve hareket metotlarını Kopek sayfasından Hayvan Class ının sayfasına almamız daha mantıklı. Zira tüm hayvanlara ait işlevler. Bu metotları hayvan class ına aldıktan sonra Maine gidip metotları çağıracağız. Ancak çağrıldığını anlamamız için Hayvan class ına bir uyarı yazdıralım:

metin, iç mekan, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şimdi Main e gidip metotları çağıralım. kopek yazıp nokta dediğimizde hayvandan inheritance olarak metodu almış olacağız, otomatik olarak yeme ve hareket lsitede görünecek, seçeceğiz:



Çalıştırdığımızda metodun çağrıldığına dair belirttiğimiz uyarı ekrana yansıdı:

metin içeren bir resim

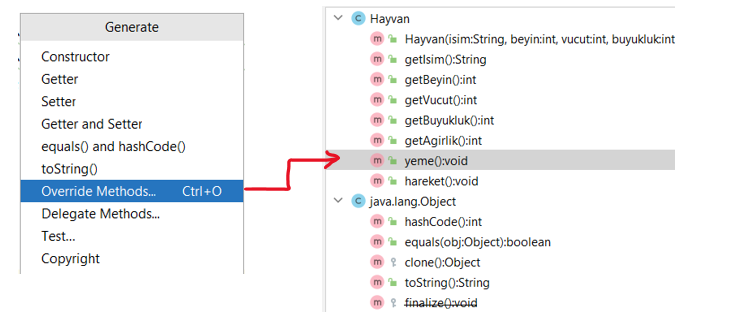
Açıklama otomatik olarak oluşturuldu

Şimdi, Hayvan a ait yeme fonksiyonu üzerinden Köpek e ait bir metot (cigneme) oluşturacağız. Bu işlem OVERRIDE işlemi olarak adlandırılır.

Önce Köpek sayfasına gidip cigneme metodunu oluşturalım ve sout ile uyarı mesajımızı yazalım.



Sonrasında Code dan Generate yapacağız: Override Methods u seçtiğimizde açılan pencere:



Bu şekilde metodumuz Kopek sayfasında geldi:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

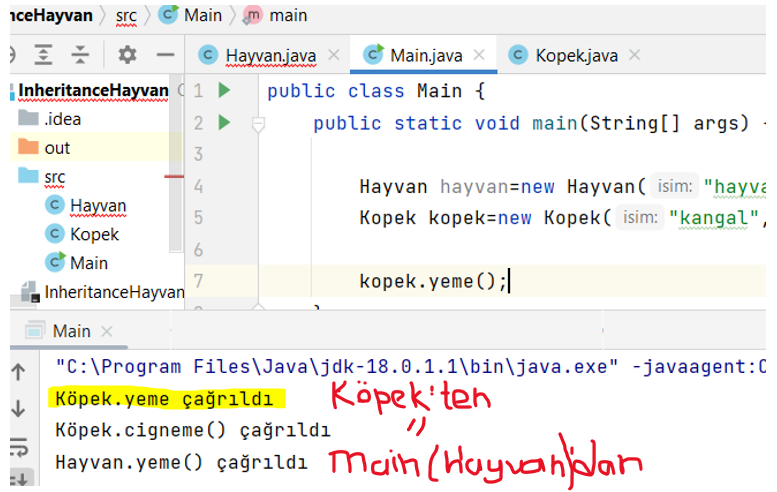
Metod süper class tan inherit edildi ancak bu haliyle bırakırsak bir şey yapmamış oluruz. Yeni bir şey eklememiz için:

Öncelikle sout ile bir bilgilendirme metodu yazalalım. Sonra da cigneme metodunu yazalım. Bu son yazacağım cigneme() metodu yemeyle aynı özellikte. Artık iki tanem yeme metodum var. Artık Kopek sayfasında doğrudan yeme() yazarsak, ilgili bulunduğumuz class taki yeme metodunu; super.yeme() yazarsak da Hayvan class ının yeme metodunu çağırmış olacağız.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şimdi tekrar çalıştıralım:

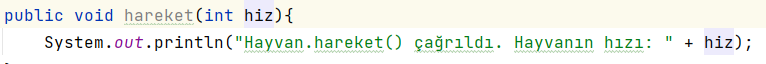


Main de değişiklik yapmadık, yeme() metodu çalıştı ve Kopek sayfasında oluşturduğumuz yapı karşılığı gelen bilgilendirme ekrana yansıdı.

Şimdi, köpek class ına *koşma()* ve *yürüme()* metodları yazalım. Bunları Hayvan Class ındaki hareket metodundan override olarak yapacağız.

Yürüme de hız önemli, bunu int hiz olarak parametre ekleyelim.

Hayvan Class ına geliyoruz ve hareket metodunun içine int hiz yazıyoruz. Bu arada bilgilendirme mesajının içine ekleme yapalım, “Hayvanın hızı:” şeklinde yazalım kontrol için:



Şimdi Kopek sayfasına geliyoruz. Burada yürüme ve koşma metodlarımızı formüle ediyoruz: Kopek sayfasında cigneme ve yeme vardı, şimdi yurume ve koşma metotlarını da oluşturmuş olduk: Bu arada hareketi çağırırken, h yazdığımızda otomatik olarak hareket geliyor. İçine değer yazarken de otomatik olarak hiz: geliyor.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şimdi Main sayfasında kopek.yurume yi çağıralım: Parametresi Kopek sayfasında girilmişti 5 olarak:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu



metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şimdi de kopek.kosma metodunu çağıralım:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Yeme metodunda güncelleme yapmıştık (super), şimdi de yurume ve koşma “hareket” le ilgili oldukları için hareket metodunda güncelleme yapalım:

Yeme ve hareket metotlarının ikisini de Hayvan sayfasında formüle etmiştik. Kopek sayfasından hareket metoduna atıf yapalım. Hayvan dan inherit edelim. Farklı özellikler oluşturalım. Mesela bütün hayvanlar bacağa sahip değiller. Bazısının var bazısının yok (yılan vb).

ÖNEMLİ: field larda olduğu gibi, metotlarda Access modifier *private* olarak belirlenebilir.



Şimdi de hareket metodunu override edeceğiz. Code + Generate + hareket:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şimdi burada hareket metodunun aynısını alıp koymuş olduk. Ekstra bir şey şu an yok. Bir yenilik katmam lazım o da *bacaklariHareketEttir* metodunu koyarak override etmeliyim.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Parametre girmem gerekiyor ki zaten kutucukta belirtmiş:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şimdi Main e gelip güncellenmiş haliyle hareket metodunu çalıştıralım: değeri 8 girelim:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Kopek.hareket metodunu çağırdığımızda, artık doğrudan hayvan Class ına gitmeyecek. Çünkü artık Kopek Class ında güncellenmiş metodumuz var. Kendi Class ında eğer o metod mevcutsa kendi Class ına gitmesi gerekir. Ekrana yansıyan dönütlerde bunu görüyoruz.

Şimdi, yurume metodu hareketle ilişkiliydi. Harekette güncelleme yapınca o da etkilendi. Buna bakalım.

Yurume metodunu önceden yapmıştık. Çağırdığımızda ekrana gelen dönüt şöyleydi:

metin, tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şimdi hareketteki güncellemeler sonrasında bakalım ne oldu:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Artık hayvan Classındaki harekete değil, yürümenin olduğu ve hareketin override edildiği Kopek Class ına gidip oradan çağırdı: Dikkat: Yukarıdaki ekran dönütü, aşağıdaki bu sayfadaki sıraya göre yansıyor.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Aynı şey kopek.kosma da da olacaktır:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şimdi konuyu teyit etmek için şöyle yapabiliriz: Kopek Class ında hareket metoduna yaptığımız override kod bloğunu komple silelim:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Ve kopek.kosma metodunu çağıralım:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu sefer sadece kendi Class ında bulunduğu için metodu Hayvan class ındaki formuyla çağırdık ve bu da ekrana da o şekilde yansıdı.

Tekrar aktif edersek yine ekranda eski dönüt yazacak.

Burada super anahtar kelimesi önemli. Kopek.kosma dediğimizde hareketi override ettiğimiz kod bloğuna gitmesini bekliyoruz. Ancak şayet kosma metodunun kodunda super anahtar kelimesini yazarsak hareket metodunun formüle edildiği Hayvan class ından çağırmış olacağız :

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şimdi kosma yı çalıştırdığımızda alt satırdaki bilgilendirmeden anlaşıldığı şekilde kod önce 1. kopek sonra da 1. hayvan class ından alıp döndürmüş:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Ancak yürüme de super anahtar kelimesini kullanmadığımız için hayvan class ındaki hareket metoduna gitmeden kopek class ındaki hareket metoduna gidecek.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu