**Memory Leak(Hafıza sızıntısı)**

Destructor ın amacı sadece cout ile mesaj vermek değil

Class’ların içinde oluşturduğun pointerları delete etmemize de yarar

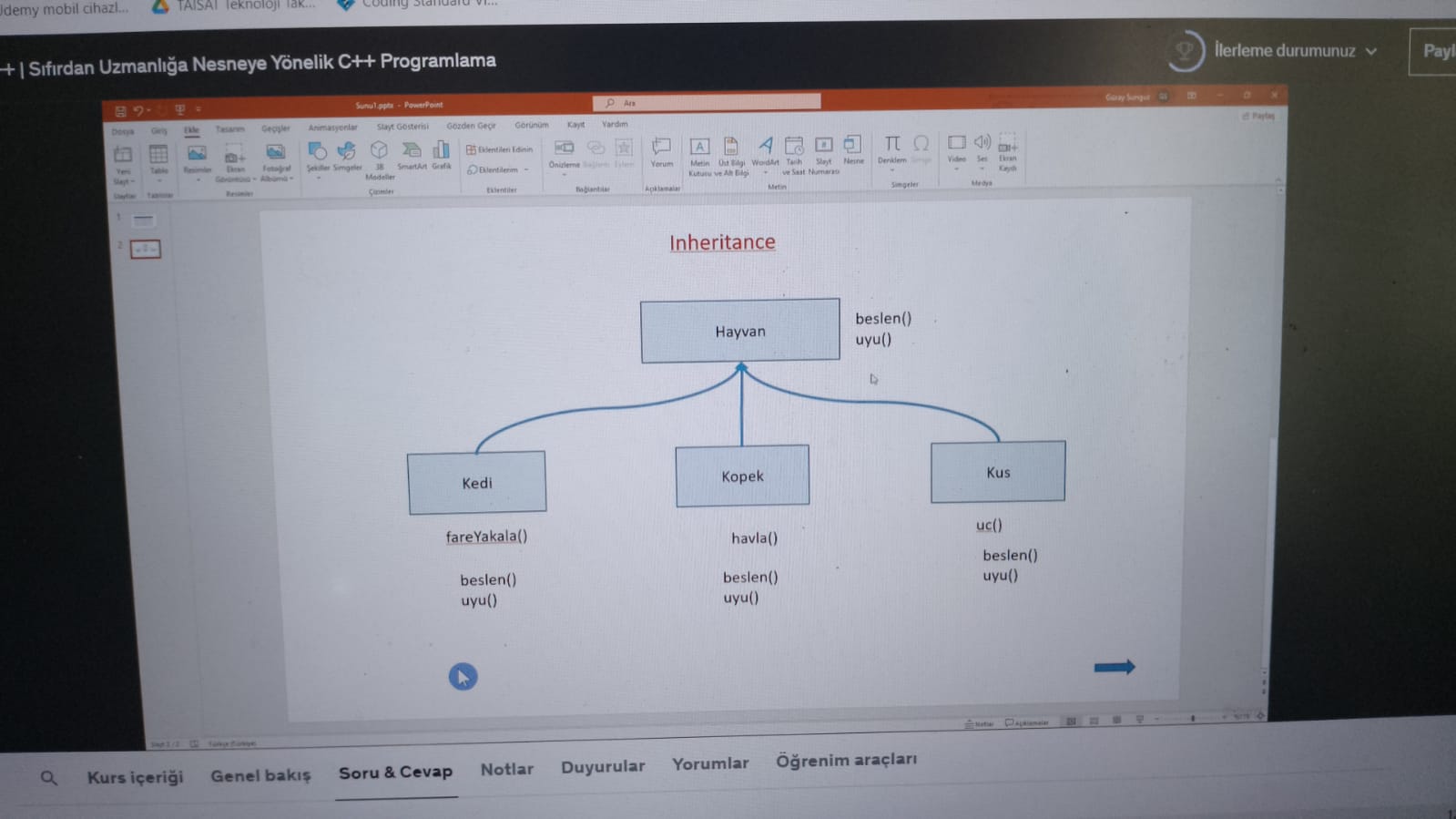
Çünkü işaret ettiği reserved diyip içerisine değer koyduğu bölge sıfırlanmıyor delete demeden

Yoksa Ramde kalmaya devam eder , başka uygulamalar açtığında ram ilgili uygulamanın herhangi bir bilgisini buraya yazmaz

metin, yazı tipi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Inheritance(Kalıtım)**

****

Kedi kopek kus birer class dır, Ama hepsi de hayvan class ın altında bir class dır.

Ortak özellikleri 🡺beslen(),uyu() gibi özellikleri tekrar tekrar diğer class’lara da bulaştırmamak için üst class’ın(Hayvan) dan kalıtım yoluyla devralıp kullanabilir. Aynı zamanda kedinin hemen altına yazdığımız gibi beslen(),uyu() metodlarını tekrar tekrar yazmamıza gerek kalmaz. Çünkü kalıtım yoluyla devralır.

ekran görüntüsü, metin, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

İlk olarak Kus ‘un Hayvan class ının bir alt class ı olduğunu gösteriyoruz. Ve Hayvan adına bir şeyler yapacağımız içinde include ediyoruz drive ı.

\_ucur , kuşlara ait bir sınıf metoduyken sadece Kus sınıfında oluşturulan nesneler bu metodu kullanacak diğer metodlar ortak olacak

metin, yazılım, multimedya yazılımı, grafik yazılımı içeren bir resim

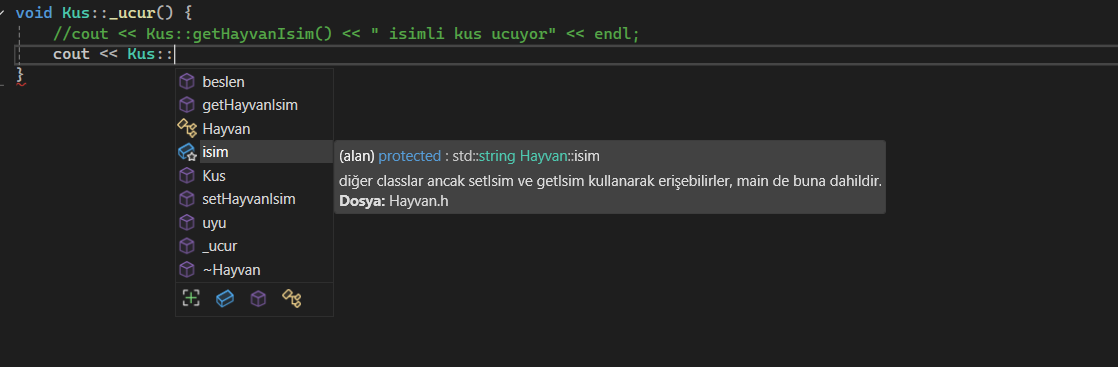
Açıklama otomatik olarak oluşturuldu

Kus constructor u Hayvan constructor una bağlanarak isim atılacak

Kus sınıfı , set&get , beslen, uyu metodlarına da erişebiliyordu, bu yüzden \_ucur metodunda getHayvanIsim metodunu kullanabildik.

-------------------------------------------------------------------------------------------------------------------------

**Protected Sınıf Üyeleri**

****

Protected üye olunca alt sınıflar ana sınıfın üyelerine de set ve get metodları kullanmadan erişebiliyor. Ama diğer alt sınıfı olmayanlar ise yine erişemiyor. Yani altclassları iyi yazanlar arasındaki set&get encapsulation metodunu aradan çıkarmış oluyor.

metin, yazı tipi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

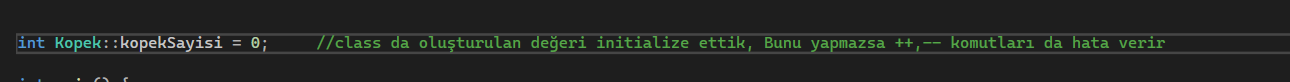
**Static Sınıf Üyeleri ve Metodları**

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Diyelim ki kaç tane ürün üretildi bunların sayımını istiyorsun, haliyle class da bir static keyword ü ile bir değişken atadın.

Tabi mainde de init değerini sıfıra eşitledikten sonra



.cpp dosyasında

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Constructor ile destructor metodlarının altına arttırma ve azaltma işlemini yaparız. Bu sayede o class da en son durumda kaç tane obje türetildi bulmuş oluruz.

metin, ekran görüntüsü, yazılım, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Türetilen nesne adetini yukarıdaki gibi almış oluyoruz.



Pointer ile de obje için memory block açıp oluşturmuş oluyoruz

metin, ekran görüntüsü, yazı tipi, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Delete edince tekrardan sayı 3 e düşüyor

Colsolun 2. Satırında delete dediğimizden kaynaklı hayvan class destructor cagirildi mesajı geliyor çünkü ram üstünden kaldırmış oluyor, delete etmezsek peki ?

metin, ekran görüntüsü, yazı tipi, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

4 obje var ama 3 tanesi delete edildi, diğeri ramin biyerlerinde kaldı , restart verene kadar pcyi , kalmaya devam edecek

Peki Statik metodlar nedir ?

Public içinde değişken(member) tanımlamak pek tavsiye edilmez. Cünkü dışarıdan başkası da erişebilir.

ekran görüntüsü, metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Örneğini yukarıda verdik. Tut ki adam sadece köpek sayısını merak ediyordu baktı ki sayimiz adında bir değişken var. He dedi tamam bu sayıyı vercek baktı 122 çıktı. Halbu ki diğer hayvanların arasında belki de köpek 22 adetti. Haliyle data ile ilgili büyük problemlere yol açar ve algoritma karışır

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şimdi diyelim ki hadi köpek sayısı diye tanımladık ama şöyle bir durum var belki yazarken farkedemeden eli kayacak bu değişkeni çağıracak. O yüzden bu değişkeni private yapıp encopsulation kurallarına uyup set get ile çağırmakta fayda var.



metin, multimedya yazılımı, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**PoliMorfisim**

Temel olarak bakarsak;

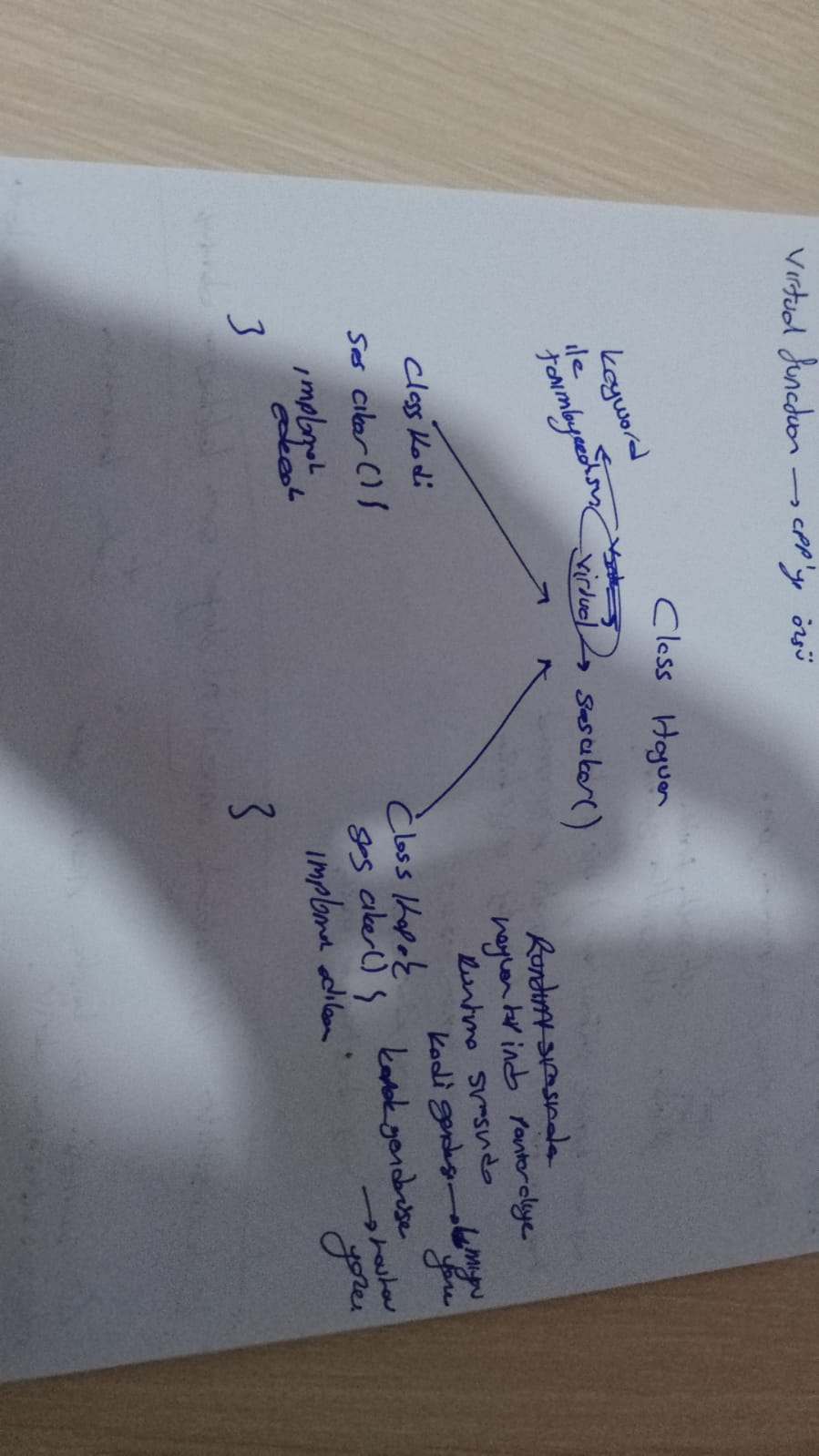
Bir ana hayvan class ı var, bunun altında da kedi ve köpek class ları var. İşte hayvan classının referansı kedi yahut köpeğe dönüşebilir.

Önce tırtıl sonradan kelebek olması gibi, geçiş yapabilir

Virtual Function a temel olarak bakarsak;

C++ a özgü bir kavramdır.

Polimorfizmi c++ de kullanabilmemiz için yazmamız gereken bir keyword, bir metod da kullanıyorsun



metin, yazılım, multimedya yazılımı, grafik yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

Yani hayvan classından bir pointer oluşturuyorsun ve bu kedi objesini tutuyor. Ses çıkar diyince kedi sınıfındaki metod çalışıyor, yazdir diyince beklenen yine kedi sınıfındaki yazdir metodu olmasına rağmen Hayvan sınıfındaki yazdir metodu çağırılıyor. Çünkü o virtual değil.

Runtime esnasında referanstan bahsedilen hayvan tipinde yaratılan pointera atanan kedi adresi ,

Hemen ardından köpeğinkini atsam Hav Hav mesajı gelir.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

Ve yine fark edildiği üzere yazdir komutu Hayvan classından , ses çıkar komutu ise kopek classından geldi

Virtual functions sayesinde polimorfizim uyguluyoruz. Polimorfizm demek çok biçimlilik demek,

Üst sınıftaki bir nesne referansın yeri geldiğinde alt sınıftaki nesne gibi davranabilmesine olanak tanır(Kedi gibi miyav miyav, köpek gibi hav hav)

Polimorfizm bizi kod kalabalığından da kurtarıyor Referans kısmı ucKezSesCıkar fonksiyon içerisinde gerçekleşiyor, mainde metodu çağırırken değil, oluşturulan pointer Hayvan tipinde ve bildiğin kedinin nesne adresi(direk kedi) gelirse miyavlama, köpek nesnesi gelirse havlama mevcut

Hem clean code kurallarına daha uygun oluyor

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Hayvan class’ında sözde sesCıkar vardı ama karışmadı çünkü virtual

ekran görüntüsü, metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Kediyi gönderince kedinin ki , köpeği gönderince köpeğinki çalıştı

**Friends Methods**

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Private olan markaya friends sistemi haricinde nasıl ilgili değişkene erişebileceğimizi yukarda gösterdik. Şimdi sıra friends sistemini kullanmada

Friend olunca aynı method class içinde olduğu için sanki class içindeki metoda erişir gibi private değişkenlere de erişebiliyor.

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, yazı tipi, ekran görüntüsü, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şimdi ise friend class lara bakalım**

ekran görüntüsü, metin, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

Ogrenci. dan sonrası gelmiyor cunkü class larımız friend değil. Private olanlara ulaşım sağlayamaz şu an.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

Not3 ü öğrenci.h da public e alınca bir başka class nesne üzerinden erişebiliyor ki bu deneyip bildiğimiz şey

Ama derdim private olanlara erişmek

metin, yazı tipi, multimedya yazılımı, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Friend class keyword ü ile dost kılmak istediği class ı tanımlıyoruz.

metin, yazılım, multimedya yazılımı, grafik yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şimdi ise private olanlara erişim sağlandı,

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

Ama hesaplama1 objesi görüldüğü gibi doğrudan ogrenci classındaki bir şeye erişemiyor. Yani ne olacaksa class dışına çıkmadan içeride hallediliyor.

