## \_\_init\_\_

Class’ın arka yüzünde mevcut olan bir gömülü fonksiyondur.

Aslında bizim her class kullanımımızda arka planda bir \_\_init\_\_ çalışır

Bize, sürekli değişken tanımlamaktansa \_\_init\_\_ içine

Yani bir nevi GLOBAL ALANA değişken tanımlayarak

Bu değişkenleri çağırmamıza ve kullanmamıza olanak sağlar.

Kullanımı ise örnekteki gibidir.

Properties (Özellik barındıran Sınıflar)

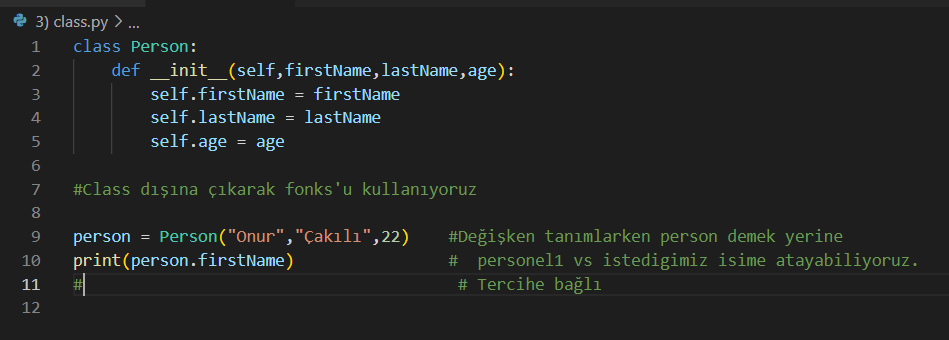
Stabil bir Classtan ziyade, aktif olan

Bilgi girilmesi gereken bir Class kullanılacaksa

Burada Özellikli Classlar tercih ederiz.

Örnegin. Personel Bilgi Girişi Alan Bir Person Class’ı

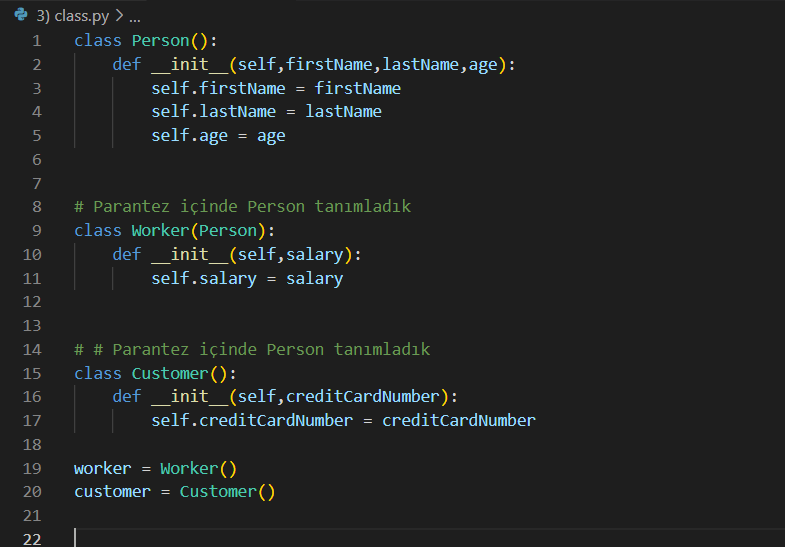
Örnk.



NOT: Bu örneğinde mantığı yine aynı şekilde çalışıyor.

Global alana değişkeni tanımlıyoruz gibi ve buralara veri girişi yapıyoruz

İnheritance (Miras) Yapısı



## inheritance(miras) yapısı

İç içe Class kullanmamızı sağlar

Adı üzerinde Bir Class’ın özelliklerini başka bir Class’a Miras bırakıyor

Açıklama: Person Class’ında ait olan ad,soyad,yaş alanları

Worker ve Customer Class’larında da kullanılmakta.

Fazladan yer kaplamaktansa Persondaki özellikleri bu diğer iki Class içinde

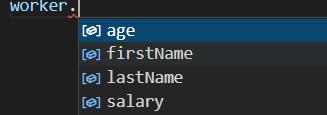
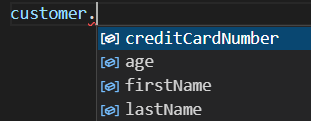
Kullan diyoruz.

Böylelikle, 2. Class’ların hem kendine has (unique) özellikleri hem de

Diğer classlardan gelen özellikleri üzerinde toplamış oluyor.

-Customer Classında ## lastName,firstName,age,creditCardNumber

Worker Classında ### lastName,firstName,age,salary

Örnk1.  Örnek2. 

Kullanırken

Class Worker (Person):

+++ şeklinde parantez içerisine 1. Class’ın adını yazmak yeterli oluyor

Böylece onun özellikleri 2. Class’ta da kullanılıyor.

NOT: Yazdırma İşlemleri ise yine normal Class ile aynı

Module Oluşturma

Modül, bir kod dosyasını başka bir kod dosyasının içine ekleyerek

Arada köprü görevi görmeye yarar.

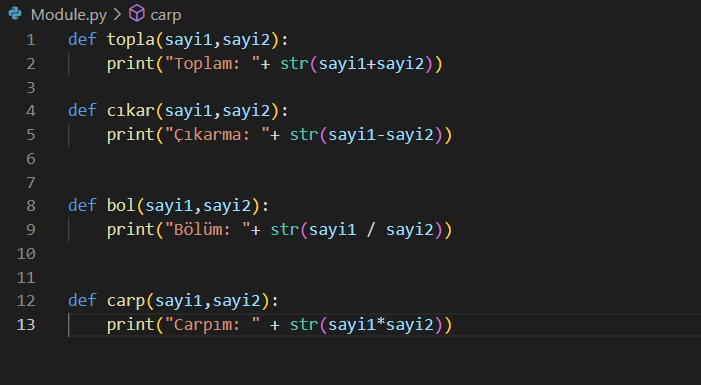
Listeler, değişkenler, fonksiyon ve classlar

Her şeyi bu şekilde kullanabiliriz

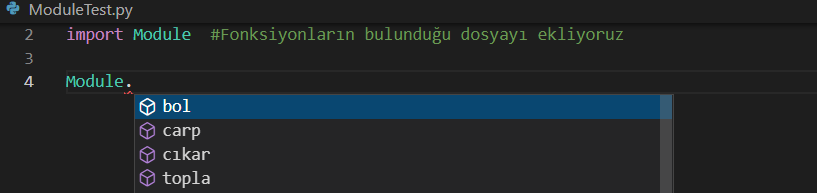
Yani Dosyalar arası etkileşim

Örnek.

1. Dosya = Module.py



1. Dosya = ModuleTest.py

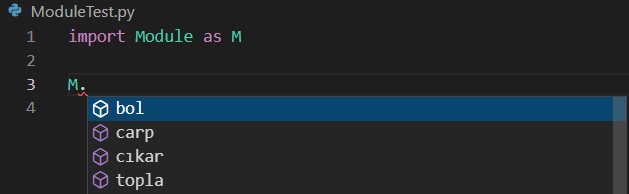


Kullanırken,

İmport < DOSYA\_ADİ >

BİLGİ:

As = “ diye ”

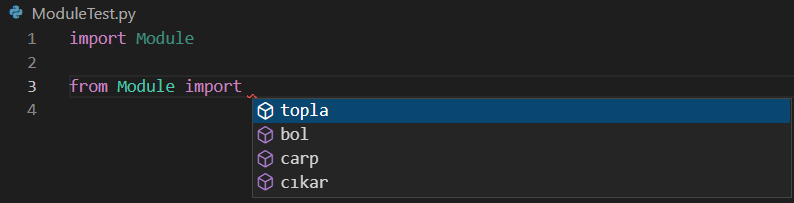


Module, takma isim vererek = ‘ M ’ diye çağırdığımda getir anlamında

SQL mantığı

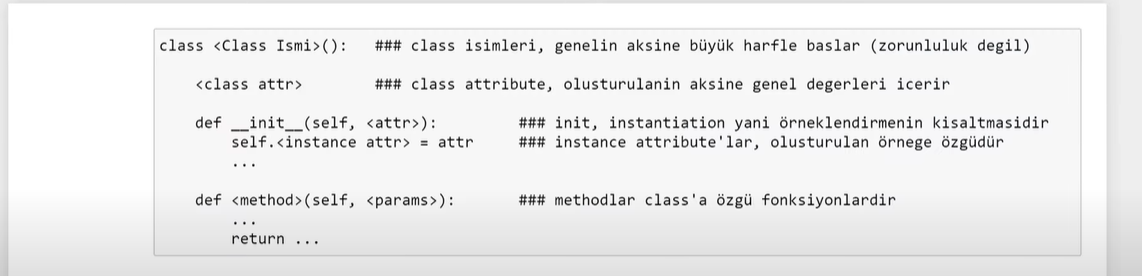
BİLGİ:

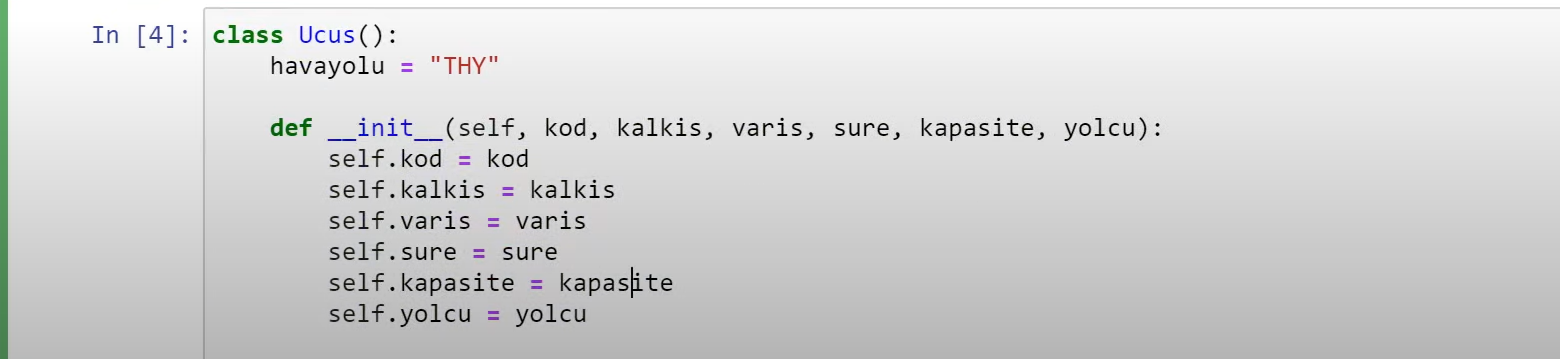
Modül’ün tamamını değilde sadece bize gerekli olan kısmı çağırmak istiyorsakta



From Module import < eklenecek\_kısım >

# Python Dersleri #25 | Class Yapısı ve OOP (Object Oriented Programming) Giriş





Class’a tanımlanan degiskeni sabit kalırken,

Fonksiyonlara tanımlanan degisken farklılık göstererek kullanılabilir ()

Ucus.havayolu = thy

Ucus1.havayolu = thy

Ucus2.havayolu = thy

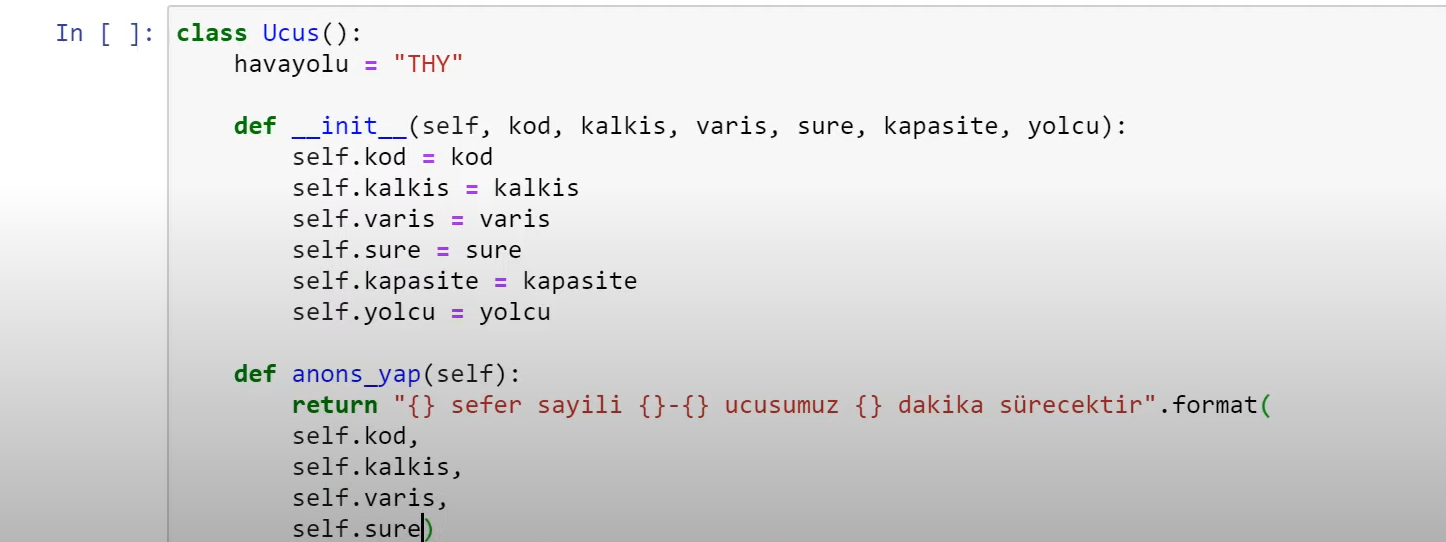
Fakat

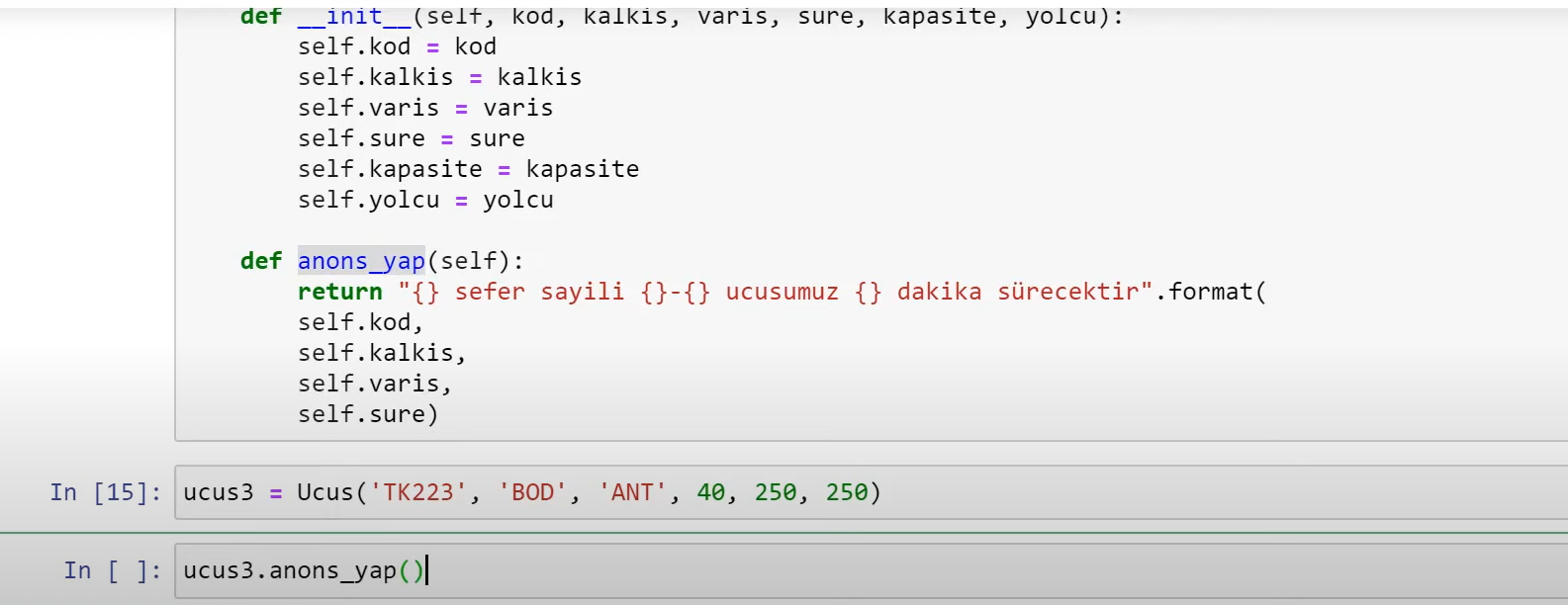
Ucus1.kapasite = 50

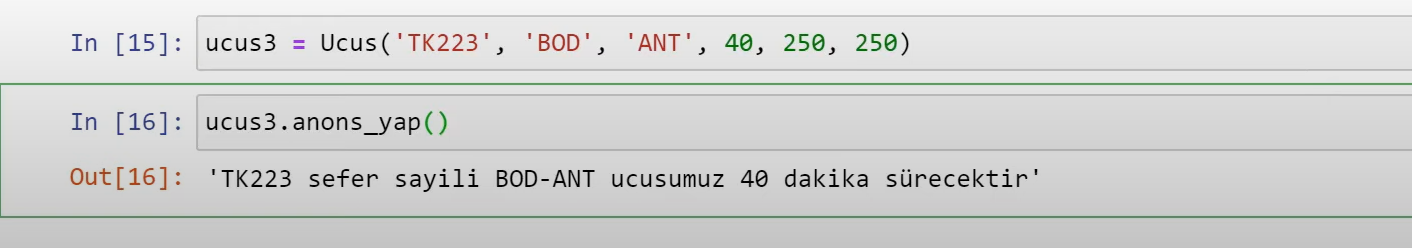
Ucus2. Kapasite = 150

Ucus3.kapasite = 40

Class içerisinde metot tanımlamak





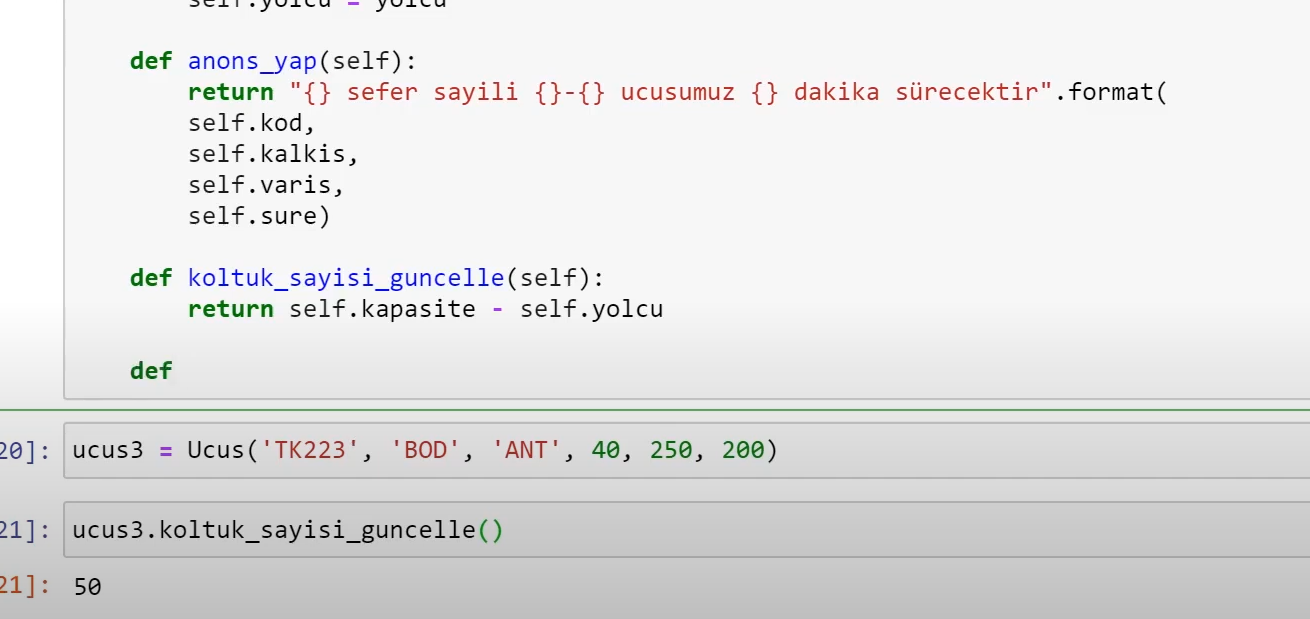


Class’in içerisinde parametreler verildiği için tekrar metot çağırınca bir parametre girilmesine gerek kalmıyor.

Methodu çağırırken

Ucus.anons\_yap() şeklinde, classa ait bir parçayı yine çağırarark methodu çalıştırabiliyoruz

Methodlar detay



.

Koltuk\_Sayisi\_güncelle metodunu çağırırken,

Dinamiklik kazanması bakımından

Self.koltuk\_sayisi\_güncelle olarak yazıyoruz.

bu yapılan işlemin amacı:

yukarıda kullanılan metodu bir nevi güncelliyoruz mevcut duruma göre

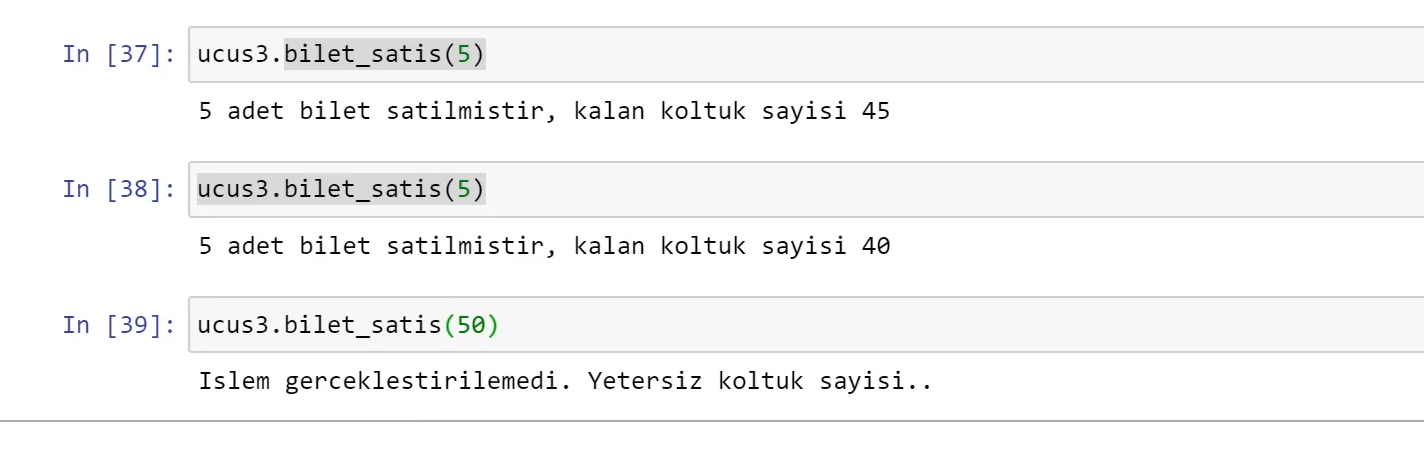
bu yüzden dinamik olması ve girilen parametre değerlerine göre işlem yapması bakımından

self. Olarak tanımlıyoruz



…………………..

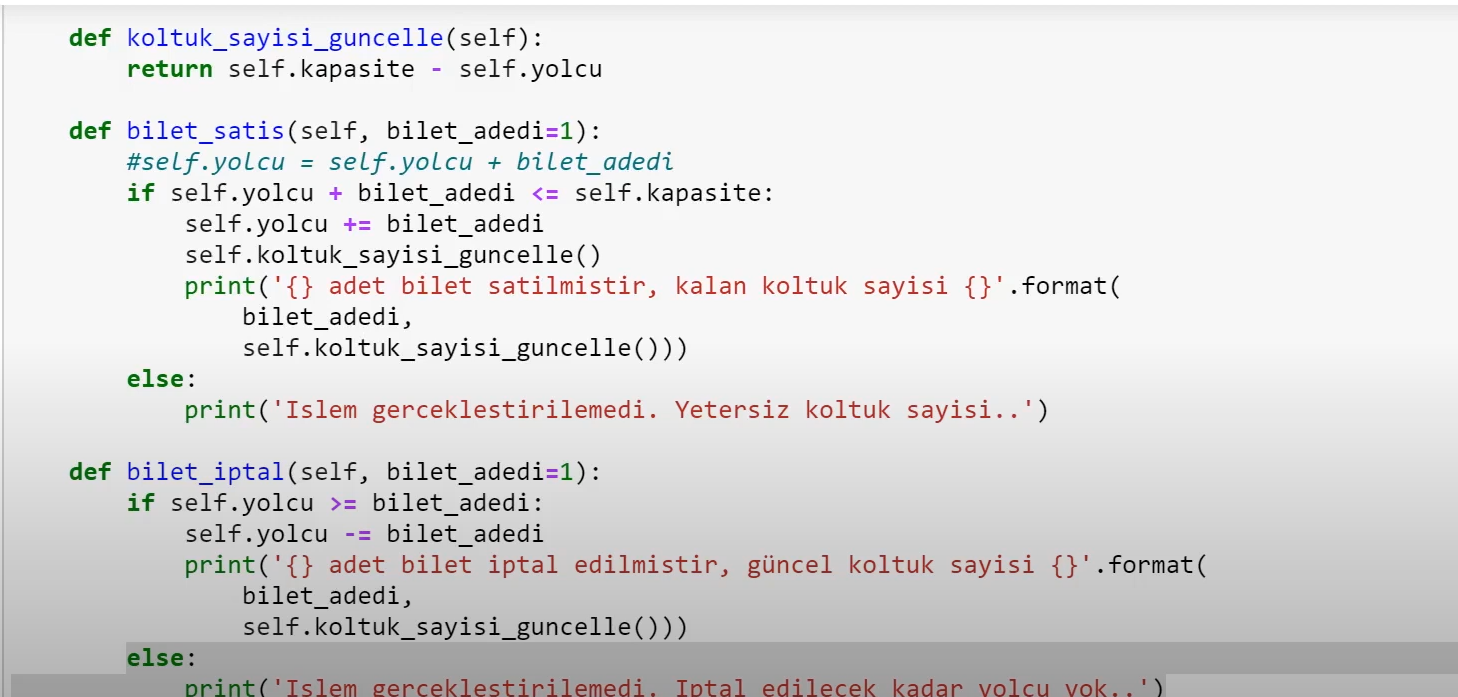
Dinamik bir class yapısı



50 kapasitesi vardı, ilk olarak 5 bilet satti

2. adımda 5 bilet daha satti (kapasitesi 40 kaldı)

3. adımda ise 50 tane bilet almak istedi ama kapasite yetersiz olduğu için işlem gerçekleştirilemedi.



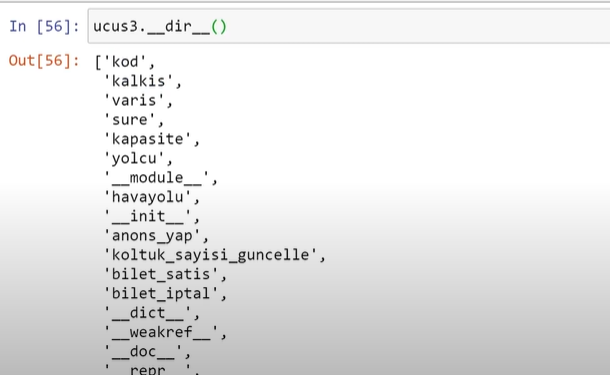
CLASSLARDA DUNDER METHODLAR VE KALITIM (İNHERİTANCE)

Dunder methodlar, \_\_init\_\_ gibi iki \_ ile başlayan hazır kalıp ve üzerinde degisklik yaparak kullanılabilecek methodlardır

Bunları görebilmek için

Örneğin.

Class\_adi.\_\_dir\_\_()



Biz görmesekte bunlar arka planda hazır gelıyor ve çalışıyor aslında

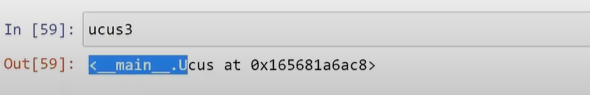
Örnek

\_\_repr\_\_ methodu mesela



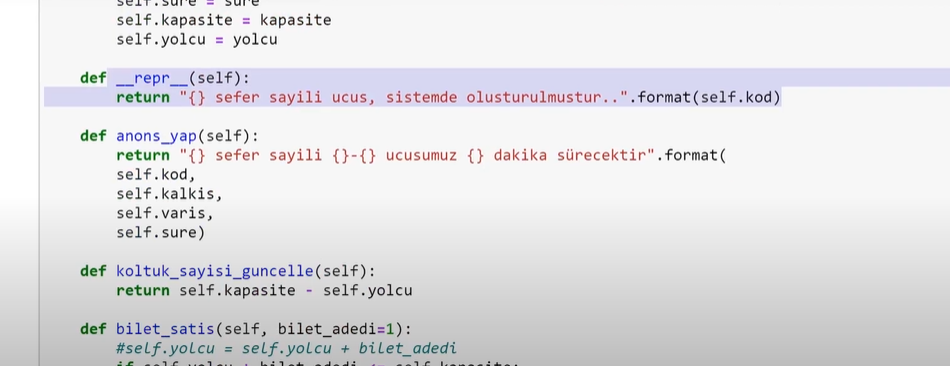
Normalde

Ucus3 şeklinde bir çağırma işlemi yapamıyoruz. Bunu yazdığımız zaman



Şeklinde bir şey dönüyor. Bu bir hata değil. Ucus3 icin gerekli işllemler oluşturuldu artık arka planda çalışıyor fakat yazdırma işlemi olmadığı ıcn bir çıktıyı göremıyoruz sadece

Ama \_\_repr\_\_ ile düzenleme yaparsak



Artık tanımladığımız yazı geri döndürülecek

