

class <Class İsmi>(): ### class isimleri, genelin aksine büyük harfle başlar (zorunluluk değil) <class attr> ### class attribute, oluşturulananın aksine genel değerleri içerir  
def \_\_init\_\_(self, <attr>): ### init, instantiation yani örneklendirmenin kısaltmasıdır self.<instance attr> = attr ### instance attribute'lar, oluşturulan örneğe özgüdür ... def  
<method>(self, <params>): ### methodlar class'a özgü fonksiyonlardır ... return ...

## class oluşturma, attribute'lar ve method'lar

```
In [1]: class Ucus():  
        havayolu = "THY" #bu atanan classa ait her şeyi tanımlıyor superior yani
```

```
In [3]: ucus1 = Ucus() #ucus1 i classımıza tanımlıyoruz artık onun bir elementi haline geliyor
```

```
In [4]: ucus1.havayolu #ucus1 classa atanınca superiorı sorgulayabiliriz
```

Out [4]: 'THY'

```
In [15]: class Ucus():  
        havayolu = "THY"  
  
        def __init__(self, kod, kalkis, varis, sure, kapasite, yolcu):  
            """  
            burada __init__ tanımlı bir fonksiyon  
            self de fonksiyonun kendi classına ait olduğunu dönüşlü olduğunu belirtiyor  
            """  
            self.kod = kod  
            self.kalkis = kalkis  
            self.varis = varis  
            self.sure = sure  
            self.kapasite = kapasite  
            self.yolcu = yolcu
```

```
In [25]: ucus2 = Ucus() #ucus2'yi Ucus classına tanıtırken instance attributeleri tamamlayarak vermemiz gerekemekte
```

```
-----TypeError-----Traceback (most recent call last)Cell In[25],  
----> 1 ucus2 = Ucus()  
TypeError: Ucus.__init__() missing 6 required positional arguments: 'kod', 'kalkis', 'varis', 'sure', 'kapasite', and 'yolcu'
```

```
In [37]: ucus2 = Ucus("TK2342", "IST", "BER", 9.5, 230, 219) #instance attributelar girilince kabul ediyor
```

```
In [27]: ucus2.varis
```

Out [27]: 'BER'

```
In [28]: ucus2.kod
```

Out [28]: 'TK2342'

```
In [29]: ucus2.sure
```

Out [29]: 9.5

```
In [30]: ucus2.havayolu
```

Out [30]: 'THY'

```
In [35]: ucus3=Ucus("TK4397", "ANK", "NYC", 23.5, 230, 228)
```

```
In [39]: class Ucus():  
        havayolu = "THY"  
  
        def __init__(self, kod, kalkis, varis, sure, kapasite, yolcu):  
  
            self.kod = kod  
            self.kalkis = kalkis  
            self.varis = varis  
            self.sure = sure  
            self.kapasite = kapasite  
            self.yolcu = yolcu  
  
        def anons(self): #yazılan fonksiyonun classa dönüşlü olacağını belirtmek için daima argüman kısmına self yazılmalı  
            return "{ sefer sayılı {}-{} uçuşumuz {} saat sürecektir.".format(  
                self.kod, #böyle alt alta yazarken de virgülü unutma!  
                self.kalkis,  
                self.varis,  
                self.sure)
```

```
In [40]: ucus3.anons()
```

Out [40]: 'TK4397 sefer sayılı ANK-NYC uçuşumuz 23.5 saat sürecektir.'

In [41]: ucus2.anons()

Out [41]: 'TK2342 sefer sayılı IST-BER uçuşumuz 9.5 saat sürecektir.'

```
In [97]: class Ucus():
        havayolu = "THY"
        #Attributelar
        def __init__(self, kod, kalkis, varis, sure, kapasite, yolcu):

            self.kod = kod
            self.kalkis = kalkis
            self.varis = varis
            self.sure = sure
            self.kapasite = kapasite
            self.yolcu = yolcu

        #Anons
        def anons(self):
            print("{} sefer sayılı {}-{} uçuşumuz {} saat sürecektir.".format(
                self.kod,
                self.kalkis,
                self.varis,
                self.sure))

        #Koltuk Sayısı Güncelleme
        def koltuk_sayisi_guncelle(self):
            return self.kapasite-self.yolcu

        #Bilet Satışı
        def bilet_satis(self, bilet_adedi=1):
            if bilet_adedi <= (self.kapasite-self.yolcu):
                self.yolcu += bilet_adedi
                self.koltuk_sayisi_guncelle()
                print("Satış işlemi başarılı. {} adet bilet satılmıştır. Güncel boş koltuk sayısı {}.".format(
                    bilet_adedi,
                    self.koltuk_sayisi_guncelle()))
            else:
                print("Satış işlemi için yeterli kadar boş koltuk sayısı mevcut değildir. {} adet boş koltuk mevcuttur.".format(
                    self.koltuk_sayisi_guncelle()))

        #Bilet İptali
        def bilet iptal(self, bilet_adedi=1):
            if bilet_adedi <= self.yolcu:
                self.yolcu -= bilet_adedi
                self.koltuk_sayisi_guncelle()
                print("İptal işlemi başarılı. {} adet bilet iptal edilmiştir. Güncel boş koltuk sayısı {}.".format(
                    bilet_adedi,
                    self.koltuk_sayisi_guncelle()))
            else:
                print("Satış işlemi için yeterli kadar dolu koltuk sayısı mevcut değildir. {} adet dolu koltuk mevcuttur.".forma
                    self.koltuk_sayisi_guncelle()))
```

```
In [126]: ucus1 = Ucus("TK8623", "IZM", "LON", 12.5, 230, 217)
        ucus2 = Ucus("TK2342", "IST", "BER", 9.5, 230, 219)
        ucus3 = Ucus("TK4397", "ANK", "NYC", 23.5, 230, 228)
```

In [108]: ucus1.koltuk\_sayisi\_guncelle()

Out [108]: 13

In [109]: ucus1.bilet\_satis(10)

Satış işlemi başarılı. 10 adet bilet satılmıştır. Güncel boş koltuk sayısı 3.

In [110]: ucus1.koltuk\_sayisi\_guncelle()

Out [110]: 3

In [111]: ucus1.bilet iptal(10)

İptal işlemi başarılı. 10 adet bilet iptal edilmiştir. Güncel boş koltuk sayısı 13.

In [112]: ucus1.koltuk\_sayisi\_guncelle()

Out [112]: 13

In [113]: ucus1.bilet\_satis(1000)

Satış işlemi için yeterli kadar boş koltuk sayısı mevcut değildir. 13 adet boş koltuk mevcuttur.

In [114]: ucus1.koltuk\_sayisi\_guncelle()

Out [114]: 13

In [115]: ucus1.bilet iptal(1000)

Satış işlemi için yeterli kadar dolu koltuk sayısı mevcut değildir. 13 adet dolu koltuk mevcuttur.

```
In [116]: ucus1.koltuk_sayisi_guncelle()
```

Out [116]: 13

```
In [119]: ucus1.bilet iptal() #eğer değer girilmezse default bilet_adi=1 devreye girer ve o kullanılır
```

İptal işlemi başarılı. 1 adet bilet iptal edilmiştir. Güncel boş koltuk sayısı 14.

```
In [120]: ucus1.bilet_satis() #eğer değer girilmezse default bilet_adi=1 devreye girer ve o kullanılır
```

Satış işlemi başarılı. 1 adet bilet satılmıştır. Güncel boş koltuk sayısı 13.

## Dunder Method'lar (Double Under Score)

```
In [121]: #__init__ gibi methodlar dunder methodtur ve classa önceden tanımlıdır
ucus3.__dir__() #bize classa tanımlı tüm attribute ve dunderları verir
```

```
Out [121]: ['kod',
'kalkis',
'varis',
'sure',
'kapasite',
'yolcu',
'__module__',
'havayolu',
'__init__',
'anons',
'koltuk_sayisi_guncelle',
'bilet_satis',
'bilet iptal',
'__dict__',
'__weakref__',
'__doc__',
'__new__',
'__repr__',
'__hash__',
'__str__',
'__getattr__',
'__setattr__',
'__delattr__',
'__lt__',
'__le__',
'__eq__',
'__ne__',
'__gt__',
'__ge__',
'__reduce_ex__',
'__reduce__',
'__subclasshook__',
'__init_subclass__',
'__format__',
'__sizeof__',
'__dir__',
'__class__']
```

```
In [123]: ucus3 #burada bize verinin tutulduğu yeri veriyor
```

Out [123]: <\_\_main\_\_.Ucus at 0x7f78001ff910>

```
In [124]: class Ucus():
    havayolu = "THY"
    #Attributelar
    def __init__(self, kod, kalkis, varis, sure, kapasite, yolcu):

        self.kod = kod
        self.kalkis = kalkis
        self.varis = varis
        self.sure = sure
        self.kapasite = kapasite
        self.yolcu = yolcu

    #Kayıt Bilgisi
    def __repr__(self):
        return "{} sefer sayılı uçuş sisteme kaydedilmiştir.".format(self.kod)

    #Anons
    def anons(self):
        print("{} sefer sayılı {}-{} uçuşumuz {} saat sürecektir.".format(
            self.kod,
            self.kalkis,
            self.varis,
            self.sure))

    #Koltuk Sayısı Güncelleme
    def koltuk_sayisi_guncelle(self):
        return self.kapasite-self.yolcu

    #Bilet Satışı
    def bilet_satis(self, bilet_adi=1):
        if bilet_adi <= (self.kapasite-self.yolcu):
            self.yolcu += bilet_adi
            self.koltuk_sayisi_guncelle()
            print("Satış işlemi başarılı. {} adet bilet satılmıştır. Güncel boş koltuk sayısı {}.".format(
                bilet_adi,
                self.koltuk_sayisi_guncelle()))
        else:
            print("Satış işlemi için yeterli kadar boş koltuk sayısı mevcut değildir. {} adet boş koltuk mevcuttur.".format(
                self.koltuk_sayisi_guncelle()))

    #Bilet İptali
    def bilet iptal(self, bilet_adi=1):
```

```

if bilet_adi <= self.yolcu:
    self.yolcu -= bilet_adi
    self.koltuk_sayisi_guncelle()
    print("İptal işlemi başarılı. {} adet bilet iptal edilmiştir. Güncel boş koltuk sayısı {}.".format(
        bilet_adi,
        self.koltuk_sayisi_guncelle()))
else:
    print("Satış işlemi için yeterli kadar dolu koltuk sayısı mevcut değildir. {} adet dolu koltuk mevcuttur.".format(
        self.koltuk_sayisi_guncelle()))

```

```

In [130]: ucu3
#Aşağıdaki kısım ile sağladık:
#Kayıt Bilgisi
# def __repr__(self):
# return "{} sefer sayılı uçuş sisteme kaydedilmiştir.".format(self.kod)

```

Out [130]: TK4397 sefer sayılı uçuş sisteme kaydedilmiştir.

```

In [128]: ucu3.__dir__()

```

```

Out [128]: ['kod',
'kalkis',
'varis',
'sure',
'kapasite',
'yolcu',
'__module__',
'havayolu',
'__init__',
'__repr__',
'anons',
'koltuk_sayisi_guncelle',
'bilet_satis',
'bilet iptal',
'__dict__',
'__weakref__',
'__doc__',
'__new__',
'__hash__',
'__str__',
'__getattr__',
'__setattr__',
'__delattr__',
'__lt__',
'__le__',
'__eq__',
'__ne__',
'__gt__',
'__ge__',
'__reduce_ex__',
'__reduce__',
'__subclasshook__',
'__init_subclass__',
'__format__',
'__sizeof__',
'__dir__',
'__class__']

```

## Inheritance (Kalıtım)

```

In [168]: class Seyehat():

    def __init__(self, kalkis, varis):
        self.kalkis=kalkis
        self.varis=varis

    def __repr__(self):
        return "{}-{} seferi sisteme kaydedilmiştir.".format(self.kalkis,self.varis)

    def anons(self):
        return "{}-{} seferine hoş geldiniz.".format(self.kalkis,self.varis)

class Otobus(Seyehat): # parantez içine yazılanın alt class ı olduğunu ve inherit ettiğini belirtiyoruz
    def __init__(self, mola, kalkis, varis):
        Seyehat.__init__(self, kalkis, varis) #kalkış ve varış bilgilerini seyahat süper classından almasını belirtiyoruz
        self.mola=mola
    def __repr__(self):
        return "{}-{} seferi sisteme kaydedilmiştir.".format(self.kalkis,self.varis)
    def anons(self):
        return "{}-{} seferine hoş geldiniz. {} ve {} duraklarında mola verilecektir.".format(
            self.kalkis,
            self.varis,
            self.mola[0],
            self.mola[1])

```

```

In [169]: seyahat1=Seyehat("ANT","BOD")
print(seyahat1)
print(seyahat1.anons())

```

ANT-BOD seferi sisteme kaydedilmiştir.  
ANT-BOD seferine hoş geldiniz.

```

In [170]: otobus1=Otobus(["FET","ALN"], "ANT", "BOD")
print(otobus1)
print(otobus1.anons())

```

ANT-BOD seferi sisteme kaydedilmiştir.  
ANT-BOD seferine hoş geldiniz. FET ve ALN duraklarında mola verilecektir.

In [ ]: