

Sınıflar

Nesne tabanlı programlamanın temelinde, yukarıdaki giriş bölümünde de adını andığımız ‘sınıf’ (class) adlı bir kavram bulunur. sınıflar, nesne üretmemizi sağlayan veri tipleridir. İşte nesne tabanlı programlama, adından da anlaşılacağı gibi, nesneler (ve dolayısıyla sınıflar) temel alınarak gerçekleştirilen bir programlama faaliyetidir.

Bir kişi işe alındığında, o kişiye dair belli birtakım bilgileri bu veritabanına işliyorsunuz. Mesela işe alınan kişinin adı, soyadı, unvanı, maaşı ve buna benzer başka bilgiler.

```
class Çalışan:  
    pass
```

Yukarıdaki, boş bir sınıf tanıımıdır. Hatırlarsanız fonksiyonları tanımlamak için def adlı bir ifadeden yararlanıyorduk. İşte sınıfları tanımlamak için de class adlı bir ifadeden yararlanıyoruz. Bu ifadenin ardından gelen Çalışan kelimesi ise bu sınıfın adıdır.

Eğer arzu ederseniz, yukarıdaki sınıfı şu şekilde de tanımlayabilirsiniz:

```
class Çalışan():  
    pass
```

fonksiyonlarda olduğu gibi, bir sınıfı kullanabilmek için de öncelikle o sınıfı tanımlamamız gerekiyor

Sınıf Nitelikleri

```
class Çalışan():  
    kabiliyetleri = []  
    unvanı = 'işçi'
```

Burada unvanı ve kabiliyetleri adlı iki değişken tanımladık. Teknik dilde bu değişkenlere ‘sınıf niteliği’ (class attribute) adı verilir.

```
class Çalışan():  
    kabiliyetleri = []  
    unvanı = 'işçi'  
    maaşı = 1500  
    memleketi = ""  
    doğum_tarihi = ""
```

‘Çalışan’ adlı bir grubun ortak niteliklerini belirledik. Elbette her çalışanın memleketi ve doğum tarihi farklı olacağı için sınıf içinde bu değişkenlere belli bir değer atamadık. Bunların birer karakter dizisi olacağını belirten bir işaret olması için yalnızca memleketi ve doğum_tarihi adlı birer boş karakter dizisi tanımladık.

Sınıf niteliklerine, doğrudan sınıf adını kullanarak erişebilmek **mümkündür**.

```
print(Çalışan.maaşı)  
print(Çalışan.memleketi)  
print(Çalışan.doğum_tarihi)
```

bu sınıfa yeni sınıf nitelikleri de ekleyebilirsiniz

```
Çalışan.isim = 'Ahmet'  
Çalışan.yaş = 40
```

Sınıfların Örneklenmesi

```
class Çalışan():  
    kabiliyetleri = []  
    unvanı = 'işçi'  
    maaşı = 1500  
    memleketi = ""  
    doğum_tarihi = ""
```

```
ahmet = Çalışan() // sınıfımızı ahmet adlı bir değişkene atadık
```

Başka bir örnek

```
class Sipariş():  
    firma = "  
    miktar = 0  
    sipariş_tarihi = "  
    teslim_tarihi = "  
    stok_adedi = 0  
jilet = Sipariş()
```

Burada class, sınıfı tanımlamamıza yarayan bir öğedir. Tıpkı fonksiyonlardaki def gibi, sınıfları tanımlamak için de class adlı bir parçacığı kullanıyoruz. Sipariş ise, sınıfımızın adı oluyor. Biz sınıfımızın adını parantezli veya parantezsiz olarak kullanma imkanına sahibiz.

Sınıfın gövdesinde tanımladığımız şu değişkenler birer sınıf niteliğidir (class attribute):

```
firma = "  
miktar = 0  
sipariş_tarihi = "  
teslim_tarihi = "  
stok_adedi = 0
```

jilet = Sipariş() komutunu verdiğimizde ise, biraz önce tanımladığımız sınıfı örnekleyip (instantiation), bunu jilet adlı bir örneğe (instance) atamış oluyoruz. Yani jilet, Sipariş() adlı sınıfın bir örneği olmuş oluyor. Bir sınıftan istediğimiz sayıda örnek çıkarabiliriz:

```
kalem = Sipariş()  
pergel = Sipariş()  
çikolata = Sipariş()  
Bu şekilde Sipariş() sınıfını üç kez örneklemiş, yani bu sınıfın bütün özelliklerini taşıyan üç farklı üye meydana getirmiş oluyoruz.  
Bu sınıf örneklerini kullanarak, ilgili sınıfın niteliklerine (attribute) erişebiliriz:
```

```
kalem = Sipariş()  
kalem.firma  
kalem.miktar  
kalem.sipariş_tarihi  
kalem.teslim_tarihi  
kalem.stok_adedi
```

eriştığımız bu nitelikler birer sınıf niteliği olduğu için, sınıfı hiç örneklemeyen, bu niteliklere doğrudan sınıf adı üzerinden de erişebiliirdik:

```
Sipariş.firma  
Sipariş.miktar  
Sipariş.sipariş_tarihi  
Sipariş.teslim_tarihi  
Sipariş.stok_adedi
```

Gelin isterseniz pratik olması bakımından bir örnek daha verelim. Elimizde şöyle bir sınıf olsun.

```
class Çalışan():  
    kabiliyetleri = []  
    unvanı = 'işçi'  
    maaşı = 1500  
    memleketi = "  
    doğum_tarihi = "
```

Burada kabiliyetleri, unvanı, maaşı, memleketi ve doğum_tarihi adlı beş adet değişken tanımladık. Teknik dilde bu değişkenlere 'sınıf niteliği' (class attribute) adı verildiğini biliyorsunuz.

Çalışan() sınıfı içindeki niteliklere erişmek için birkaç tane örnek çıkaralım:

```
ahmet = Çalışan()  
mehmet = Çalışan()  
ayşe = Çalışan()
```

Bu şekilde Çalışan() sınıfının üç farklı örneğini oluşturmuş olduk. Bu sınıfın niteliklerine, oluşturduğumuz bu örnekler üzerinden erişebiliriz:

```
print(ahmet.kabiliyetleri)  
print(ahmet.unvanı)  
print(mehmet.maaşı)  
print(mehmet.memleketi)  
print(ayşe.kabiliyetleri)  
print(ayşe.doğum_tarihi)
```

Çıkardığımız örnekler aracılığıyla sınıf nitelikleri üzerinde değişiklik de yapabiliyoruz:

```
ahmet.kabiliyetleri.append('prezantabl')
```

hafta 3 (b)

Python nesne tabanlı bir dildir. Python da nesneleri (Object) oluşturmak için sınıflar (Class) kullanılır. Sınıflar, değişkenlerden ve fonksiyonlardan oluşur.

Fonksiyonlar, değişkenler gibi herhangi bir bilgiyi saklamaz - her fonksiyon çalıştırıldığında yeni başlar. **Bununla birlikte, bazı işlevler ve değişkenler birbirleriyle çok yakından ilgilidir ve birbirleriyle çok fazla etkileşime girmesi gerekir.**

Gerekli olan, birbirleriyle etkileşim kurabilmeleri için birbirine yakın bir yerde bulunan fonksiyonları ve değişkenleri gruplamanın bir yoludur.

Class (Sınıf):

Herhangi bir nesnenin özelliklerini (attributes) içeren kullanıcının tanımladığı bir prototip koddur. Bu özellikler members (elemanları) ve methods (fonksiyonları) olarak geçer ve bunlara erişim sağlanırken nesneden sonra bir nokta kullanılır.

Class variable (Sınıf değişkeni):

Bu değişken sınıftan türetilmiş tüm örnekler tarafından kullanılan ortak değişkendir. Class'ın içerisinde tanımlanır. Class'ın dışında onun metodları (alt fonksiyonları) tarafından tanımlanamaz. Class değişkenleri ondan türetilmiş örnekler (instance) tarafından kullanılamaz.

Instance variable (Örnek değişkeni):

Bu değişken bir metodun içinde tanımlanan ve sadece sınıfın o anki örneğine ait olan değişkendir.

Instance (örnek):

Belli bir sınıftan türetilmiş bağımsız bir birey nesnedir. Bir nesne hangi sınıfa ait ise onun bir örnek nesnesidir.

Method (metod):

Bir sınıf içinde tanımlanan özel bir fonksiyon türüdür.

Object (Nesne):

Class tarafından oluşturulan örnek bir veri yapısıdır. Yani bir kopyasıdır.

Defining a class:

```
# Defining a class  
class classname:  
    [statement 1]  
    [statement 2]  
    [statement 3]
```

[etc.]

Here is an example that creates the definition of a Shape:

#An example of a class

class Shape:

```
def __init__(self, x, y):  
    self.x = x  
    self.y = y  
    self.description = "This shape has not been described yet"  
    self.author = "Nobody has claimed to make this shape yet"  
def area(self):  
    return self.x * self.y  
def perimeter(self):  
    return 2 * self.x + 2 * self.y  
def describe(self, text):  
    self.description = text  
def authorName(self, text):  
    self.author = text  
def scaleSize(self, scale):  
    self.x = self.x * scale  
    self.y = self.y * scale
```

rectangle = Shape(100, 45)

#finding the area of your rectangle:

print(rectangle.area())

#4500

#finding the perimeter of your rectangle:

print(rectangle.perimeter())

#290

#describing the rectangle

rectangle.describe("A wide rectangle, more than twice\as wide as it is tall")

#making the rectangle 50% smaller

rectangle.scaleSize(0.5)

#re-printing the new area of the rectangle

print(rectangle.area())

#1125.0