

KLASE

Klasa - predložak u kojem se čuvaju podaci o nekom elementu: automobil i vozač, te ima implementirane neke radnje nad podacima

- **elementi klase:**
- svojstva (atributi) – čuvaju vrijednosti važne za neki element koji opisujemo (marka automobila, maksimalna brzina, ime vozača,...)
- metode – radnje koje izvodimo nad elementima (uključivanje motora, dodavanje gasa, kočenje,...)
- objekt – konkretizacija klase – nakon što klasi dodamo vrijednosti svojstava, konkretni element koji možemo pokazati

class je ključna riječ i obavezna je prilikom definiranja klase.

- **self** nije ključna riječ, nego je konvencija za ime prvog parametra metode. Prilikom poziva metode na instanci objekta Python će automatski postaviti taj objekt kao prvi parametar.
- Ako klasa ima metodu koja se zove **__init__** Python će je automatski izvršiti pri stvaranju novog objekta. Ova metoda je zgodna za inicijaliziranje atributa.

Deklaracija:

Class ImeKlase:

Definiranje klase

Definiciju klase čini niz metoda (funkcije) te svojstva (varijable)

- svaka metoda obavezno ima parametar **self** – preko njega dolazimo do svih svojstava i metoda klase

PRIMJERI:

Primjer 1:

```
#deklaracija klase/nema atributa u tijelu klase
#nije potrebno unositi atribut u klasu Posuda()
class Posuda:
#poziv funkcije unosa biskvita
    def insertBiskvit(self, biskvit):
        self.biskvit = biskvit
#poziv funkcije ispisa biskvita
    def getBiskvit(self):
        print (self.biskvit)
```

Deklaracija podrazumijeva naziv klase **Posuda** s dvije funkcije koje omogućavaju unos i ispis (**insertBiskvit**, **getBiskvit**). Ova klasa nema inicijaliziranih parametara na početku funkcijom **def __init__**. U drugom dijelu programa je napravljen poziv klase.

```
#pozivi klase i funkcija
mojaPosuda=Posuda()
mojaPosuda.insertBiskvit('mojBiskvit')
mojaPosuda.getBiskvit()
```

Unutar iste skripte je napravljen poziv klase. Kao prvo kreirali smo neku varijablu **mojaPosuda** u nju smo smjestili naziv **Posuda()** i na taj način kreirali objekt ili instancu klase. Kada kod ispisa dobijemo:

```
<__main__.Pravokutnik instance at 0x0296C198>
```

To znači da je kreirana jedna instanca (objekt) klase. Budući da nemamo parametara unaprijed definiranih, moramo pozvati funkciju **insertBiskvit('parametar')** koja nam omogućava unos parametra **biskvit**, nakon toga pozivamo funkciju **getBiskvit()** bez upisa parametara i dobivamo ispis onog što smo unijeli funkcijom **insertBiskvit('parametar')**.

Primjer 2:

```
#deklaracija klase
class Auto:
#inicijalizacija objekta
    def __init__(self, model, color):
        self.model = model
        self.color = color
#prikaz objekta
    def display(self):
        print ('Auto je', self.color, self.model)
#promjena boje
    def paint(self, new_color):
        self.color = new_color
-
```

Kod ovog primjera uz deklaraciju klase imamo i funkciju **def __init__** koja omogućava definiranje parametara klase: **model i color**. Nakon toga imamo funkciju **display** kojom ispisujemo parametre nekog objekta klase. Funkcijom **paint** omogućavamo mijenjanje boje kod **color** kao jednog parametra objekta klase **Auto**.

Poziv se nalazi u drugom dijelu programa:

```
#-----
#poziv klase i metoda(funkcija)
#stvaranje 1. objekta klase
car_1 = Auto('VW', 'pink')
car_1.display()
#stvaranje 2. objekta klase
car_2 = Auto('Mercedes', 'white')
car_2.display()
car_2.paint('black')
car_2.display()
```

Primjer 3:

```
#deklaracija klase
class Dog:
#stvaranje objekta
    def __init__(self, ime, boja):
        self.ime = ime
        self.boja = boja
#prikaz objekta
    def display(self):
        print("Podaci o psu", self.ime, self.boja)
#promjena boje psa
    def farba(self, nova_boja):
        self.boja=nova boja
```

Klasa je potpuno identičen deklaracije kao i prethodni primjer (Primjer 2) . Imamo dva parametra za pojedini objekt, funkciju prikaza parametara obejkta i mogućnost promjene boje. Poziv se nalazi u drugom dijelu programa.

```
#poziv klase i metoda.
pas_1=Dog('fido','smedja')
pas_1.display()
#poziv metode za promjenu boje psa
pas_1.farba('ljubicasta')
pas_1.display()
```

Primjer 4:

```
#deklaracija klase
class Pravokutnik:
#stvaranje objekta klase
    def insertStranice(self, a,b):
        self.a = a
        self.b = b
#racunanje površine
    def getPovrsina(self):
        print ("povrsina",self.a*self.b)
#racunanje opsega
    def getOpseg(self):
        print ("opseg", (2*self.a)+(2*self.b))
```

Sljedeća klasa je deklarirana s 2 parametra stranicom a i b za računanje opsega i površine pravokutnika. Imamo tri funkcije: **1. insertStranice**, **2. getPovrsina** i **3.getOpseg**. Prva omogućava proizvoljna unos brojeva stranica pravokutnika što ujedno predstavljaju parametre objekta klase. Zatim druge dvije funkcije računaju površinu i opseg pravokutnika. Kod poziva klase i kreiranja objekta klase, ne upisujemo parametre jer oni nisu ionako određeni u samoj klasi nego se pozivaju u funkciji **insertStranice**.

```
#pozivi klase
p=Pravokutnik()
#stvaranje objekta klase
p.insertStranice(12,4)
p.getPovrsina()
p.getOpseg()
```