

Objektumok 1.

Osztály, példány, attribútum, metódus, inicializálás

© Tóth József, 2015

<http://infotanar.blog.hu>

Osztály

- Új adattípus, amely ...
 - több adatot együtt tárol, és ...
 - az adatokkal kapcsolatos műveleteket is tartalmazhat.
- Például a string és a lista is osztály.
- Az osztály alapján létrehozott objektumokat **példányoknak** nevezzük.

Feladatok

- Készíts osztályt a koordináta rendszer egy pontjának adataihoz (x és y koordináta)!
- Hozz létre egy pont példányt az osztály alapján!
- Állítsd be a pont koordinátáit (2,3), majd írasd ki őket!

```
class Pont:  
    'egy pont adatai'
```

```
p1 = Pont()  
p1.x = 2  
p1.y = 3  
print(p1.x, p1.y)
```

obj1.py

Osztály
definíciója
(nagybetűvel
kezdődik!)

Példány
(objektum)
létrehozása

Példány-
(objektum-)
változók =
attribútumok

A zárójel kötelező!

Kimenet

2 3

Feladat

- Hozz létre még egy pont példányt, állítsd be a koordinátáit $(-5,6)$, majd írasd ki a két pont x , illetve y koordinátáit!

```
class Pont:  
    'egy pont adatai'
```

```
p1 = Pont()  
p1.x = 2  
p1.y = 3  
p2 = Pont()  
p2.x = -5  
p2.y = 6  
print(p1.x, p2.x)  
print(p1.y, p2.y)
```

Másik
példány
(objektum)
létrehozása

Példány-
(objektum-)
változók =
atribútumok

Kimenet

2 -5
3 6

Metódusok

- Olyan függvények, amelyeket az osztályban adunk meg, és az objektum adott példányával végeznek valamilyen műveletet.
- Egységbe záras: az objektum attribútumokat és metódusokat is tartalmaz.
- A metódusok első paramétere mindig a self!

Feladat

- Készíts a Pont osztályba egy távolság nevű metódust, amely kiírja a pont origótól mért távolságát! (A távolságot a Pithagorasz tétellel számítsa ki.)
- Írasd ki mindkét pont origótól mért távolságát!


```
class Pont:
    'egy pont adatai'

    def tavolsag(self):
        'pont távolsága az origótól'
        return (self.x**2 + self.y**2)**0.5
```

Metódus.
Első
paramétere
mindig: **self**

```
p1 = Pont()
p1.x = 2
p1.y = 3
p2 = Pont()
p2.x = -5
p2.y = 6
print(p1.tavolsag())
print(p2.tavolsag())
```

Metódus
hívása

Kimenet

```
3.605551275463989
7.810249675906654
```

Feladat

- Készíts mozgat néven még egy metódust, amely a pont x és y koordinátáját módosítja a kapott dx és dy értékekkel!
- Mozgasd a metódus segítségével a p2 pontot! (dx=2,dy=-1)

```
class Pont:
    'egy pont adatai'

    def tavolsag(self):
        'pont távolsága az origótól'
        return (self.x**2 + self.y**2)**0.5

    def mozgat(self, dx, dy):
        'pont mozgatása dx és dy értékekkel'
        self.x += dx
        self.y += dy

p1 = Pont()
p1.x = 2
p1.y = 3
p2 = Pont()
p2.x = -5
p2.y = 6
p2.mozgat(2, -1)
print(p2.x, p2.y)
print(p2.tavolsag())
```

Metódus
paramé-
terekkel.
Első
paramétere
mindig: **self**

Kimenet

-3 5
5.830951894845301

Metódus
hívása para-
méterekkel

Inicializáló metódus

- A példány létrehozásakor beállítja az attribútumok értékeit.
- Neve kötelezően: `__init__(self, ...)`
- Ezt szokás használni, nem a közvetlen beállítást.
- Így biztosan nem felejtjük el beállítani az attribútumokat.

Feladat

- Készíts inicializáló metódust a Pont osztályba, amely a kapott értékekre állítja be a pont koordinátáit!
- Ha valamelyik koordináta értékét nem kapja meg, akkor az legyen 0!
- Írd át a főprogramot is ennek megfelelően!

```
class Pont:
    'egy pont adatai'

    def __init__(self, x=0, y=0):
        'Kezdőértékek beállítása'
        self.x = x
        self.y = y

    def tavolsag(self):
        'pont távolsága az origótól'
        return (self.x**2 + self.y**2)**0.5

    def mozgat(self, dx, dy):
        'pont mozgatása dx és dy értékekkel'
        self.x += dx
        self.y += dy

p1 = Pont(2, 3)
p2 = Pont(-5, 6)
p2.mozgat(2, -1)
print(p2.x, p2.y)
print(p2.tavolsag())
```

Inicializáló
metódus.
Első
paraméter
mindig: **self**

Példányok
létrehozása
kezdőértékekkel

Kimenet

-3 5
5.830951894845301

Önálló feladat

- Készíts a Pont osztályba egy új metódust csere névvel, amely felcseréli a pont x és y koordinátáit!
- Hívd meg a metódust a p1 pontra, majd írasd ki a p1 koordinátáit a többi adat kiírása előtt!

```
class Pont:
    'egy pont adatai'

    def __init__(self, x=0, y=0):
        'Kezdőértékek beállítása'
        self.x = x
        self.y = y

    def tavolsag(self):
        'pont távolsága az origótól'
        return (self.x**2 + self.y**2)**0.5

    def mozgat(self, dx, dy):
        'pont mozgatása dx és dy értékekkel'
        self.x += dx
        self.y += dy

    def csere(self):
        self.x, self.y = self.y, self.x

p1 = Pont(2, 3)
p1.csere()
print(p1.x, p1.y)
p2 = Pont(-5, 6)
p2.mozgat(2, -1)
print(p2.x, p2.y)
print(p2.tavolsag())
```

Megoldás

Kimenet

```
3 2
-3 5
5.830951894845301
```