

Додавање атрибута

Пример: додати атрибуте инстанцираном објекту просте класе

```
class Tacka:
    pass

t1 = Tacka()
t2 = Tacka()
t1.x = 5
t1.y = 4
t2.x = 3
t2.y = 6
print(t1.x, t1.y)
print(t2.x, t2.y)
```

Даје:

5 4

3 6

T1 и t2 су инстанцирани објекти класе Tacka().

Програм креира празну класу Tacka, без података и без понашања.

Затим се креирају две инстанце класе Tacka и свакој од њих се додели x и y координата за позицију тачке у две димензије.

То се изводи доделом вредности атрибуту објекта коришћењем синтаксе: објекат.атрибут = вредност.

Ово се назива дот нотација.

Вредност која се додељује може бити основни тип података, уграђени тип или неки други објекат, па чак и функција или нека друга класа.

Такође се може рећи да су x и y **промењиве инстанце** (instance variables) које садрже податке које су јединствене за дату инстанцу.

Додавање понашања (методе)

Пошто је ООП засновано на интеракцији између објеката, потребно је додати неке акције (понашања) које мењају вредности атрибута.

Пример:

```
class Tacka:
    def resetovanje(self):
        self.x = 0
        self.y = 0
```

```
t = Tacka()
t.resetovanje()
print(t.x, t.y)
```

Даје: 0 0

У класу Tacka се додаје метода resetovanje (не користи аргументе) која поставља вредности за атрибуте x и y на 0.

У Пајтону, методе се формирају идентично као и функције.

Свака метода започиње са службеном речи def испред имена методе са заградама (листа параметара) и двотачком. Од следеће линије започиње тело методе које је увучено у односу на заглавље методе.

Обраћање самом себи (self)

Једина разлика између метода и функција је што све методе имају један обавезан параметар који се назива по обичају self (иако се може назвати и другачије).

Параметар self се употребљава у методи само као референца на објекат над којим се подиже сама метода.

Помоћу методе self се може приступити свим атрибутима и методама које су дефинисане на објекту.

Када се позива метода не мора се писати и self аргумент (нпр, t.resetovanje(self)) пошто се то по дифолту и дешава, тј Пајтон аутоматски придружује self методи која се позива.

Подизање функције на класи

Метода је само функција која се реализује на класи.

Уместо да се позива метода на објекту, метода се може подићи и на класи и експлицитно додати објекат као self аргумент:

```
class Tacka:
    def resetovanje(self):
        self.x = 0
        self.y = 0
```

```
t = Tacka()
Tacka.resetovanje(t)
print(t.x, t.y)
```

Излаз кода је исти као претходни пошто се заправо одиграо исти процес.

Ако се не укључи self аргумент у дефиницију класе (def resetovanje():) добија се порука о грешци.

Вишеструки параметри у методи

На претходном примеру се додаје нова метода која омогућава постављање тачке на било коју позицију а не само на координатни почетак.

```
class Tacka:
    def pomeranje(self, x, y):
        self.x = x
        self.y = y

    def resetovanje(self):
        self.pomeranje(0, 0)
```

```
tacka1 = Tacka()
tacka2 = Tacka()
tacka1.resetovanje()
tacka2.pomeranje(5,0)
print(tacka1.x, tacka1.y)
print(tacka2.x, tacka2.y)
```

Даје:

0 0

5 0

Метода pomeranje има два аргумента (x, y) и поставља вредности на self објекат, као што је то раније радила resetovanje метода.

Сада метода resetovanje позива методу pomeranje, пошто је ресетовање исто што и померање само на прецизно дефинисану тачку у простору.

То значи да метода resetovanje сада користи методу pomeranje тако што поставља њене аргументе на вредност 0.

Израда лабораторијских вежби:

Задатак 033: Инстанцирати празну класу Tacka објектима tacka_1 и tacka_2. Коришћењем дот нотације доделити x и y атрибутима ових објеката вредности 0 и 0 (tacka_1) па 5 и 3 (tacka_2).

```
class Tacka:
    '''Predstavljaja tacku u 2D prostoru.'''
```

```
tacka_1 = Tacka()
tacka_2 = Tacka()
tacka_1.x = 0
tacka_1.y = 0
tacka_2.x = 5
tacka_2.y = 3
print(tacka_1.x, tacka_1.y) #0 0
print(tacka_2.x, tacka_2.y) #5 3
```

Задатак 034: Креирати класу Tacka. Креирати методу resetovanje класе Tacka. Нека метода нема аргументе већ само додељује атрибутима x и y, вредност 0.

```
class Tacka:
    '''Predstavlja tacku u 2D prostoru.'''
    def resetovanje(self):
        self.x = 0
        self.y = 0
```

Задатак 035: Креирати класу Tacka и у њој методу resetovanje која нема аргументе већ само додељује атрибутима x и y, вредност 0. Инстанцирати класу објектом tacka_1 и приступањем методи класе приказати атрибуте x и y објекта.

```
class Tacka:
    '''Predstavlja tacku u 2D prostoru.'''
    def resetovanje(self):
        self.x = 0
        self.y = 0
```

```
tacka_1 = Tacka()
tacka_1.resetovanje()
print(tacka_1.x, tacka_1.y)
```

Задатак 036: Креирати класу Tacka и њену методу pomeranje. Ова метода има два аргумента, x и y, којима се објекат tacka_1 поставља на координате 5, 3 у 2Д простору. Приказати позицију објекта.

```
class Tacka:
    def pomeranje(self, x, y):
        self.x = x
        self.y = y
```

```
tacka_1 = Tacka()
tacka_1.pomeranje(5, 3)
print(tacka_1.x, tacka_1.y)
```

Задатак 037: Креирати класу Tacka и њене методе pomeranje и resetovanje. Метода pomeranje има два аргумента, x и y, који као атрибути указују сами на себе. Метода resetovanje нема аргументе, већ позива преко саме себе методу pomeranje са вредностима 0, 0 као аргументима. Објекти tacka_1 и tacka_2 су инстанце класе Tacka при чему tacka_1 има координате 0,0 а tacka_2 има координате 5, 3.

```
class Tacka:
    def pomeranje(self, x, y):
        self.x = x
        self.y = y

    def resetovanje(self):
        self.pomeranje(0, 0)
```

```
tacka_1 = Tacka()
tacka_2 = Tacka()
tacka_1.resetovanje()
tacka_2.pomeranje(5, 3)
print(tacka_1.x, tacka_1.y)      #0 0
print(tacka_2.x, tacka_2.y)     #5 3
```

Задаци за самосталан рад:

- 31.** Креирати класу `LeteciInsekt`, инстанцирати класу `LeteciInsekt` објектима `muva` и `komaras`. Доделити атрибут `leti` објектима са Буловим вредностима.
- 32.** Креирати класу `LeteciInsekt` методом `kretanje()`, без аргумената. Доделити атрибут `leti` методи `kretanje()` и поставити одговарајућу Булову вредност. Инстанцирати класу `Insekt` објектима `muva` и `komaras`. Приказати вредност атрибута објекта.
- 33.** Креирати класу `Taska` и њену методу `pomeranje` са два аргумента. Објекат `taska_1` се поставља на координте 5, 3 а објекат `taska_2` се поставља на координте 6, 5, у 2Д простору. Приказати позиције оба објекта.
- 34.** Креирати класу `Taska` и њене методе `pomeranje` и `resetovanje`. Метода `pomeranje` има два аргумента, `x` и `y`, који као атрибути указују сами на себе. Метода `resetovanje` нема аргументе, већ позива преко саме себе методу `pomeranje` са вредностима 0, 0 као аргументима. Објекти `taska_1` и `taska_2` су инстанце класе `Taska` и оба су на почетку постављени у координатни почетак. Приказати тренутну позицију обе тачке ако су се помериле за 5 позиција по `x` оси и 4 позиције по `y` оси али у супротним смеровима једна од друге.