



برنامه سازی پیشرفته (برنامه نویسی شیء‌گرا: برخی نکات)

صادق اسکندری - دانشکده علوم ریاضی، گروه علوم کامپیوتر

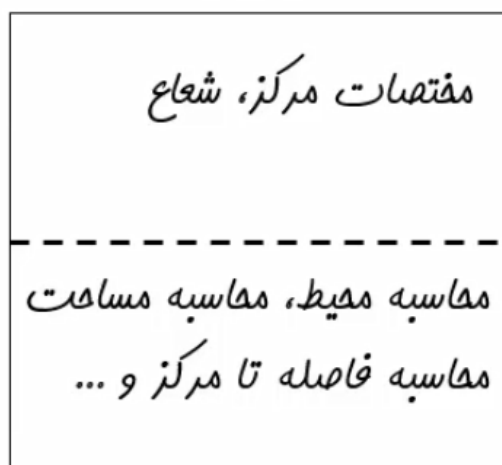
eskandari@guilan.ac.ir



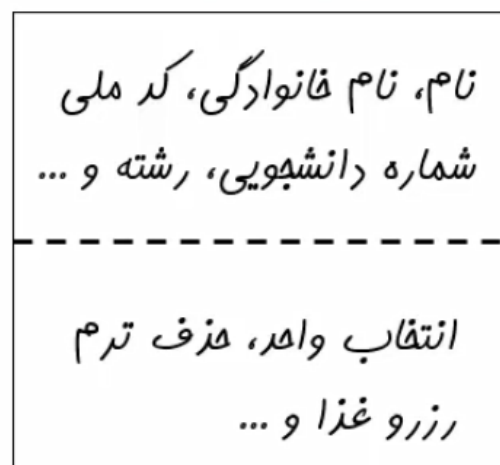


برای تعریف یک نوع داده جدید، از مفهوم کلاس (class) استفاده می شود.

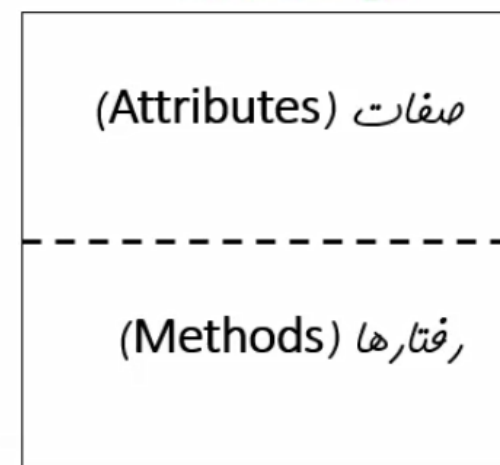
مثال: دایره



مثال: دانشجو



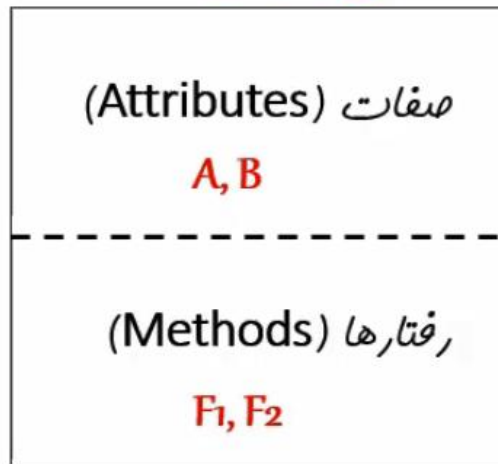
نوع داده جدید: X





کد نویسی یک کلاس در پایتون

نوع داده جدید: X



```
class X:
    def __init__(self, a, b):
        self.A = a
        self.B = b

    def F1(self, params):
        #F1 Body

    def F2(self, params):
        #F2 Body
```

سازنده (Constructor): اشیاء از این نوع داده را مقدار دهی اولیه می کند. یعنی یک متغیر (شیء) از نوع X را با صفات A=a و B=b ایجاد می کند.

یادآوری



به متغیرهایی که از نوع داده جدید (داده کلاسی) ایجاد می‌کنیم، یک نمونه یا یک شیء از آن نوع داده می‌گوییم.

```
class X:
    def __init__(self, a, b):
        self.A = a
        self.B = b

    def F1(self, params):
        #F1 Body

    def F2(self, params):
        #F2 Body
```

```
Obj1 = X(10,"aabb")
Obj2 = X(a = 44, b = "jfdk")
```

ایجاد اشیاء مختلف از نوع داده X

```
Obj1.A = 18
Obj2.B = 32

Obj1.F1(...)
```

با استفاده از نقطه، می‌توان به صفات و متدهای یک شیء دسترسی داشت

👹 **توجه مهم:** یک کلاس، یک الگو برای یک نوع داده است ولی یک شیء یک نمونه واقعی از آن نوع داده است.

آوری



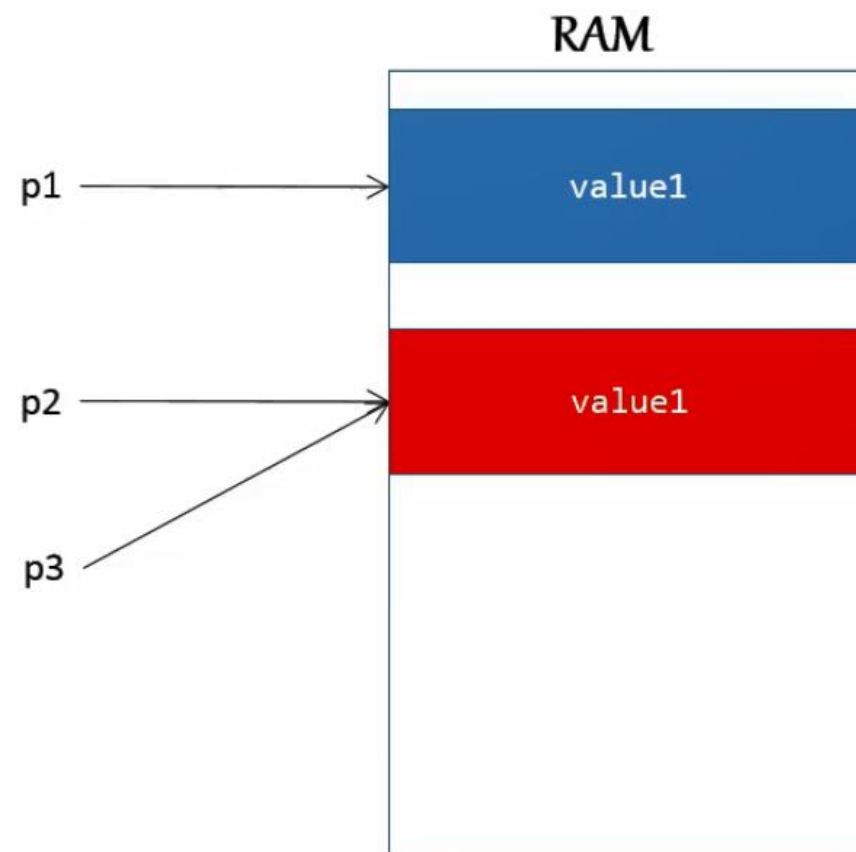
در زبانهای برنامه نویسی نیز مفهوم مقایسه سطحی و عمیق به صورت زیر قابل بیان است:

مقایسه سطحی دو شیء عبارت است از مقایسه مقادیر آنها

بنابراین مقایسه سطحی **p1** با **p2** دارای پاسخ **True** خواهد بود. زیرا، هر دو دارای مقادیر یکسان **value1** هستند.

مقایسه عمیق دو شیء عبارت است از مقایسه آدرس های آنها

بنابراین مقایسه عمیق **p1** با **p2** دارای پاسخ **False** خواهد بود. زیرا، اشیاء آنها یکی نیستند. ولی مقایسه عمیق **p2** با **p3** دارای پاسخ **True** خواهد بود زیرا هر دو به یک پیز اشاره می کنند.



بهر آوری

در پایتون، عملگرهای اصلی جهت مقایسه اشیاء عبارتند از **عملگر رابطه ای ==** و **کلمه کلیدی is**

نوع داده	تأثیر عملگر ==	تأثیر عملگر is
اولیه	سطحی <pre>l1 = [1,2,3] l2 = [1,2,3] l1 == l2 #True</pre>	عمیق <pre>l1 = [1,2,3] l2 = [1,2,3] l1 is l2 #False</pre>
کلاسی	عمیق <pre>r1 = Rectangle(Point(10,10), 10,20) r2 = Rectangle(Point(10,10), 10,20) print(r1 == r2) #False</pre>	عمیق <pre>r1 = Rectangle(Point(10,10), 10,20) r2 = Rectangle(Point(10,10), 10,20) print(r1 is r2) #False</pre>

سوال: اگر بفواهیم دو شیء کلاسی را به صورت سطحی مقایسه کنیم، چه باید کرد؟ 🤖

جواب: چنین مقایسه ای باید به شکل یک متد درون تعریف کلاس گنجانده شود. 😊

کلاس: تبدیل یک شیء به رشته

فرض کنید می‌خواهید مفتصات اشیاء ایجاد شده از نوع Point را چاپ کنید:

```
import math
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y
    ...
```

```
p1 = Point(12.4, 13.5)
p2 = Point(93,12.4)
```

```
print(p1)
```

```
print('({:.2f},{:.2f})'.format(p1.x, p1.y))
print('({:.2f},{:.2f})'.format(p2.x, p2.y))
```

```
<__main__.Point object at 0x00000000055FB348>
<__main__.Point object at 0x00000000052C5B08>
```

```
(12.40,13.50)
(93.00,12.40)
```

راهکار اول: استفاده از تابع print پایتون

در اجرای مفتلف، اشیاء در مکان‌های مفتلفی از حافظه ایجاد می‌شوند.

این راهکار برای کلاسهای پیچیده‌ای که دارای ویژگی‌های فراوانی است، می‌تواند فست‌کننده باشد. 🤖

کلاس: تبدیل یک شیء به رشته

فرض کنید می‌خواهید مفتمات اشیاء ایجاد شده از نوع Point را چاپ کنید:

```
import math
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y
    ...
    def __str__(self):
        return '%.2f,%.2f'%(self.x, self.y)

p1 = Point(12.4, 13.5)
p2 = Point(93,12.4)
print(p1)
print(p2)
```

راهکار دوام: معرفی شیء به وسیله فودش
با استفاده از متد `__str__`

(12.40,13.50)

(93.00,12.40)

کلاس: تبدیل یک شیء به رشته

متد `__str__` برای کلاس `Rectangle`

```
def __str__(self):  
    return 'Rectangle ==> center: %s, width: %d , height:%d'%(self.center, self.width, self.height)
```



```
rec1 = Rectangle(Point(10,10), w=10, h=20)  
rec2 = Rectangle()  
print(rec1)  
print(rec2)
```

```
Rectangle ==> center: (10.00,10.00), width: 10 , height: 20  
Rectangle ==> center: (0.00,0.00), width: 1 , height: 1
```

کلاس: ایجاد کپی از یک شیء

سوال: فروبی کد زیر چیست؟

```
1 rec1 = Rectangle(Point(10,10), w=10, h=20)
2 rec2 = rec1
3
4 print(rec1)
5 print(rec2)
6
7 rec1.width = 32
8 rec2.height = 5
9
10 print(rec1)
11 print(rec2)
```

Rectangle ==> center: (10.00,10.00), width: 10 , height: 20
Rectangle ==> center: (10.00,10.00), width: 10 , height: 20

Rectangle ==> center: (10.00,10.00), width: 32 , height: 5
Rectangle ==> center: (10.00,10.00), width: 32 , height: 5

عملگر تفصیص، به جای ایجاد یک کپی از شیء، یک اشاره گر جدید به همان شیء ایجاد می کند. ☹️☹️

کلاس: ایجاد کپی از یک شیء

سوال: بنابراین، چگونه می توان یک کپی از یک شیء ایجاد کرد؟

پاسخ: با استفاده از متد `deepcopy()` در پکیج `copy`

```
1 import copy
2 rec1 = Rectangle(Point(10,10), w=10, h=20)
3 rec2 = copy.deepcopy(rec1)
4
5 print(rec1)
6 print(rec2)
7
8 rec1.width = 32
9 rec2.height = 5
10
11 print(rec1)
12 print(rec2)
```

```
Rectangle ==> center: (10.00,10.00), width: 10 , height: 20
Rectangle ==> center: (10.00,10.00), width: 10 , height: 20
```

```
Rectangle ==> center: (10.00,10.00), width: 32 , height: 20
Rectangle ==> center: (10.00,10.00), width: 10 , height: 5
```

کلاس: یک نکته کوچک

پایتون یک زبان کاملاً شیء گرا است. بنابراین، تمامی متغیرهایی که تعریف می کنیم، شیء هستند، حتی متغیرهای از نوع داده های اولیه. بنابراین، این متغیرها نیز دارای برفی ویژگیها و رفتارها (متد) هستند.

```
a = 10874
print(a.bit_length())
print(a.real)
print(a.imag)
```

```
14
10874
0
```

```
s = 'hello'
print(s.upper())
print(s.count('l'))
```

```
HELLO
2
```

```
lst = [6,2,3,8,12,5]
lst.append(2)
print(lst)
lst.sort()
print(lst)
```

```
[6, 2, 3, 8, 12, 5, 2]
[2, 2, 3, 5, 6, 8, 12]
```

```
def fun(x):
    return x**2
```

```
a=fun.__call__(54)
print(a)
print(fun.__name__)
```

```
2916
fun
```



تمرین: متدها و ویژگی های دیگر انواع داده لیست، رشته و دیکشنری را بررسی نمایید.

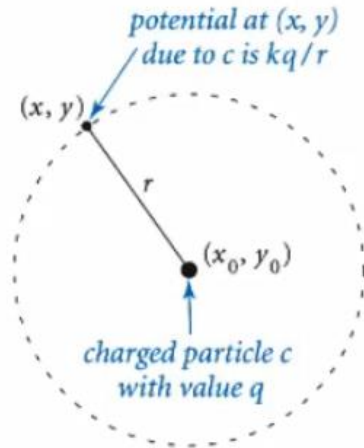
تمرین

بر اساس قانون کلمبوس، پتانسیل حاصل از یک ذره باردار در نقطه (x,y) برابر $V = \frac{kq}{r}$ است که در آن q اندازه بار الکتریکی ذره، r فاصله نقطه (x,y) از ذره و $k = 8.99 \times 10^9$ یک مقدار ثابت است.



بنابراین، یک بار الکتریکی دارای ویژگی‌ها و رفتارهای زیر است:

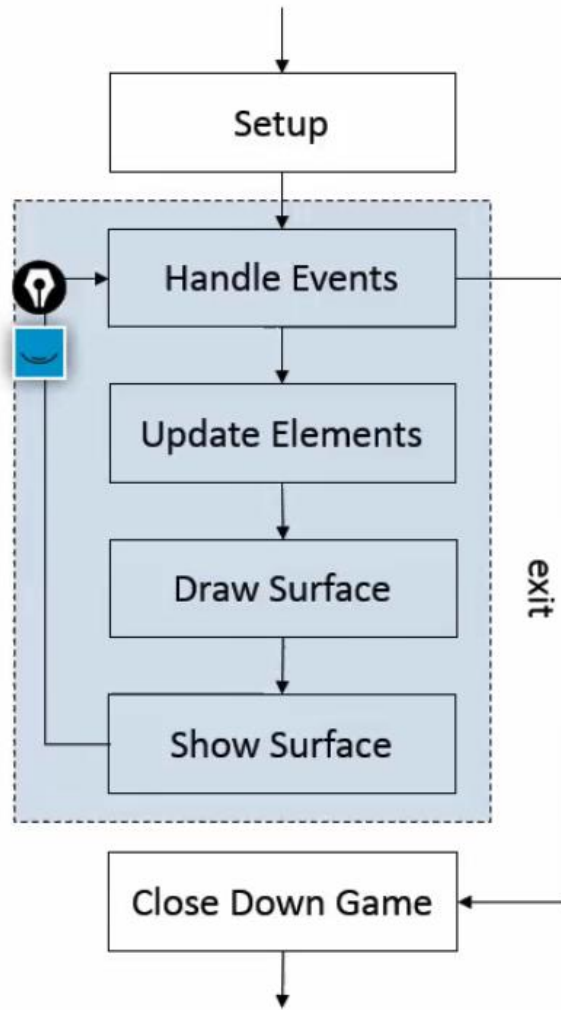
نوع داده: Charge



مفتصات مرکز بار الکتریکی (x_0, y_0)
اندازه بار الکتریکی (q_0)

Potential_at(p): پتانسیل حاصل از این بار، در نقطه p
__str__(): معرفی نوع داده

کلاس Charge را بنویسید.



pygame

پرفه بازی در pygame