Python programming language

Week 10. Python Classes and Objects

Alisher Ismailov
Lecturer

Department of Information technology
Andijan Branch of Tashkent Financial Institute
Email: alisherismailov534@gmail.com

Python dasturlash tili

10-Mavzu. Python Classes and Objects

Ma’ruzachi: Alisher Ismoilov

Axborot texnologiyalari kafedrasi

Toshkent moliya instituti Andijon filiali

Elektron pochta: alisherismailov534@gmail.com

10-Mavzu. Python Classes and Objects

Reja:

- Class yaratish
- Obyekt yaratish
- __init__() funksiyasi
- Obyekt funksiyalari
- Obyekt xususiyatlarini o‘zgartirish
- Obyekt xususiyatlarini o‘chirish
- Obyektlarni o‘chirish
- Parent class (ota class)ini yaratish
- Child (bola) classini yaratish
- __init__() funksiyasini Child classiga qo‘shish
- super() funksiyasidan foydalanish
- Child (bola) classiga o‘zgaruvchilar qo‘shish
- Child classiga funksiyalarni qo‘shish

Python dasturlash tilida Class va Ob’ektlar

Python dasturlash tili obyektga yo‘naltirilgan dasturlash tili hisoblanadi. Obyektga yo‘naltirilgan dasturlash tilining asosiy yutug‘i bu yozilgan dasturlash kodini qayta ishlatish mumkin. Buning uchun yoziladigan dasturlash kodi odatda classlarga yoziladi. Python dasturlash tilida Class (sinf) bu o‘zida ma’lum funksiyalarni jamlagan dasturlash kodiga aytiladi. Class o‘z nomiga ega bo‘ladi. Yaratilgan classdan boshqa classda foydalanish uchun dasturchi yaratilgan class uchun obyekt yaratishi kerak. Obyekt bu o‘zgaruvchiga o‘xshash narsa, farqi obyekt class ichidagi ma’lumotlarga (o‘zgaruvchi, funksiyalar) kirishga huquqi bo‘ladi. Demak yaratilgan Class ichidagi ma’lumotlarni chaqirish uchun biz obyektdan foydalanamiz. Class va Obyekt yaratish va ulardan qanday foydalanish amaliy mashg‘ulotlar ushbu mavzuda keltirilgan.

Python dasturlash tilida Class va Ob’ektlar

- **Class yaratish**

```
class Matn:
    matn = “dasturlash”
```

Python dasturlash tilida Class va Ob’ektlar

- **Obyekt yaratish**
- Class obyekti – bu maxsus o‘zgaruvchi hisoblanadi. Class obyekti belgilangan **class** ichidagi barcha ma’lumotlarga kirishga huquqi bor o‘zgaruvchi hisoblanadi. Masalan, quyidagi suratda **obyekt_Dasturlash** nomli obyekt yaratdik va bu **obyekt_Dasturlash** obyekti **Dasturlash** classi ichidagi barcha o‘zgaruvchi va funksiyalarga kirishi mumkin. Biz Mening_class ichidagi sonni ekranga chop etdik.

Python dasturlash tilida Class va Ob’ektlar

- **Obyekt yaratish**

```
class Dasturlash:
        dasturlashtili = “Python”

obyekt_Dasturlash = Dasturlash()
print(obyekt_Dasturlash.dasturlashtili)
```

Python dasturlash tilida Class va Ob’ektlar

- **__init__() funksiyasi**
- Class (Sinf)larning ma’nosini tushunish uchun biz python dasturlash tilidagi **__init__()** funksiyasini tushunishimiz kerak. Barcha class (sinf)larda **__init__()** funksiyasi mavjud bo‘lib, u har doim class (sinf) ishga tushirilganda bajariladi. Obyekt xususiyatlariga qiymatlarni belgilash yoki obyekt yaratilayotganda bajarilishi kerak bo‘lgan boshqa operatsiyalar uchun **__init__()** funksiyasidan foydalanishingiz mumkin:

Python dasturlash tilida Class va Ob’ektlar

- **__init__() funksiyasi**

```
class Inson:
        def __init__(shaxs, ism, yosh):
            shaxs.ism = ism
            shaxs.yosh = yosh
p1 = Inson(“Alisher”, 30)

print(p1.ism)
print(p1.yosh)
```

Python dasturlash tilida Class va Ob’ektlar

- **Obyekt funksiyalari**
- Class ichidagi funksiyalarni obyektlar orqali chaqirishimiz va foydalanishimiz mumkin.
- Inson class (sinf)ida funksiya yarataylik va p1 obyekti orqali chaqirib ishlatdik:

Python dasturlash tilida Class va Ob’ektlar

- **Obyekt funksiyalari**

```
class Inson:
        def __init__(shaxs, ism, yosh):
            shaxs.ism = ism
            shaxs.yosh = yosh

        def meningfunksiyam(shaxs):
             print(“Mening ismim ”+shaxs.ism)

p1 = Inson(“Alisher”, 30)

p1.meningfunksiyam()
```

Python dasturlash tilida Class va Ob’ektlar

- **Obyekt orqali o‘zgaruvchi qiymatini o‘zgartirish**
- Siz quyidagi kabi obyekt orqali class ichidagi o‘zgaruvchilarni qiymatini o‘zgartishingiz mumkin. Quyidagi suratda obyektdan foydalanib biz yosh o‘zgaruvchisini qiymatini o‘zgartidik. (30 dan 20 ga)

Python dasturlash tilida Class va Ob’ektlar

- **Obyekt orqali o‘zgaruvchi qiymatini o‘zgartirish**
- class Inson:
- 	def __init__(shaxs, ism, yosh):
- 		shaxs.ism = ism
- 		shaxs.yosh = yosh
-
- 	def meningfunksiyam(shaxs):
- 		 print(“Mening ismim ”+shaxs.ism)
-
- p1 = Inson(“Alisher”, 30)
- p1.yosh = 20
-
- print(p1.yosh)

Python dasturlash tilida Class va Ob’ektlar

- **Obyekt xususiyatlarini o‘chirish**
- *Xususiyat – class ichidagi o‘zgaruvchi va funksiyalar hisoblanadi.*
- **del** kalit so‘zidan foydalanib, obyektlardagi xususiyatlarni o‘chirishingiz mumkin:

```
class Inson:
  def __init__(shaxs, ism, yosh):
    shaxs.ism = ism
    shaxs.yosh = yosh

  def meningfunksiyam(shaxs):
    print("Meing ismim " + shaxs.yosh)

p1 = Inson("Alisher", 30)
del p1.yosh

print(p1.yosh)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_4380/3855746675.py in <module>
     10 del p1.yosh
     11
---> 12 print(p1.yosh)

AttributeError: 'Inson' object has no attribute 'yosh'
```

Python dasturlash tilida Class va Ob’ektlar

- **Obyektlarni o‘cherish**
- Obyektlarni **del** kalit so‘zidan foydalanib o‘chirishingiz mumkin:

```
In [35]: class Inson:
           def __init__(shaxs, ism, yosh):
             shaxs.ism = ism
             shaxs.yosh = yosh

           def meningfunksiyam(shaxs):
             print("Meing ismim " + shaxs.yosh)

         p1 = Inson("Alisher", 30)
         del p1

         print(p1.yosh)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_4380/1044287728.py in <module>
      10 del p1
      11
---> 12 print(p1.yosh)

NameError: name 'p1' is not defined
```

Python dasturlash tilida Class va Ob’ektlar

- **pass kalit so‘zi**
- Class (sinf) bo‘sh bo‘lishi mumkin emas, lekin agar sizda biron sababga ko‘ra kontentsiz class (sinf) mavjud bo‘lsa, xatolikka yo‘l qo‘ymaslik uchun **pass** kalit so‘zini kiriting.
- class Inson:
- pass

Python dasturlash tilida Class va Ob’ektlar

Inheritance boshqa classdan barcha funksiyalar va paramaterlarni chaqirib oladigan class yaratish imkonini beradi. Inheritance 1969-yilda Simula dasturlash tili uchun ixtiro qilingan.

Inherentance konsepsiyasini ishlatishda ikkita class mavjud.

Parent(ota) class - bu asosiy class hisoblanadi, asosiy funksiya va parameterlar ushbu classda bo‘ladi.

Child (bola) class - bu asosiy classdan funksiya va parameterlarni inherent qiladigan class hisoblanadi.

Python dasturlash tilida Class va Ob’ektlar

- **Parent class (ota class)ini yaratish**

```
class Inson:
    def __init__(shaxs, ism, familya):
        shaxs.ism = ism
        shaxs.familya = familya
     def ChopEtish(shaxs):
        print(shaxs.ism, shaxs.familya)
x = Inson(“Ulugbek”, “Dehqonov”)
x.ChopEtish()
```

Python dasturlash tilida Class va Ob’ektlar

- **Child (bola) classini yaratish**

- Funksionallikni boshqa classdan **inherent** qilib oladigan class yaratish uchun parent (ota) classni child classini yaratishda uning parametri sifatida joylashtiriladi, quyidagidek.
- **class Ustoz(Inson)**
-
- Inson classidan o‘zgaruvchilar va funksiyalarni inherent qilib oladigan Ustoz nomli class yaratamiz:
-
- Yuqorida Inson nomli parent (ota) classini yaratdik.

Python dasturlash tilida Class va Ob’ektlar

- **Child (bola) classini yaratish**

```
class Inson:
    def __init__(shaxs, ism, familya):
        shaxs.ism = ism
        shaxs.familya = familya

    def ChopEtish(shaxs):
        print(shaxs.ism, shaxs.familya)
```

Python dasturlash tilida Class va Ob’ektlar

- Child classi yaratish quyidagidek yozilishi mumkin. Ustoz nomli child classini yaratamiz:

```
class Ustoz(Inson):
    pass
```

Python dasturlash tilida Class va Ob’ektlar

- Endi Ustoz child (bola) classi Inson parent (ota) classidagi barcha o‘zgaruvchi va funksiyalarni qabul qildi. Ustoz child classidan foydalanib ChopEtish() funskiyasini ishlatib ko‘rishimiz mumkin.
- Quyidagi misolda obyekt yaratish uchun **Ustoz** classidan foydalanamiz va keyin **ChopEtish()** funksiyasini chaqirib ishlatamiz:

Python dasturlash tilida Class va Ob’ektlar

```
class Inson:
    def __init__(shaxs, ism, familya):
        shaxs.ism = ism
        shaxs.familya = familya

    def ChopEtish(shaxs):
        print(shaxs.ism, shaxs.familya)

class Ustoz(Inson):
    pass

x = Ustoz("Kamoldin", "Jalilov")
x.ChopEtish()
```

Python dasturlash tilida Class va Ob’ektlar

- **__init__() funksiyasini Child classiga qo‘shish**

Yuqoridagi misollarda biz Parent classidan o‘zgaruvchilar va funksiyalarni **inherent** qilib oladigan Child classini yaratish jarayonini ko‘rdik. Ushbu bo‘limda **__init__** funksiyasini child classiga qo‘shish jarayonini ko‘rib chiqamiz. **__init__** funksiyasi child (bola) classida dasturchiga yangi parametr va o‘zgaruvchi yaratish imkonini beradi.

Ustoz classiga **__init__()** funksiyasini qo‘shamiz:

- class Ustoz(Inson):
- def __init__(shaxs, ism, familya):

__init__() funksiyasini child classiga qo‘shsangiz, child classi parent classning **__init__()** funksiyasini inherent qilmaydi.

Python dasturlash tilida Class va Ob’ektlar

- Ustoz classiga yangi paramatr qo‘shib sinab ko‘ramiz. tug‘ilgan yil nomli yangi parameter qo‘shamiz:

```
class Ustoz(Inson):
        def __init__(ustoz, ism, familya, tug_yil):
                ustoz.ism = ism
                ustoz.familya = familya
                ustoz.tug_yil = tug_yil
        def ChopEtish(ustoz):
                print(ustoz.ism, ustoz.familya, ustoz.tug_yil)
u = Ustoz(“Zafar”, “Jo`rayev”, “1962”)
u.ChopEtish()
```

Python dasturlash tilida Class va Ob’ektlar

- Agarda parent classining **__init__()** funksiyasining inherentini saqlab qolish kerak bo‘lsa, child classi __init__ funksiyasidan keyin, parent classining **__init__()** funksiyasini chaqirish kerak:

```
class Ustoz(Inson):
            def __init__(shaxs, ism, familya):
   Inson.__init__(shaxs, ism, familya)
        shaxs.ism = ism
        shaxs.familya = familya

   def ChopEtish(shaxs):
      print(shaxs.ism, shaxs.familya)

class Ustoz(Inson):
   pass

x = Ustoz("Nilufar", "Abdurakhmonova")
x.ChopEtish()
```

Python dasturlash tilida Class va Ob’ektlar

- **super() funksiyasidan foydalanish**

- Python dasturlash tili yana, ***child*** classini parent classidan barcha funksiyalar va o‘zgaruvchilarni inherent qilib oladigan **super()** funksiyasiga ega. Buning uchun child classidagi ***__init__*** funksiyasidan so‘ng quyidagi kodni kiritish kerak:
- **super().__init__(ism, familya)**

Python dasturlash tilida Class va Ob’ektlar

```
class Ustoz(Inson):
    def __init__(shaxs, ism, familya):
        super().__init__(ism, familya)
        shaxs.ism = ism
        shaxs.familya = familya

    def ChopEtish(shaxs):
        print(shaxs.ism, shaxs.familya)

x = Ustoz("Muftoh", "Hakimov")
x.ChopEtish()
```

Python dasturlash tilida Class va Ob’ektlar

- **super()** funksiyasidan foydalanganda parent class element nomini (shaxs) yozishingiz shart emas, super() funksiyasi avtomatik tarzda parent classidan funksiyalar va o‘zgaruvchilarni inherent qilib oladi.

Python dasturlash tilida Class va Ob’ektlar

- **Child classiga funksiyalar qo‘shish**

- Ustoz (child) classiga **tanishtiruv()** deb nomlangan funksiyani qo‘shish jarayoni quyidagidek bo‘ladi:

```
class Ustoz(Inson):
    def __init__(shaxs, ism, familya, yil):
        super().__init__(ism, familya)
        shaxs.ishBoshlaganYil = yil

    def tanishtiruv(shaxs):
            print(shaxs.ism, shaxs.familya, shaxs.ishBoshlaganYil,
             “yildan buyon o`qituvchi lavozimida ishlab kelmoqda”)
```

Python dasturlash tilida Class va Ob’ektlar

- Agar siz Parent classidagi funksiya bilan bir xil nomdagi funksiyani Child classiga qo‘shsangiz, Parent classi funksiyasining inherenti bekor qilinadi. Masalan, Inson (parent) classida ChopEtish() nomli funksiya mavjud, Ustoz (child) classiga ham ChopEtish() nomli funksiya yaratamiz va bu funskiya orqali faqatgina ism o‘zgaruvchisini ekranga chop etamiz.

Python dasturlash tilida Class va Ob’ektlar

```
class Ustoz(Inson):
    def __init__(shaxs, ism, familya):
        super().__init__(ism, familya)
        shaxs.ism = ism
        shaxs.familya = familya

    def ChopEtish(shaxs):
        print(shaxs.ism)

x = Ustoz("Yoqubjon", "Qurbonov")
x.ChopEtish()
```

Foydalanilgan adabiyotlar

1. Mastering Object-Oriented Python: Build powerful applications with reusable code using OOP design patterns and Python 3.7, 2nd Edition, Steven F. Lott, Packt Publishing (June 14, 2019)

2. Learning Python, 5th Edition Fifth Edition, Mark Lutz , O'Reilly Media, June 12, 2013

3. Python Programming for Beginners: The Ultimate Guide for Beginners to Learn Python Programming: Crash Course on Python Programming for Beginners, AMZ Publishing, Independently published (July 13, 2021)

4. https://www.python.org/

5. https://www.w3schools.com/

6. https://www.codecademy.com/catalog/language/python

7. https://realpython.com/

8. https://www.anaconda.com/

E’tiboringiz uchun rahmat!