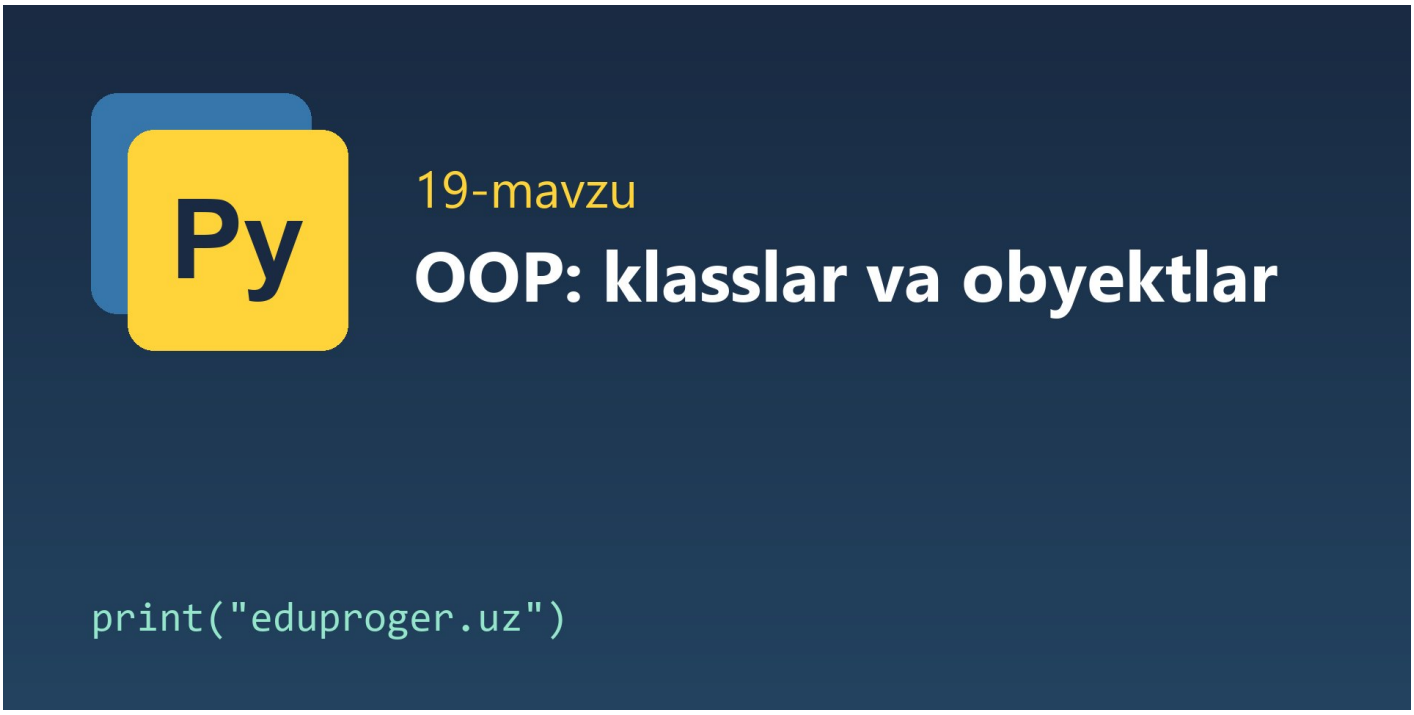
19. OOP: klasslar va obyektlar



Katta dasturlarda ma'lumot va u bilan ishlaydigan funksiyalarni birga saqlash qulay. Masalan, «talaba»da ism va ballar (ma'lumot) hamda «o'rtacha ball» (amal) bor. Bunday birlik **obyekt** deyiladi, uning «chizmasi» esa — **klass**. Bu yondashuv **OOP** (obyektga yo'naltirilgan dasturlash) deyiladi.

1. Klass va obyekt

Klass — qolip (chizma), obyekt — shu qolipdan yasalgan aniq nusxa. Pythonda biz allaqachon obyektlar bilan ishlaganmiz: `"salom"` — str klassining obyekti, `[1, 2]` — list klassining obyekti.

```
print(type("salom"), type([1, 2]))
```

```
class Mushuk:
    pass

m1 = Mushuk()
m2 = Mushuk()
print(type(m1).__name__, m1 is m2)
```

Natija:

```
<class 'str'> <class 'list'>
Mushuk False
```

2. __init__ va self

`__init__` — obyekt yaratilganda avtomatik chaqiriladigan maxsus metod (konstruktor). Birinchi parametr `self` — yaratilayotgan obyektning o'zi. `self.ism = ism` obyektga **atribut** (xususiyat) qo'shadi:

```
class Talaba:
    def __init__(self, ism, sinf):
        self.ism = ism
        self.sinf = sinf
        self.ballar = []

t = Talaba("Malika", 9)
print(t.ism, t.sinf, t.ballar)
t.sinf = 10
print(t.sinf)
```

Natija:

```
Malika 9 []
10
```

3. Metodlar

Klass ichidagi funksiya — **metod**. U ham birinchi parametr sifatida `self` oladi va obyekt atributlari bilan ishlaydi:

```
class Talaba:
    def __init__(self, ism):
        self.ism = ism
        self.ballar = []

    def ball_qosh(self, ball):
        self.ballar.append(ball)

    def ortacha(self):
        return sum(self.ballar) / len(self.ballar) if self.ballar else 0

t = Talaba("Sardor")
t.ball_qosh(85)
t.ball_qosh(92)
print(t.ism, t.ballar, t.ortacha())
```

Natija:

```
Sardor [85, 92] 88.5
```

4. __str__ — obyektni chiroyli chiqarish

```
class Nuqta:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def __str__(self):
        return f"({self.x}, {self.y})"
```

```
    def masofa(self, boshqa):
        return ((self.x - boshqa.x) ** 2 + (self.y - boshqa.y) ** 2) ** 0.5

a = Nuqta(0, 0)
b = Nuqta(3, 4)
print(a, b)
print(a.masofa(b))
```

Natija:

```
(0, 0) (3, 4)
5.0
```

5. Klass atributi

`self.` siz, klass ichida yozilgan o'zgaruvchi BARCHA obyektlar uchun umumiy:

```
class Talaba:
    maktab = "eduproger"
    soni = 0

    def __init__(self, ism):
        self.ism = ism
        Talaba.soni += 1

a = Talaba("Ali")
b = Talaba("Vali")
print(a.maktab, b.maktab, Talaba.soni)
```

Natija:

```
eduproger eduproger 2
```

6. Misol: bank hisobi

Obyekt o'z holatini o'zi himoya qiladi: balansni to'g'ridan-to'g'ri o'zgartirish o'rniga metodlar orqali — tekshiruv bilan:

```
class Hisob:
    def __init__(self, egasi, balans=0):
        self.egasi = egasi
        self.balans = balans

    def kirim(self, summa):
        self.balans += summa

    def chiqim(self, summa):
        if summa > self.balans:
            print("Mablag' yetarli emas")
            return False
        self.balans -= summa
        return True

    def __str__(self):
```

```
        return f"{self.egasi}: {self.balans} so'm"

h = Hisob("Aziz", 100000)
h.kirim(50000)
h.chiqim(200000)
h.chiqim(30000)
print(h)
```

Natija:

```
Mablag' yetarli emas
Aziz: 120000 so'm
```

7. Obyektlar ro'yxati

```
class Mahsulot:
    def __init__(self, nom, narx, soni):
        self.nom, self.narx, self.soni = nom, narx, soni

    def jami(self):
        return self.narx * self.soni

savat = [Mahsulot("non", 4000, 3), Mahsulot("sut", 12000, 1), Mahsulot("choy",
for m in savat:
    print(f"{m.nom:6} {m.jami():>7}")
print("Jami:", sum(m.jami() for m in savat))
qimmat = max(savat, key=lambda m: m.narx)
print("Eng qimmat:", qimmat.nom)
```

Natija:

```
non      12000
sut      12000
choy     50000
Jami: 74000
Eng qimmat: choy
```

8. OOP ning asosiy g'oyalari

G'oya	Ma'nosi
Inkapsulyatsiya	Ma'lumot va u bilan ishlaydigan metodlar bitta obyektda; tashqaridan metodlar orqali ishlanadi
Meros	Yangi klass mavjud klassning imkoniyatlarini oladi va kengaytiradi (keyingi mavzu)
Polimorfizm	Turli klasslarning bir xil nomli metodi o'zicha ishlaydi (keyingi mavzu)
Abstraksiya	Murakkablik yashiriladi: foydalanuvchi faqat kerakli metodlarni ko'radi

9. Xulosa

- `class Nom:` — qolip; `Nom(...)` — obyekt yaratish.
- `__init__(self, ...)` — boshlang'ich atributlar; `self.atribut`.
- Metod — klass ichidagi funksiya, birinchi parametri `self`.
- `__str__` — print() da chiroyli ko'rinish; klass atributi — hamma obyekt uchun umumiy.