

# Python'da Klass Inkapsulyatsiyasi

---

Inkapsulyatsiya — bu obyektga oid dasturlashda muhim tushuncha bo'lib, ma'lumotlar va funksiyalarni bir joyga jamlash va ularni tashqi aralashuvdan himoya qilishni anglatadi. Python'da inkapsulyatsiya orqali sinf ichidagi atributlar va metodlarni tashqi koddan yashirish mumkin. Bu dastur xavfsizligi va barqarorligini oshiradi.

## Oddiy inkapsulyatsiya (private atribut)

Kod:

```
class Oquvchi:
    def __init__(self, ism, baho):
        self.izm = ism
        self.__baho = baho # Private atribut

    def baho_korish(self):
        return self.__baho

ali = Oquvchi("Ali", 5)
print(ali.baho_korish())
```

---

## Private atributga to'g'ridan-to'g'ri murojaat qilish (xatolik)

Kod:

```
ali = Oquvchi("Ali", 5)
print(ali.__baho) # AttributeError: 'Oquvchi' object has no
attribute '__baho'
```

---

## Getter va Setter metodlari bilan foydalanish

Kod:

```
class Oquvchi:
    def __init__(self, ism):
        self.izm = ism
        self.__baho = 0

    def baho_olish(self):
        return self.__baho
```

```
def baho_qoyish(self, yangi_baho):
    if 1 <= yangi_baho <= 5:
        self.__baho = yangi_baho

ali = Oquvchi("Ali")
ali.baho_qoyish(4)
print(ali.baho_olish())
```

---

### Property dekoratori bilan ishlash

Kod:

```
class Oquvchi:
    def __init__(self, ism):
        self.ism = ism
        self.__yosh = 0

    @property
    def yosh(self):
        return self.__yosh

    @yosh.setter
    def yosh(self, qiymat):
        if qiymat > 0:
            self.__yosh = qiymat

ali = Oquvchi("Ali")
ali.yosh = 15
print(ali.yosh)
```

---

### Private metoddan foydalanish (ichki hisob-kitob)

Kod:

```
class Oquvchi:
    def __init__(self, ballar):
        self.__ballar = ballar

    def umumiy_baho(self):
        return self.__hisobla()

    def __hisobla(self):
```

```
return sum(self.__ballar) // len(self.__ballar)
```

```
ali = Oquvchi([4, 5, 5, 3])  
print(ali.umumiy_baho())
```

---

### **Inkapsulyatsiya nima uchun kerak?**

Inkapsulyatsiya dastur kodini yaxshiroq tashkil qilishga, xatoliklarni kamaytirishga va noto'g'ri foydalanishlarning oldini olishga yordam beradi. Shuningdek, u maxfiy (private) atributlar va metodlar yordamida ma'lumotlarni himoya qiladi. Getter va setter metodlar esa bu ma'lumotlarga nazoratli kirishni ta'minlaydi.