

Python dasturlash tilida Inkapsulyatsiya

1. Inkapsulyatsiya nima?

Inkapsulyatsiya — bu obyektga yo'naltirilgan dasturlash (OOP) tamoyili bo'lib, obyekt ichidagi ma'lumotlar va funksiyalarni tashqi muhitdan yashirish orqali faqat maxsus metodlar orqali ularga murojaat qilishni ta'minlaydi. Python tilida inkapsulyatsiya `private`, `protected` va `public` atributlar orqali amalga oshiriladi.

2. Python'dagi access modifierlar

- `public`: hech qanday chiziqcha bilan yozilmagan. Hammaga ochiq.
- `_protected`: bitta pastki chiziq bilan boshlanadi. Faqat class va subclassdan kirish mumkin (shunchaki konvensiya).
- `__private`: ikki pastki chiziq bilan boshlanadi. Class tashqarisidan kira olmaymiz.

3. Misol: BankAccount classi

```
class BankAccount:
    def __init__(self, balance):
        self.__balance = balance

    def deposit(self, amount):
        if amount > 0:
            self.__balance += amount

    def withdraw(self, amount):
        if 0 < amount <= self.__balance:
            self.__balance -= amount

    def get_balance(self):
        return self.__balance
```

Bu yerda `__balance` private qilib qo'yilgan. Biz unga faqat metodlar orqali murojaat qilamiz.

4. Nega kerak?

- Ma'lumotlarni himoya qilish uchun

- Kodni strukturaviy va boshqariladigan qilish uchun
- Foydalanuvchiga faqat kerakli interfeysni berish

5. Inkapsulyatsiyaga oid 3 ta misol

Misol 1: Telefon

```
class Phone:
    def __init__(self):
        self.__battery_level = 100

    def use(self, minutes):
        self.__battery_level -= minutes * 0.5
        if self.__battery_level < 0:
            self.__battery_level = 0

    def get_battery(self):
        return self.__battery_level
```

Misol 2: Student

```
class Student:
    def __init__(self, name):
        self.name = name
        self.__grades = []

    def add_grade(self, grade):
        if 0 <= grade <= 100:
            self.__grades.append(grade)

    def average(self):
        return sum(self.__grades) / len(self.__grades)
```

Misol 3: Developer

```
class Developer:
    def __init__(self, name, salary):
        self.name = name
        self.__salary = salary

    def set_salary(self, salary):
```

```
if salary >= 0:  
    self.__salary = salary  
  
def get_salary(self):  
    return self.__salary
```