

# Створення об'єктів та класів

## Створення класу

В мові програмування Python класи створюються дуже легко: пишемо `class`, його назву та вписуємо його властивості, які будуть передаватися об'єктам.

```
class A():  
    властивості
```

Якщо клас є дочірним (тобто він сам буде належати якомусь класу), то батьківський клас вказується у скобках.

Щоб написати властивість для класу, потрібно створити поле.

*Поле класу (атрибут)* – це змінна, зв'язана з класом або об'єктом. Всі дані об'єкта зберігаються в його полях. Доступ до полів здійснюється по імені. Зазвичай тип даних кожного поля задається в описі класу, членом якого є поле.

```
class A():  
    m = 20  
    l = 10
```

При цьому наслідування буде передавати ці поля дочірнім класам та об'єктам створеного класу.

## Створення об'єкту

Створення об'єкту відбувається шляхом простого виклику класу. При цьому скобки писати обов'язково. Також об'єкту потрібно присвоїти змінну.

```
n = A()
```

Для окремого об'єкту ми можемо змінити або додати властивість, тобто переписати поле.

```
n = A()  
n.l = 30  
n.k = 50
```

Однак, у програмуванні майже не використовується додавання властивості для окремого об'єкту, так як у великому проекті різні властивості будуть вносити хаос в код.

## Створення функцій класу

Для додавання функції класу пишемо функцію у тілі класу. При написанні функції обов'язковим першим параметром є змінна, яка позначає об'єкт.

```
class A():
    m = 20
    l = 10
    def dodavannya(self):
        return self.m+3
```

За негласним правилом зазвичай це є self, але ця змінна може мати будь-яку назву. При цьому у дужках при виклику функції ми не вказуємо об'єкт. Це робиться тому, що функція належить класу. При читанні запису виклику функції інтерпретатор шукає функцію у властивостях об'єкту, не знаходить та шукає її у властивостях класу цього об'єкта. Коли він знаходить функцію, запис, незримо для нас, перетворюється на виклик функції класу з вказаним об'єктом в якості першого параметру. Всі інші вказані параметри зміщуються на наступну позицію.

### Атрибут \_\_dict\_\_

В мові програмування Python є атрибут об'єктів \_\_dict\_\_. При його виклику він повертає словник з властивостями об'єкту у ключах та їх значеннями у значеннях словника.

```
print(n.__dict__)

{'l': 30, 'k': 50}
```

Але при застосуванні атрибута до об'єкта у нього додаються тільки змінені або додані властивості, так як всі інші належать саме класу.

Атрибут \_\_dict\_\_ можна застосовувати до класів, але при цьому запис також повертає властивості, які записані за замовчуванням:

```
print(A.__dict__)

{'__module__': '__main__', 'm': 20, 'l': 10, 'dodavannya': <function A.dodavannya at 0x000001E7B337F3A0>, '__dict__': <attribute '__dict__' of 'A' objects>, '__weakref__': <attribute '__weakref__' of 'A' objects>, 'doc': None}
```

### **Контрольні питання:**

- 1) Як створити клас?
- 2) Як створити дочірній клас?
- 3) Як створити об'єкт, який відноситься до класу?
- 4) Що таке атрибут?
- 5) Навіщо існує параметр self у функціях класу?