

# Ознайомлення з ООП. Успадкування

**№ уроку:** 1 **Курс:** Python Базовий

**Засоби навчання:** Персональний комп'ютер/ноутбук стандартної продуктивності

## Огляд, мета та призначення уроку

Ознайомитись з поняттям об'єктно-орієнтованого програмування (ООП) та його особливостями у Python. Розглянути один із чотирьох базових принципів ООП - успадкування.

## Вивчивши матеріал даного заняття, учень зможе:

- Розуміти, що таке ООП і навіщо воно потрібне.
- Розуміти, що таке успадкування в ООП та як його застосувати на практиці.

## Зміст уроку

1. Про що курс
2. Що таке ООП
3. Що таке успадкування
4. Приклади використання ООП та успадкування на практиці

## Резюме

- ООП або об'єктно-орієнтоване програмування – це одна з багатьох парадигм у програмуванні та в мові Python вона також реалізована, і є дуже корисним та потужним інструментом у розробці програм. ООП будується на 4 основних принципах: успадкування, інкапсуляція, поліморфізм та абстракція.
- Ключовими поняттями в ООП є клас та об'єкт класу. Клас - загальне поняття (рослина, предмет меблів), а об'єкт чи об'єкт класу - це вже конкретне поняття (кульбаба (рослина), стілець (предмет меблів)). Ця концепція є основою ООП.
- ООП не є єдиним стилем програмування, можна писати програми без нього. Але володіння ним дає незаперечну перевагу в якості написання коду і несе в собі масу корисних властивостей.
- Спадкування є простою ідеєю. Ви можете створити ієрархію класів, де є материнські (від яких успадковуються) та дочірні класи (які успадковуються від материнських).
- Ключова ідея успадкування така сама, як і в біології. Дочірні класи-спадкоємці "переймають" від материнських їхні атрибути та методи. Тобто вам не потрібно їх заново прописувати. Але який сенс успадкування того самого без змін? Ніякого.
- Якщо ви використовуєте успадкування, завжди повинні в дочірніх класах щось змінювати або додавати.
- Наприклад, материнський клас Car (автомобіль) може їздити (вперед, назад, повертати). Якщо ви успадкуєте від нього клас FlyingCar (літаюча машина), то можете йому додати функцію польоту, при цьому літати вона зможе так само - вперед, назад і повертати, як і материнський клас.

- Також існує множинне спадкування, коли материнських класів у дочірнього не один, а кілька. Наприклад, клас SuperUser з розширеними правами може бути успадкований від класу Admin і User, при цьому маючи функції користувача та права адміністратора.

### Закріплення матеріалу

- Що таке ООП?
- Навіщо потрібне ООП?
- Що таке спадкування?
- Чи може бути множинне успадкування?

### Додаткове завдання

Вивчити теорію спадкування. Як успадкування допомагає скоротити дублювання коду? Дайте відповідь самі собі на це питання.

### Самостоятельная деятельность учащегося

1. Написати клас Cat, створити атрибути size, color, cat\_type.
2. При створенні об'єкта класу передавати в конструктор color та cat\_type, які записуються у відповідні атрибути.
3. Зробити метод set\_size, коли self.cat\_type це "indoor", то self.size = 'small' інакше self.size='undefined'. Протестуйте різні варіанти.
4. Створити клас Tiger, успадкований від класу Cat.
5. Перевизначити метод set\_size таким чином, щоб якщо self.cat\_type це 'wild', то self.size = 'big' інакше self.size='undefined'.

### Рекомендовані ресурси

- [Inheritance \(успадкування, англ\)](#)
- [ООП](#) (про принципи ООП та про нього в цілому)