

# Абстракція

**№ уроку:** 4 **Курс:** Python Базовий

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

## Огляд, мета та призначення уроку

Познайомитись з тим, що таке абстракція і як вона реалізована у Python. У цьому уроці буде розглянуто саме поняття абстракції і після, на практичних прикладах, буде показано, як вона реалізована в Python.

## Вивчивши матеріал даного заняття, учень зможе:

- Розуміти, що таке абстракція, і навіщо вона потрібна.
- Знати, як вона реалізована в Python і вміти застосувати її на практиці.

## Зміст уроку

1. Що таке абстракція
2. Як вона застосовна на практиці
3. Вирішення задач

## Резюме

- Завершальне поняття в ООП – це абстракція. Дане поняття не має реалізацію через ключові слова чи спеціальні символи. Абстракція є скомбінованим принципом із раніше вивчених успадкування, інкапсуляції та поліморфізму.
- Абстракція в ООП - це приховування складності реалізації програмного продукту від кінцевого користувача шляхом створення простих і зрозумілих інтерфейсів (мова про так звані класи).
- Основну ідею абстракції можна легко переплутати з інкапсуляцією. Адже приховування даних користувача з'являється в обох визначеннях. У разі інкапсуляції – це просте приховування атрибутів та методів, а абстракція – це приховування цілих класів чи груп класів шляхом побудови архітектури програмного продукту (саме за рахунок ООП).
- Уміння мислити абстрактно є ключем до розуміння ідеї ООП.
- По суті, всі прийоми ООП є простими (ви їх швидко зрозумієте), але найскладніший момент – це проектування архітектури (як організувати класи? які їм дати атрибути та методи? як все пов'язати в єдине ціле?). Ця навичка обов'язково прийде до вас з практикою та досвідом.

## Закріплення матеріалу

- Що таке абстракція?
- Навіщо потрібна абстракція?
- Що таке архітектура програмного продукту? Чи існує спеціальний синтаксис для абстракції в Python?

## Додаткове завдання

Вивчити теорію абстракції у ООП. Як абстракція ООП допомагає у розробці програмних продуктів? Яким є її призначення?

## Самостійна діяльність учня

1. Організуйте архітектуру програми "База даних" (псевдо). У ролі бази даних у вас буде клас Database, який зберігатиме дані у вигляді змінної списку.
2. Клас Database повинен мати методи `read_data(criteria)`, `write_data(element)`.
3. Для елементів даних напишіть клас Data. У цьому випадку ми зберігатимемо дані про користувачів. Data матиме атрибути: `country`, `name`, `age`, `gender`, `height`, `weight`.
4. У класі Database метод `read_data` прийматиме на вхід аргумент `criteria`, який є словником виду `{"age": 25}`, після чого метод поверне окремий список всіх елементів, у яких дана умова істина.

Підказка: щоб отримати в об'єкта класу значення його атрибуту як у словника, використовуйте наступний синтаксис: `your_class_instance.dict.get('name')`.

PS: організуйте правильну інкапсуляцію. Ви повинні додавати елементи до класу Database тільки через метод `write`, але ніяк не безпосередньо через атрибут `elements`.

## Рекомендовані ресурси

- <https://www.faceprep.in/python/abstraction-in-python/#:~:text=Abstraction%20in%20Python%3F,Abstraction%20in%20Python%20is%20the%20process%20of%20hiding%20the%20real,using%20abstract%20classes%20and%20methods.>
- [Абстракція в ООП](#)
- <https://www.rupython.com/27563-27563.html>