

6

ІНФОРМАТИКА

6

Створення програм із використанням функцій користувача. Практична робота 8

За навчальною програмою 2017 року



Урок 31

На *рисунку*
наведено
алгоритм роботи
з програмою в
середовищі
Python у вікні
програми.



Введення тексту
програми

Збереження
файлу
програми

Виконання
програми

Перегляд
результату
у вікні IDLE

Метод поділу на підзадачі полягає в тому, щоб у ході розроблення алгоритму розв'язування складної задачі спочатку визначити основні кроки, тобто розробити стратегію розв'язування задачі (**що зробити?**).

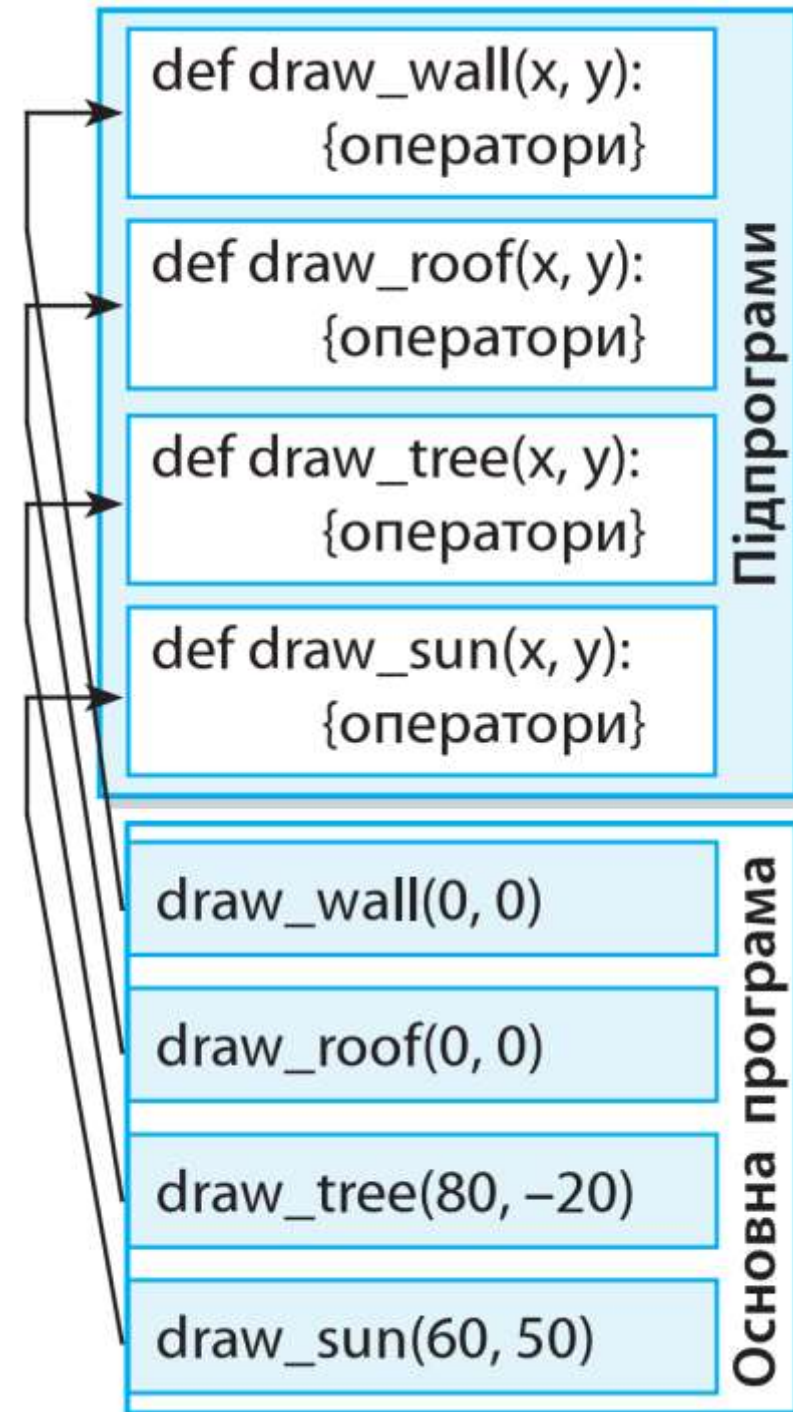
Далі потрібно розробити шляхи та методи реалізації намічених кроків (**як зробити?**).



Метод поділу складної задачі на підзадачі

Алгоритм розв'язування кожної підзадачі можна оформити як **функцію** — окремий іменований блок програмного коду.

Функції реалізують алгоритми розв'язування підзадач складної задачі. Їх можна порівняти з невеликими програмами, що вбудовані в основну програму.





У **Python** є можливість створювати й використовувати власні функції, так звані **функції користувача** (під користувачем розуміють програміста — того, хто пише програму, а не того, хто потім її використовує).



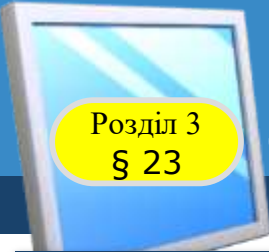
Розглянуті раніше функції:

draw_roof

draw_tree

draw_sun

є функціями користувача.



Функція складається з трьох частин:

імені

параметрів

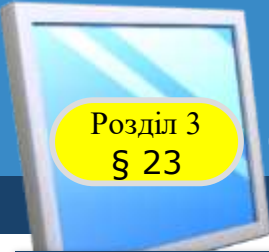
тіла

У **Python** функції визначаються за допомогою зарезервованого слова **def**.

Опис функції:

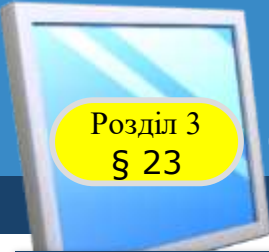
***def ім'я_функції(<перелік параметрів>):
тіло функції***

Тіло функції являє собою послідовність операторів, які будуть виконані після викликання функції.



Ознайомимося з правилами створення функцій у *Python*.

- **Блок функції починається з ключового слова *def*, після якого пишуть назву функції й круглі дужки *()*.**
- **Усередині дужок пишуть параметри — імена змінних, які отримують значення під час виклику функції. Ці параметри ще називають формальними. Якщо функції не треба передавати значення, необхідно записати порожні дужки.**
- **Після дужок ставиться двокрапка і з нового рядка з відступом записуються оператори тіла функції.**



*Після того як функцію створено, її можна викликати з іншої функції або безпосередньо з оболонки **Python**.*

*Щоб **викликати функцію**, потрібно:*

1) ввести ім'я функції й додати дужки;

*2) у дужках перелічити **аргументи функції** — значення, які треба присвоїти формальним параметрам.*

Якщо формальних параметрів немає, то при виклику після імені функції потрібно писати порожні дужки.



Не можна використати змінні, створені в тілі функції, після того, як ця функція завершить роботу. Ці змінні існують тільки під час її виконання. У таких випадках говорять, що область видимості змінних обмежена.



Область видимості — це та частина програмного коду, в якій змінна доступна для використання.

Змінні, створені всередині тіла функції, є локальними — їх «не видно» з інших функцій і з основної програми.



Якщо змінна створена в основній програмі, вона є глобальною і її можна використовувати у всіх наступних командах і в будь якій функції.

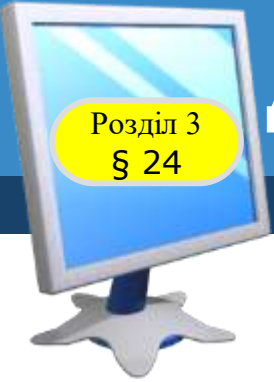
Якщо потрібно змінити глобальну змінну всередині функції, то слід використовувати ключове слово *global*.

```
def info():  
    global a  
    a += 1  
    b = 22
```

→ локальна змінна

```
a = 44  
info()  
print('a =', a)
```

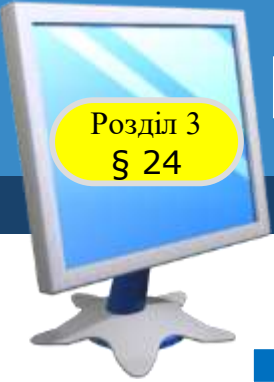
→ глобальна змінна



Домашнє завдання



***Проаналізувати
§ 24, ст. 153-154***



Практична робота 8

***Створення програм із
використанням
функцій користувача***





**Сторінка
153-154**



6

ІНФОРМАТИКА

Дякую за увагу!

6

За навчальною програмою 2017 року



Урок 31