

Урок 13

Класи та об'єкти

Як користуватися класами та об'єктами у Python

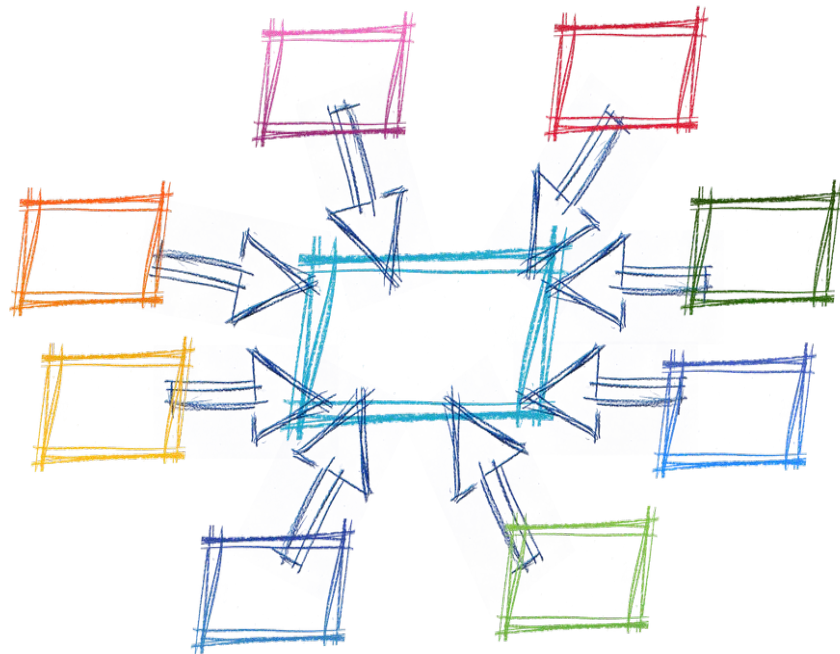


Об'єктно-орієнтоване програмування

це певний підхід у програмуванні та створенні програм.

Згідно з цим напрямом - все, що є у Python, це **об'єкти**, які взаємодіють між собою.

Всі об'єкти створюються за допомогою **класів**.



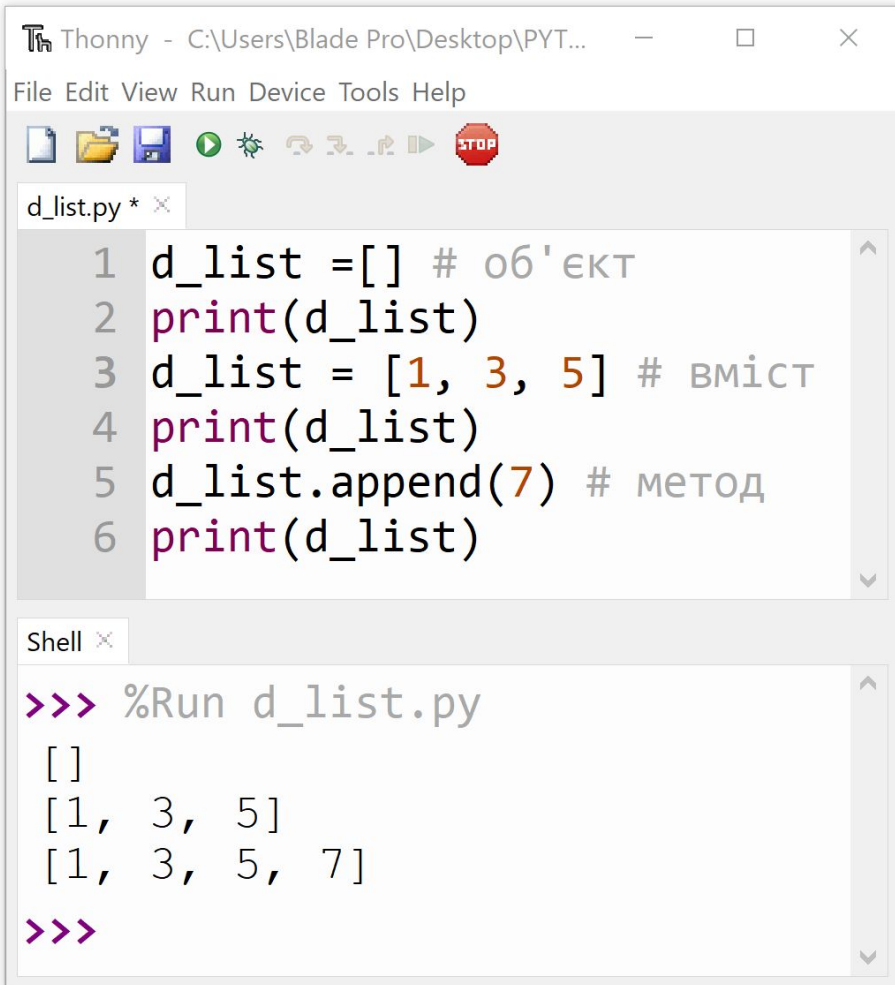
Об'єкт

це **контейнер**, який має певні **характеристики** (дані або стан) та **виконує певні дії** (поведінка).

Приклад:

список **d_list**, який містить дані - набір чисел **1, 3, 5** та може виконувати певні дії за допомогою методів, наприклад **.append**

Різним об'єктам властива різна поведінка (різні методи).



The screenshot shows the Thonny Python IDE interface. The title bar indicates the file path: C:\Users\Blade Pro\Desktop\PYT... The menu bar includes File, Edit, View, Run, Device, Tools, and Help. The toolbar contains icons for opening files, saving, running, and stopping. The main editor window displays a Python script named d_list.py with the following code:

```
1 d_list = [] # об'єкт
2 print(d_list)
3 d_list = [1, 3, 5] # вміст
4 print(d_list)
5 d_list.append(7) # метод
6 print(d_list)
```

Below the editor is the Shell window, which shows the output of running the script:

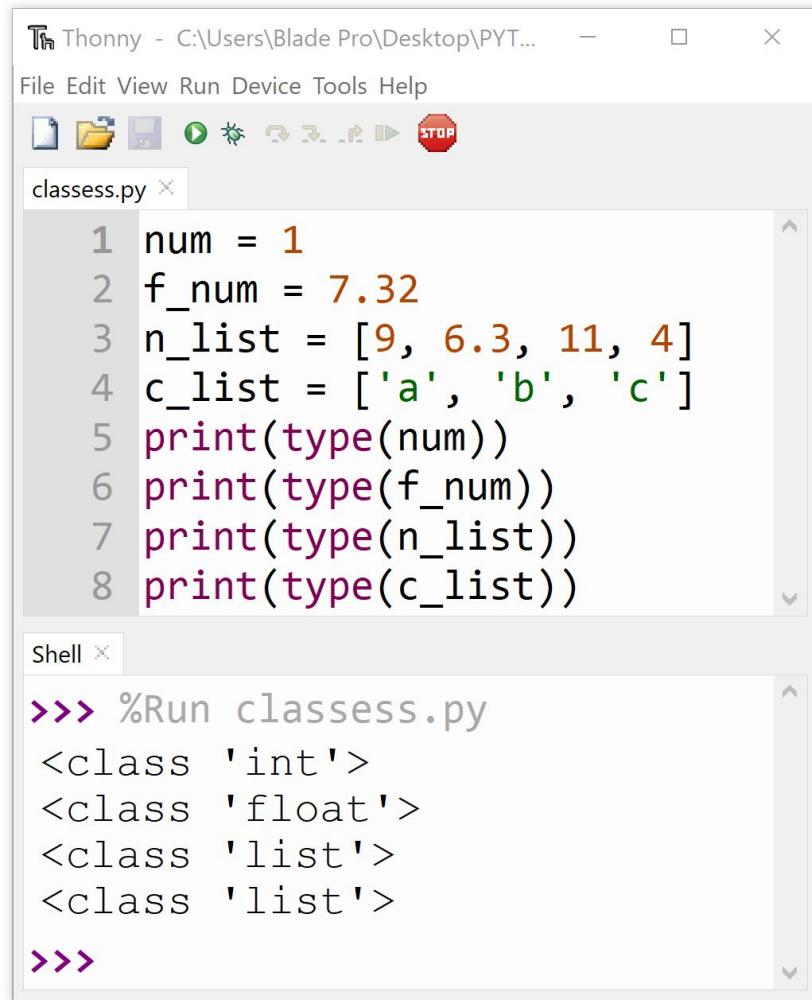
```
>>> %Run d_list.py
[]
[1, 3, 5]
[1, 3, 5, 7]
>>>
```

Клас

це спосіб групування (класифікація) або шаблон певних характеристик об'єкта.

Дізнатись клас об'єкта можна за допомогою функції **type()**.

Клас можна порівняти з видами тварин: птахи (горобець, орел), ссавці (кішка, жирафа), а представників конкретних видів - з об'єктами.



The screenshot shows the Thonny Python IDE interface. The title bar reads "Thonny - C:\Users\Blade Pro\Desktop\PYT...". The menu bar includes "File", "Edit", "View", "Run", "Device", "Tools", and "Help". The toolbar contains icons for file operations and execution. The editor window, titled "classess.py", contains the following Python code:

```
1 num = 1
2 f_num = 7.32
3 n_list = [9, 6.3, 11, 4]
4 c_list = ['a', 'b', 'c']
5 print(type(num))
6 print(type(f_num))
7 print(type(n_list))
8 print(type(c_list))
```

The Shell window at the bottom shows the execution output:

```
>>> %Run classess.py
<class 'int'>
<class 'float'>
<class 'list'>
<class 'list'>

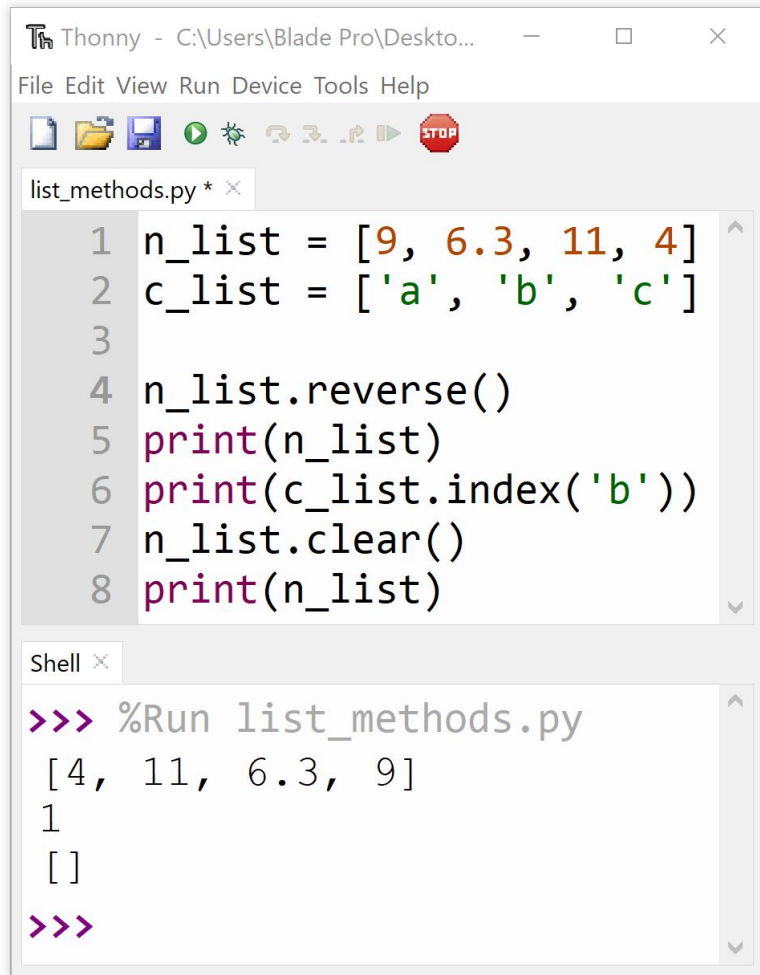
>>>
```

Класи у Python

У Python існують вбудовані класи об'єктів, наприклад: **int**, **str**, **list**.

За допомогою цих класів можна створювати об'єкти зі стандартними характеристиками.

Якщо потрібен об'єкт з нестандартними властивостями, які не мають вбудовані класи, можна створити необхідний клас самостійно.



The screenshot shows the Thonny Python IDE interface. The title bar reads 'Thonny - C:\Users\Blade Pro\Desktop...'. The menu bar includes 'File', 'Edit', 'View', 'Run', 'Device', 'Tools', and 'Help'. Below the menu is a toolbar with icons for file operations and execution. The main editor window, titled 'list_methods.py *', contains the following Python code:

```
1 n_list = [9, 6.3, 11, 4]
2 c_list = ['a', 'b', 'c']
3
4 n_list.reverse()
5 print(n_list)
6 print(c_list.index('b'))
7 n_list.clear()
8 print(n_list)
```

Below the editor is a 'Shell' window. It shows the command prompt output after running the script:

```
>>> %Run list_methods.py
[4, 11, 6.3, 9]
1
[]
>>>
```

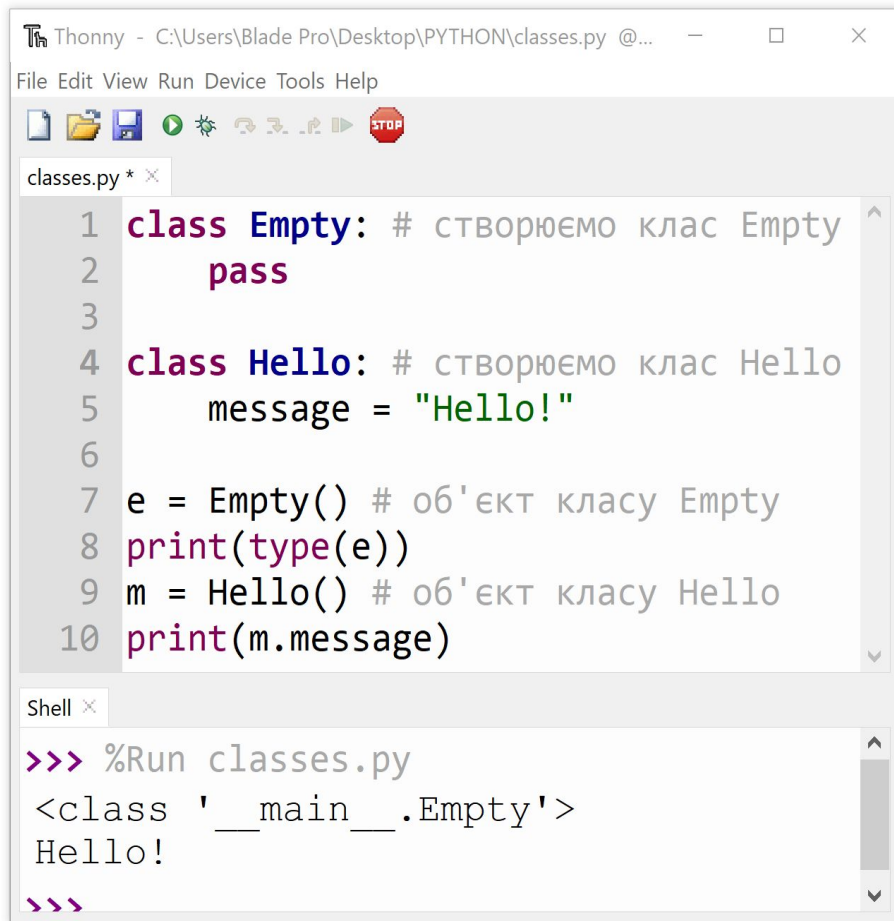
Створення класу

Свій клас можна створити за допомогою ключового слова **class**, потім вказується назва класу з

Великої літери і двокрапка :

Якщо ми створюємо порожній клас, треба додати команду **pass**.

Якщо ми хочемо надати певні характеристики майбутнім об'єктам цього класу, вони вказуються з нового рядка в блоці класу.



The screenshot shows the Thonny Python IDE interface. The main editor window displays a Python script named `classes.py` with the following code:

```
1 class Empty: # створюємо клас Empty
2     pass
3
4 class Hello: # створюємо клас Hello
5     message = "Hello!"
6
7 e = Empty() # об'єкт класу Empty
8 print(type(e))
9 m = Hello() # об'єкт класу Hello
10 print(m.message)
```

Below the editor is a Shell window showing the output of running the script:

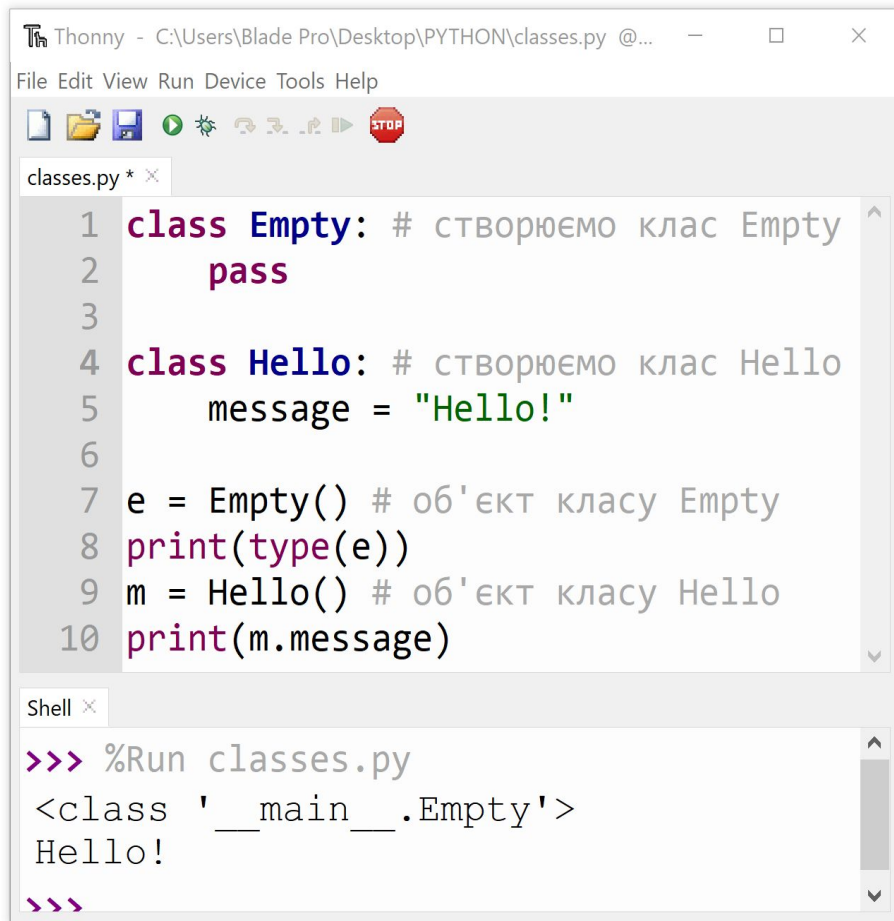
```
>>> %Run classes.py
<class '__main__.Empty'>
Hello!
>>>
```

Екземпляр класу

це об'єкт, що відноситься до певного класу.

Щоб створити об'єкт класу, треба створити змінну зі значенням назви класу з дужками.

Новостворений об'єкт володіє всіма властивостями класу, які були прописані при створенні цього класу.



The screenshot shows the Thonny Python IDE interface. The main editor window displays a Python script named `classes.py` with the following code:

```
1 class Empty: # створюємо клас Empty
2     pass
3
4 class Hello: # створюємо клас Hello
5     message = "Hello!"
6
7 e = Empty() # об'єкт класу Empty
8 print(type(e))
9 m = Hello() # об'єкт класу Hello
10 print(m.message)
```

Below the editor, the Shell window shows the output of running the script:

```
>>> %Run classes.py
<class '__main__.Empty'>
Hello!
>>>
```



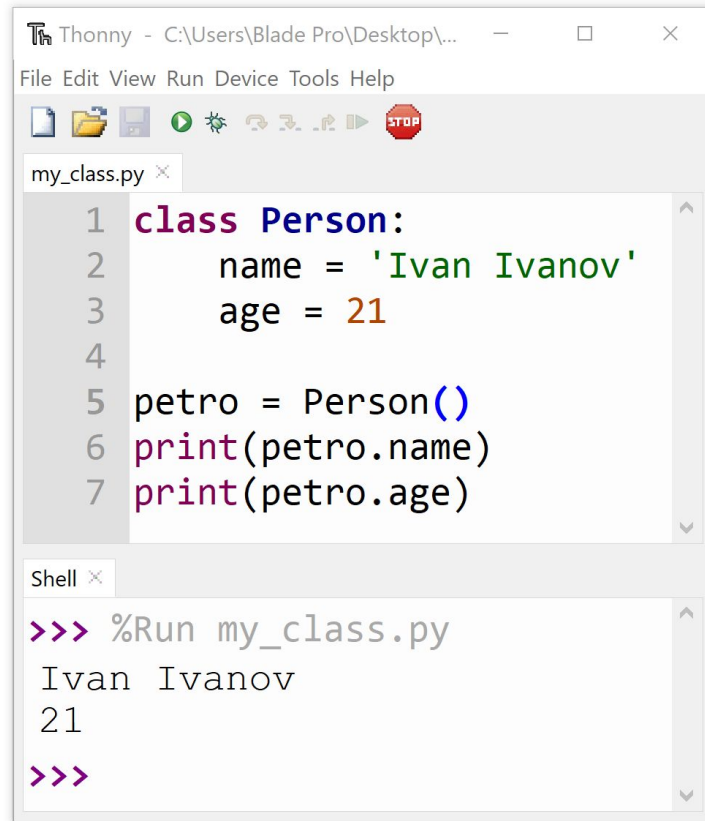
Практична робота

- 1) Створіть новий файл (**New**)
- 2) Створіть клас з назвою **Person**, який містить наступні дані:

name = "Ivan Ivanov"

age = 21

- 3) Створіть об'єкт (екземпляр) класу **Person()** з ім'ям **petro**
- 4) Збережіть файл (**Save**) під назвою **my_class.py**
- 5) Виведіть значення **name** та **age** об'єкта **petro** у вікно **Shell**



The image shows a screenshot of the Thonny Python IDE. The window title is "Thonny - C:\Users\Blade Pro\Desktop\...". The menu bar includes "File", "Edit", "View", "Run", "Device", "Tools", and "Help". The toolbar contains icons for file operations and execution. The editor window, titled "my_class.py", contains the following Python code:

```
1 class Person:
2     name = 'Ivan Ivanov'
3     age = 21
4
5 petro = Person()
6 print(petro.name)
7 print(petro.age)
```

Below the editor is a "Shell" window. It shows the command prompt with the command `>>> %Run my_class.py` and the output:

```
Ivan Ivanov
21
>>>
```

Якщо все зроблено правильно - ви побачите такий результат

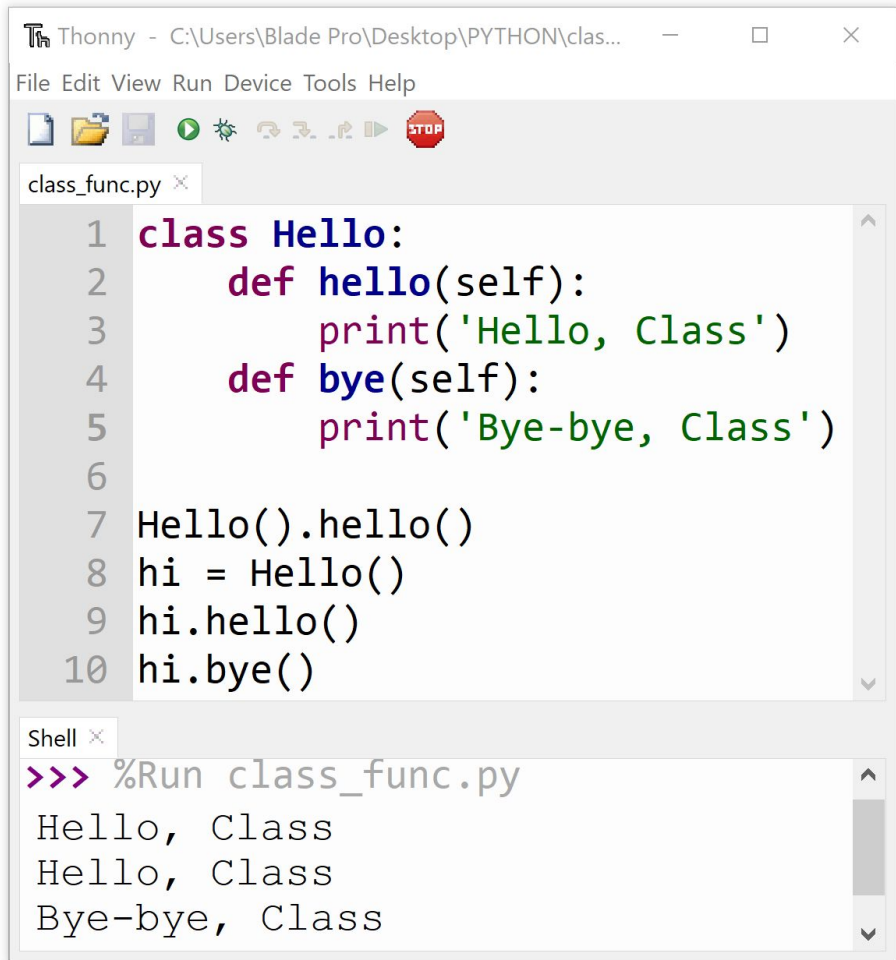
Функції класів

Для класів можна створювати функції.

Для створення функції використовують команду **def**, після якої вказується аргумент **self** у дужках та тіло у вкладеному блоці після двокрапки.

Функції, які належать класу, називають **методами**.

Щоб викликати метод класу, треба вказати назву класу та назву функції



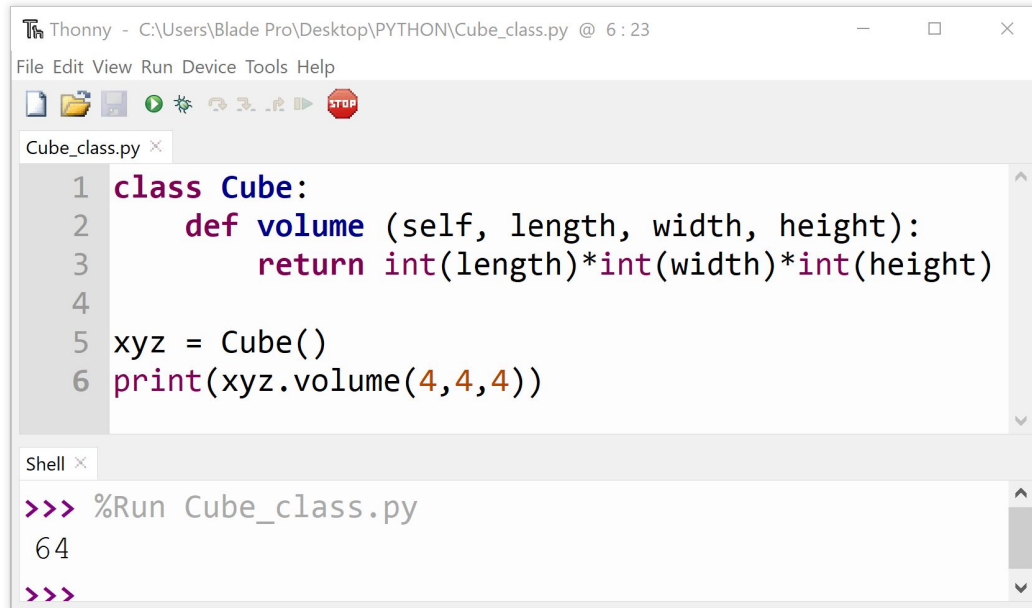
The screenshot shows the Thonny Python IDE interface. The main editor window displays a Python script named `class_func.py` with the following code:

```
1 class Hello:
2     def hello(self):
3         print('Hello, Class')
4     def bye(self):
5         print('Bye-bye, Class')
6
7 Hello().hello()
8 hi = Hello()
9 hi.hello()
10 hi.bye()
```

Below the editor is a Shell window showing the output of running the script:

```
>>> %Run class_func.py
Hello, Class
Hello, Class
Bye-bye, Class
```

Методи класів з параметрами



The screenshot shows the Thonny Python IDE interface. The title bar indicates the file path: C:\Users\Blade Pro\Desktop\PYTHON\Cube_class.py. The menu bar includes File, Edit, View, Run, Device, Tools, and Help. The toolbar contains icons for file operations and execution. The editor window displays the following Python code:

```
1 class Cube:
2     def volume (self, length, width, height):
3         return int(length)*int(width)*int(height)
4
5 xyz = Cube()
6 print(xyz.volume(4,4,4))
```

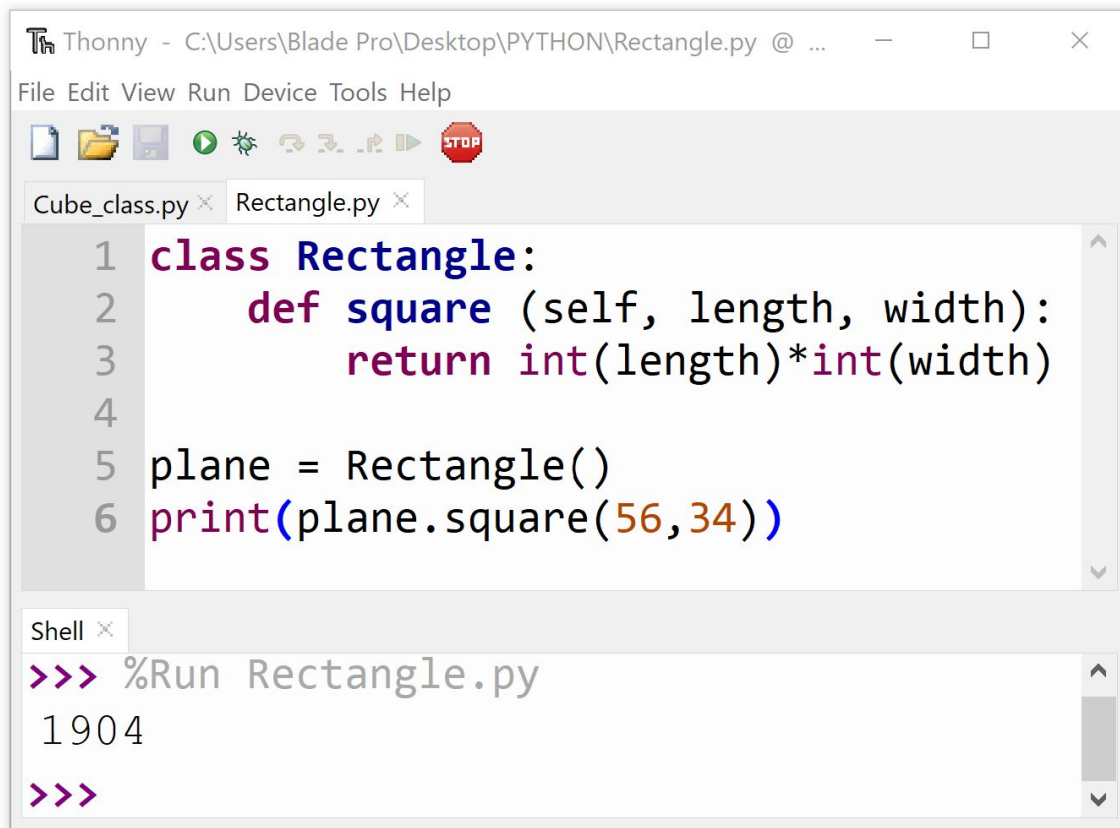
Below the editor is a Shell window showing the execution output:

```
>>> %Run Cube_class.py
64
>>>
```



Практична робота

- 1) Створіть новий файл (**New**)
- 2) Створіть клас **Rectangle**, який містить функцію **square** для розрахунку площі прямокутника, має два параметри - довжину (**length**) та ширину (**width**) та повертає результат множення цих параметрів
- 3) Використайте цей клас, щоб створити об'єкт **plane** з розмірами **56** та **34**. Використовуючи метод **square** розрахуйте площу об'єкта **plane**
- 4) Збережіть файл (**Save**) під назвою **rectangle.py**
- 5) Виведіть результат у вікно **Shell** за допомогою функції **print()**



The image shows a screenshot of the Thonny Python IDE. The title bar indicates the file path: C:\Users\Blade Pro\Desktop\PYTHON\Rectangle.py. The menu bar includes File, Edit, View, Run, Device, Tools, and Help. Below the menu is a toolbar with icons for opening files, saving, running, and stopping. The main editor window displays a Python script with the following code:

```
1 class Rectangle:
2     def square (self, length, width):
3         return int(length)*int(width)
4
5 plane = Rectangle()
6 print(plane.square(56,34))
```

Below the editor is a Shell window showing the execution output:

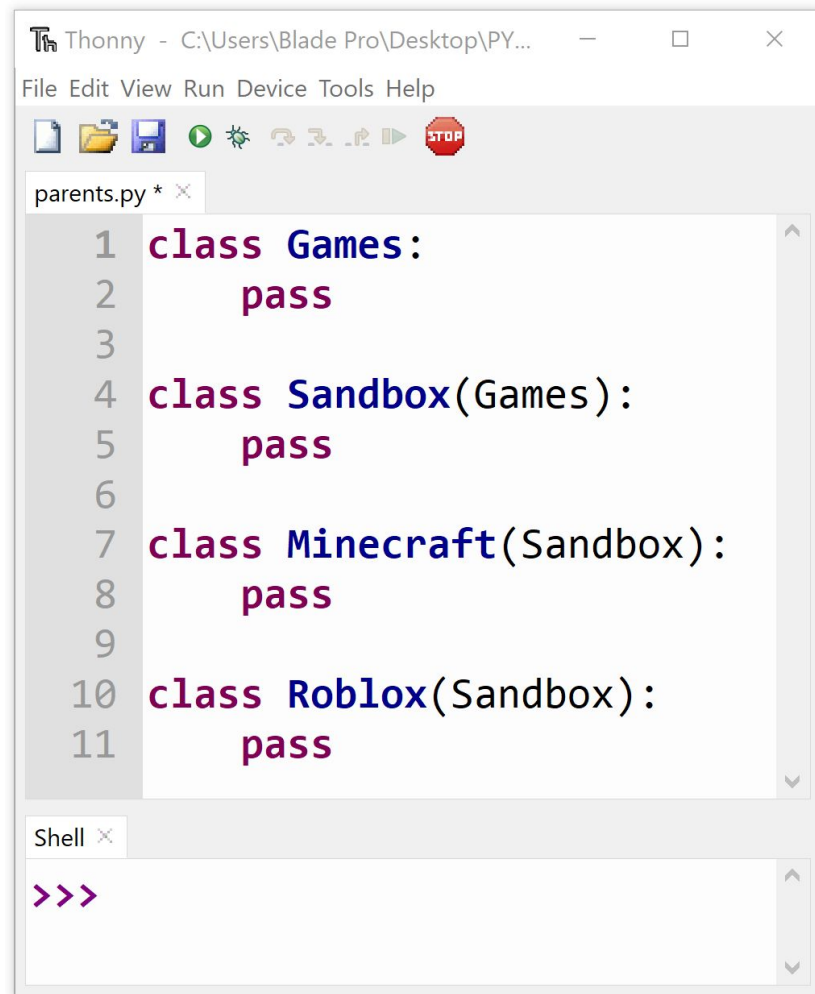
```
>>> %Run Rectangle.py
1904
>>>
```

Якщо все зроблено правильно - ви побачите такий результат

Батьки й нащадки

Якщо клас є частиною іншого класу, підпорядковується йому, то його називають **нащадком**, а клас, до якого він належить - **батьком**.

Щоб створити клас-нащадок, треба при його створенні у дужках вказати ім'я батьківського класу.



```
Thonny - C:\Users\Blade Pro\Desktop\PY...
File Edit View Run Device Tools Help

parents.py *
1 class Games:
2     pass
3
4 class Sandbox(Games):
5     pass
6
7 class Minecraft(Sandbox):
8     pass
9
10 class Roblox(Sandbox):
11     pass

Shell
>>>
```



Підсумки

Дізнались, що таке **об'єкти** та **класи**, які вони бувають та для чого вони використовуються

Навчилися створювати **власні класи**

Дізнались, як створювати **екземпляри класів**