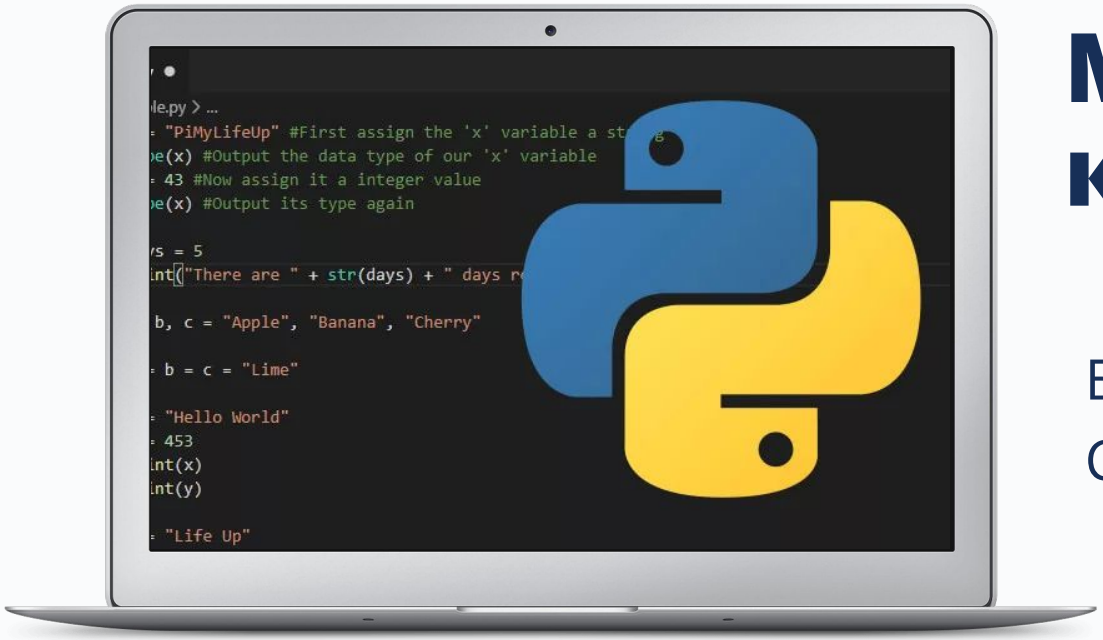


Методи класу

Бутенко
Сергій

План лекції

- Поняття методу
- Методи об'єкта
- Методи класу
- “Магічні” методи
- Статичні методи

Поняття методу

Методи - це функції, які визначені всередині класу.

Типи методів:

- методи об'єкта
- методи класу
- статичні методи

Методи об'єкта

Методи об'єкта - це функції, які визначені всередині класу та визначають стан об'єкта

Синтаксис створення методу об'єкта:

def <ім'я методу>(self[, <параметри>]):

    [self.<атрибут> ...]

    [self.<метод>()]

Синтаксис виклику методу об'єкта:

<ім'я об'єкта>.<ім'я методу>([<параметри>])

Приклад 1. Використання методу об'єкта

shopMethod.py

PYTHON_BASICS > shopMethod.py > ...

```
1  class ShopWorker:
2      count_workers = 0
3      def __init__(self, name1='', age1=0):
4          self.name = name1
5          self.age = age1
6          ShopWorker.count_workers +=1
7      def add_year(self):
8          self.age+=1
9
10 print("всіх працівників ", ShopWorker.count_workers)
11
12 worker_one = ShopWorker('Іван', 25)
13 print("Працівник 1: ",worker_one.name," ", worker_one.age,
14 " всіх працівників ",  worker_one.count_workers)
15 worker_one.add_year()
16 print("Працівник 1: ",worker_one.name," ", worker_one.age,
17 " всіх працівників ",  worker_one.count_workers)
18
19 worker_two = ShopWorker('Петро', 32)
20 print("Працівник 2: ",worker_two.name," ", worker_two.age,
21 " всіх працівників ",  worker_two.count_workers)
22 worker_two.add_year()
23 print("Працівник 2: ",worker_two.name," ", worker_two.age,
24 " всіх працівників ",  worker_two.count_workers)
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE JUPYTER Python

```
всіх працівників  0
Працівник 1:  Іван   25  всіх працівників  1
Працівник 1:  Іван   26  всіх працівників  1
Працівник 2:  Петро   32  всіх працівників  2
Працівник 2:  Петро   33  всіх працівників  2
```

Опис класу **ShopWorker**
Створення атрибуту класу **count_workers**
Конструктор класу

Опис методу **add_year** для збільшення віку об'єкта

Вивід атрибуту класу **ShopWorker**

Створення об'єкту **worker_one** та ініціалізація його полів
Вивід атрибутів об'єкту **worker_one** та атрибуту класу
Виклик методу збільшення віку для об'єкта **worker_one**
Вивід атрибутів об'єкту **worker_one** та атрибуту класу

Створення об'єкту **worker_two** та ініціалізація його полів
Вивід атрибутів об'єкта **worker_two** та атрибуту класу
Виклик методу збільшення віку для об'єкта **worker_two**
Вивід атрибутів об'єкта **worker_two** та атрибуту класу

Результат роботи коду

Приклад 2. Використання методу об'єкта

shopMethod1.py

PYTHON_BASICS > shopMethod1.py > ...

```
1  class ShopWorker:
2      count_workers = 0
3      def __init__(self, name1='', age1=0):
4          self.name = name1
5          self.age = age1
6          ShopWorker.count_workers +=1
7      def add_year(self, year):
8          self.age+=year
9
10 print("всіх працівників ", ShopWorker.count_workers)
11
12 worker_one = ShopWorker('Іван', 25)
13 print("Працівник 1: ",worker_one.name," ", worker_one.age,
14 " всіх працівників ",  worker_one.count_workers)
15 worker_one.add_year(3)
16 print("Працівник 1: ",worker_one.name," ", worker_one.age,
17 " всіх працівників ",  worker_one.count_workers)
18
19 worker_two = ShopWorker('Петро', 32)
20 print("Працівник 2: ",worker_two.name," ", worker_two.age,
21 " всіх працівників ",  worker_two.count_workers)
22 worker_two.add_year(2)
23 print("Працівник 2: ",worker_two.name," ", worker_two.age,
24 " всіх працівників ",  worker_two.count_workers)
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE JUPYTER Python

```
всіх працівників  0
Працівник 1:  Іван   25  всіх працівників  1
Працівник 1:  Іван   28  всіх працівників  1
Працівник 2:  Петро   32  всіх працівників  2
Працівник 2:  Петро   34  всіх працівників  2
```

Опис класу **ShopWorker**
Створення атрибуту класу **count_workers**
Конструктор класу

Опис методу **add_year** для збільшення віку об'єкта

Вивід атрибуту класу **ShopWorker**

Створення об'єкту **worker_one** та ініціалізація його полів
Вивід атрибутів об'єкту **worker_one** та атрибуту класу

Виклик методу збільшення віку для об'єкта **worker_one**
Вивід атрибутів об'єкту **worker_one** та атрибуту класу

Створення об'єкту **worker_two** та ініціалізація його полів
Вивід атрибутів об'єкта **worker_two** та атрибуту класу

Виклик методу збільшення віку для об'єкта **worker_two**
Вивід атрибутів об'єкта **worker_two** та атрибуту класу

Результат роботи коду

Методи класу

Методи класу - це методи, які визначають поведінку саме класу загалом, що впливає всі об'єкти цього класу

`cls` - це обов'язковий параметр методу класу, який вказує на сам клас.

`@classmethod` - це декоратор, що позначає метод класу.

Синтаксис методу класу:

```
@classmethod
  def <ім'я методу>(cls[,<параметр1>[,<параметр2> …]]):
        <команди методу>
```

Виклик методу класу:

```
<ім'я класу>. <ім'я методу>([<параметри>])
<ім'я об'єкта>. <ім'я методу>([<параметри>])
```

Приклад методу класу:

```
@classmethod
   def naming_shop(cls, name):
        cls.name_shop= name
        return cls.name_shop
```

“Магічні” методи

“Магічні” методи - це методи, які в імені мають дві риски підкреслення на початку та в кінці імені методу

Приклади магічних методів:

`__new__` , `__init__` , `__str__`

Метод `__str__(self)` використовують для перетворення об'єкта до рядкового представлення, викликається, коли об'єкт передається функціям print() і str();

Приклад:

```
def __str__(self):
    str_out="Працівник : "+self.name+" "+str(self.age)
    str_out+=" всіх працівників "+str(ShopWorker.count_workers)
    return str_out
```

Статичні методи

Статичні методи - це методи, які не змінюють стан ні класу, ні об'єкта.

Синтаксис написання статистичного методу

```
@staticmethod
 def <ім'я методу>([<параметр1> [,<параметр2>...]]):
       <команди методу>
```

Виклик методу:

```
<ім'я класу>.<ім'я методу>([параметри])
```

Приклад статичного методу:

```
@staticmethod
def info():
  print('В магазині працює: ', ShopWorker.count_workers, ' працівників')
```

Приклад 3. Клас з різними методами

shopMethod2.py

PYTHON_BASICS > shopMethod2.py > ...

```python
class ShopWorker:
    count_workers = 0
    def __init__(self, name1='', age1=0):
        self.name = name1
        self.age = age1
        ShopWorker.count_workers +=1
        self.id=ShopWorker.count_workers
    def add_year(self, year):
        self.age+=year
    def __str__(self):
        str_out="Працівник "+str(self.id)+": "+self.name+" "+str(self.age)
        str_out += " всіх працівників " + str(ShopWorker.count_workers)
        return str_out
    @staticmethod
    def info():
        print('В магазині працює: ', ShopWorker.count_workers, ' працівників')
    @classmethod
    def naming_shop(cls, name):
        cls.name_shop= name
        return cls.name_shop
ShopWorker.info()
worker_one = ShopWorker('Іван', 25)
print (worker_one)
worker_one.add_year(3)
print (worker_one)
worker_two = ShopWorker('Петро', 32)
print (worker_two)
worker_two.add_year(2)
print (worker_two)
print ('Назва магазину: ',worker_one.naming_shop('Fara'))
print ('Назва магазину: ',ShopWorker.name_shop)
print ('Назва магазину: ',ShopWorker.naming_shop('Para'))
```

Опис класу **ShopWorker**
Створення атрибуту класу **count_workers**
Конструктор класу

Опис методу **add_year** для збільшення віку об'єкта

Опис методу **__str__**

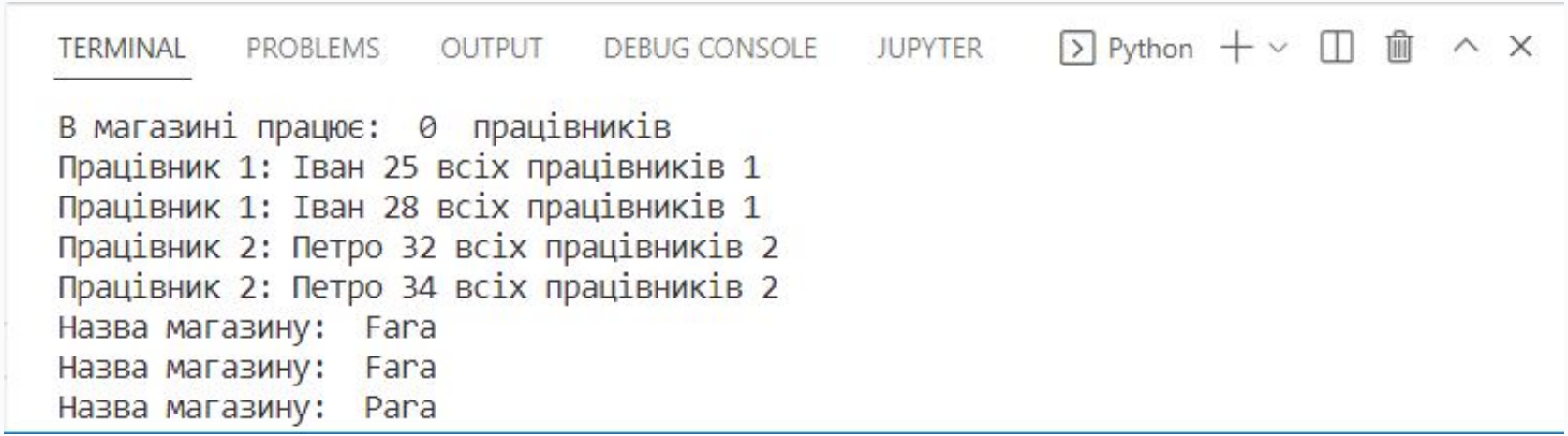
Статичний метод info()

Метод класу naming_shop(cls, name):

Вивід атрибуту класу **ShopWorker**

Створення об'єкту **worker_one**
Вивід атрибутів об'єкту **worker_one**
Виклик методу збільшення віку для об'єкта **worker_one**
Повторний вивід атрибутів об'єкту **worker_one**
Створення об'єкту **worker_two**
вивід атрибутів об'єкту **worker_two**
Виклик методу збільшення віку для об'єкта **worker_two**
Повторний вивід атрибутів об'єкту **worker_two**
Виклик методу класу **naming_shop** через об'єкт **worker_one** та
Вивід результату, вивід атрибута класу **name_shop** ,
Виклик методу класу **naming_shop** через назву
класу та вивід результату.

Приклад 3. Результат роботи програми



Підсумки

Методи - це функції, які визначені всередині класу.

Типи методів:
- методи об'єкта
- методи класу
- статичні методи.

Методи об'єкта - це функції, які визначені всередині класу та визначають стан об'єкта та мають обов'язковий параметр `self`.

Методи класу - це методи, які визначають поведінку саме класу загалом з обов'язковим параметром `cls`. Позначаються декоратором `@classmethod` .

Статичні методи - це методи, які не змінюють стан ні класу, ні об'єкта. Позначаються декоратором `@staticmethod`.

“Магічні” методи - це методи, які в імені мають дві риски підкреслення на початку та в кінці імені методу.