

**Мета:** навчитись розрізняти основні елементи теорії ООП: клас, екземпляр класу, атрибут, ме-тод, конструктор класу; вміти користуватись цими елементами при складанні програм звикористанням об'єктно-орієнтованого підходу на мові Python в інтерактивному режимі IDLE.

**Обладнання:** ОС Windows, Microsoft Office2013, середовище програмування IDLE

### Хід роботи

1 Використовуючи головне меню ОС Windows запустив інтерактивний режим інтерпретатора Python – середовище програмування IDLE

2 Створив в своєму каталозі папку ЛР16 для збереження наступних завдань

3 За допомогою команди File→NewFile (Ctrl+N) створив нові текстові документи з відповідними назвами ЛР\_16(No завдання) для розробки наступних кодів програми

4 Виконав індивідуальні завдання виданого варіанту (варіант 2)

4.1 Створить програму з конструктором, за допомогою якої із слів «Дніпро», «- моє», «місто» виводиться повідомлення «Дніпро – моє місто» (слова оформити за допомогою трьох відповідних змінних)

Код програми:

```
class WordSurr:
```

```
    w1 = "Дніпро"
```

```
    w2 = "- моє"
```

```
    w3 = "місто"
```

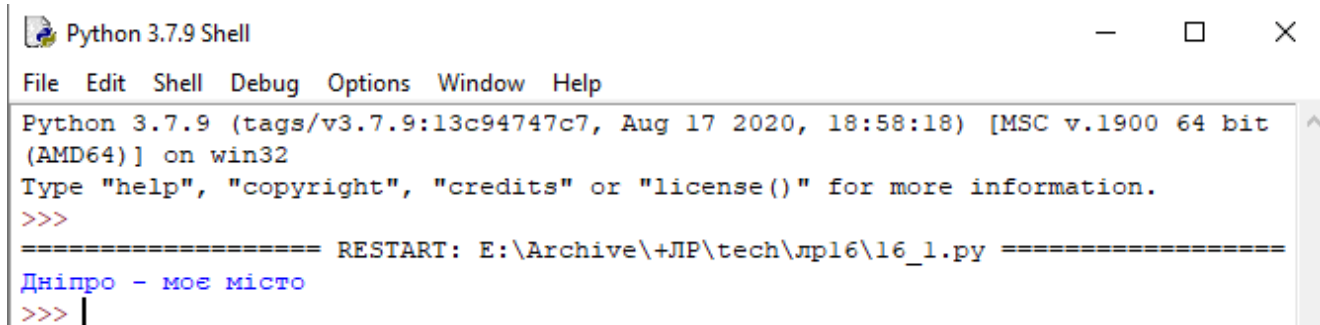
```
    def __init__(self):
```

```
        print(self.w1 + ' ' + self.w2 + ' ' + self.w3)
```

```
w = WordSurr()
```

Результат зображений на скріншоті 1

					<b>ДТЗЕ.121РПЗ00ТЕХ1600ЛР</b>			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Мітленко Д.А.			Використання конструктора класу		Лім.	Арк.
Перевір.		Саприкіна І.Г.						1
Реценз.							Група ПЗ-19-1/9	
Н. Контр.		Саприкіна І.Г.						
Затверд.		Саприкіна І.Г.						
							Аркушів	6



```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\Archive\+ЛП\tech\лп16\16_1.py =====
Дніпро - моє місто
>>> |
```

Скріншот 1

4.2 Створіть програму з конструктором та методом класу для обчислення наступного виразу  $(a+c)^3$ , де змінні  $a$  і  $c$  генеруються в заданому діапазоні в основній програмі

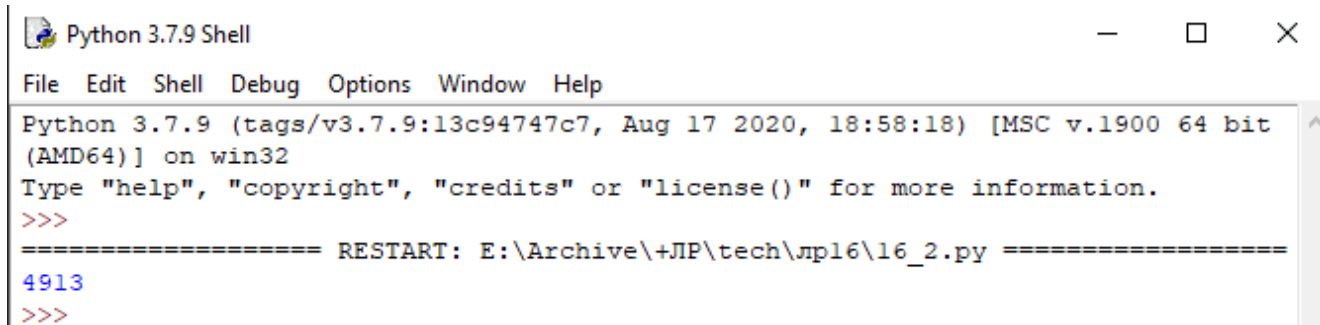
Код програми:

```
from random import randint

class Expression:
    a = 0
    c = 0
    def __init__(self,a,c):
        self.a = a
        self.c = c
    def calc(self):
        return (self.a + self.c) ** 3

obj = Expression(randint(1,10),randint(-4,12))
print(obj.calc())
```

Результат зображений на скріншоті 2



```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\Archive\+ЛП\tech\лп16\16_2.py =====
4913
>>>
```

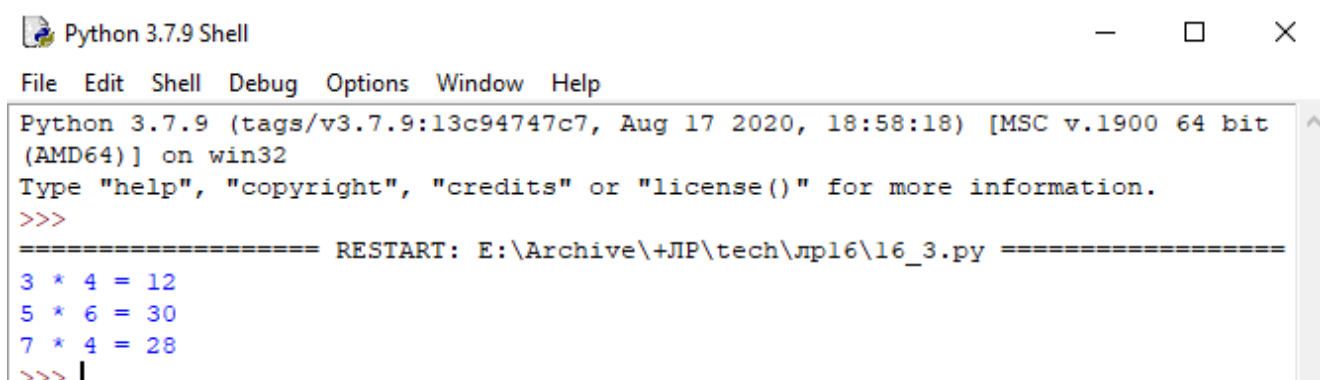
Скріншот 2

4.3 Конструктор має такі значення параметрів: self, a1= 3, a2= 4. Створіть програму множення значень параметрів конструктора за умови, що він викликається з першого об'єкта без аргументів, з другого – з аргументами 5 і 6, а з третього – з аргументом 7

Код програми:

```
class Class:
    def __init__(self,a1=3,a2=4):
        print("%i * %i ="%(a1,a2),a1*a2)
obj = Class()
obj2 = Class(5,6)
obj3 = Class(7)
```

Результат зображений на скріншоті 3



```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\Archive\+JLP\tech\jlp16\l6_3.py =====
3 * 4 = 12
5 * 6 = 30
7 * 4 = 28
>>> |
```

Скріншот 3

4.4 Конструктор має такі значення параметрів: self, n1= «Операційна», n2 = «система». За допомогою інструкції input() вводиться слово. Розробіть програму, за допомогою якої якщо введена буква L виводиться повідомлення – «Операційна система Linux», якщо введена буква W – повідомлення «Операційна система Windows», будь-яка інша буква – повідомлення про помилку

Код програми:

```
class OperatingSys:
    def __init__(self,n1="Операційна",n2="система"):
        sym = input('Символ: ')
        if sym == "L": self.p = n1 + ' ' + n2 + ' ' + "Linux"
```

```

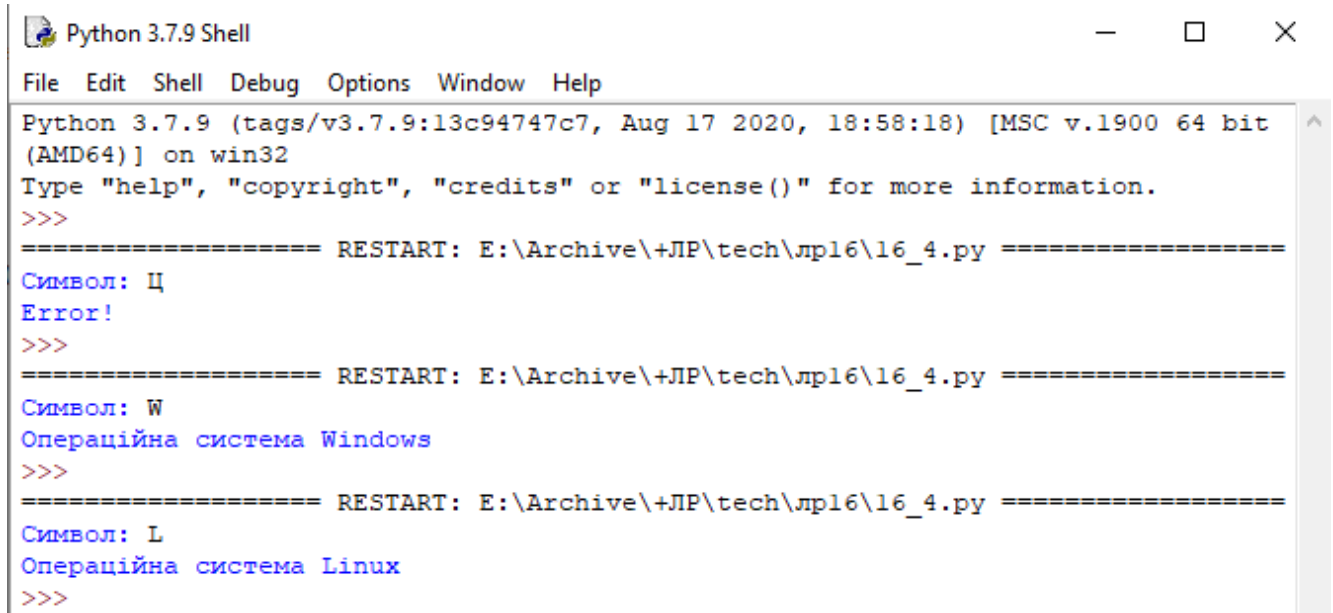
elif sym == "W": self.p = n1 + ' ' + n2 + ' ' + "Windows"
else: self.p = "Error!"

obj = OperatingSys()

print(obj.p)

```

Результат зображений на скріншоті 4



```

Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\Archive\+ЛП\tech\лп16\l6_4.py =====
Символ: C
Error!
>>>
===== RESTART: E:\Archive\+ЛП\tech\лп16\l6_4.py =====
Символ: W
Операційна система Windows
>>>
===== RESTART: E:\Archive\+ЛП\tech\лп16\l6_4.py =====
Символ: L
Операційна система Linux
>>>

```

Скріншот 4

4.5 Дано конус висотою  $h$  та різними радіусами основи  $r$ . Розробити програму з використанням конструктора класу і вбудованих стандартних методів за допомогою яких обчислити об'єм 4 різних конусів. Передбачити повідомлення про те що радіус конуса може бути тільки додатнім числом

Код програми:

```

class Conus:
    h,r = 0,0
    def __init__(self):
        print('Радіус конуса може бути тільки додатнім числом')
    def calc(self):
        return round((
            int(self.__getattrubute__('r')) *
            int(self.__getattrubute__('h')) *
            3.14

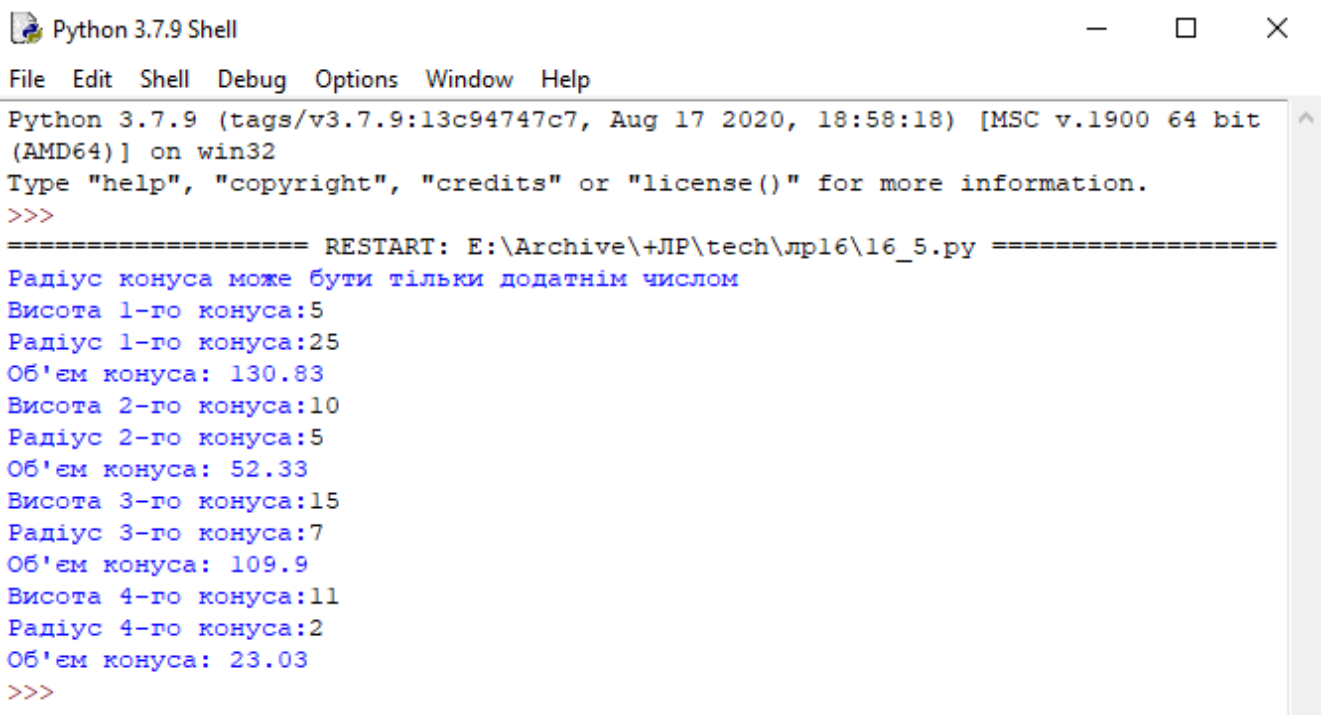
```

```

        )/3,2)
obj = Conus()
for i in range(4):
    h = input('Висота %i-го конуса:%%(i+1))
    r = input('Радіус %i-го конуса:%%(i+1))
    obj.__setattr__('h',h)
    obj.__setattr__('r',r)
    print("Об'єм конуса:",obj.calc())

```

Результат зображений на скріншот 5



```

Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\Archive\+LP\tech\lpl6\l6_5.py =====
Радіус конуса може бути тільки додатнім числом
Висота 1-го конуса:5
Радіус 1-го конуса:25
Об'єм конуса: 130.83
Висота 2-го конуса:10
Радіус 2-го конуса:5
Об'єм конуса: 52.33
Висота 3-го конуса:15
Радіус 3-го конуса:7
Об'єм конуса: 109.9
Висота 4-го конуса:11
Радіус 4-го конуса:2
Об'єм конуса: 23.03
>>>

```

Скріншот 5

### Відповіді на контрольні запитання

1 Що називають конструктором класу в ООП? Яку загальну структуру має метод `__init__`?

Конструктор - це метод, який викликається для автоматичного створення екземпляра класу. Загальна структура конструктору:

```

def __init__(self [,<параметр 1>,...,<параметр N>]):
    тіло методу

```

2 У яких випадках викликається метод `__init__`? Наведіть загальну

		Мітленко Д.А.			ДТЗЕ.121РПЗ00ТЕХ1600ЛР	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		5

структуру інструкції створення об'єкта класу з використанням конструктора?

Метод викликається, коли, при створення, класу необхідно передати певні значення:

екземпляр\_класу = ім'я\_класу (аргумент1,..., аргументN)

3 Які дії виконуються в процесі створення об'єкта за наявності в класі конструктора?

При створенні об'єкта також виконуються дії в конструкторі класу.

4 Що передається параметру self в процесі створення об'єкта?

self - це посилання на сам об'єкт, який створюється.

5 Як можна присвоїти значення параметрам методу \_\_init\_\_ в самому методі?

Це можна зробити задавши параметрам стандартні значення

**Висновок:** я навчився розрізняти основні елементи теорії ООП: клас, екземпляр класу, атрибут, ме-тод, конструктор класу; вміти користуватись цими елементами при складанні програм звикористанням об'єктно-орієнтованого підходу на мові Python в інтерактивному режимі IDLE

		Мітленко Д.А.			<i>ДТЗЕ.121РПЗ00ТЕХ1600ЛР</i>	Арк.
						6
Змін.	Арк.	№ докум.	Підпис	Дата		