

**Мета:** навчитись користуватись композиційним і модульним принципом при складанні програм з використанням об'єктно-орієнтованого підходу на мові Python в інтерактивному режимі IDLE

**Обладнання:** ОС Windows, Microsoft Office 2013, 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 ЛР19 для збереження наступних завдань

3 За допомогою команди File→NewFile (Ctrl+N) створив нові текстові документи з відповідними назвами ЛР\_19(No завдання) для розробки наступних кодів програми

4 Виконав індивідуальні завдання виданого варіанту (варіант 1)

4.1 На письмовому столі розміром 1 на 1 лежать аркуш паперу формату А4 і книжка формату А5. Розробіть на основі композиційного підходу програму обчислення площі, яку займають книжка і аркуш, та вільну площу поверхні столу.

Код програми:

```
class Book:
```

```
    def __init__(self,x,y):
```

```
        self.sq = x * y
```

```
class Sheet:
```

```
    def __init__(self,x,y):
```

```
        self.sq = x * y
```

```
class Table:
```

```
    def __init__(self,l,p):
```

```
        self.sq = l * p
```

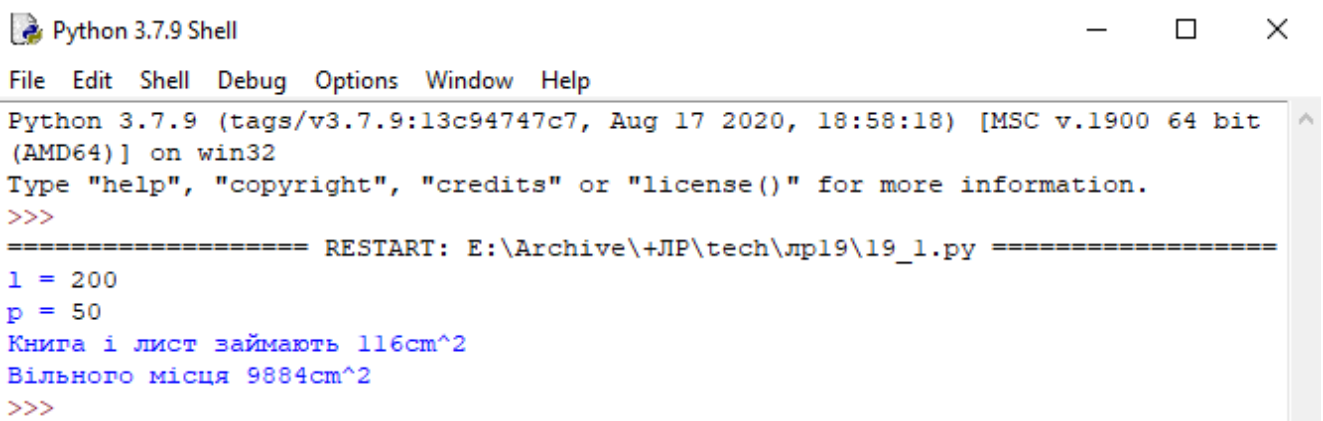
					<b>ДТЗЕ.121РПЗ00ТЕХ1900ЛР</b>				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.		Мітленко Д.А.			Композиційний і модульний принципи розробки програм			Лім.	Арк.
Перевір.		Саприкіна І.Г.							Аркушів
Реценз.									
Н. Контр.		Саприкіна І.Г.						1	8
Затверд.		Саприкіна І.Г.						Група ПЗ-19-1/9	

```

def objects(self,x1,y1,x2,y2):
    self.book = Book(x1,y1)
    self.sheet = Sheet(x2,y2)
def calculate(self):
    self.calculate = self.sq - (self.book.sq + self.sheet.sq)
def out(self):
    print('Книга і лист займають %ісм^2'%(self.book.sq + self.sheet.sq))
    print('Вільного місця %ісм^2'%self.calculate)
obj1 = Table(int(input('l = ')),int(input('p = ')))
obj1.objects(12,7,8,4)
obj1.calculate()
obj1.out()

```

Результат зображений на скріншоті 1



```

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\jlp19\19_1.py =====
l = 200
p = 50
Книга і лист займають 116см^2
Вільного місця 9884см^2
>>>

```

Скріншот 1

4.2 У порожній басейн розміром  $l$ ,  $p$ ,  $h$  одна помпа починає закачувати воду продуктивністю  $V_1$  м<sup>3</sup> за годину, й одночасно інша помпа починає викачувати воду з продуктивністю  $V_2$  м<sup>3</sup> за годину ( $V_1 > V_2$ ). Розробіть на основі композиційного підходу програму визначення об'єму води в басейні через 3 години та об'єму води, який ще може бути закачаний в басейн

Код програми:

```

class V1:
    def __init__(self,v):

```

```

        self.v = v
class V2:
    def __init__(self,v):
        self.v = v
class Pool:
    def __init__(self,l,p,h):
        self.v = l * p * h
    def objects(self,v1,v2):
        self.v1 = V1(v1)
        self.v2 = V2(v2)
    def calculate(self):
        self.calculate = self.v - (3 * (self.v1.v - self.v2.v))
    def out(self):
        print("Об'єм води: %iл"%self.calculate)
        print("Ще може бути закачано: %i"%abs(self.v - self.calculate))
obj1 = Pool(int(input('l = ')),int(input('p = ')),int(input('h = ')))
obj1.objects(170,120)
obj1.calculate()
obj1.out()

```

Результат зображено на скріншоті 2

```

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\лп19\19_2.py =====
l = 10
p = 20
h = 30
Об'єм води: 5850л
Ще може бути закачано: 150
>>>

```

Скріншот 2

4.3 Розробіть модуль обчислення середнього арифметичного значення заданого масиву цілих чисел розмірності 19 згенерованих у відповідному діапазоні (інтервал ввести з клавіатури в діалоговому режимі). Використайте цей модуль для обчислення середнього значення трьох масивів з різними значеннями елементів

Код головного файлу:

```
import module_19_3
obj1 = module_19_3.Module(int(input('from:')),int(input('to:')))
obj2 = module_19_3.Module(int(input('from:')),int(input('to:')))
obj3 = module_19_3.Module(int(input('from:')),int(input('to:')))
o1 = obj1.generate()
o2 = obj2.generate()
o3 = obj3.generate()
print("Середнє значення об'єкту 1:",obj1.avg())
print("Середнє значення об'єкту 2:",obj2.avg())
print("Середнє значення об'єкту 3:",obj3.avg())
```

Код файлу module\_19\_3.py:

```
from random import randint
class Module:
    def __init__(self,f,t):
        self.r = (f,t)
    def generate(self):
        self.generate = [randint(self.r[0],self.r[1]) for i in range(19)]
    def avg(self):
        return int(sum(self.generate)/len(self.generate))
```

Результат зображено на скріншоті 3

		Мітленко Д.А.			<b>ДТЗЕ.121РПЗ00ТЕХ1900ЛР</b>	Арк.
						4
Змін.	Арк.	№ докум.	Підпис	Дата		

```

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\лп19\19_3.py =====
from:17
to:200
from:20
to:50
from:1
to:2
Середнє значення об'єкту 1: 93
Середнє значення об'єкту 2: 36
Середнє значення об'єкту 3: 1
>>>

```

Скріншот 3

4.4 Розробіть модуль обчислення площ та периметрів будь-яких видів трикутників (рівносторонній, рівнобедрений, прямокутний). Використайте цей модуль для обчислення довільних декількох трикутників, вид яких вказують в режимі діалогу

Код головного файлу:

```

import module_19_4 as modu

print('Меню:\n 1.Рівносторонній\n 2.Рівнобедрений\n 3.Прямокутний\n 4.Вихід')

while True:
    inp = int(input('Виб> '))
    if inp == 1:
        obj = modu.TriangleWorker(int(input('a = ')),t=0)
        print('Периметр:',obj.perimeter())
        print('Площа:',obj.area(0))
    elif inp == 2:
        obj = modu.TriangleWorker(int(input('a = ')),int(input('b = ')),t=1)
        print('Периметр:',obj.perimeter())
        print('Площа:',obj.area(1))
    elif inp == 3:

```

```

obj = modu.TriangleWorker(int(input('a = ')),int(input('b = ')),int(input('c = ')),t=2)

print('Периметр:',obj.perimeter())

print('Площа:',obj.area(2))

elif inp == 4:

    break

```

Код файлу module\_19\_4.py:

```

from math import sqrt

```

```

class TriangleWorker:

```

```

    def __init__(self,a=1,b=1,c=1,t=0):

```

```

        if t == 0:

```

```

            self.size = (a,a,a)

```

```

        elif t == 1:

```

```

            self.size = (a,a,b)

```

```

        elif t == 2:

```

```

            self.size = (a,b,c)

```

```

    def perimeter(self):

```

```

        return sum(self.size)

```

```

    def area(self,t):

```

```

        side = self.size

```

```

        if t in [0,1,2]:

```

```

            return {

```

```

                0: round(side[0]/(2*sqrt(3))),

```

```

                1: round((side[1]/4)*sqrt(4 * side[0]**2 - side[1]**2)),

```

```

                2: round((side[0] * side[1]) / 2)

```

```

            }[t]

```

```

        else:

```

```

            return False

```

Результат зображений на скріншоті 4

		Мітленко Д.А.			ДТЗЕ.121РПЗ00ТЕХ1900ЛР	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		6

```

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\+ЛР\tech\лр19\19_4.py =====
Меню:
  1.Рівносторонній
  2.Рівнобедрений
  3.Прямокутний
  4.Вихід
Виб> 1
a = 3
Периметр: 9
Площа: 1
Виб> 2
a = 22
b = 10
Периметр: 54
Площа: 210
Виб> 3
a = 10
b = 9
c = 8
Периметр: 27
Площа: 45
Виб> 4
>>>

```

Скріншот 4

### Відповіді на контрольні запитання

1 В чому полягає композиційний підхід в програмуванні? Який клас називають класом контейнером?

Полягає він в наступному: є клас-контейнер, який включає в себе виклики інших класів. Контейнери - це будь-який об'єкт, який містить будь-яку кількість інших об'єктів.

2 Із якою метою застосовують композиційний підхід? Поясніть сутність композиційного підходу на прикладі з лабораторної роботи.

Композиція в програмуванні – це вбудовування інших об'єктів в об'єкт-контейнер і використання їх для реалізації методів класу-контейнера.

3 За допомогою яких інструкцій імпортуються модулі в програму? Яка різниця між інструкціями import та from?

За допомогою import та from. Import використовується для повного імпорту

		Мітленко Д.А.			<b>ДТЗЕ.121РПЗ00ТЕХ1900ЛР</b>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		7

модуля. From – для імпорту певних функцій

4 Яку структуру має інструкція import? У якому каталозі застосовується пошук модуля по замовчуванню?

import назва\_модуля. Пошук модуля виконується в каталозі, куди встановлено Python або в тому ж каталозі, де і файл, в який здійснюється імпорт

5 Як здійснюється імпортування модулів, що містять лише методи?

from назва\_модуля import назва\_методу,назва\_методу...

6 Із якою методи використовують засоби документування? За допомогою яких символів позначається текст документування?

“”” опис модуля “””

7 Для чого застосовується метод \_\_doc\_\_?

Щоб вивести документацію модуля.

**Висновок:** я навчився користуватись композиційним і модульним принципом при складанні програм з використанням об’єктно-орієнтованого підходу на мові Python в інтерактивному режимі IDLE

		Мітленко Д.А.			<i>ДТЗЕ.121РПЗ00ТЕХ1900ЛР</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		8