

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра «Електронних обчислювальних машин»



Звіт
з лабораторної роботи № 9
з дисципліни: «Кросплатформенні засоби програмування»
на тему: «Основи об'єктно-орієнтованого
програмування у Python»

Виконав:

студент групи КІ-306

Ширий Б. І.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

МЕТОДИЧНІ ВІДОМОСТІ РОБОТИ

МЕТА

Оволодіти навиками реалізації парадигм об'єктно-орієнтованого програмування використовуючи засоби мови Python.

ЗАВДАННЯ

№1

Написати та налагодити програму на мові Python згідно варіанту, де в мене базовий клас – це «Шлюпка на веслах», а похідний – це «Моторний човен».

Програма має задовольняти наступним вимогам:

- класи програми мають розміщуватися в окремих модулях в одному пакеті;
- точка входу в програму (main) має бути в окремому модулі
- мають бути реалізовані базовий і похідний класи предметної області згідно варіанту;
- програма має містити коментарі

№2

Для розробленої програми згенерувати документацію

№3

Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.

№4

Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації та завантажити його у ВНС.

№5

Дати відповідь на контрольні запитання.

ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

ВИХІДНИЙ КОД

Написав програму та навів її модулі на рисунках 2.1, 2.2 та 2.3.

Лістинг 2.1. Клас «Шлюпка на веслах».

```
# Модуль boat.py - базовий клас "Шлюпка на веслах"
class Rowboat:
    def __init__(self, length, color):
        self.length = length
        self.color = color

    def row(self):
        print(f"Рухаємося на веслах шлюпкою довжиною {self.length} метрів, кольору {self.color}.")

    def anchor(self):
        print("Закидаємо якор.")
```

Лістинг 2.2. Клас «Моторний човен».

```
# Модуль motorboat.py - похідний клас "Моторний човен"
from boat import Rowboat

class Motorboat(Rowboat):
    def __init__(self, length, color, engine_type):
        super().__init__(length, color)
        self.engine_type = engine_type

    def start_engine(self):
        print(f"Запускаємо мотор {self.engine_type}.")

    def move_fast(self):
        print("Рухаємося з використанням мотора з великою швидкістю.")
```

Лістинг 2.3. Точка входу в програму.

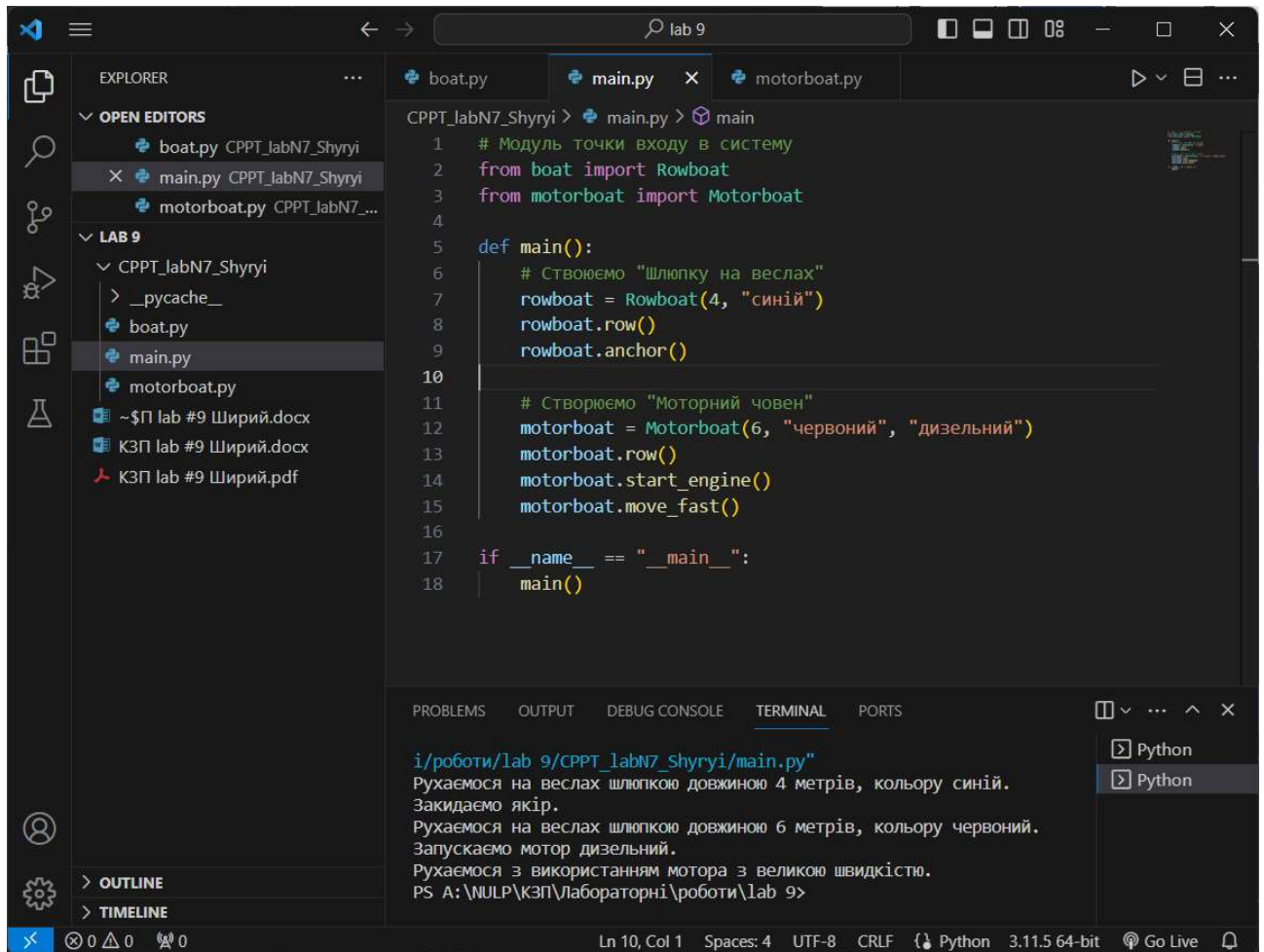
```
# Модуль точки входу в систему
from boat import Rowboat
from motorboat import Motorboat

def main():
    # Створюємо "Шлюпку на веслах"
    rowboat = Rowboat(4, "синій")
    rowboat.row()
    rowboat.anchor()
    # Створюємо "Моторний човен"
    motorboat = Motorboat(6, "червоний", "дизельний")
    motorboat.row()
    motorboat.start_engine()
    motorboat.move_fast()

if __name__ == "__main__":
    main()
```

РЕЗУЛЬТАТИ ВИКОНАННЯ

Вивід програми при запуску наведений на рисунку 2.1.



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left shows the project structure for 'CPPT_labN7_Shyryi', including files 'boat.py', 'main.py', and 'motorboat.py'. The main editor displays the code in 'main.py':

```
1 # Модуль точки входу в систему
2 from boat import Rowboat
3 from motorboat import Motorboat
4
5 def main():
6     # Створюємо "Шлюпку на веслах"
7     rowboat = Rowboat(4, "синій")
8     rowboat.row()
9     rowboat.anchor()
10
11     # Створюємо "Моторний човен"
12     motorboat = Motorboat(6, "червоний", "дизельний")
13     motorboat.row()
14     motorboat.start_engine()
15     motorboat.move_fast()
16
17 if __name__ == "__main__":
18     main()
```

The TERMINAL pane at the bottom shows the output of the program:

```
i/роботи/lab 9/CPPT_labN7_Shyryi/main.py
Рухаємося на веслах шлюпкою довжиною 4 метрів, кольору синій.
Закидаємо якор.
Рухаємося на веслах шлюпкою довжиною 6 метрів, кольору червоний.
Запускаємо мотор дизельний.
Рухаємося з використанням мотора з великою швидкістю.
PS A:\NULP\КЗП\лабораторні\роботи\lab 9>
```

Рисунок 2.1. Вивід програми.

ВІДПОВІДІ НА КОНТРОЛЬНІ ПИТАННЯ

ЩО ТАКЕ МОДУЛІ?

Модулі в Python - це файли, які містять функції, класи та змінні, які можна використовувати в інших програмах.

ЯК ІМПОРТУВАТИ МОДУЛЬ?

Щоб імпортувати модуль, використовуйте ключове слово `import`, приклад наведено у лістингу 2.4.

Лістинг 2.4.

```
import module_name
```

ЯК ОГЛОСИТИ КЛАС?

Для оголошення класу використовується ключове слово `class`, приклад наведено у лістингу 2.5.

Лістинг 2.5.

```
class MyClass:  
    # Тіло класу
```

ЩО МОЖЕ МІСТИТИСЯ У КЛАСІ?

У класі можуть міститися атрибути (змінні), методи (функції), конструктори та інші класи.

ЯК НАЗИВАЄТЬСЯ КОНСТРУКТОР КЛАСУ?

Конструктор класу називається `__init__`.

ЯК ЗДІЙСНИТИ СПАДКУВАННЯ?

Для здійснення спадкування в Python використовується наступний синтаксис, приклад наведено у лістингу 2.6.

Лістинг 2.6.

```
class ChildClass(ParentClass):  
    # Тіло похідного класу
```

ЯКІ ВИДИ СПАДКУВАННЯ ІСНУЮТЬ?

В Python існують одиночне спадкування (один клас успадковує від іншого) і багат шарове спадкування (клас успадковує від кількох інших класів).

ЯКІ НЕБЕЗПЕКИ Є ПРИ МНОЖИННОМУ СПАДКУВАННІ, ЯК ЇХ УНИКНУТИ?

Небезпеки при множинному спадкуванні включають конфлікти імен, надмірну складність, і можливу непередбачувану поведінку. Їх можна уникнути, використовуючи належне управління класами і методами.

ЩО ТАКЕ КЛАСИ-ДОМІШКИ?

Класи-домішки (mixin classes) - це спеціальні класи, які містять методи, які можна використовувати для розширення функціональності інших класів.

ЯКА РОЛЬ ФУНКЦІЇ SUPER() ПРИ СПАДКУВАННІ?

Функція `super()` використовується при спадкуванні для виклику методів батьківського класу в похідному класі.

ВИСНОВОК

Виконано лабораторну роботу з програмування на мові Python. Створено базовий клас "Шлюпка на веслах" і похідний клас "Моторний човен". Реалізовано програму, де класи розміщені в окремих модулях в одному пакеті, точка входу розміщена в окремому модулі, і програма містить коментарі для пояснення структури і функціональності коду.