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



**Звіт**

з лабораторної роботи №9

з дисципліни: “Кросплатформні засоби програмування”

на тему: «Основи об’єктно-орієнтованого програмування у Python»

**Виконала:**

Ст. гр. КІ-306

Покидько Юлія

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

**Львів – 2023**

**Мета:**оволодіти навиками реалізації парадигм об’єктно-орієнтованого програмування використовуючи засоби мови Python.

**Завдання (варіант № 15) :**

1. Написати та налагодити програму на мові Python згідно варіанту. 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. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.

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

4. Дати відповідь на контрольні запитання.

**Базовий клас ⎯ Корабель, похідний клас ⎯ Фрегат.**

**Хід роботи :**

**lab9.py**

*# lab9.py*from package1.Ship import Ship  
from package1.Frigate import Frigate  
  
*# Define the main function*def main():  
 *# Create Ship and Frigate objects* my\_ship = Ship("Nostromo", **190**, **29**)  
 my\_frigate = Frigate("Defender", **150**, **25**, "high")  
  
 *# Call methods to print information about these objects* print("Information about my ship:")  
 my\_ship.print\_info()  
  
 print("**\n**Information about my frigate:")  
 my\_frigate.print\_info()  
  
*# Run the main function when this script is executed*if \_\_name\_\_ == "\_\_main\_\_":  
 main()

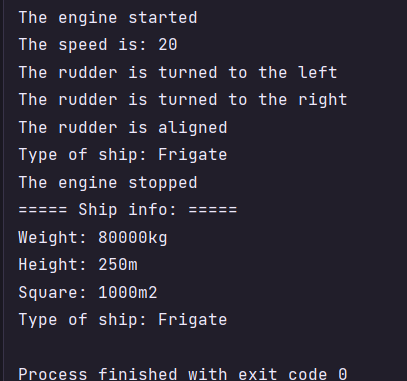
**Ship.py**

*# Ship.py  
  
# Define the Ship class*class Ship:  
 *# Initialize the ship with name, length, and max\_speed attributes* def \_\_init\_\_(self, name, length, max\_speed):  
 self.name = name  
 self.length = length  
 self.max\_speed = max\_speed  
  
 *# Method to simulate the ship sailing* def sail(self):  
 print(f"The **{**self.name**}** is sailing at **{**self.max\_speed**}** knots.")  
  
 *# Method to simulate the ship anchoring* def anchor(self):  
 print(f"The **{**self.name**}** is anchored.")  
  
 *# Method to simulate the ship sounding its horn* def sound\_horn(self):  
 print("Honk! Honk!")  
  
 *# Method to print information about the ship* def print\_info(self):  
 print(f"Ship Name: **{**self.name**}**")  
 print(f"Length: **{**self.length**}** meters")  
 print(f"Maximum Speed: **{**self.max\_speed**}** knots")  
  
*# Test the Ship class*if \_\_name\_\_ == "\_\_main\_\_":  
 my\_ship = Ship("Titanic", **200**, **30**)  
 my\_ship.print\_info()  
 my\_ship.sail()  
 my\_ship.anchor()  
 my\_ship.sound\_horn()

**Frigate.py**

*# Frigate.py*from package1.Ship import Ship  
  
*# Define the Frigate class, inheriting from Ship*class Frigate(Ship):  
 *# Initialize the frigate with name, length, max\_speed, and firepower attributes* def \_\_init\_\_(self, name, length, max\_speed, firepower):  
 super().\_\_init\_\_(name, length, max\_speed)  
 self.firepower = firepower  
  
 *# Method to simulate the frigate firing its cannons* def fire\_cannons(self):  
 print(f"The **{**self.name**}** is firing its cannons with **{**self.firepower**}** firepower!")  
  
 def print\_info(self):  
 super().print\_info()  
 print(f"Firepower: **{**self.firepower**}**")  
   
*# Test the Frigate class*if \_\_name\_\_ == "\_\_main\_\_":  
 my\_frigate = Frigate("Defender", **150**, **25**, "high")  
 my\_frigate.print\_info()  
 my\_frigate.sail()  
 my\_frigate.anchor()  
 my\_frigate.fire\_cannons()

**Вивід:**



**Висновок:**

Під час виконання даної лабораторної роботи я оволоділа навиками реалізації парадигм об’єктно-орієнтованого програмування використовуючи засоби мови Python.