Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра ЕОМ



**Звіт**

з лабораторної роботи №9

з дисципліни: “Кросплатформні засоби програмування”

на тему: “ФАЙЛИ ТА ВИКЛЮЧЕННЯ У PYTHON”

Варіант 2

Виконав: ст. гр. КІ-301

Гавриляк Д. В.

Прийняв:

Майдан М. В.

Львів – 2023

**Мета:**

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

**Завдання(Варіант 2)**

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

Варіант завдання: Базовий клас: Космічний корабель , похідний: Багаторазовий космічний корабель

**Код програми:**

Main.py

from warship import SpaceShip, Warship  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 # Створення об'єкта SpaceShip  
 my\_space\_ship = SpaceShip(brand="Galactica", material="Titanium", warp\_speed=7, capacity=1000, status\_engine=True)  
 # Виведення інформації про корабель  
 print("=== Initial Information ===")  
 my\_space\_ship.get\_info()  
 print("===========================")  
  
 # Зміна деяких параметрів  
 my\_space\_ship.set\_brand("Voyager")  
 my\_space\_ship.set\_warp\_speed(9)  
 my\_space\_ship.set\_engine\_status(False)  
  
 # Виведення оновленої інформації про корабель  
 print("\n=== Updated Information ===")  
 my\_space\_ship.get\_info()  
 print("===========================")  
 # Створення об'єкта Warship  
 my\_warship = Warship(brand="Dreadnought", material="Adamantium", warp\_speed=5, capacity=1500, status\_engine=True,  
 weapon\_power=500, shield\_strength=1000, missile\_count=20)  
  
 # Виведення інформації про військовий корабель  
 print("=== Warship Information ===")  
 my\_warship.get\_info()  
 print("============================")  
  
 # Зміна деяких параметрів  
 my\_warship.set\_warp\_speed(7)  
 my\_warship.set\_weapon\_power(800)  
 my\_warship.set\_shield\_strength(1200)  
 my\_warship.set\_missile\_count(25)  
  
 # Виведення оновленої інформації про військовий корабель  
 print("\n=== Updated Warship Information ===")  
 my\_warship.get\_info()  
 print("===============================")

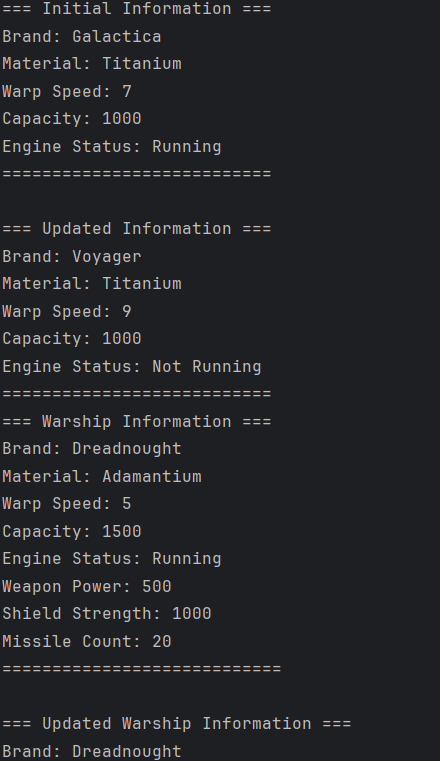
warship.py

from starship import SpaceShip  
  
  
class Warship(SpaceShip):  
 def \_\_init\_\_(self, brand, material, warp\_speed, capacity, status\_engine, weapon\_power, shield\_strength, missile\_count):  
 # Викликаємо конструктор батьківського класу SpaceShip  
 super().\_\_init\_\_(brand, material, warp\_speed, capacity, status\_engine)  
 self.\_weapon\_power = weapon\_power  
 self.\_shield\_strength = shield\_strength  
 self.\_missile\_count = missile\_count  
  
 # Додані методи для отримання значень щита та кількості ракет  
 def get\_shield\_strength(self):  
 return self.\_shield\_strength  
  
 def get\_missile\_count(self):  
 return self.\_missile\_count  
  
 # Додані методи для встановлення значень щита та кількості ракет  
 def set\_shield\_strength(self, shield\_strength):  
 self.\_shield\_strength = shield\_strength  
  
 def set\_missile\_count(self, missile\_count):  
 self.\_missile\_count = missile\_count  
  
 # Доданий метод для отримання потужності зброї  
  
 def get\_weapon\_power(self):  
 return self.\_weapon\_power  
  
 # Доданий метод для встановлення потужності зброї  
  
 def set\_weapon\_power(self, weapon\_power):  
 self.\_weapon\_power = weapon\_power  
  
 # Перевизначений метод для виведення інформації про військовий корабель  
 def get\_info(self):  
 # Викликаємо метод батьківського класу для отримання загальної інформації  
 super().get\_info()  
 print("Weapon Power:", self.\_weapon\_power)  
 print("Shield Strength:", self.\_shield\_strength)  
 print("Missile Count:", self.\_missile\_count)

starship.py

class SpaceShip:  
 def \_\_init\_\_(self, brand, material, warp\_speed, capacity, status\_engine):  
 self.\_brand = brand  
 self.\_material = material  
 self.\_warp\_speed = warp\_speed  
 self.\_capacity = capacity  
 self.\_status\_engine = status\_engine  
  
 # Getter methods  
 def get\_brand(self):  
 return self.\_brand  
  
 def get\_material(self):  
 return self.\_material  
  
 def get\_warp\_speed(self):  
 return self.\_warp\_speed  
  
 def get\_capacity(self):  
 return self.\_capacity  
  
 def is\_engine\_running(self):  
 return self.\_status\_engine  
  
 # Setter methods  
 def set\_brand(self, brand):  
 self.\_brand = brand  
  
 def set\_material(self, material):  
 self.\_material = material  
  
 def set\_warp\_speed(self, warp\_speed):  
 self.\_warp\_speed = warp\_speed  
  
 def set\_capacity(self, capacity):  
 self.\_capacity = capacity  
  
 def set\_engine\_status(self, status\_engine):  
 self.\_status\_engine = status\_engine  
  
 # Method to get information about the spaceship  
 def get\_info(self):  
 print("Brand:", self.\_brand)  
 print("Material:", self.\_material)  
 print("Warp Speed:", self.\_warp\_speed)  
 print("Capacity:", self.\_capacity)  
 print("Engine Status:", "Running" if self.\_status\_engine else "Not Running")

**Результат:**



**Висновок:**

Я ознайомився з основними принципами мови програмування python та оволодів навиками застосування них. Закріпив теорію на практиці.