Івано-Франківський національний технічний університет нафти і газу

Кафедра комп’ютерних систем і мереж

**ЛАБОРАТОРНА РОБОТА №8**

з дисципліни Практикум Python

Тема: Розробка програм з ієрархією класів. Організація

класів з використанням успадкування в Python

Виконав:

Ст. гр. КІ-23-2 (Боднар Р.В.)

Перевірив: (Кропивницький Д. Р.)

м. Івано-Франківськ

2025

**Мета роботи:** Ознайомитися з особливостями реалізації наслідування атрибутів класу в ООП на мові Python

**Варіант 1**

**Завдання:**

Створити ієрархію класів у відповідності з вказаною предметною областю. Передбачити можливість роботи з довільним числом записів, а також реалізувати: конструктори; функції виведення інформації на екран; функції пошуку потрібної інформації за конкретною ознакою. При розробці програми слід здійснити захищення даних для ізоляції елементів-даних класу від підпрограм, в яких цей клас використовується.

1. Створити базовий клас "Транспортний засіб". На його основі реалізувати класи "Літак", "Автомобіль" та "Корабель". Класи повинні мати можливість задавати та отримувати параметри засобів пересування (вартість, швидкість, рік випуску тощо) задати за допомогою полів. Для літака повинна бути визначена висота, для літака та корабля – кількість пасажирів, для корабля – порт приписки.

**Хід роботи**

**Код розроблених класів та їх використання:**

class Vehicle:

    def \_\_init\_\_(self, make, model, year, price, speed):

        self.\_\_make = make

        self.\_\_model = model

        self.\_\_year = year

        self.\_\_price = price

        self.\_\_speed = speed

    def get\_make(self):

        return self.\_\_make

    def get\_model(self):

        return self.\_\_model

    def get\_year(self):

        return self.\_\_year

    def get\_price(self):

        return self.\_\_price

    def get\_speed(self):

        return self.\_\_speed

    def set\_price(self, price):

        self.\_\_price = price

    def set\_speed(self, speed):

        self.\_\_speed = speed

    def display\_info(self):

        return (f"Марка: {self.\_\_make}, Модель: {self.\_\_model}, "

                f"Рік: {self.\_\_year}, Ціна: {self.\_\_price}$, "

                f"Швидкість: {self.\_\_speed} км/год")

class Airplane(Vehicle):

    def \_\_init\_\_(self, make, model, year, price, speed, altitude, passengers):

        super().\_\_init\_\_(make, model, year, price, speed)

        self.\_\_altitude = altitude

        self.\_\_passengers = passengers

    def display\_info(self):

        return (super().display\_info() +

                f", Висота: {self.\_\_altitude} м, Пасажирів: {self.\_\_passengers}")

class Car(Vehicle):

    def \_\_init\_\_(self, make, model, year, price, speed, fuel\_type):

        super().\_\_init\_\_(make, model, year, price, speed)

        self.\_\_fuel\_type = fuel\_type

    def display\_info(self):

        return (super().display\_info() +

                f", Тип палива: {self.\_\_fuel\_type}")

class Ship(Vehicle):

    def \_\_init\_\_(self, make, model, year, price, speed, passengers, port):

        super().\_\_init\_\_(make, model, year, price, speed)

        self.\_\_passengers = passengers

        self.\_\_port = port

    def display\_info(self):

        return (super().display\_info() +

                f", Пасажирів: {self.\_\_passengers}, Порт приписки: {self.\_\_port}")

vehicles = [

    Airplane("Boeing", "747", 2015, 150\_000\_000, 900, 10000, 416),

    Car("Tesla", "Model S", 2022, 80\_000, 250, "Електро"),

    Ship("Royal Caribbean", "Harmony", 2018, 1\_000\_000\_000, 45, 6780, "Маямі")

]

for v in vehicles:

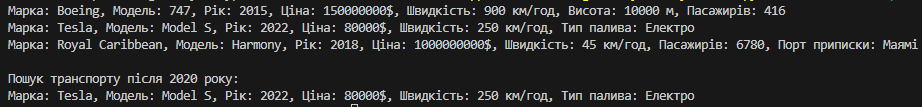
    print(v.display\_info())

print("\nПошук транспорту після 2020 року:")

for v in vehicles:

    if v.get\_year() >= 2020:

        print(v.display\_info())

****

**Рисунок 1 - Результат виконання завдання**

**Висновок:** У ході виконання лабораторної роботи було створено ієрархію класів із використанням механізму успадкування в мові Python. Реалізовано базовий клас “Транспортний засіб” та похідні класи ”Літак”, “Автомобіль” і “Корабель”, для яких передбачено специфічні властивості. Забезпечено інкапсуляцію даних шляхом використання приватних атрибутів та доступу до них через геттери й сеттери. Створено конструктори, функції для відображення інформації та пошуку записів за заданими критеріями. Також реалізовано можливість роботи з довільною кількістю об’єктів шляхом використання списків.