Міністерство освіти і науки України

Національний університет “Львівська політехніка ”

Кафедра ЕОМ



**ЗВІТ**

до лабораторної роботи №9

З дисципліни: «Кросплатформні засоби програмування»

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

Варіант - 7

Виконала:

ст. гр. КІ-305

Дзера А. Р.

Прийняв:

доц. каф. ЕОМ

Іванов Ю. С.

Львів – 2023

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

**Завдання**

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 ⎯ комп’ютер, похідний клас ⎯ ноутбук.

**Код програми**

**Файл Main.py**

from Computer import Computer  
from Laptop import Laptop  
  
def main():  
 computer = Computer("HP", "Intel Core i5", 8)  
 laptop1 = Laptop("Dell", "Intel Core i7", 16, 10)  
 laptop2 = Laptop("Lenovo", "AMD Ryzen 5", 12, 8)  
  
 print("Computer Info:")  
 computer.display\_info()  
  
 print("\nLaptop 1 Info:")  
 laptop1.display\_info()  
  
 print("\nLaptop 2 Info:")  
 laptop2.display\_info()  
  
 print("\nUpdating Laptop 1 specs...")  
 laptop1.update\_specs(processor="Intel Core i9", ram=32, battery\_life=12)  
 laptop1.display\_info()  
  
 print("\nComparing Laptop 1 and Laptop 2 specs...")  
 if laptop1.compare\_specs(laptop2):  
 print("Laptop 1 and Laptop 2 have the same processor and RAM.")  
 else:  
 print("Laptop 1 and Laptop 2 have different processor or RAM.")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

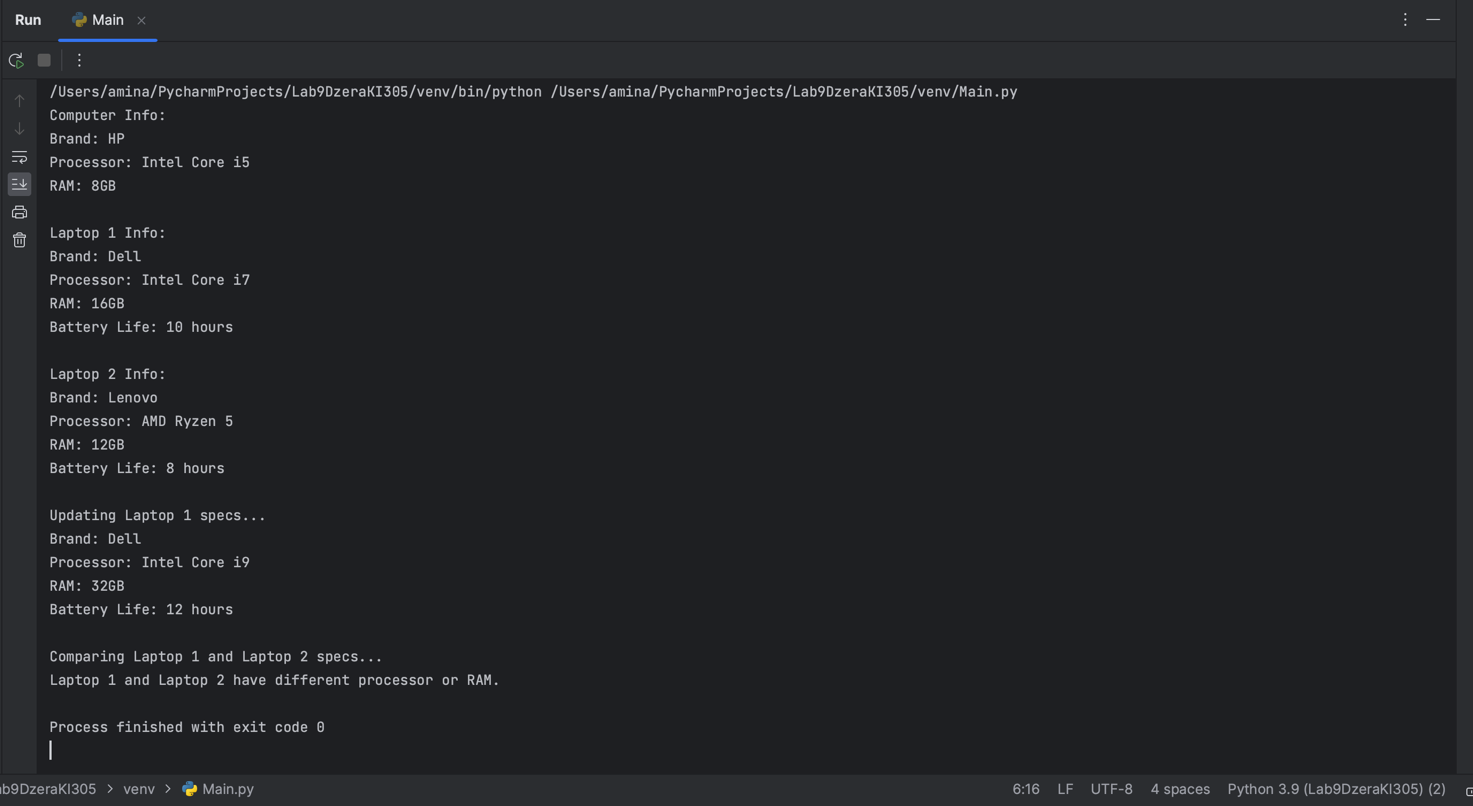
**Файл Computer.py**

class Computer:  
 def \_\_init\_\_(self, brand, processor, ram):  
 self.brand = brand  
 self.processor = processor  
 self.ram = ram  
  
 def display\_info(self):  
 print(f"Brand: {self.brand}")  
 print(f"Processor: {self.processor}")  
 print(f"RAM: {self.ram}GB")  
  
 def update\_specs(self, processor=None, ram=None):  
 if processor:  
 self.processor = processor  
 if ram:  
 self.ram = ram

**Файл Laptop.py**

from Computer import Computer  
  
class Laptop(Computer):  
 def \_\_init\_\_(self, brand, processor, ram, battery\_life):  
 super().\_\_init\_\_(brand, processor, ram)  
 self.battery\_life = battery\_life  
  
 def display\_info(self):  
 super().display\_info()  
 print(f"Battery Life: {self.battery\_life} hours")  
  
 def update\_specs(self, processor=None, ram=None, battery\_life=None):  
 super().update\_specs(processor, ram)  
 if battery\_life:  
 self.battery\_life = battery\_life  
  
 def compare\_specs(self, other\_laptop):  
 if self.processor == other\_laptop.processor and self.ram == other\_laptop.ram:  
 return True  
 else:  
 return False

**Результати роботи програми**



**Контрольні питання**

1. Що таке модулі?

*Модулем* у Python називається файл з розширенням \*.py. Ці файли можуть містити звичайні скрипти, змінні, функції, класи і їх комбінації.

1. Як імпортувати модуль?

Для цього використовуються два оператори – import та from-import.

1. Як оголосити клас?

Клас оголошується за допомогою ключового слова class після якого йде назва класу.

1. Що може міститися у класі?

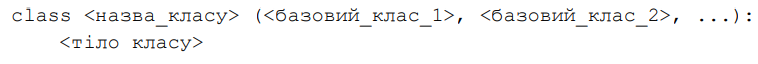
* дані, які належать класу (статичні дані-члени класу);
* дані, які належать об’єкту класу;
* методи, які належать класу (статична методи);
* методи, які належать об’єкту класу.

1. Як називається конструктор класу?

Роль конструктора відіграє метод \_\_init\_\_(self).

1. Як здійснити спадкування?

Для реалізації спадкування класи, які слід успадкувати вказуються у круглих дужках через кому після назви класу, який оголошується:

****

1. Які види спадкування існують?

* Одинарне спадкування
* Багатократне спадкування
* Інтерфейси
* Абстрактні класи
* Множинне спадкування
* Заміщення

1. Які небезпеки є при множинному спадкуванні, як їх уникнути?

* Амбігвітність – уникнути можна забезпечивши чітку структуру спадкування
* Складність ієрархій класів – уникнути можна дотримуючись принципу “розділення відповідальностей”
* Залежність від порядку – уникнути можна спроектувавши класи так, вони не залежали від порядку успадкування
* Дублювання коду – уникнути можна використовуючи абстрактні класи або інтерфейси для виділення спільного функціоналу та розділення його між класами
* Складність відлагодження та обслуговування – уникнути можна називаючи класи та методи відповідно до їх функції

1. Що таке класи-домішки?

*Домішки або Mixin* – це шаблон проектування, в якому деякий метод базового класу використовує метод, який не визначається у цьому класі. Цей метод призначений для реалізації іншим базовим класом. Клас-домішка або mixin class – це клас, який використовується у цьому шаблоні, надаючи функціональні можливості (методи), але не призначений для самостійного використання у вигляді об’єктів класу. В ідеальному випадку класи-домішки не мають власної ієрархії спадкування і не мають полів, а мають лише методи.

1. Яка роль функції super() при спадкуванні?

Функція *super()* грає важливу роль при спадкуванні в об'єктно-орієнтованому програмуванні. Вона дозволяє виконувати методи батьківського класу в контексті підкласу.

**Висновок**

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