

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 №13

Виконав:
ст. гр. КІ-203
Панченко Д. В.

Прийняв:
доцент кафедри ЕОМ
Іванов Ю. С.

Львів 2023

Мета: 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: 13: Базовий клас – Телефон, похідний – Мобільний телефон.

Вихідний код програми:

Файл Phone.py:

```
# -*- coding: cp1251 -*-

# Базовий клас телефон
class Phone:

    # Конструктор
    def __init__(self, brand, model):
        self.brand = brand
        self.model = model

    # Метод "здійснити дзвінок"
    def make_call(self, number):
        print(f"{self.brand} {self.model} здійснює дзвінок за номером {number}\n")
```

Файл MobilePhone.py:

```
# -*- coding: cp1251 -*-

from Phone import Phone

# Похідний клас мобільний телефон
class MobilePhone(Phone):

    # Конструктор
    def __init__(self, brand, model, os):
        super().__init__(brand, model)
        self.os = os

    # Метод "завантажити застосунок"
    def install_app(self, app_name):
        print(f"Завантаження {app_name} на пристрій {self.brand} {self.model}\n")
```

Файл Main.py:

```
# -*- coding: cp1251 -*-

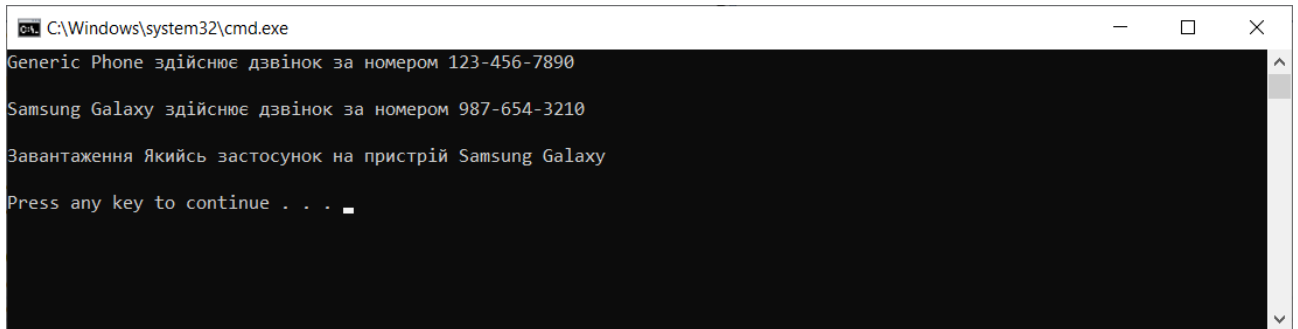
from Phone import Phone
from MobilePhone import MobilePhone

# Точка входу в програму main
if __name__ == "__main__":

    # Створення об'єкту телефон
    phone = Phone("Generic", "Phone")
    phone.make_call("123-456-7890")

    # Створення об'єкту мобільний телефон
    mobile_phone = MobilePhone("Samsung", "Galaxy", "Android")
    mobile_phone.make_call("987-654-3210")
    mobile_phone.install_app("Якийсь застосунок")
```

Результат виконання програми:



Відповіді на контрольні запитання:

1. Що таке модулі?

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

2. Як імпортувати модуль?

Оператор import дозволяє імпортувати модуль повністю, та доступатися до нього через назву модуля. Вона може бути вказана у будь-якому місці програми перед звертанням до елементів, які у ній містяться, але зазвичай її вказують на початку модуля.

3. Як оголосити клас?

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

4. Що може міститися у класі?

Клас може містити:

- 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.

5. Як називається конструктор класу?

Роль конструктора відіграє метод `__init__(self, <параметри>)`.

6. Як здійснити спадкування?

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

```
class <назва_класу> (<базовий_клас_1>, <базовий_клас_2>, ...):  
    <тіло класу>
```

7. Які види спадкування існують?

Python дозволяє реалізовувати як одинарне так і множинне спадкування.

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

При множинному спадкуванні можуть виникати конфлікти імен, що призводять до невизначеності викликів методів. Для уникнення цього, рекомендується ретельно планувати ієрархію класів, використовувати зрозумілі імена та, за можливістю, обирати композицію замість множинного спадкування.

9. Що таке класи-домішки?

Клас-домішка або `mix-in class` – це клас, який використовується у цьому шаблоні, надаючи функціональні можливості (методи), але не призначений для самостійного використання у вигляді об'єктів класу.

10. Яка роль функції `super()` при спадкуванні?

Для виклику конструктора базового класу у тілі конструктора похідного класу слід викликати функцію `super()`, який повертає проксі об'єкт базового класу, та явно викликати з-під нього конструктор базового класу. Також функцію `super()` слід використовувати для звертання до інших членів базового класу.

Висновок: Я оволодів навиками реалізації парадигм об'єктно-орієнтованого програмування використовуючи засоби мови Python.