

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»

Виконав:

студент групи КІ-307

Возний А. О.

Перевірив:

доцент кафедри 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: $y = \cos(x)/\sin(x)$

Лістинг програми:

Клас Cat

```
class Cat:
    def __init__(self, name, age):
        """
        Конструктор класу Cat.
        """
        self.name = name
        self.age = age

    def make_sound(self):
        """
        Метод, який виводить звук, що видає кіт.
        """
        print("Meow!")

    def info(self):
        """
        Метод, який виводить інформацію про кота.
        """
        print(f"Name: {self.name}, Age: {self.age}")
```

Клас ExperimentalCat

```
from Cat import Cat

class ExperimentalCat(Cat):
    def __init__(self, name, age, experiment):
        """
        Конструктор класу ExperimentalCat.
        """
```

```

        super().__init__(name, age)
        self.experiment = experiment

    def conduct_experiment(self):
        """
        Метод, який виводить результати експерименту.
        """
        print(f"Experiment: {self.experiment} - Successful!")

    def info(self):
        """
        Перевизначений метод, який виводить інформацію про експериментального
        кота.
        """
        print(f"Name: {self.name}, Age: {self.age}, Experiment:
{self.experiment}")

```

Головний модуль

```

from ExperimentalCat import *

def main():
    # Створення базового кота
    basic_cat = Cat(name="Whiskers", age=3)
    print("Basic Cat:")
    basic_cat.info()
    basic_cat.make_sound()

    print("\n-----\n")

    # Створення експериментального кота
    experimental_cat = ExperimentalCat(name="Fluffy", age=2,
experiment="Teleportation")
    print("Experimental Cat:")
    experimental_cat.info()
    experimental_cat.make_sound()
    experimental_cat.conduct_experiment()

if __name__ == "__main__":
    main()

```

Результат виконання програми:

```

Basic Cat:
Name: Whiskers, Age: 3
Meow!

-----

Experimental Cat:
Name: Fluffy, Age: 2, Experiment: Teleportation
Meow!
Experiment: Teleportation - Successful!

Process finished with exit code 0

```

Рис. 1. Результат виконання програми

Відповіді на контрольні питання:

1. Що таке модулі?

Модулі - це файли, які містять код для використання в інших програмах.

2. Як імпортувати модуль?

Імпорт модуля здійснюється за допомогою ключового слова "import" та імені модуля.

3. Як оголосити клас?

Клас оголошується ключовим словом "class", за яким слідує ім'я класу.

4. Що може міститися у класі?

У класі можуть міститися атрибути (змінні) та методи (функції).

5. Як називається конструктор класу?

Конструктор класу називається "init".

6. Як здійснити спадкування?

Спадкування здійснюється за допомогою наслідування від іншого класу.

7. Які види спадкування існують?

Існують одинарне та багат шарове спадкування.

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

При множинному спадкуванні можуть виникати конфлікти та нерозбіжності, які слід уникнути шляхом коректної організації класів.

9. Що таке класи-домішки?

Класи-домішки - це класи, які містять методи, які можна використовувати у інших класах.

10. Яка роль функції super() при спадкуванні?

Функція super() використовується для виклику методів батьківського класу при спадкуванні.

Висновок: під час виконання цієї лабораторної роботи я оволодів навиками реалізації парадигм об'єктно-орієнтованого програмування використовуючи засоби мови Python.