Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



Звіт

з лабораторної роботи № 9

з дисципліни: «Кросплатформенні засоби програмування»

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

**Виконав:**

студент групи КІ-307

Трофименко Є.Е.

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2023

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

**Завдання (варіант № 22)**

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.

**Базовий клас:** *Автомат.*

**Похідний клас:** *Штурмова гвинтівка.*

**Вихідний код програми**

Модуль automate.py:

#Базовий клас Автомат

class Automate:

#Конструктор класу Автомат

def \_\_init\_\_(self, model, caliber):

self.model = model

self.caliber = caliber

#Метод класу Автомат

def fire(self):

print(f"Firing {self.model}with ammo {self.caliber}")

Модуль assault\_rifle.py:

from automate import Automate

#Похідний клас Штурмова гвинтівка

class AssaultRifle(Automate):

#Конструктор класу Штурмова гвинтівка

def \_\_init\_\_(self, model, caliber, magazine\_capacity):

super().\_\_init\_\_(model, caliber)

self.magazine\_capacity = magazine\_capacity

#Метод класу Штурмова гвинтівка

def reload(self):

print(f"Reloading {self.model} with {self.magazine\_capacity} rounds")

Модуль main.py:

from automate import Automate

from assault\_rifle import AssaultRifle

#Точка входу в програму

def main():

automate = Automate("M4", "5.56mm")

automate.fire()

assault\_rifle = AssaultRifle("AK-47", "7.62mm", 30)

assault\_rifle.fire()

assault\_rifle.reload()

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Результат виконання програми**

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

**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.*

1. Як оголосити клас?

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

1. 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.*

1. Як називається конструктор класу?

*Роль конструктора відіграє метод \_\_init\_\_(self, <параметри>).*

1. Як здійснити спадкування?

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

1. Які види спадкування існують?

*Python дозволяє реалізовувати як одинарне так і множинне спадкування.*

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

*У звязку з тим, що конструктор (метод \_\_init()\_\_) у Python не викликає автоматично конструкторів базового класу, то їх слід викликати самостійно.*

*Якщо базовий клас не містить полів, які слід ініціалізовувати, то виклик конструктора цього базового класу можна пропустити, проте це не рекомендується робити.*

*Якщо методи \_\_init()\_\_ різних класів приймають різні параметри, то:*

*1. Конструктори усіх класів необхідно реалізувати таким чином, щоб вони додатково приймали параметром словник параметрів у форматі ключ-значення (\*\*kwargs). У цей словник попадатимуть усі параметри, які передані конструктору, але не згадані явно перед \*\*kwargs.*

*2. Викликати конструктори базових класів для усіх класів в ієрархії. Це необхідно для коректної роботи механізму MRO (розглядається далі).*

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.*

1. 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*

На данній лабораторній роботі я оволодів навиками реалізації парадигм об’єктно-орієнтованого програмування використовуючи засоби мови Python. Створив модулі, що реалізують предметну область, згідно індивідуального завдання.