Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра ЕОМ



**Звіт**

З лабораторної роботи №9

Варіант№8

З дисципліни: «Кросплатформні засоби програмування»

На тему: «ОСНОВИ ОБ’ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ У PYTHON»

Виконав: ст. гр. КІ-304

Кісіль В. І.

Прийняв:

Олексів М.В.

Львів 2023

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

**Теоретичні відомості**

**Модулі**

Модулем у Python називається файл з розширенням \*.py. Ці файли можуть містити

звичайні скрипти, змінні, функції, класи і їх комбінації. Python дозволяє структурувати

код програм у різні модулі та доступатися до класів, функцій і змінних, які у них

знаходяться з інших модулів. Для цього використовуються два оператори – import та

from-import.

**Оператор import**

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. Для звертання до елементів модуля треба вказати назву модуля і після крапки вказати до якого елементу ви хочете звернутися.

Синтаксис:

import назва\_модуля

назва\_модуля.елемент\_модуля

Приклад:

Файл my\_print\_module.py

def hello\_print():

print(“Hello”)

Файл printer.py

import my\_print\_module

my\_print\_module.hello\_print ()

Для зручності оригінальну назву модуля можна змінити на псевдонім і доступатися

до елементів модуля за допомогою псевдоніма.

Синтаксис:

import оригінальна\_назва\_модуля as псевдонім

Приклад:

import my\_print\_module as my\_printer

**Класи**

Клас оголошується за допомогою ключового слова class після якого йде назва

класу. 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.

Члени класу є лише публічні, проте Python забезпечує механізми, які дозволяють організувуати області видимості близькі за своєю суттю до protected і private. Це робиться шляхом використання нижнього підкреслення у назві членів класу. Одинарне нижнє підкреслення перед назвою члену класу робить за своїми властивостями схожим на захищений член класу, а подвійне – схожим на приватний член класу. Всі члени класу, що йому належать мають відступ у розмірі одного табулятора, або 4-ох пробілів від початку оголошення класу.

Статичні члени-дані класу оголошуються в класі як назва змінної і її початкове значення.

Дані, які належать об’єкту класу оголошуються в конструкторі з використанням ключового слова self, яке є посиланням на об’єкт класу:

self.<назва\_змінної> = <початкове значення>

Статичні методи класу оголошуються в класі за правилами оголошення функцій.

Не статичні методи оголошуються в класі за правилами оголошення функцій, перший

параметр якої є обов’язково self:

def <назва\_методу>(self, <параметри>):

тіло методу

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

Доступ до статичних членів класу відбуваєтсья за допомогою назви класу:

<назва класу>.<назва статичного члену класу>.

Доступ до не статичних членів класу відбуваєтсья за допомогою назви об’єкту:

<назва об’єкту>.<назва не статичного члену класу>.

Параметри self передавати у метод при виклику не потрібно. Він передається

неявно як і у інших мовах програмування.

Якщо клас не містить ніяких членів, то він має мати замість них ключову слово

pass.

class <назва класу>:

pass

Видалення окпремих членів класу або об’єкту загалом здійснюється за допомогою

оператора del:

del <назва\_об’єкту.назва\_члену\_даних> # видалення

властивості

del <назва\_об’єкту> # видалення об’єкту

**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 8. Фотоапарат**

**Похідний клас 8. Цифрова відеокамера**

**Код програми:**

Button.py

class Button:

def \_\_init\_\_(self, res=1000000):

self.\_\_btnResource = res

def clickButton(self):

self.\_\_btnResource = self.\_\_btnResource - 1

def getButtonResource(self):

return self.\_\_btnResource

Storage.py

class Storage:

def \_\_init\_\_(self, standart\_photoNumber = 0, standart\_maxPhotoNumber = 100000):

self.\_\_photoNumber = standart\_photoNumber

self.\_\_maxPhotoNumber = standart\_maxPhotoNumber

def deletePhoto(self):

if self.\_\_photoNumber == 0:

print("Error. Storage is empty")

else:

self.\_\_photoNumber = self.\_\_photoNumber - 1

def makePhoto(self):

if self.\_\_photoNumber == self.\_\_maxPhotoNumber:

print("Error. Storage is full")

else:

self.\_\_photoNumber = self.\_\_photoNumber + 1

def getPhotoNumber (self):

return self.\_\_photoNumber

def getMaxPhotoNumber (self):

return self.\_\_maxPhotoNumber

FocusControl.py

class FocusControl:

def \_\_init\_\_(self, standart\_length = 15, standart\_minlength = 1, standart\_maxlength = 30 ):

self.\_\_length = standart\_length

self.\_\_minlength = standart\_minlength

self.\_\_maxlength = standart\_maxlength

def scrollUp (self):

if self.\_\_length==self.\_\_maxlength :

print("Error. Length is already max")

else:

self.\_\_length = self.length+1

def scrollDown (self):

if self.length == self.\_\_minlength:

print("Error. Length is already min")

else:

self.\_\_length=self.\_\_length+1

def getFocusDistance(self):

return self.\_\_length

def getMinFocusDistance(self):

return self.\_\_minlength

def getMaxFocusDistance(self):

return self.\_\_maxlength

Camera.py

from Button import Button

from Storage import Storage

from FocusControl import FocusControl

class Camera:

def \_\_init\_\_(self, res = 1000):

self.\_\_button = Button(res)

self.\_\_storage = Storage()

self.\_\_focusDistance = FocusControl()

def makePhoto (self, quantity = 1):

count = 0

while count < quantity:

self.\_\_button.clickButton()

self.\_\_storage.makePhoto()

count = count + 1

def deletePhoto (self, quantity = 1):

count = 0

while count < quantity:

self.\_\_storage.deletePhoto()

count = count + 1

def viewStorage (self):

print(f"{self.\_\_storage.getPhotoNumber()} memori units are currently occupied in storage")

def distanceUp (self):

self.\_\_focusDistance.scrollUp()

def distanceDown (self):

self.\_\_focusDistance.scrollDown()

def viewDistance (self):

print(f"{self.\_\_focusDistance.getFocusDistance()} is distance now ")

def viewFocusParameters(self):

print(f"Minimal distance is {self.\_\_focusDistance.getMinFocusDistance()}")

print(f"Maximal distance is {self.\_\_focusDistance.getMaxFocusDistance()}")

print(f"Now distance is {self.\_\_focusDistance.getFocusDistance()}")

VideoCamera.py

from Camera import Camera

class VideoCamera (Camera):

def \_\_init\_\_(self,weight = 1.7, res = 1000000):

super().\_\_init\_\_(res)

self.\_\_weight = weight

def getWeight(self):

return self.\_\_weight

def makeVideo(self, time = 1):

count = 0

while count < time:

self.makePhoto(30)

count = count + 1

def deleteVideo(self,time = 1):

count = 0

while count < time:

self.deletePhoto(30)

count = count + 1

Main.py

from Camera import Camera

from VideoCamera import VideoCamera

my\_camera = Camera()

my\_camera.viewDistance()

my\_camera.viewFocusParameters()

my\_camera.makePhoto()

my\_camera.viewStorage()

my\_video\_camera = VideoCamera()

my\_video\_camera.makeVideo()

my\_video\_camera.viewStorage()

my\_video\_camera.makePhoto(3)

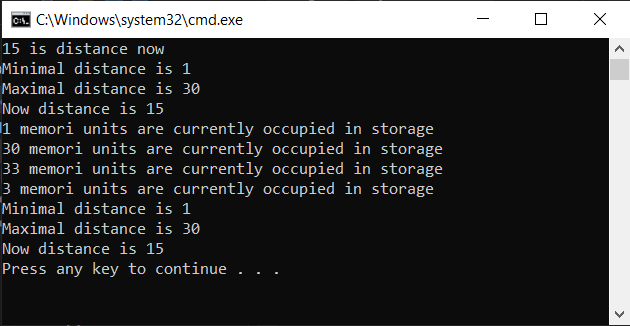
my\_video\_camera.viewStorage()

my\_video\_camera.deleteVideo()

my\_video\_camera.viewStorage()

my\_video\_camera.viewFocusParameters()

**Результат роботи програми:**



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