Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



Звіт

з лабораторної роботи № 8

з дисципліни: «Кросплатформенні засоби програмування»

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

**Виконав:**

студент групи *КІ-307*

*Сімчук Святослав*

**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 18: Сканер

Файл scanner.py

class Scanner:  
 def \_\_init\_\_(self, model, is\_color):  
 self.model = model  
 self.is\_color = is\_color  
 self.is\_turned\_on = False  
  
 def scan(self, document):  
 if self.is\_turned\_on:  
 print(f"{self.model} scanning {document} in {'color' if self.is\_color else 'black and white'}.")  
 else:  
 print(f"{self.model} is turned off. Cannot scan.")  
  
 def is\_duplex(self):  
 return False  
  
 def dpi(self):  
 return 300 if self.is\_color else 600  
  
 def turn\_on(self):  
 self.is\_turned\_on = True  
 print(f"{self.model} is turned on.")  
  
 def turn\_off(self):  
 self.is\_turned\_on = False  
 print(f"{self.model} is turned off.")

Файл copier.py

from scanner import Scanner  
  
  
class Copier(Scanner):  
 def \_\_init\_\_(self, model, is\_color, is\_duplex):  
 super().\_\_init\_\_(model, is\_color)  
 self.is\_duplex\_mode = is\_duplex  
  
 def copy(self, document):  
 if self.is\_turned\_on:  
 print(f"{self.model} copying {document} in {'color' if self.is\_color else 'black and white'}.")  
 if self.is\_duplex\_mode:  
 print("Duplex copying is enabled.")  
 else:  
 print("Single-sided copying is used.")  
 else:  
 print(f"{self.model} is turned off. Cannot copy.")  
  
 def is\_duplex(self):  
 return self.is\_duplex\_mode  
  
 def copy\_content\_from\_files(self, source\_files, destination\_file):  
 if self.is\_turned\_on:  
 with open(destination\_file, 'w') as dest:  
 for source\_file in source\_files:  
 with open(source\_file, 'r') as source:  
 dest.write(source.read())  
 print(f"{self.model} copied content from multiple files to {destination\_file}.")  
 else:  
 print(f"{self.model} is turned off. Cannot copy content.")

Файл main.py

from scanner import Scanner  
from copier import Copier  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 scanner = Scanner("OfficeJet Pro 8710", is\_color=True)  
 copier = Copier("Xerox WorkCentre 6515", is\_color=True, is\_duplex=True)  
  
 scanner.turn\_on()  
 scanner.scan("Document 1")  
 print(f"Scanner DPI: {scanner.dpi()}")  
 scanner.turn\_off()  
  
 copier.turn\_on()  
 copier.copy("Document 2")  
 print(f"Copier DPI: {copier.dpi()}")  
 print(f"Copier isDuplex: {copier.is\_duplex()}")  
 copier.turn\_off()  
  
 source\_files = ["file1.txt", "file2.txt"]  
 destination\_file = "combined\_file.txt"  
 copier.turn\_on()  
 copier.copy\_content\_from\_files(source\_files, destination\_file)  
 copier.turn\_off()

Результати:

Файл file1.txt

Hello

Файл file2.txt

World

Файл combined\_file.txt

Hello World

Висновок:

Під час виконання лабораторної роботи №9 оволодів навиками реалізації парадигм об’єктно-орієнтованого програмування використовуючи засоби мови Python.