Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Звіт про виконання лабораторної роботи №9

з дисципліни «Створення та рефакторінг програмно-інформаційного продукту засобами Python»

Виконала:  
студентка групи РІ-31  
Заплетнюк Ю.І.

Прийняв:  
Щербак С.С.

**Мета роботи:** розробка програмно-інформаційного продукту засобами Python.

**План роботи:**

Завдання 1. Створити скрипт запуску лабораторних робіт 1-8 (Runner) з єдиним меню для управління додатками використовуючи патерн FACADE https://refactoring.guru/uk/design-patterns/facade

Завдання 2. Зробити рефакторінг додатків, які були зроблені в лб 1-8, для підтримки можливості запуску через Runner

Завдання 3. Зробити рефакторинг додатків, які були зроблені в лб 1-8, використовуючи багаторівневу архітектуру додатків (див. приклад нижче) та всі принципи об’єктно-орієнтованого підходу

Завдання 4. Створити бібліотеку класів, які повторно використовуються у всіх лабораторних роботах та зробити рефакторінг додатків для підтримки цієї бібліотеки. Таких класів в бібліотеці має буде як найменш 5

Завдання 5. Додати логування функцій в класи бібліотеки програмного продукту використовуючи https://docs.python.org/uk/3/howto/logging.html

Завдання 6. Додати коментарі до програмного коду та сформувати документацію програмного продукту засобами pydoc. Документація має бути представлена у вигляді сторінок тексту на консолі, подана у веб-браузері та збережена у файлах HTML

Завдання 7. Документація та код програмного продукту має бути розміщено в GIT repo

Завдання 8. Проведіть статичний аналіз коду продукту засобами PYLINT https://pylint.readthedocs.io/en/stable/ та виправте помилки, які були ідентифіковані. Первинний репорт з помилками додайте до звіту лабораторної роботи

Завдання 9. Підготуйте звіт до лабораторной роботи

**Код програмної реалізації**

**task.py**

import sys

import os

sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_))))

from lab1.runner.runner import ConsoleCalculator

from lab2.classes.calculator import Calculator

from lab3.classes.art\_generatop import ArtGenerator

from lab4.classes.art\_generator import ArtRawGenerator

from lab5.classes.shape import Shape

from lab6.tests.tests import TestCalculator

from lab7.api.app import ApiApp

from lab8.classes.data\_loader import DataLoader

class RunnerFacade:

def \_\_init\_\_(self):

self.programs = {

"1": ConsoleCalculator,

"2": Calculator,

"3": ArtGenerator,

"4": ArtRawGenerator,

"5": Shape,

"6": TestCalculator,

"7": ApiApp,

"8": DataLoader,

}

def run\_program(self, program\_number):

program = self.programs.get(program\_number)

if program:

program.run()

else:

print("Incorrect choice. Try again.")

**runner.py**

import sys

import os

# додаємо до системного шляху директорію, яка містить модуль lab9 для коректного імпорту

sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_))))

from lab9.task import RunnerFacade # імпортуємо клас RunnerFacade з модуля lab9.task

def runner():

runner\_instance = RunnerFacade() # створення екземпляра класу RunnerFacade

while True:

# виведення меню на екран

print("\n Menu ")

print("1) Run Program 1")

print("2) Run Program 2")

print("3) Run Program 3")

print("4) Run Program 4")

print("5) Run Program 5")

print("6) Run Program 6")

print("7) Run Program 7")

print("8) Run Program 8")

print("0) Exit")

# отримання вибору користувача

choice = input("Select a menu option: ")

if choice == "0":

print("Exiting!") # повідомлення про вихід

break # завершення циклу та програми

# виклик методу для запуску вибраної програми

runner\_instance.run\_program(choice)

runner() # виклик функції для запуску меню

На рис. 1-3 показано результат виконання завдання.

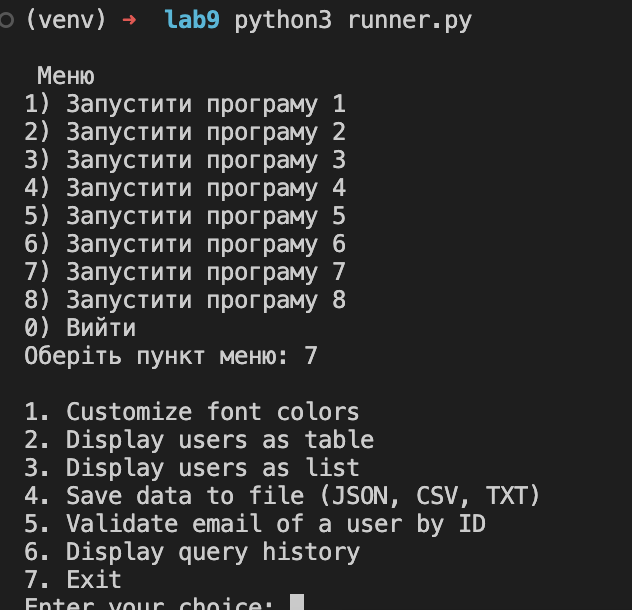


Рис. 1 Результат виконання завдання

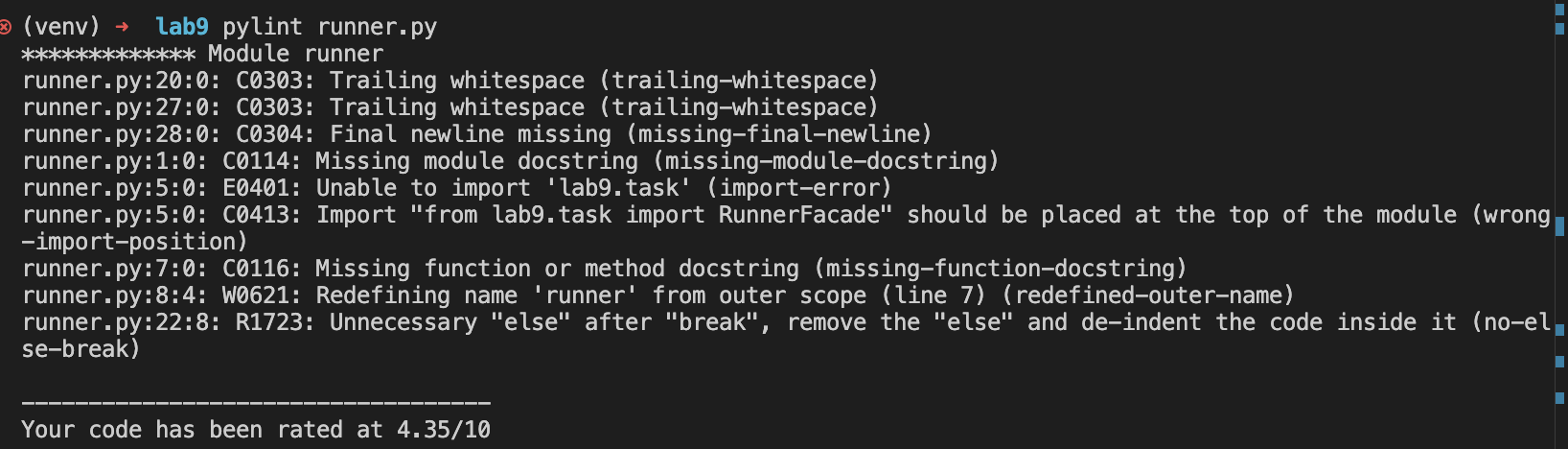


Рис. 2 Статичний аналіз коду



Рис. 3 Документація програмного коду

**Висновки:** виконавши лабораторну роботу, було розроблено програмно-інформаційний продукт засобами Python та створено скрипт запуску лабораторних робіт 1-8 за допомогою патерту Facade. Також проведено статичний аналіз коду засобами PYLINT.