МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра інформаційних систем та мереж

**Звіт**

До лабораторної роботи №9

**Створення та рефакторінг програмно-інформаційного продукту засобами Python**

Виконав:

ст. гр. ІТ-21сп

**Ярослав МАРКІВСЬКИЙ**

Прийняв:

доц. каф. ІСМ

**Сергій ЩЕРБАК**

Львів ‒ 2023

**Лабораторна робота №9**

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

**Хід 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 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

Файл **menu\_builder.py**

import json  
from abc import ABC, abstractmethod  
  
  
class Menu(ABC):  
  
  
 def \_\_init\_\_(self):  
 with open('config.json', 'r') as config\_file:  
 self.\_config = json.load(config\_file)  
 self.\_config = self.\_config["variables"]  
  
 @abstractmethod  
 def run(self):

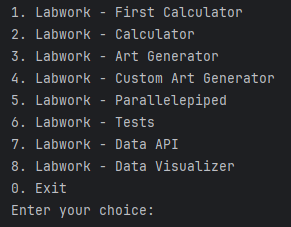
Файл **menu\_facade.py**

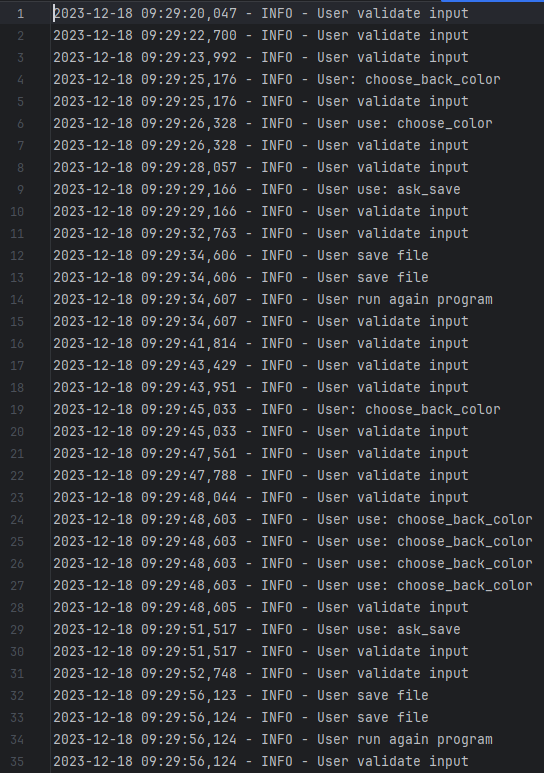
from UI.MenuBuilder.labwork\_1.first\_calculator\_menu import FirstCalculatorMenu  
from UI.MenuBuilder.labwork\_2.calculator\_menu import CalculatorMenu  
from UI.MenuBuilder.labwork\_3.art\_menu import ArtMenu  
from UI.MenuBuilder.labwork\_4.custom\_art\_menu import CustomArtMenu  
from UI.MenuBuilder.labwork\_5.parallelepiped\_menu import ParallelepipedMenu  
from UI.MenuBuilder.labwork\_6.test\_menu import TestMenu  
from UI.MenuBuilder.labwork\_7.data\_api\_menu import DataAPIMenu  
from UI.MenuBuilder.labwork\_8.data\_visualizer\_menu import DataVisualizerMenu  
  
class MenuFacade:  
  
 def \_\_init\_\_(self):  
 self.\_\_menus = [("Labwork - First Calculator", FirstCalculatorMenu()),  
 ("Labwork - Calculator", CalculatorMenu()),  
 ("Labwork - Art Generator", ArtMenu()),  
 ("Labwork - Custom Art Generator", CustomArtMenu()),  
 ("Labwork - Parallelepiped", ParallelepipedMenu()),  
 ("Labwork - Tests", TestMenu()),  
 ("Labwork - Data API", DataAPIMenu()),  
 ("Labwork - Data Visualizer", DataVisualizerMenu())]  
 self.\_\_finish\_number = 0  
  
 def print\_menu\_options(self):  
  
 for index, (name, \_) in enumerate(self.\_\_menus, start=1):  
 print(f"{index}. {name}")  
 print(f"{self.\_\_finish\_number}. Exit")  
  
 def start(self):  
 while True:  
 self.print\_menu\_options()  
 choice = input("Enter your choice: ")  
 try:  
 choice = int(choice)  
 if choice == self.\_\_finish\_number:  
 break  
 if not 1 <= choice <= len(self.\_\_menus):  
 raise ValueError  
 \_, menu = self.\_\_menus[choice - 1]  
 menu.run()  
 except ValueError:  
 print("Invalid choice. Enter again!")  
def start(self):  
 while True:  
 self.print\_menu\_options()  
  
 choice = input("Enter your choice: ")  
  
 try:  
 choice\_int = int(choice)  
 except ValueError:  
 print("Invalid choice. Enter again!")  
 continue  
  
 if choice\_int == self.\_\_finish\_number:  
 break  
  
 if not 1 <= choice\_int <= len(self.\_\_menus):  
 print("Invalid choice. Enter again!")  
 continue  
  
 menu\_name, menu = self.\_\_menus[choice\_int - 1]  
  
 menu.run()

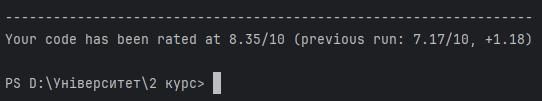
Файл **logger.py**

import logging  
  
logging.basicConfig(level=logging.INFO, filename="log.log", filemode="w",  
 format="%(asctime)s - %(levelname)s - %(message)s")

Результат виконання всіх завдань:







**Рис.1** Результат виконання

**Висновок:** розробив програмно-інформаційний продукт засобами Python.