Міністерство освіти і науки України

Національний університет «Львівська політехніка»



**ЗВІТ**

про виконання лабораторної роботи № 9

з курсу:

«Спеціалізовані мови програмування»

**Виконала:**

студентка гр. ІТ-31

Анастасія КАМІНОВСЬКА

**Прийняв:**

Сергій ЩЕРБАК

Львів 2024

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

**Мета роботи:** 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

import pandas as pd

"""Required modules"""

import os

import unittest

from logging.config import dictConfig

from share.console\_io.console\_output import ConsoleOutput

from share.console\_io.console\_input import ConsoleInput

from share.config import Config

from share.menu import Menu, menu\_item

from subsystems.lab1.main import main as lab1

from subsystems.lab2.main import main as lab2

from subsystems.lab3.src.ascii\_art\_maker import AsciiArtMaker

from subsystems.lab4.src.core.app import App

from subsystems.lab4.src.core.console\_inputter import ConsoleInputter

from subsystems.lab4.src.core.graphic import Graphic

from subsystems.lab5.main import main as lab5

from subsystems.lab6.main import main as lab6

from subsystems.lab7.main import main as lab7

from subsystems.lab8.main import main as lab8

class Runner(Menu):

    """Facade for running Python scripts through a console menu"""

    def \_\_init\_\_(self):

        super().\_\_init\_\_("Python Runner", "Select a subsystem to run")

    def run(self):

        """Start to show the menu to allow script selection"""

        self.show()

    @menu\_item(title="Calulator")

    def lab1(self) -> None:

        """Lab 1 Standard Python Calculator"""

        lab1()

    @menu\_item(title="OOP Calulator")

    def lab2(self) -> None:

        """Lab2 OOP Calculator"""

        lab2()

    @menu\_item("ASCII Arts")

    def lab3(self) -> None:

        """Lab3 ACSII Arts"""

        art\_maker = AsciiArtMaker()

        art\_maker.run()

    @menu\_item("ASCII Art without libs")

    def lab4(self) -> None:

        """Lab4 ASCII Art no libraries"""

        inputter = ConsoleInputter()

        size = os.get\_terminal\_size()

        outputter = Graphic(size.columns, size.lines)

        app = App(inputter, outputter)

        app.run()

    @menu\_item("ASCII Art 3D")

    def lab5(self) -> None:

        """Lab5 ASCII Art 3D"""

        lab5()

    @menu\_item("Unit tests")

    def lab6(self) -> None:

        """Lab6 Unit tests with oop calulator"""

        lab6()

    @menu\_item("API")

    def lab7(self) -> None:

        """Lab7 Api"""

        lab7()

    @menu\_item("CSV visualizer")

    def lab8(self) -> None:

        """Lab8 CSV visualizer"""

        lab8()

def main() -> None:

    """ Main function to run as script from shell """

    config = Config("config.json")

    dictConfig(config["logging"])

    runner = Runner()

    runner.run()

if '\_\_main\_\_' == \_\_name\_\_:

    main()

    # menu = ConsoleMenu("Runner")

    # menu.append\_item(FunctionItem("Calculator", lab1.main.main))

    # menu.append\_item(FunctionItem("OOP Calculator", lab2.main.main))

    # menu.append\_item(FunctionItem("ASCII Art", lab3.main.main))

    # menu.append\_item(FunctionItem("2D ASCII Art without additional libraries", lab4.main.main))

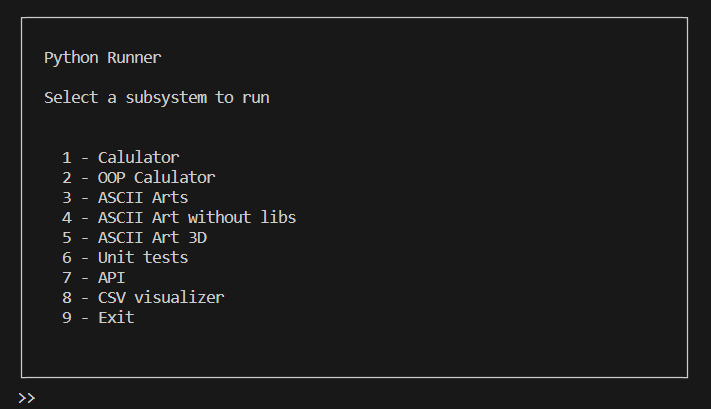
    # menu.append\_item(FunctionItem("3D ASCII Arts", lab5.main.main))

    # menu.append\_item(FunctionItem("Unit tests", lab6.main.main))

    # menu.append\_item(FunctionItem("API requests", lab7.main.main))

    # menu.append\_item(FunctionItem("Visualize csv files", lab8.main.main))

    # menu.show()



*Рис.1 – Результат виконання програми*

**Висновок:** на даній лабораторній роботі розроблено програмно-інформаційний продукт засобами Python.