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

Ірина ПЕРХУН

**Прийняв:**

Сергій ЩЕРБАК

Львів 2023

**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 лабораторної роботи*

Було створено runner.py та використано патерн FAÇADE:

"""Runner in facade pattern to run all subsystems"""

import unittest

from classes.lab1.main import main as lab1\_main

from classes.lab2.main import main as lab2\_main

from classes.lab3.main import main as lab3\_main

from classes.lab4.main import main as lab4\_main

from classes.lab5.main import main as lab5\_main

import classes.lab6.test\_calculator

from classes.lab7.main import main as lab7\_main

from classes.lab8.main import main as lab8\_main

from logs.logger import setup\_logger, logger

from shared.console\_input import UserInputHandler

class LabFacade:

    """Class for facade pattern"""

    @staticmethod

    def run\_lab\_1():

        """

        Run lab 2.

        """

        logger.info("Running lab 1")

        lab1\_main()

    @staticmethod

    def run\_lab\_2():

        """

        Run lab 2.

        """

        logger.info("Running lab 2")

        lab2\_main()

    @staticmethod

    def run\_lab\_3():

        """

        Run lab 3.

        """

        logger.info("Running lab 3")

        lab3\_main()

    @staticmethod

    def run\_lab\_4():

        """

        Run lab 4.

        """

        logger.info("Running lab 4")

        lab4\_main()

    @staticmethod

    def run\_lab\_5():

        """

        Run lab 5.

        """

        logger.info("Running lab 5")

        lab5\_main()

    @staticmethod

    def run\_lab\_6():

        """

        Run lab 6.

        """

        unittest.main(classes.lab6.test\_calculator)

    @staticmethod

    def run\_lab\_7():

        """

        Run lab 7.

        """

        logger.info("Running lab 7")

        lab7\_main()

    @staticmethod

    def run\_lab\_8():

        """

        Run lab 8.

        """

        logger.info("Running lab 8")

        lab8\_main()

if \_\_name\_\_ == "\_\_main\_\_":

    setup\_logger()

    lab\_facade = LabFacade()

    while True:

        lab\_number = 0

        while True:

            lab\_number = UserInputHandler.get\_integer\_input("Choose the lab number (1-8): ")

            if 0 <= lab\_number <= 8:

                break

            else:

                print("Entered number does not correspond to any lab work (0-8)")

        if lab\_number == 0:

            break

        else:

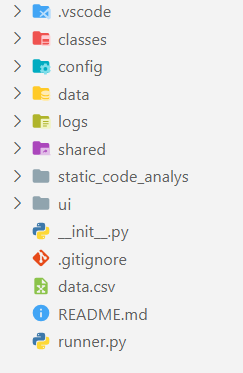
            getattr(lab\_facade, f'run\_lab\_{lab\_number}')()

        another\_lab = UserInputHandler.get\_yes\_no\_input("Do you want to perform another lab? (yes/no): ")

        if another\_lab == 'no':

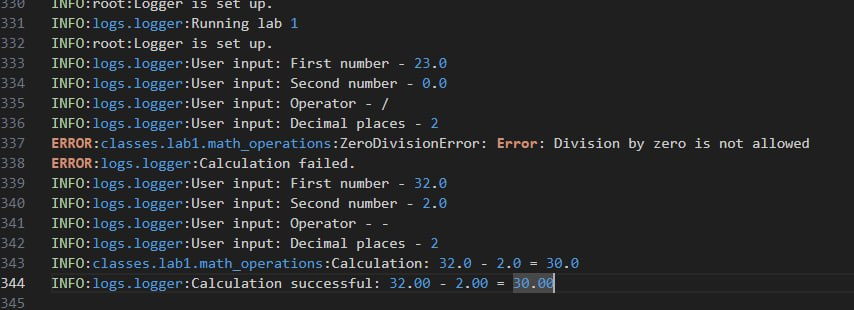
            break

Був проведений рефакторинг відповідно до якого структура проекту виглядає наступним чином:



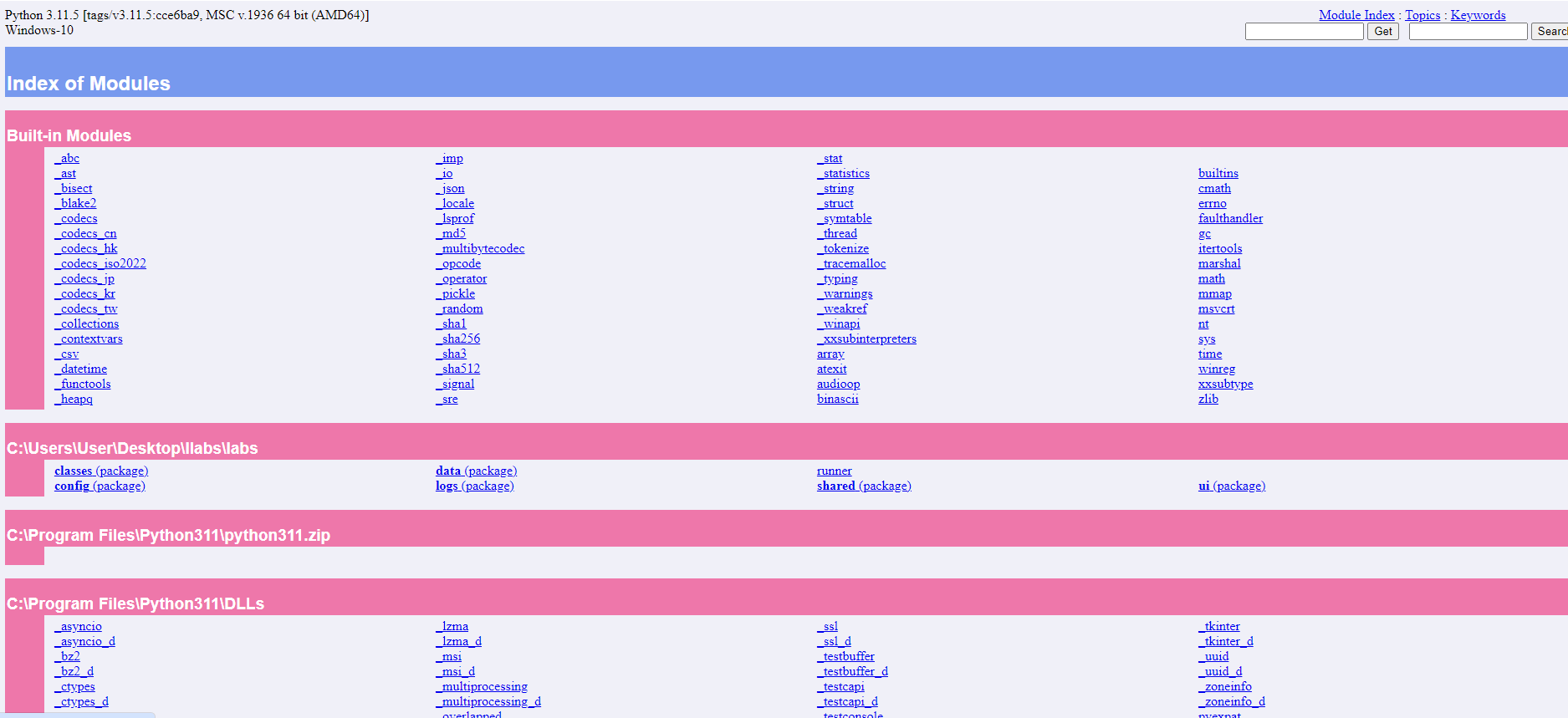
*Рис. 1 – Структура проекту*

Було створено логери для лабораторних робіт:



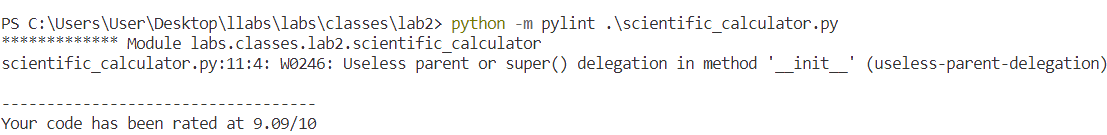
*Рис. 2 – Logger for lab1*

Здійснено документацію засобами pydoc:



*Рис. 3 – Результат документації*

Проведений статичний аналіз коду продукту засобами PYLINT:



*Рис. 4 – Перевірка одного з файлів*

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