

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**  
*Кафедра інформаційних систем та мереж*

**ЗВІТ**

про виконання лабораторної роботи № 9  
«Створення та рефакторинг програмно-інформаційного  
продукту засобами Python»  
з дисципліни "Спеціалізовані мови програмування”

**Виконала:**

ст. гр. ІТ-32,  
Троцько О. М.

**Прийняв:**

Щербак С. С.

**ЛЬВІВ – 2023**

**Мета:** 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 рефакторинг 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 рефакторинг додатків для підтримки цієї бібліотеки. Таких класів в бібліотеці має бути як найменш 5.

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

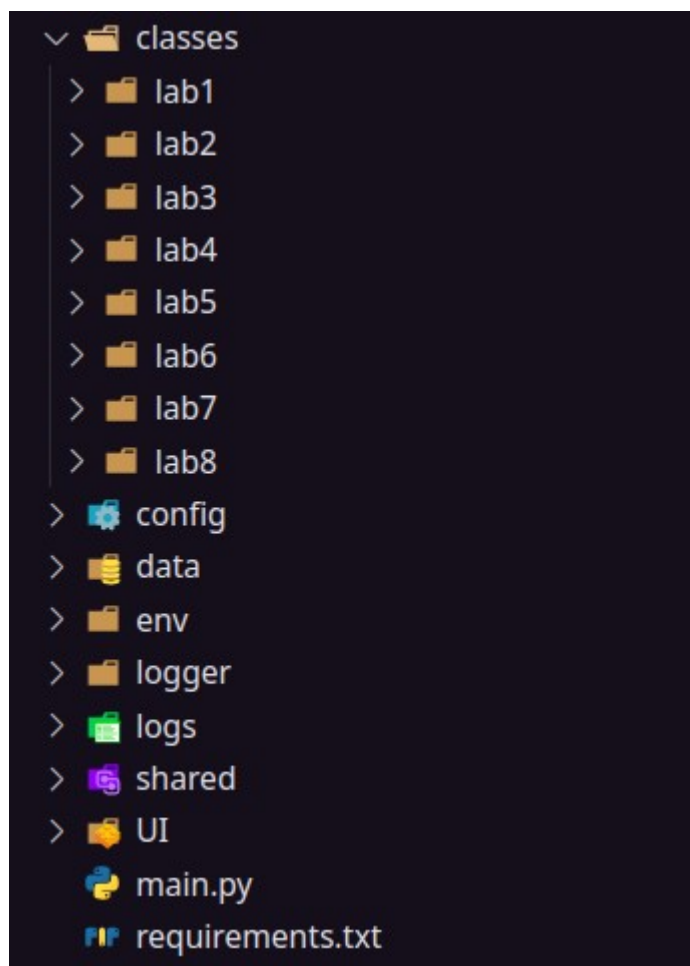
**Завдання 6.** Додати коментарі до програмного коду та сформувати документацію програмного продукту засобами roudoc. Документація має бути представлена у вигляді сторінок тексту на консолі, подана у веб-браузері та збережена у файлах HTML.

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

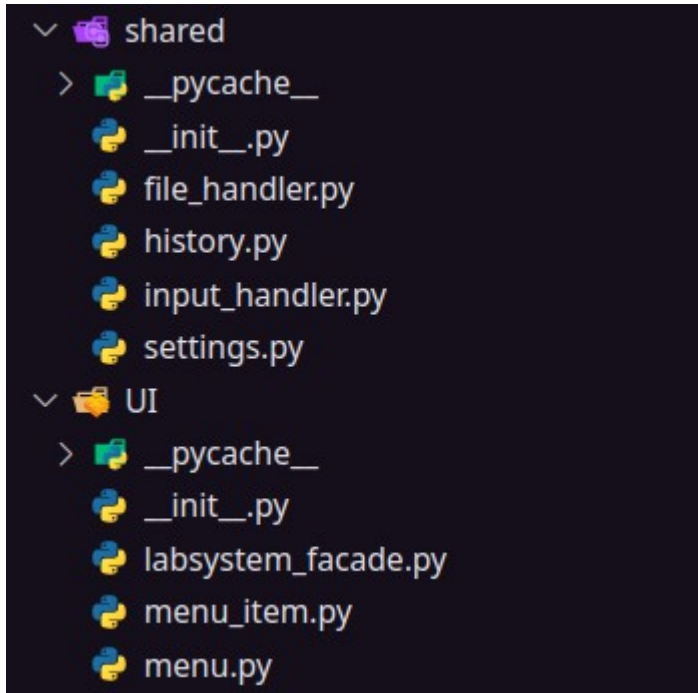
**Завдання 8.** Проведіть статичний аналіз коду продукту засобами PYLINT <https://pylint.readthedocs.io/en/stable/> та виправте помилки, які були ідентифіковані. Первинний репорт з помилками додайте до звіту лабораторної роботи.

**Завдання 9.** Підготуйте звіт до лабораторної роботи.

Архітектура додатка:



## Бібліотека класів:



## Код програми:

**# logger.py**

"""

Module to set up logging configuration and create a logger instance.

This module configures the logging system with a specific format and level, and creates a logger instance for use in other modules.

Attributes:

logger (logging.Logger): The logger instance configured for the module.

Usage:

Import this module and use the 'logger' instance to log messages with the configured settings.

"""

```
import logging
```

```
logging.basicConfig(  
    level=logging.INFO,  
    format='%(asctime)s [%(levelname)s]: %(message)s',
```

```

        datefmt='%d-%m-%Y %H:%M:%S',
        filename='logs/logs.log',
        filemode='w'
    )

logger = logging.getLogger(__name__)

```

# **# labsystem\_facade.py**

```

"""

```

```

Module: lab_system_facade

```

This module provides the Lab System Facade, which initializes and runs a menu for executing different lab modules.

```

"""

```

```

import importlib
import os
from UI.menu import Menu
from UI.menu_item import Item
from logger.logger import logger

```

```

class LabSystemFacade:

```

```

    """

```

```

    A class representing the Lab System Facade.

```

```

    Attributes:

```

```

    - labs_menu (Menu): The main menu for navigating through different lab modules.
    """

```

```

    def __init__(self):

```

```

        """

```

```

        Initializes a LabSystemFacade object.

```

```

        """

```

```

        self.labs_menu = Menu("Labs Menu")

```

```

        self._initialize_labs_menu()

```

```

    def _initialize_labs_menu(self):

```

```

        """

```

```

        Initializes the labs menu dynamically based on available lab modules.

```

```

        """

```

```

        labs = [f for f in os.listdir("classes") if
os.path.isdir(os.path.join("classes", f))]
        labs.sort()

        for index, lab in enumerate(labs, start=1):
            try:
                module = importlib.import_module(f'classes.{lab}.runner')
                self.labs_menu.add_item(Item(str(index), f'Lab {index} ',
module.run))
            except ImportError:
                pass

        self.labs_menu.add_item(Item(0, "Exit"))

    def run_program(self):
        """
        Runs the Lab System Facade, logging the start and end of the program.
        """
        logger.info("The program has started.")
        self.labs_menu.run()
        logger.info("The program has ended.")

# main.py
"""
Module: main

This module provides a simple script to run the Lab System Facade.
"""
from UI.labsystem_facade import LabSystemFacade

if __name__ == "__main__":
    """
    Initializes and runs the Lab System Facade.
    """
    facade = LabSystemFacade()
    facade.run_program()

```

Оцінка pylint до виправлень:

```
-----  
Your code has been rated at 7.04/10
```

Оцінка pylint після виправлень:

```
-----  
Your code has been rated at 9.84/10 (previous run: 9.84/10, +0.00)
```

**GitHub Repository:** <https://github.com/trolchiha/SPL-labs.git>

**Висновок:** під час виконання лабораторної роботи навчилася розробляти програмно-інформаційний продукт засобами Python.