МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж

Лабораторна робота №9

з дисципліни

СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ

на тему

Створення та рефакторінг програмно-інформаційного

продукту засобами Python

Виконала:

ст. гр. РІ-31

Оксана ЛЕСЮК

Прийняв:

Сергій Сергійович ЩЕРБАК

Львів-2024

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

**Завдання:**

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 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

Приклад організаціі Layered Architecture (LA) of Applications Базові рівні додатка: • UI • BLL • DAL • Sources (DB, Files,…)

Деталі можна прочитати за посиланням https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html

**Хід роботи**

**LabRunner.py:**

class LabRunner:

def \_\_init\_\_(self, max\_lab\_number=9):

self.max\_lab\_number = max\_lab\_number

def run\_lab(self, lab\_number):

try:

module\_name = f"ui.lab{lab\_number}.lab{lab\_number}\_menu"

main\_function = getattr(importlib.import\_module(module\_name), "main")

if callable(main\_function):

main\_function()

print(f"Lab {lab\_number} executed successfully.")

return True

else:

print(f"Error: 'main' function not found in lab {lab\_number}")

return False

except ImportError as e:

print(f"Error importing lab {lab\_number}: {e}")

return False

**Runner.py**:

class Runner:

def \_\_init\_\_(self, max\_lab\_number=9):

self.lab\_runner = LabRunner(max\_lab\_number)

self.input\_handler = UserInputHandler(max\_lab\_number)

def run(self):

while (lab\_number := self.input\_handler.get\_user\_input()) is not None:

self.lab\_runner.run\_lab(lab\_number)

if name == "main":

Runner().run()

**UserInputHandler.py:**

class UserInputHandler:

def \_\_init\_\_(self, max\_lab\_number):

self.max\_lab\_number = max\_lab\_number

def get\_user\_input(self):

while True:

user\_input = input(f"\nSelect a lab (1-{self.max\_lab\_number}) or 'q' to quit: ").lower()

if user\_input == 'q':

return None

if user\_input.isdigit() and 1 <= int(user\_input) <= self.max\_lab\_number:

return int(user\_input)

print(f"Please enter a number between 1 and {self.max\_lab\_number}.")

**Посилання на гіт-репозиторій:**

https://github.com/lesiukoksana/SMP-labs

**Висновок:**

У ході виконання лабораторної роботи було успішно реалізовано програмно-інформаційний продукт за допомогою Python, що відповідає визначеним завданням. Основна мета роботи була досягнута шляхом створення зручного інтерфейсу запуску лабораторних робіт (Runner), рефакторингу коду попередніх лабораторних завдань, розробки бібліотеки класів та інтеграції об'єктно-орієнтованого підходу із застосуванням багаторівневої архітектури.