

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

Виконав:
студент групи РІ-21сп
Владислав ДМИТРЕНКО

Львів – 2024

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 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 всі 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 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 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

Результати тестування:

```
PS C:\Users\Blxxd\Documents\GitHub\DegraCalc> python .\runner.py
Оберіть дію:
1. Запустити лабораторну роботу
2. Запустити всі тести (test_calculator.py i unit_test.py)
Ваш вибір (1 або 2):
```

Рис. 1. Користувацький інтерфейс

```
2. Запустити всі тести (test_calculator.py i unit_test.py)
Ваш вибір (1 або 2): 1
Оберіть лабораторну роботу для запуску:
1. lab1
2. lab2
3. lab3
4. lab4
5. lab5
6. lab6
7. lab7
8. lab8
Введіть номер лабораторної роботи (1-8):
```

Рис. 2. Вивід списку лабораторних робіт

```
Введіть номер лабораторної роботи (1-8): 8
Запускаю lab8...
1. Display extreme values
2. Show Temperature Line Chart
3. Show Temperature vs Humidity Scatter Plot
4. Show Humidity Histogram
5. Show Multiple Subplots
6. Export Chart as PNG
Select an option (or 'q' to quit):
```

Рис. 3. Запуск лабораторної роботи

```
PS C:\Users\Blxxd\Documents\GitHub\DegraCalc> python .\runner.py
Оберіть дію:
1. Запустити лабораторну роботу
2. Запустити всі тести (test_calculator.py i unit_test.py)
Ваш вибір (1 або 2): 2
Запускаю тести test_calculator.py...

test_addition_negative_numbers (__main__.TestCalculator.test_addition_negative_numbers) ... ok
✓Test 'test_addition_negative_numbers (__main__.TestCalculator.test_addition_negative_numbers)' passed successfully.
test_addition_positive_and_negative (__main__.TestCalculator.test_addition_positive_and_negative) ... ok
✓Test 'test_addition_positive_and_negative (__main__.TestCalculator.test_addition_positive_and_negative)' passed successfully.
```

Рис. 4. Вигляд всіх діаграм разом

Текст ui і інтеракції

```
def get_lab_choice():

    print("Оберіть лабораторну роботу для запуску:")
    lab_dirs = [f"lab{i}" for i in range(1, 9)]
    for i, lab in enumerate(lab_dirs, 1):
        print(f"{i}. {lab}")
    choice = input("Введіть номер лабораторної роботи (1-8): ")
    return int(choice) if choice.isdigit() and 1 <= int(choice) <= 8 else
None

def run_lab(lab_number):
```

```

base_dir = os.path.dirname(os.path.abspath(__file__))
lab_name = f"lab{lab_number}"
lab_path = os.path.join(base_dir, "..", lab_name, "main.py")

if os.path.exists(lab_path):
    print(f"Запускаю {lab_name}...")
    try:
        subprocess.run(["python", lab_path], check=True)
    except subprocess.CalledProcessError as e:
        print(f"Помилка при виконанні {lab_path}: {e}")
else:
    print(f"Файл {lab_path} не знайдено!")

def interact():

    print("Оберіть дію:")
    print("1. Запустити лабораторну роботу")
    print("2. Запустити всі тести (test_calculator.py і unit_test.py)")
    choice = input("Ваш вибір (1 або 2): ")

    if choice == "1":
        lab_choice = get_lab_choice()
        if lab_choice:
            run_lab(lab_choice)
        else:
            print("Невірний вибір. Будь ласка, введіть номер від 1 до 8.")
    elif choice == "2":
        run_tests()
    else:
        print("Невірний вибір. Введіть 1 або 2.")

def run_tests():

    base_dir = os.path.dirname(os.path.abspath(__file__))

    test_calculator_path = os.path.join(base_dir, "..", "lab6", "tests",
"test_calculator.py")
    unit_test_path = os.path.join(base_dir, "..", "lab7", "tests",
"unit_test.py")

    for test_path in [test_calculator_path, unit_test_path]:
        if os.path.exists(test_path):
            print(f"Запускаю тести {os.path.basename(test_path)}...\n")
            try:
                subprocess.run(["python", test_path], check=True)
            except subprocess.CalledProcessError as e:
                print(f"Помилка під час виконання тестів у {test_path}: {e}")
        else:
            print(f"Файл {test_path} не знайдено!")

```

Висновки: В ході виконання лабораторної роботи було створено проєкт, який об'єднує всі лабораторні роботи, та дозволяє запуснути їх з одного раннера, також виконано рефакторинг та описана документація коду.