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

Андрій ПІГУЛЯК

Львів – 2024

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

**Хід роботи**

1. Створити Runner з меню для запуску лабораторних робіт (1-8) з використанням патерну Facade для уніфікованого управління додатками.
2. Провести рефакторинг лабораторних робіт (1-8), додавши підтримку запуску через Runner.
3. Виконати рефакторинг додатків (1-8), впровадивши багаторівневу архітектуру (наприклад, рівні Presentation, Business Logic, Data Access) та дотримуючись принципів ООП.
4. Розробити бібліотеку класів для спільного використання в усіх лабораторних роботах. Бібліотека має містити не менше 5 класів. Виконати рефакторинг програм для інтеграції цієї бібліотеки.
5. Додати логування у класи бібліотеки, щоб відстежувати виконання функцій.
6. Додати коментарі до коду та створити документацію засобами pydoc. Документація має бути:

* Доступна у вигляді тексту в консолі,
* Відображена у веб-браузері,
* Збережена у файлах HTML.

1. Виконати статичний аналіз коду засобами Pylint, виправити помилки, та зберегти початковий звіт з аналізу у звіті до лабораторної роботи.
2. **Опис і структура програми:**

Програма складається з головного модуля main.py, який надає користувачу доступ до окремих лабораторних робіт через головне меню. Для кожної лабораторної роботи реалізовано окремий фасад, що інкапсулює її логіку та забезпечує простий інтерфейс для її виконання.

**Структура директорій і файлів**

* + 1. **Фасади лабораторних робіт**
* Lab1Facade: Інкапсулює функціональність консольного калькулятора.
* Lab2Facade: Використовує подібний калькулятор, оптимізований для іншої задачі.
* Lab3Facade: Реалізує генерацію ASCII-арту з вибором шрифту, кольору, розміру та символів.
* Lab4Facade: Забезпечує створення ASCII-арту з можливістю зміни кольору чи символів та збереження в файл.
* Lab5Facade: Реалізує 3D-об'єкти (куб, піраміда) з можливістю обертання, зміни масштабу, кольору та збереження в файл.
* Lab7Facade: Надає доступ до роботи з API, обробки даних та журналювання.
* Lab8Facade: Забезпечує візуалізацію даних про вино за допомогою графіків.
  + 1. **Головне меню (main.py)**
* Відображає список доступних лабораторних робіт.
* Надає інтерфейс для вибору роботи та її виконання.
* Передбачає цикл роботи програми до вибору опції "Вихід".

**Використані патерни та принципи**

* **Патерн *Фасад***: Кожна лабораторна робота реалізована як фасад для спрощення виклику та інкапсуляції складної логіки.
* **Принцип єдиної відповідальності** (*SRP*): Фасади відповідають лише за інтерфейс взаємодії із відповідною лабораторною роботою.
* **Принцип відкритості/закритості** (*OCP*): Легке додавання нових лабораторних робіт через створення нового фасаду.

1. **Функціонал:**
   1. **Запуск програми**

* Користувач обирає номер лабораторної роботи в головному меню.
* Наприклад, обравши "3", програма запускає фасад Lab3Facade, що відповідає за генерацію ASCII-арту.
  1. **Виконання лабораторної роботи**
* Фасад викликає всі необхідні модулі та функції для виконання роботи.
* Користувач взаємодіє з відповідним інтерфейсом (наприклад, вводить текст для генерації ASCII-арту чи обирає колір).
  1. **Завершення**
* Після виконання завдання користувач може повернутись до головного меню або завершити роботу програми.

**3. Код програми**

1. **Lab3\_facade.py**

class Lab3Facade:  
  
 def \_\_init\_\_(self):  
  
 def gather\_input(self):  
 self.text = get\_user\_input()  
 self.font = choose\_font()  
 self.color = choose\_color()  
 self.alignment = choose\_alignment()  
  
 def generate\_ascii(self):  
 self.ascii\_art = generate\_ascii\_art(self.text, font=self.font)  
 self.colored\_ascii\_art = colored(self.ascii\_art, self.color)  
 self.formatted\_ascii\_art = format\_ascii\_art(self.colored\_ascii\_art, self.alignment)  
  
 def scale\_and\_customize(self):  
 self.max\_width, self.max\_height = choose\_size()  
 self.scaled\_ascii\_art = scale\_ascii\_art(self.formatted\_ascii\_art, self.max\_width, self.max\_height)  
 self.symbols = choose\_symbols()  
 self.custom\_ascii\_art = replace\_with\_symbols(self.scaled\_ascii\_art, self.symbols)  
  
 def preview\_and\_save(self):  
 print("\nПопередній перегляд ASCII-арту:")  
 print(self.custom\_ascii\_art)  
  
 user\_choice = input("\n Введіть 'y' для продовження або 'n' для перезапуску: ").strip().lower()  
  
 if user\_choice == 'y':  
 save\_option = input("Зберегти ASCII-арт у файл? (y/n): ").lower()  
 if save\_option == 'y':  
 save\_to\_file(self.custom\_ascii\_art)  
 else:  
 print("Процес перезапускається...")

1. **lab5\_facade.py**

class Lab5Facade:  
  
 def execute(self):  
 angle\_x = 0  
 angle\_y = 0  
 current\_size = 1.5  
  
 colors, default\_color = load\_colors\_from\_json()  
  
 shape\_type = get\_shape\_type()  
 shape = Cube(current\_size, default\_color) if shape\_type == "cube" else Pyramid(current\_size, default\_color)  
  
 shape\_representation = shape.draw(angle\_x, angle\_y)  
 print(shape\_representation)  
  
 while True:  
 scale\_factor = get\_scale\_factor()  
 if scale\_factor == 'exit':  
 print("Вихід з програми.")  
 break  
  
 current\_size \*= scale\_factor  
 shape.size = current\_size  
 color\_choice = get\_color\_choice(colors)  
 if color\_choice:  
 shape.color = colors[color\_choice]  
  
 shape\_representation = shape.draw(angle\_x, angle\_y)  
 print(shape\_representation)  
  
 save\_prompt = get\_save\_prompt()  
 if save\_prompt == 'y':  
 filename = get\_filename()  
 save\_to\_file(filename, shape\_representation.replace(f"{shape.color}", "").replace("\033[0m", ""))  
 print(f"Фігуру збережено у файл {filename}.")

1. **lab5\_facade.py**

class Lab8Facade:  
 def \_\_init\_\_(self):  
 self.visualizer = WineVisualizer()  
  
 def execute(self):  
 print("=== Лабораторна робота 8: Візуалізація даних про вино ===")  
 self.visualizer.start()

1. **main.py**

def main():  
while True:  
 print("\n=== Головне меню ===")  
 print("1. Лабораторна 1: Консольний калькулятор")  
 print("8. Лабораторна 8: Візуалізація вина")  
 print("0. Вихід")  
  
 choice = input("Ваш вибір: ").strip()  
  
 if choice == '0':  
 print("Дякуємо за використання програми!")  
 break  
  
 elif choice == '1':  
 lab1\_facade = Lab1Facade()  
 lab1\_facade.execute()  
  
 elif choice == '8':  
 lab8\_facade = Lab8Facade()  
 lab8\_facade.execute()  
  
 else:  
 print("Неправильний вибір, спробуйте ще раз.")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**4.** **Приклад роботи:**

**Приклад 1:** Запуск програми

A screenshot of a computer

Description automatically generated

Рис. 1 Вигляд головного меню

**Приклад 2:** Вибір запуску програм



Рис. 2 Запуск пʼятої лабораторної

**Висновок:** У рамках дев’ятої лабораторної роботи ми реалізували програму, яка об'єднує результати попередніх лабораторних робіт за допомогою патерну проектування "Фасад". Цей підхід дозволив спростити взаємодію користувача з програмою, забезпечивши єдиний інтерфейс для доступу до кожної функціональної частини. Усі лабораторні роботи були адаптовані до нового підходу.